# Package 'fitdistcp'

November 24, 2025

Type Package

**Title** Distribution Fitting with Calibrating Priors for Commonly Used Distributions

Version 0.2.3

Maintainer Stephen Jewson < stephen.jewson@gmail.com>

**Imports** stats, mev, extraDistr, gnorm, fdrtool, pracma, rust, actuar, fExtremes

**Depends** R (>= 3.5.0)

**Description** Generates predictive distributions based on calibrating priors for various commonly used statistical models, including models with predictors. Routines for densities, probabilities, quantiles, random deviates and the parameter posterior are provided. The predictions are generated from the Bayesian prediction integral, with priors chosen to give good reliability (also known as calibration). For homogeneous models, the prior is set to the right Haar prior, giving predictions which are exactly reliable. As a result, in repeated testing, the frequencies of out-of-sample outcomes and the probabilities from the predictions agree. For other models, the prior is chosen to give good reliability. Where possible, the Bayesian prediction integral is solved exactly. Where exact solutions are not possible, the Bayesian prediction integral is solved using the Datta-Mukerjee-Ghosh-Sweeting (DMGS) asymptotic expansion. Optionally, the prediction integral can also be solved using posterior samples generated using Paul Northrop's ratio of uniforms sampling package ('rust'). Results are also generated based on maximum likelihood, for comparison purposes. Various model selection diagnostics and testing routines are included. Based on ``Reducing reliability bias in assessments of extreme weather risk using calibrating priors", Jewson, S., Sweeting, T. and Jewson, L. (2024); <doi:10.5194/ascmo-11-1-2025>.

License MIT + file LICENSE

BugReports https://github.com/stephenjewson/fitdistcp/issues

URL https://www.fitdistcp.info

**Encoding UTF-8** 

LazyData true
LazyData duc
RoxygenNote 7.3.3
Suggests knitr, rmarkdown
NeedsCompilation no
Author Stephen Jewson [aut, cre] (ORCID:
<pre><https: 0000-0002-6011-6262="" orcid.org="">)</https:></pre>
Repository CRAN
<b>Date/Publication</b> 2025-11-24 07:40:02 UTC

= 0 = 1	24
	24
	24
	25
J = 1	26
	33
···· <b>)</b> —     · · · · · · · · · · · · · · · · ·	33
<b>7</b> =	34
	34
<b>7</b> =	35
cauchy_fdd	
<b>7</b> —	36
<b>7</b> —	36
<b>7</b> —	37
	37
<b>7</b> =	38
	38
	39
	40
cauchy_logfddd	40
cauchy_loglik	41
cauchy_logscores	41
cauchy_mulf	42
	42
	43
	43
cauchy_p1_f1f	51
cauchy_p1_f1fa	52
cauchy_p1_f1fw	52
cauchy_p1_f2f	53
cauchy_p1_f2fa	53
cauchy_p1_f2fw	54
cauchy_p1_fd	54
cauchy_p1_fdd	55
	55

cauchy_p1_ldda	 56
cauchy_p1_lddd	 57
cauchy_p1_lddda	 57
cauchy_p1_lmn	 58
cauchy_p1_lmnp	 59
cauchy_p1_logf	 59
cauchy_p1_logfdd	60
cauchy_p1_logfddd	61
cauchy_p1_loglik	61
cauchy_p1_logscores	62
cauchy_p1_means	62
cauchy_p1_mu1f	63
cauchy_p1_mu2f	64
cauchy_p1_p1f	64
cauchy_p1_p1f	65
cauchy_p1_predictordata	66
	66
cauchy_p1_waic	
cauchy_p2f	67
cauchy_waic	68
crhpflat_dmgs_cpmethod	69
d010exp_example_data_v1	69
d011pareto_k2_example_data_v1	69
d020halfnorm_example_data_v1	69
d025unif_example_data_v1	70
d030norm_example_data_v1	70
d031norm_dmgs_example_data_v1	70
d032gnorm_k3_example_data_v1	 70
d035lnorm_example_data_v1	 70
d036lnorm_dmgs_example_data_v1	 71
d040logis_example_data_v1	 71
d041lst_k3_example_data_v1	 71
d042cauchy_example_data_v1	
d050gumbel_example_data_v1	
d051frechet_k1_example_data_v1	
d052weibull_example_data_v1	
d053gev k3 example data v1	
d055exp_p1_example_data_v1_t	72
d055exp_p1_example_data_v1_x	72
d055exp_p1_example_data_v1_x	73
d056pareto_p1k2_example_data_v1_x	73
	73
d060norm_p1_example_data_v1_t	
d060norm_p1_example_data_v1_x	73
d061lnorm_p1_example_data_v1_t	73
d061lnorm_p1_example_data_v1_x	74
d062logis_p1_example_data_v1_t	74
d062logis_p1_example_data_v1_x	74
d063lst_p1k3_example_data_v1_t	74
d063lst_p1k3_example_data_v1_x	 74

du64caucny_p1_example_data_v1_t
$d064 cauchy\_p1\_example\_data\_v1\_x \ldots \ldots$
$d070 gumbel\_p1\_example\_data\_v1\_t \ . \ . \ . \ . \ . \ . \ . \ . \ . \ $
$d070gumbel\_p1\_example\_data\_v1\_x  \dots  \dots  \dots  .  .  .  .  .  .  .$
$d071 frechet\_p2k1\_example\_data\_v1\_t \dots \dots$
d071frechet_p2k1_example_data_v1_x 76
d072weibull_p1_example_data_v1_t
d072weibull_p1_example_data_v1_x 76
$d073 weibull\_p2\_example\_data\_v1\_t \dots \dots$
d073weibull_p2_example_data_v1_x
d074gev_p1k3_example_data_v1_t
d074gev_p1k3_example_data_v1_x
d080norm_p12_example_data_v1_t1
d080norm_p12_example_data_v1_t2
d080norm_p12_example_data_v1_x
d081lst_p12k3_example_data_v1_t1
d081lst_p12k3_example_data_v1_t2
d081lst_p12k3_example_data_v1_x
d082weibull_p12_example_data_v1_t1
d082weibull_p12_example_data_v1_t2
d082weibull_p12_example_data_v1_x
d100gamma_example_data_v1
d101invgamma_example_data_v1
d102invgauss_example_data_v1
d105burr_example_data_v1
•
d110gev_example_data_v1
d120gpd_k1_example_data_v1
d150gev_p1_example_data_v1_t
d150gev_p1_example_data_v1_x
d151gev_p12_example_data_v1_t
d151gev_p12_example_data_v1_x
d152gev_p123_example_data_v1_t
$d152 gev\_p123\_example\_data\_v1\_x  .  .  .  .  .  .  .  .  .  $
deauchysub
dcauchy_p1
dcauchy_p1sub
deriv_copyfdd
deriv_copyld2
deriv_copyldd
deriv_copylddd
dexpsub
dexp_p1
dexp_p1sub
dfrechetsub
dfrechet_p2k1
dfrechet_p2k1sub
dgammasub
dgevsuh 89

dgev_k3sub														
dgev_p1														
dgev_p12														
dgev_p123														
dgev_p123sub														
dgev_p12sub		 					 				 			 92
dgev_p1k3		 					 				 			 93
dgev_p1k3sub		 					 				 			 94
dgev_p1n		 					 				 			 95
dgev_p1nsub		 					 				 			 95
dgev_p1sub		 					 				 			 96
dgnorm_k3sub														
dgpdsub														
dgumbelsub														
dgumbel_p1														
dgumbel_p1sub														
dhalfnormsub														
dinvgammasub														
dinvgausssub														
dlnormsub														
dlnorm_dmgssub .														
dlnorm_p1														
dlnorm_p1sub														
-														
dlogis2sub														
dlogis_p1														
dlogis_p1sub														
dlst_k3sub														
dlst_p1k3														
dlst_p1k3sub														
dmgs														
dnormsub														
dnorm_dmgssub														
dnorm_p1														
dnorm_p12		 					 							 108
dnorm_p12dmgs .														
dnorm_p1sub														
dnorm_p1_formula		 					 				 			 110
dpareto_k2_sub		 					 				 			 110
dpareto_p1k2		 					 				 			 111
dpareto_p1k2sub .		 					 				 			 111
dunif_formula		 					 				 			 112
dweibullsub		 					 				 			 112
dweibull_p2		 					 				 			 113
dweibull_p2sub														
exp_cp														
exp_f1fa														
exp_f2fa														
exp_fd														
	 •	 	 •	•	 •	 •	 	 •	 •	 •	 	•		 

exp_fdd	121
exp_ldda	122
exp_lddda	123
exp_logf	123
exp logfdd	124
exp_logfddd	124
exp_logscores	
exp_p1fa	
exp_p1_cp	
exp_pl_flfa	
exp_pl_flfw	
exp_p1_f2fa	
exp_p1_f2fw	
exp_p1_fd	
exp_p1_fdd	
exp_p1_ldda	
exp_p1_lddda	
exp_p1_logf	
exp_p1_logfdd	
exp_p1_logfddd	
exp_p1_loglik	
exp_p1_logscores	
exp_p1_means	
exp_p1_mulfa	
exp_p1_mu1fa	
exp_p1_mu2ia	
1-1 -1	
exp_p1_p2fa	
exp_p1_pd	
exp_pl_pdd	
exp_pl_predictordata	
exp_pl_waic	
exp_p2fa	
exp_pd	
exp_pdd	
exp_waic	
findnt	
fixgevrange	
fixgpdrange	
frechet_k1_cp	
frechet_k1_f1fa	
$frechet\_k1\_f2fa \ \dots \ $	
$frechet\_k1\_fd \ \dots $	
$frechet\_k1\_fdd \dots $	
frechet_k1_ldda	
$frechet\_k1\_lddda \ \ldots \ \ldots$	
$frechet\_k1\_logf \ldots \ldots \ldots \ldots \ldots \ldots$	
$frechet\_k1\_logfdd\dots$	
frechet_k1_logfddd	159

frechet_k1_mu1fa	. 159
frechet_k1_mu2fa	. 160
frechet_k1_p1fa	. 160
frechet_k1_p2fa	. 161
frechet_k1_pd	. 161
frechet_k1_pdd	. 162
frechet_k1_waic	. 162
frechet_loglik	. 163
frechet_logscores	
frechet_means	. 164
frechet_p2k1_cp	. 164
frechet_p2k1_f1fa	
frechet_p2k1_f1fw	
frechet_p2k1_f2fa	
frechet_p2k1_f2fw	
frechet_p2k1_fd	
frechet_p2k1_fdd	
frechet_p2k1_ldda	
frechet_p2k1_lddda	
frechet_p2k1_logf	
frechet_p2k1_logfdd	
frechet_p2k1_logfddd	
frechet_p2k1_loglik	
frechet_p2k1_logscores	
frechet_p2k1_means	
frechet_p2k1_mu1fa	
<b></b>	
frechet_p2k1_mu2fa	
frechet_p2k1_p1fa	
frechet_p2k1_p2fa	
frechet_p2k1_pd	
frechet_p2k1_pdd	
frechet_p2k1_predictordata	
frechet_p2k1_waic	
gamma_cp	
gamma_f1f	
gamma_f1fa	
gamma_f2f	
gamma_f2fa	
gamma_fd	
gamma_fdd	
gamma_gg	
gamma_gmn	. 196
gamma_ldd	. 196
gamma_ldda	. 197
gamma_lddd	. 198
gamma_lddda	. 198
gamma_lmn	. 199
gamma_lmnp	. 199

gamma_logf	
gamma_logfdd	
gamma_logfddd	201
gamma_loglik	202
gamma_logscores	202
gamma_means	203
gamma_mulf	203
gamma_mu2f	204
gamma_p1f	204
gamma_p2f	205
gamma_waic	206
gev_boot	206
gev_checkmle	
gev_cp	
gev_f1fa	
gev_f2fa	
gev_fd	
gev_fdd	
gev_k12_ppm_minusloglik	
gev_k3_cp	
gev_k3_f1fa	
gev_k3_f2fa	
gev_k3_fd	
gev_k3_fdd	
gev_k3_ldda	
gev_k3_lddda	
gev_k3_logf	
gev_k3_logfdd	
gev_k3_logfddd	
gev_k3_loglik	
gev_k3_means	
gev_k3_mu1fa	
gev_k3_mu2fa	
gev_k3_pd	
gev_k3_pdd	
gev k3 waic	
gev ld12a	
gev_lda	
gev_ldda	
gev_lddda	
gev_logf	
gev_logfd	
gev_logfdd	
gev_logfddd	
gev_loglik	
gev_means	
gev_mu1fa	
gev mu2fa	

gev_p123_checkmle	
gev_p123_cp	
gev_p123_f1fa	
gev_p123_f1fw	
gev_p123_f2fa	
gev_p123_f2fw	
gev_p123_fd	
gev_p123_fdd	
gev_p123_ldda	
gev_p123_lddda	
gev_p123_logf	
gev_p123_logfdd	
gev_p123_logfddd	258
gev_p123_loglik	259
gev_p123_means	259
gev_p123_mu1fa	260
gev_p123_mu2fa	261
gev_p123_pd	261
gev_p123_pdd	
gev_p123_predictordata	
gev_p123_setics	
gev_p123_waic	
gev_p12k3_f1fa	
gev_p12k3_f1fw	
gev_p12k3_f2fa	
gev_p12k3_f2fw	
gev_p12k3_fd	
gev_p12k3_fdd	
gev_p12k3_ldda	
gev_p12k3_lddda	
gev_p12k3_logfdd	
gev_p12k3_logfddd	
gev_p12k3_mu1fa	
gev_p12k3_mu2fa	
gev p12k3 pd	
gev_p12k3_pdd	
gev_p12_checkmle	
gev_p12_cp	
gev_p12_t1fa	
gev_p12_f1fw	
gev_p12_f1fw	
gev_p12_f2fw	
gev_p12_i2iw	
gev_p12_fdd	
· ·	
gev_p12_ldda	
gev_p12_lddda	
gev_p12_logf	
gev_p12_logfdd	289

gev_p12_logfddd	
gev_p12_loglik	
gev_p12_means	. 291
gev_p12_mu1fa	. 292
gev_p12_mu2fa	
gev_p12_pd	
gev_p12_pdd	
gev_p12_predictordata	
gev_p12_setics	
gev_p12_waic	
gev_pla_flfa	
gev_pla_flfw	
gev_pla_f2fa	
gev_p1a_f2fw	
gev_pla_fd	
gev_pla_fdd	
gev_p1a_ldda	
gev_p1a_lddda	
gev_p1a_logfdd	
gev_p1a_logfddd	
gev_p1a_mu1fa	
gev_p1a_mu2fa	
gev_p1a_muzia	
gev_p1a_pdd	
gev_p1b_f1fa	
gev_p1b_f1fw	
gev_plb_f2fa	
gev_p1b_f2fw	
gev_plb_fd	
gev_plb_fdd	
gev_p1b_ldda	
gev_p1b_lddda	
gev_p1b_logfdd	
gev_plb_logfddd	
gev_plb_mu1fa	
gev_p1b_mu2fa	
gev_p1b_pd	
gev_p1b_pdd	
gev_plc_flfa	
gev_p1c_f1fw	
gev_p1c_f2fa	
gev_p1c_f2fw	
gev_p1c_fd	
gev_p1c_fdd	
gev_p1c_ldda	
gev_p1c_lddda	
gev_p1c_logfdd	
gev_p1c_logfddd	. 321

gev_plc_mulfa	
gev_p1c_mu2fa	. 322
$gev\_p1c\_pd \ \dots $	. 323
$gev\_p1c\_pdd  \dots \dots$	. 324
gev_p1k3_cp	. 324
gev_p1k3_f1fa	. 332
gev_p1k3_f1fw	. 333
gev_p1k3_f2fa	. 333
gev_p1k3_f2fw	. 334
gev_p1k3_fd	. 334
gev_p1k3_fdd	. 335
gev_p1k3_ldda	. 336
gev_p1k3_lddda	. 336
gev_p1k3_logf	. 337
gev_p1k3_logfdd	. 337
gev_p1k3_logfddd	. 338
gev_p1k3_loglik	. 338
gev_p1k3_means	
gev_p1k3_mu1fa	
gev_p1k3_mu2fa	
gev_p1k3_pd	
gev_p1k3_pdd	
gev_p1k3_predictordata	
gev_p1k3_waic	
gev_pln_checkmle	. 344
gev_p1n_cp	. 344
gev_pln_logf	. 353
gev_p1n_loglik	. 354
gev_p1n_means	. 354
gev_p1n_n1_exampledata	. 355
gev_p1n_n2_exampledata	
gev_p1n_predictordata	. 356
gev_pln_setics	. 357
gev_p1n_waic	. 357
gev_p1_checkmle	. 358
gev_p1_cp	. 358
gev_p1_logf	
gev_p1_loglik	. 368
gev_p1_means	
gev_p1_predictordata	
gev_pl_setics	
gev_p1_waic	
gev_pd	. 370
gev_pdd	
gev_pwm_params	
gev_setics	
gev_waic	
gnorm_k3_cp	

Contents Contents

gnorm_k3_f1f	
gnorm_k3_f1fa	
gnorm_k3_f2f	382
gnorm_k3_f2fa	
gnorm_k3_fd	383
gnorm_k3_fdd	383
gnorm_k3_ldd	384
gnorm_k3_ldda	385
gnorm_k3_lddd	385
gnorm_k3_lddda	386
gnorm_k3_lmn	386
gnorm_k3_logf	387
gnorm_k3_logfdd	387
gnorm_k3_logfddd	388
gnorm_k3_loglik	388
gnorm_k3_logscores	389
gnorm_k3_mu1f	
gnorm_k3_mu2f	
gnorm_k3_p1f	
gnorm_k3_p2f	
gnorm_lmnp	
gnorm_waic	
gpd_k13_f1fa	
gpd_k13_f2fa	
gpd_k13_fd	
gpd_k13_fdd	
gpd_k13_ldda	
gpd_k13_lddda	
gpd_k13_logfdd	
gpd_k13_logfddd	
gpd_k13_mu1fa	
gpd_k13_mu2fa	
gpd_k13_pd	
gpd_k13_pdd	
gpd_k1_checkmle	
gpd_k1_cp	
gpd_k1_f1fa	408
gpd_k1_f2fa	
gpd_k1_fd	
gpd_k1_fdd	
gpd_k1_ldda	
gpd_k1_lddda	
gpd_k1_logf	
gpd_k1_logfdd	
gpd_k1_logfddd	
gpd_k1_loglik	
gpd_k1_lognk	
gpd_k1_means	
2DU NI 111UITA	41.)

gpd_k1_mu2fa	415
gpd_k1_pd	416
gpd_k1_pdd	416
gpd_k1_setics	417
gpd_k1_waic	417
gumbel_cp	
gumbel_f1fa	
gumbel_f2fa	
gumbel_fd	
gumbel_fdd	
gumbel ldda	
gumbel_lddda	
gumbel_logf	
gumbel_logfdd	
gumbel_logfddd	
gumbel_loglik	
gumbel_logscores	
gumbel_means	
gumbel_mu1fa	
_	
gumbel_mu2fa	
gumbel_p1fa	
gumbel_p1_cp	
gumbel_p1_f1fa	
gumbel_p1_f1fw	
gumbel_p1_f2fa	
gumbel_p1_f2fw	
gumbel_p1_fd	
gumbel_p1_fdd	
gumbel_p1_ldda	
gumbel_p1_lddda	
gumbel_p1_logf	
gumbel_p1_logfdd	
gumbel_p1_logfddd	
gumbel_p1_loglik	
gumbel_p1_logscores	
gumbel_p1_means	
gumbel_p1_mu1fa	
gumbel_p1_mu2fa	
gumbel_p1_p1fa	447
gumbel_p1_p2fa	448
gumbel_p1_pd	448
gumbel_p1_pdd	449
gumbel_p1_predictordata	449
gumbel_p1_waic	450
gumbel_p2fa	451
gumbel_pd	
gumbel_pdd	
gumbel_waic	

Contents Contents

halfnorm_cp	
halfnorm_f1f	
halfnorm_f1fa	
halfnorm_f2f	
halfnorm_f2fa	
halfnorm_fd	
halfnorm_fdd	
halfnorm_gg	
halfnorm_gg11	
halfnorm_1111	
halfnorm_ldd	
halfnorm_ldda	
halfnorm_lddd	
halfnorm_lddda	
halfnorm_logf	
halfnorm_logfdd	
halfnorm_logfddd	8
halfnorm_loglik	
halfnorm_logscores	9
halfnorm_means	9
halfnorm_mu1f	0
halfnorm_mu2f	0
halfnorm_p1f	1
halfnorm_p2f	1
halfnorm_waic	2
ifvectorthenmatrix	2
invgamma_cp	'3
invgamma_f1f	0
invgamma_f1fa	0
invgamma_f2f	1
invgamma_f2fa	1
invgamma_fd	2
invgamma_fdd	2
invgamma_ldd	3
invgamma_ldda	
invgamma_lddd	
invgamma lddda	
invgamma_lmn	5
invgamma lmnp	
invgamma_logf	
invgamma_logfdd	
invgamma_logfddd	
invgamma_loglik	
invgamma_logscores	
invgamma mu1f	
invgamma_mu2f	
invgamma_p1f	
invgamma_p2f	

invgamma_waic	
invgauss_cp	 492
invgauss_f1f	 499
invgauss_f1fa	 500
invgauss_f2f	
invgauss f2fa	
invgauss_fd	
invgauss_fdd	
invgauss_ldd	
invgauss_ldda	
invgauss_lddd	
invgauss_lddda	
invgauss_lmn	
invgauss_lmnp	
invgauss_logf	
invgauss_logfdd	
invgauss_logfddd	
invgauss_loglik	
invgauss_logscores	
invgauss_neans	
invgauss_mu1f	
invgauss_mu2f	
invgauss_p1f	
invgauss_p11	
invgauss_waic	
ipf2p	
VI I	
jpf3p	
jpf4p	
lnorm_cp	
lnorm_dmgs_cp	
lnorm_dmgs_loglik	
lnorm_dmgs_logscores	
lnorm_dmgs_means	
lnorm_dmgs_waic	
lnorm_f1fa	
lnorm_f2fa	
lnorm_fd	
lnorm_fdd	
lnorm_ldda	
lnorm_lddda	
lnorm_logf	
lnorm_logfdd	
lnorm_logfddd	
lnorm_logscores	
lnorm_mu1fa	
lnorm_mu2fa	
lnorm_p1fa	
lnorm_p1_cp	 535

Contents Contents

lnorm_p1_f1fa	42
lnorm_p1_f1fw	42
lnorm_p1_f2fa	43
lnorm_p1_f2fw	43
lnorm_p1_fd	44
Inorm_p1_fdd	
lnorm_p1_ldda	
Inorm_p1_lddda	
lnorm_p1_logf	
Inorm_p1_logfddd	
lnorm_p1_logscores	
lnorm_p1_mu1fa	
lnorm_p1_mu2fa	
lnorm_p1_p1fa	
lnorm_p1_p2fa	
lnorm_p1_pd	
lnorm_p1_pdd	
lnorm p1 predictordata	
lnorm_p1_waic	
lnorm_p2fa	
lnorm_pd	
lnorm_pdd	
Inorm_waic	
logis_cp	
logis_f1fa	
logis_f2fa	
logis_fd	
logis_fdd	
logis_ldda	
logis_lddda	
logis_logf	
logis_logfdd	
logis logfddd	
logis loglik	
logis_logscores	
logis_nogscores	
<b>c</b> –	
logis_mu2fa	
logis_p1fa	
logis_p1_cp	
logis_p1_f1fa	
logis_p1_f1fw	
logis_p1_f2fa	
logis_p1_f2fw	
logis_p1_fd	
logis_p1_fdd	
logis_p1_ldda	78

logis_p1_lddda	
logis_p1_logf	
logis_p1_logfdd	
logis_p1_logfddd	. 580
logis_p1_loglik	. 581
logis_p1_logscores	. 581
logis_p1_means	. 582
logis_p1_mu1fa	. 582
logis_p1_mu2fa	. 583
logis_p1_p1fa	. 583
logis_p1_p2fa	. 584
logis_p1_pd	. 584
logis_p1_pdd	. 585
logis_p1_predictordata	. 585
logis_p1_waic	. 586
logis_p2fa	. 587
logis_pd	. 587
logis_pdd	. 588
logis_waic	. 588
lst_k3_cp	. 589
lst_k3_f1f	. 596
lst_k3_f1fa	. 597
lst_k3_f2f	. 597
lst_k3_f2fa	. 598
lst_k3_fd	. 598
lst_k3_fdd	. 599
lst_k3_ldd	. 599
lst_k3_ldda	. 600
lst_k3_lddd	. 600
lst_k3_lddda	. 601
lst_k3_lmn	. 601
lst_k3_lmnp	. 602
lst_k3_logf	. 603
lst_k3_logfdd	. 603
lst_k3_logfddd	. 604
lst_k3_loglik	. 604
lst_k3_logscores	. 605
lst_k3_mu1f	. 605
lst_k3_mu2f	. 606
lst_k3_p1f	. 607
lst_k3_p2f	
lst_k3_waic	. 608
lst_p1k3_cp	. 609
lst_p1k3_f1f	. 617
lst_p1k3_f1fa	
lst_p1k3_f1fw	
lst_p1k3_f2f	
lst_p1k3_f2fa	

Contents Contents

manpredictor		
Ist_plk3_Idd         621           Ist_plk3_Idd         622           Ist_plk3_Iddd         623           Ist_plk3_Iddd         623           Ist_plk3_Iddd         624           Ist_plk3_Imn         625           Ist_plk3_Imnp         625           Ist_plk3_logfd         626           Ist_plk3_logfddd         627           Ist_plk3_logfddd         627           Ist_plk3_logis         628           Ist_plk3_mulf         629           Ist_plk3_mulf         629           Ist_plk3_mulf         630           Ist_plk3_plf         631           Ist_plk3_predictordata         632           Ist_plk3_perdictordata         632           Ist_plk3_waic         633           makebetat0         634           makebetat0         634           makewhate         635           makeq         636           maket0         635           makeq         636           make_se         638           make_waic         639           man         639           manlodd         650           manlodd         658           manlugf	lst_p1k3_f2fw	. 620
lst_plk3_ldd         622           lst_plk3_ldda         623           lst_plk3_lddd         623           lst_plk3_lddd         624           lst_plk3_lmn         625           lst_plk3_logf         626           lst_plk3_logfdd         627           lst_plk3_logfddd         627           lst_plk3_logik         628           lst_plk3_logik         628           lst_plk3_logik         629           lst_plk3_logscores         629           lst_plk3_mu1f         629           lst_plk3_plf         631           lst_plk3_plf         631           lst_plk3_predictordata         632           lst_plk3_setics         633           lst_plk3_setics         633           lst_plk3_waic         633           makebetatm         635           makeq         636           maket0         636           maket0         636           maketesid0         637           make_se         638           make_waic         638           make_se         638           make_waic         639           manf         650 <td< td=""><td><math display="block">lst\_p1k3\_fd \ \dots </math></td><td>. 621</td></td<>	$lst\_p1k3\_fd \ \dots $	. 621
Ist_p1k3_lddd         623           Ist_p1k3_lddd         623           Ist_p1k3_lmn         624           Ist_p1k3_lmn         625           Ist_p1k3_logf         626           Ist_p1k3_logfdd         627           Ist_p1k3_logfddd         627           Ist_p1k3_logik         628           Ist_p1k3_logik         628           Ist_p1k3_logscores         629           Ist_p1k3_mulf         629           Ist_p1k3_m2f         630           Ist_p1k3_p2f         631           Ist_p1k3_pedictordata         632           Ist_p1k3_setics         633           Ist_p1k3_waic         633           makebetat0         635           makebetat0         635           makewnuhat0         635           makevaic         637           make_waic         638           make_waic         638           make_waic         638           manke_waic         638           manke_waic         638           manke_maic         638           manke_maic         638           manke_main         650           manlodd         650	$lst\_p1k3\_fdd  \dots \dots$	. 621
Ist_plk3_lddd         623           Ist_plk3_lddda         624           Ist_plk3_lmn         625           Ist_plk3_logf         626           Ist_plk3_logfdd         627           Ist_plk3_logfddd         627           Ist_plk3_loglik         628           Ist_plk3_loglik         628           Ist_plk3_mulf         629           Ist_plk3_mulf         630           Ist_plk3_plf         631           Ist_plk3_pf         631           Ist_plk3_predictordata         632           Ist_plk3_setics         633           Ist_plk3_waic         633           makebetat0         634           makewhuhat0         635           maker         636           maketresid0         637           make_maic         638           make_waic         638           manke_waic         638           manke_wai         639           man1f         649           man2f         650           mandoub         651           manf         651           manlogf         658           manlogf         659           manloglik	$lst\_p1k3\_ldd  \dots \dots$	. 622
Ist_plk3_ldda         624           Ist_plk3_lmn         625           Ist_plk3_lmp         625           Ist_plk3_logf         626           Ist_plk3_logfdd         627           Ist_plk3_logfdd         627           Ist_plk3_logfidd         627           Ist_plk3_logfidd         628           Ist_plk3_logscores         629           Ist_plk3_mulf         629           Ist_plk3_mu2f         630           Ist_plk3_pf         631           Ist_plk3_predictordata         632           Ist_plk3_waic         633           Ist_plk3_waic         633           makebetat0         634           makebetat0         634           makewaic         636           maket0         635           makeq         636           maket0         637           make_waic         638           make_waic         639           man         639           man1f         649           man2f         650           mandudd         651           manf         651           manlogf         658           manlogf         659 <td><math display="block">lst\_p1k3\_ldda \ \dots </math></td> <td>. 623</td>	$lst\_p1k3\_ldda \ \dots $	. 623
Ist_plk3_lmn         625           Ist_plk3_lmp         625           Ist_plk3_logf         626           Ist_plk3_logfdd         627           Ist_plk3_logfidd         627           Ist_plk3_logkik         628           Ist_plk3_logscores         629           Ist_plk3_mulf         629           Ist_plk3_mulf         630           Ist_plk3_plf         631           Ist_plk3_predictordata         632           Ist_plk3_perdictordata         632           Ist_plk3_setics         633           Ist_plk3_waic         633           makebetat0         634           makebetat0         635           makeumuhat0         635           makeq         636           maket0         636           maketersid0         636           make_waic         638           make_waic         638           make_waic         638           manf         649           manlof         650           manlodd         651           manlodd         658           manlunn         659           manloglik         660           manloglik<	lst_p1k3_lddd	. 623
Ist_p1k3_logf         625           Ist_p1k3_logfd         626           Ist_p1k3_logfddd         627           Ist_p1k3_loglik         628           Ist_p1k3_logscores         629           Ist_p1k3_mu1f         629           Ist_p1k3_mu2f         630           Ist_p1k3_p1f         631           Ist_p1k3_p2f         631           Ist_p1k3_sectics         633           Ist_p1k3_waic         633           makebetat0         634           makebetatm         635           makeut         635           makeut         636           maketvsid0         637           make_waic         638           make_waic         639           man         639           man1f         649           man2f         650           mancheckmle         650           man(dd         658           man(ldd         658	lst_p1k3_lddda	. 624
Ist_p1k3_logfdd         626           Ist_p1k3_logfddd         627           Ist_p1k3_logfddd         628           Ist_p1k3_loglik         628           Ist_p1k3_logscores         629           Ist_p1k3_mu1f         629           Ist_p1k3_mu2f         630           Ist_p1k3_p2f         631           Ist_p1k3_predictordata         632           Ist_p1k3_setics         633           Ist_p1k3_waic         633           makebetat0         634           makebetat1         635           makevotata         636           maket0         635           maket0         636           make_waic         637           make_waic         638           make_waic         638           man         639           man1f         649           man0st         650           manddd         658           manldd         658           manlodd         658           manlogf         659           manlogfik         660           manpredictor         661	lst_p1k3_lmn	. 625
Ist_p1k3_logfdd         627           Ist_p1k3_logfddd         627           Ist_p1k3_logfidd         628           Ist_p1k3_logscores         629           Ist_p1k3_mu1f         629           Ist_p1k3_mu2f         630           Ist_p1k3_p1f         631           Ist_p1k3_p2f         631           Ist_p1k3_setics         633           Ist_p1k3_waic         633           makebetat0         634           makebetat0         635           makeq         636           maketc0         636           maketresid0         637           make_waic         638           make_waic         638           make_waic         639           man         639           man1f         649           manboot         650           mancheckmle         650           manddd         658           manlnn         659           manlogf         659           manlogfi         659           manlogfi         659           manlogfin         659           manlogicores         660           manpredictor         661 <td>lst_p1k3_lmnp</td> <td>. 625</td>	lst_p1k3_lmnp	. 625
Ist_p1k3_logfddd         627           lst_p1k3_logisk         628           lst_p1k3_logscores         629           lst_p1k3_mu1f         629           lst_p1k3_mp2f         630           lst_p1k3_p2f         631           lst_p1k3_peacticordata         632           lst_p1k3_setics         633           lst_p1k3_waic         633           makebetat0         634           makebetatm         635           makeq         636           maket0         636           maket0         637           make_cwaic         637           make_maic         638           make_waic         638           make_waic         639           man         639           man1f         649           man2f         650           mandsub         651           manf         651           manfun         659           manlnn         659           manlnn         659           manlogf         659           manlogfik         660           manpredictor         661	lst_p1k3_logf	. 626
Ist_plk3_logsik         628           lst_plk3_logscores         629           lst_plk3_mulf         629           lst_plk3_mu2f         630           lst_plk3_pplf         631           lst_plk3_pzedictordata         632           lst_plk3_setics         633           lst_plk3_waic         633           makebetat0         634           makebetat0         635           makerunhat0         635           maket0         636           maketresid0         637           make_cwaic         637           make_maic         638           make_waic         638           make_waic         639           man         639           man1f         649           man2f         650           manboot         650           mandsub         651           manf         651           manln         659           manlogid         658           manlogif         659           manlogik         660           manlogscores         660           manpredictor         661	lst_p1k3_logfdd	. 627
Ist_plk3_mulf         629           lst_plk3_mulf         630           lst_plk3_plf         631           lst_plk3_p2f         631           lst_plk3_predictordata         632           lst_plk3_setics         633           lst_plk3_waic         633           makebetat0         634           makebetat0         635           makenuhat0         635           makeq         636           maket0         636           maketvesid0         637           make_waic         638           make_se         638           make_se         638           make_waic         639           man1f         649           man2f         650           mandboot         650           mandckmle         650           manddd         658           manlnn         659           manlnn         659           manlogf         659           manlogfik         660           manpredictor         661	lst_p1k3_logfddd	. 627
lst_plk3_mulf         629           lst_plk3_mu2f         630           lst_plk3_plf         631           lst_plk3_p2f         631           lst_plk3_predictordata         632           lst_plk3_setics         633           lst_plk3_waic         633           makebetat0         634           makebetatm         635           makeq         636           maket0         636           maket, cwaic         637           make_se         638           make_se         638           make_se         638           make_waic         639           man         639           man1f         649           man2f         650           manboot         650           mankeckmle         650           manldd         651           manld         658           manlnn         659           manloglik         660           manlogscores         660           manpredictor         661	lst_p1k3_loglik	. 628
lst_plk3_mulf         629           lst_plk3_mu2f         630           lst_plk3_plf         631           lst_plk3_p2f         631           lst_plk3_predictordata         632           lst_plk3_setics         633           lst_plk3_waic         633           makebetat0         634           makebetatm         635           makeq         636           maket0         636           maket, cwaic         637           make_se         638           make_se         638           make_se         638           make_waic         639           man         639           man1f         649           man2f         650           manboot         650           mankeckmle         650           manldd         651           manld         658           manlnn         659           manloglik         660           manlogscores         660           manpredictor         661	_i	
lst_p1k3_mu2f         630           lst_p1k3_p1f         631           lst_p1k3_p2f         631           lst_p1k3_predictordata         632           lst_p1k3_setics         633           lst_p1k3_waic         633           makebetat0         634           makebetatm         635           makeq         636           maket0         636           maket0         636           make_cwaic         637           make_se         638           make_waic         638           make_waic         639           man1f         649           man2f         650           manboot         650           mandsub         651           manf         651           manldd         658           manlun         659           manloglik         660           manlogscores         660           manpredictor         661	_i	
lst_plk3_plf         631           lst_plk3_predictordata         632           lst_plk3_setics         633           lst_plk3_waic         633           makebetat0         634           makebetatm         635           makeq         636           maket0         636           maketresid0         637           make_maic         638           make_waic         638           make_waic         639           man1f         649           man2f         650           manboot         650           mancheckmle         650           manldd         651           manldd         658           manlnn         659           manlogik         659           manlogik         660           manmeans         660           manpredictor         661	<b></b> -	
lst_p1k3_p2f         631           lst_p1k3_predictordata         632           lst_p1k3_setics         633           lst_p1k3_waic         633           makebetat0         634           makebetatm         635           makeq         636           maket0         636           maketvesid0         637           make_cwaic         637           make_se         638           make_waic         638           make_waic         639           man         639           man1f         649           man2f         650           mandoot         650           mandabub         651           manf         651           manldd         658           manllann         659           manlogf         659           manloglik         660           manmeans         660           manpredictor         661	<u>*</u>	
Ist_p1k3_predictordata         632           Ist_p1k3_setics         633           Ist_p1k3_waic         633           makebetatt         634           makebetattm         635           makeq         635           maket0         636           maket0         636           maketresid0         637           make_maic         638           make_se         638           make_waic         639           man         639           man1f         649           man2f         650           mandsub         651           manfd         651           manldd         658           manlldd         658           manlogf         659           manlogfi         659           manlogscores         660           manpredictor         661		
lst_plk3_setics       633         lst_plk3_waic       633         makebetat0       634         makebetatm       635         makemuhat0       635         maket0       636         maketresid0       637         make_ewaic       638         make_se       638         make_waic       639         man       639         man1f       649         man2f       650         mancheckmle       650         mandsub       651         manf       651         manldd       658         manllodf       658         manlogf       659         manloglik       660         manneans       660         manpredictor       661	<b>-1 -1</b>	
Ist_p1k3_waic       633         makebetat0       634         makebetatm       635         makeq       635         maket       636         maketo       636         maketresid0       637         make_waic       638         make_se       638         make_waic       639         man       639         man1f       649         man2f       650         manboot       650         mandsub       651         manf       651         manldd       658         manlldd       658         manlogf       659         manlogfis       659         manloglik       660         manmeans       660         manpredictor       661	<b>-1 -1</b>	
makebetat0       634         makebetatm       635         makemuhat0       635         makeq       636         maket0       636         maketresid0       637         make_maic       638         make_se       638         make_waic       639         man       639         man1f       649         man2f       650         manboot       650         mancheckmle       650         manddd       658         manldd       658         manlodf       658         manlogf       659         manlogfik       660         manneans       660         manpredictor       661	<u> </u>	
makebetatm       635         makeq       636         maket0       636         maketresid0       637         make_cwaic       637         make_maic       638         make_waic       638         man       639         man 1f       649         man2f       650         manboot       650         mancheckmle       650         manldd       651         manldd       658         manllnn       659         manlogf       659         manlogfi       659         manlogfik       660         manneans       660         manpredictor       661	<u>*</u>	
makemuhat0       635         makeq       636         maket0       636         maketresid0       637         make_cwaic       637         make_maic       638         make_se       638         make_waic       639         man       639         man1f       649         man2f       650         manboot       650         mancheckmle       650         mandsub       651         manf       651         manldd       658         manldd       658         manlogf       659         manlogfik       660         manlogscores       660         manpredictor       661		
makeq       636         maket0       636         maketresid0       637         make_cwaic       637         make_maic       638         make_se       638         make_waic       639         man       639         man1f       649         man2f       650         mancheckmle       650         mandsub       651         manf       651         manldd       658         manlunn       659         manlogf       659         manloglik       660         manneans       660         manpredictor       661		
maket0       636         maketresid0       637         make_cwaic       637         make_maic       638         make_se       638         make_waic       639         man       639         man1f       649         man2f       650         manboot       650         mancheckmle       650         mandsub       651         manf       651         manldd       658         manlun       659         manlogf       659         manloglik       660         manneans       660         manpredictor       661		
maketresid0       637         make_cwaic       637         make_maic       638         make_se       638         make_waic       639         man       639         man1f       649         man2f       650         manboot       650         mancheckmle       650         mandsub       651         manf       651         manldd       658         manlldd       658         manlogf       659         manlogfi       659         manloglik       660         manneans       660         manpredictor       661	1	
make_cwaic       637         make_maic       638         make_se       638         make_waic       639         man       639         man1f       649         man2f       650         manboot       650         mancheckmle       650         mandsub       651         manf       651         manldd       658         manlnn       659         manlogf       659         manloglik       660         manneans       660         manpredictor       661		
make_maic       638         make_se       638         make_waic       639         man       639         man1f       649         man2f       650         mancheckmle       650         mancheckmle       650         mandsub       651         manf       651         manldd       658         manlddd       658         manlogd       659         manlogf       659         manloglik       660         manlogscores       660         manpredictor       661		
make_se       638         make_waic       639         man       639         man1f       649         man2f       650         manboot       650         mancheckmle       650         mandsub       651         manf       651         manldd       658         manlddd       658         manlogf       659         manlogfik       659         manloglik       660         manneans       660         manpredictor       661	<del>-</del>	
make_waic       639         man       639         man1f       649         man2f       650         manboot       650         mancheckmle       650         mandsub       651         manf       651         manldd       658         manlddd       658         manlnn       659         manlogf       659         manloglik       660         manneans       660         manpredictor       661	<del>-</del>	
man       639         man1f       649         man2f       650         manboot       650         mancheckmle       650         mandsub       651         manf       651         manldd       658         manlddd       658         manlnn       659         manlogf       659         manloglik       660         manneans       660         manpredictor       661	<del>-</del>	
man1f       649         man2f       650         manboot       650         mancheckmle       650         mandsub       651         manf       651         manldd       658         manlddd       658         manlnn       659         manlogf       659         manloglik       660         manneans       660         manpredictor       661	<del>-</del>	
man2f       650         manboot       650         mancheckmle       650         mandsub       651         manf       651         manldd       658         manlddd       658         manlnn       659         manlogn       659         manlogf       659         manloglik       660         manneans       660         manpredictor       661		
manboot       650         mancheckmle       650         mandsub       651         manf       651         manldd       658         manlddd       658         manlnn       659         manlogn       659         manlogf       659         manloglik       660         manneans       660         manpredictor       661		
mancheckmle       650         mandsub       651         manf       651         manldd       658         manlddd       658         manlnn       659         manlogf       659         manloglik       660         manlogscores       660         manpredictor       661	man2f	. 650
mandsub       651         manf       651         manldd       658         manlddd       658         manlnn       659         manlogf       659         manloglik       660         manlogscores       660         manmeans       660         manpredictor       661		
manf       651         manldd       658         manlddd       658         manlnn       659         manlogf       659         manloglik       660         manlogscores       660         manmeans       660         manpredictor       661	mancheckmle	. 650
manldd       658         manlddd       658         manlnn       659         manlogf       659         manloglik       660         manlogscores       660         manmeans       660         manpredictor       661	mandsub	. 651
manlddd       658         manlnn       659         manlogf       659         manloglik       660         manlogscores       660         manmeans       660         manpredictor       661	manf	. 651
manlnn       659         manlogf       659         manloglik       660         manlogscores       660         manmeans       660         manpredictor       661	manldd	. 658
manInnn       659         manlogf       659         manloglik       660         manlogscores       660         manmeans       660         manpredictor       661	manlddd	. 658
manlogf659manloglik660manlogscores660manmeans660manpredictor661	manlnn	. 659
manloglik660manlogscores660manmeans660manpredictor661	manlnnn	. 659
manloglik660manlogscores660manmeans660manpredictor661	manlogf	. 659
manlogscores	e	
manmeans	<u> </u>	
manpredictor	e	
•		
1114111700101	manyector	

manwaic	. 661
movexiawayfromzero	. 662
ms_flat_1tail	. 662
ms_flat_2tail	. 664
ms_predictors_1tail	. 665
ms_predictors_2tail	. 666
nopdfcdfmsg	. 667
norm_boot	. 668
norm_cp	. 668
norm_dmgs_cp	
norm_dmgs_loglik	. 680
norm_dmgs_logscores	. 680
norm_dmgs_means	. 681
norm_dmgs_waic	
norm_f1fa	. 682
norm_f2fa	. 683
norm_fd	
norm fdd	
norm_lddda	
norm_logfdd	
norm_logscores	
norm_ml_params	
norm_mu1fa	
norm_mu2fa	
norm_p12_boot	
norm_p12_checkmle	
norm_p12_cp	
norm_p12_exampledata	
norm p12 f1fa	
norm p12 f1fw	
norm p12 f2fa	
norm p12 f2fw	
norm p12 fd	
norm_p12_fdd	
norm_p12_ldda	
norm_p12_lddda	
norm_p12_logf	
norm_p12_logfdd	
norm_p12_logfddd	
norm_p12_loglik	
norm_p12_logscores	
norm_p12_mu1fa	
norm_p12_mu2fa	
norm_p12_nid2ia	
norm_p12_p11a	707

norm_p12_pd	. 708
norm_p12_pdd	. 709
norm_p12_predictordata	. 709
norm_p12_setics	. 710
norm p12 waic	. 711
norm plfa	. 712
norm p1 cp	. 712
norm pl flfa	
norm_p1_f1fw	
norm_p1_f2fa	
norm p1 f2fw	
norm p1 fd	
norm_p1_fdd	
norm_p1_ldda	
norm_p1_lddda	
norm_p1_logf	
norm_p1_logfdd	
norm_p1_logfddd	
norm_p1_loglik	
norm_p1_logscores	
norm_p1_mlparams	
norm_p1_mu1fa	
norm_p1_mu2fa	
norm_p1_p1fa	. 729
norm_p1_p2fa	. 730
norm_p1_pd	. 730
norm_pl_pdd	. 731
norm_p1_predictordata	
norm_p2fa	
norm_pd	
norm_pdd	
norm_unbiasedv_params	
norm_waic	
pareto_k2_cp	
pareto k2 f1fa	
pareto_k2_f1fa	
<u> </u>	
pareto_k2_fd	
pareto_k2_fdd	
pareto_k2_ldda	
pareto_k2_lddda	
pareto_k2_logf	
pareto_k2_logfdd	
pareto_k2_logfddd	
pareto_k2_logscores	
pareto_k2_ml_params	. 748
pareto_k2_mu1fa	. 748
pareto_k2_mu2fa	. 749

pareto_k2_p1fa	. 749
pareto_k2_p2fa	. 750
pareto_k2_pd	. 750
pareto_k2_pdd	. 751
pareto_k2_waic	. 752
pareto_p1k2_cp	. 752
pareto_p1k2_f1fa	. 760
pareto_p1k2_f1fw	. 760
pareto_p1k2_f2fa	. 761
pareto_p1k2_f2fw	. 761
pareto_p1k2_fd	. 762
pareto_p1k2_fdd	. 762
pareto_p1k2_ldda	. 763
pareto_p1k2_lddda	. 763
pareto_p1k2_logf	. 764
pareto_p1k2_logfdd	. 764
pareto_p1k2_logfddd	. 765
pareto_p1k2_loglik	. 765
pareto_p1k2_logscores	. 766
pareto_p1k2_means	. 766
pareto_p1k2_mu1fa	. 767
pareto_p1k2_mu2fa	. 768
pareto_p1k2_p1fa	. 768
pareto_p1k2_p2fa	. 769
pareto_p1k2_pd	. 769
pareto_p1k2_pdd	
pareto_p1k2_predictordata	. 770
pareto_p1k2_waic	. 771
pcauchy_p1	. 772
pexp_p1	. 772
pfrechet_p2k1	. 773
pgev_p1	
pgev_p12	
pgev_p123	. 774
pgev_p1k3	. 775
pgev_p1n	
pgumbel_p1	
plnorm_p1	. 777
plogis_p1	. 777
plst_p1k3	. 778
pnorm_p1	. 778
pnorm_p12	. 779
pnorm_p1_formula	. 779
ppareto_p1k2	. 780
punif_formula	. 780
pweibull_p2	. 781
qcauchy_p1	. 781
qexp_p1	. 782

Contents Contents

qfrechet_p2k1	 	 			 						 . 782
qgamma_k1_ppm	 	 			 						 . 783
qgamma_ppm	 	 			 						 . 784
ggev_k12_ppm	 	 			 						 . 785
ggev_mpd_ppm	 	 			 						 787
gev_p1											
gev_p12											
gev_p123											
gev_p1k3											
gev_p1n3											
gev_p1 gev_p1_ppm											
gev_pr_ppm											
gev_ppm											
gumbel_p1											
norm_p1											
logis_p1											
lst_p1k3											
norm_p1											
norm_p12											
norm_p1_formula .	 	 			 						 799
tt_ppm	 	 			 						 . 799
pareto_p1k2	 	 			 						 801
ınif_formula	 	 			 						 801
weibull_p2	 	 			 						 802
ltest	 	 			 						 802
ltest2	 	 			 						 806
ltest2 cases											
ltest2_makeep											
ltest2_plot											
eltest2_predict											
eltest2_simulate											
eltest_makeep											
eltest_makemaxep											
-											
eltest_params											
eltest_predict											
eltest_simulate											
gev_minmax											
gev_p123_minmax .											
gev_p12_minmax .											
gev_p1n_minmax .											
gev_p1_minmax	 	 			 						 820
gpd_k1_minmax	 	 			 						 821
hp_dmgs_cpmethod	 	 			 						 821
ust_pumethod											
estppm_plot											
unif_cp											
weibull_cp											
weibull_cp											

weibull_f2fa
weibull_fd
weibull_fdd
weibull_ldda
weibull_lddda
weibull_logf
weibull_logfdd
weibull_logfddd
weibull_loglik
weibull_logscores
weibull_means
$weibull\_mu1fa  .  .  .  .  .  .  .  .  .  $
weibull_mu2fa
weibull_p1fa
weibull_p2fa
weibull_p2_cp
$weibull\_p2\_f1fa \dots \dots$
$weibull\_p2\_f1fw  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  $
$weibull\_p2\_f2fa \dots \dots$
$weibull\_p2\_f2fw \qquad \dots \qquad $
$weibull\_p2\_fd \dots \dots$
$weibull\_p2\_fdd \dots $
$weibull\_p2\_ldda  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  $
$weibull\_p2\_lddda \ \ldots \ \ldots \ \ldots \ 854$
$weibull\_p2\_logf \dots \dots$
$weibull\_p2\_logfdd \dots $
$weibull\_p2\_logfddd\dots$
$weibull\_p2\_loglik \ldots \ldots \\ 856$
weibull_p2_logscores
weibull_p2_means
$weibull\_p2\_mu1fa  .  .  .  .  .  .  .  .  .  $
$weibull\_p2\_mu2fa  .  .  .  .  .  .  .  .  .  $
$weibull\_p2\_p1fa  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  $
$weibull\_p2\_p2fa \qquad \dots \qquad \dots \qquad 860$
$weibull\_p2\_pd  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  \dots  $
$weibull\_p2\_pdd \ \dots \$
$weibull\_p2\_predictordata \ . \ . \ . \ . \ . \ . \ . \ . \ . \ $
weibull_p2_waic
weibull_pd
weibull_pdd
weibull_waic
865

Index

 ${\tt adhoc\_dmgs\_cpmethod}$ 

Generates a comment about the method

# Description

Generates a comment about the method

# Usage

```
adhoc_dmgs_cpmethod()
```

#### Value

String

analytic\_cpmethod

Generates a comment about the method

# Description

Generates a comment about the method

# Usage

```
analytic_cpmethod()
```

# Value

String

 ${\tt bayesian\_dq\_4terms\_v1} \quad \textit{Evaluate DMGS equation 3.3}$ 

# Description

Evaluate DMGS equation 3.3

# Usage

```
bayesian_dq_4terms_v1(lddi, lddd, mu1, pidopi1, pidopi2, mu2, dim)
```

calc\_revert2ml 25

# Arguments

lddi inverse of second derivative of observed log-likelihood

1ddd third derivative of observed log-likelihood

mu1 DMGS mu1 vector

pidopi1 first part of the prior term

pidopi2 second part of the prior term

mu2 DMGS mu2 matrix
dim number of parameters

#### Value

Vector

calc\_revert2ml determine revert2ml or not

# Description

determine revert2ml or not

# Usage

```
calc_revert2ml(v5h, v6h, t3)
```

# Arguments

v5h fifth parameter v6h sixth parameter

t3 a vector of predictors for the shape

#### Value

Logical

cauchy\_cp

Cauchy Distribution Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qcauchy_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    d1 = 0.01,
    fd2 = 0.01,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    rust = FALSE,
    nrust = 1e+05,
    debug = FALSE,
    aderivs = TRUE
)
```

```
n,
 х,
 d1 = 0.01,
  fd2 = 0.01,
 rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
dcauchy_cp(
 х,
 y = x,
 d1 = 0.01,
 fd2 = 0.01,
  rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
pcauchy_cp(
 х,
 y = x,
 d1 = 0.01,
 fd2 = 0.01,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
tcauchy_cp(n, x, d1 = 0.01, fd2 = 0.01, debug = FALSE)
```

# **Arguments** ×

X	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)

dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
y	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Cauchy distribution has probability density function

$$f(x; \mu, \sigma) = \frac{1}{\pi \sigma} \left( 1 + \left( \frac{x - \mu}{\sigma} \right)^2 \right)^{-1}$$

where x is the random variable and  $\mu, \sigma > 0$  are the parameters.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (non-homogeneous models)**

This model is not homogeneous, i.e. it does not have a transitive transformation group, and so there is no right Haar prior and no method for generating exactly reliable predictions. The cp outputs are generated using a prior that has been shown in tests to give reasonable reliability. See Jewson et al. (2025a) for discussion of the prior and test results. For non-homogeneous models, reliability is generally poor for small sample sizes (<20), but is still much better than maximum likelihood. For small sample sizes, it is advisable to check the level of reliability using the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),

- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (1st\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
#
# example 1
x=fitdistcp::d042cauchy_example_data_v1
p=c(1:9)/10
q=qlogis_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qcauchy_cp)",
main="Cauchy: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

cauchy\_f1f 33

caucily i i i	cauc	hv	f1f
---------------	------	----	-----

DMGS equation 3.3, f1 term

# Description

DMGS equation 3.3, f1 term

# Usage

```
cauchy_f1f(y, v1, d1, v2, fd2)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

# Value

Matrix

cauchy_	f1	fa

The first derivative of the density

# Description

The first derivative of the density

# Usage

```
cauchy_f1fa(x, v1, v2)
```

# Arguments

x a vector of training dat	a values
----------------------------	----------

v1 first parameter v2 second parameter

#### Value

Vector

34 cauchy\_f2fa

00110	h.,	£24
cauc	IIV .	$\Gamma \angle \Gamma$

 $DMGS\ equation\ 3.3, f2\ term$ 

# Description

DMGS equation 3.3, f2 term

# Usage

```
cauchy_f2f(y, v1, d1, v2, fd2)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

# Value

3d array

cauchy_	f2fa

The second derivative of the density

# Description

The second derivative of the density

# Usage

```
cauchy_f2fa(x, v1, v2)
```

# Arguments

v1 first parameter v2 second parameter

#### Value

Matrix

cauchy\_fd 35

cauchy_fd	First derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
cauchy_fd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Vector

cauchy_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
cauchy_fdd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Matrix

36 cauchy\_ldda

cauchy_	ヿゐゐ
cauchy	- 1 aa

Second derivative matrix of the normalized log-likelihood

# Description

Second derivative matrix of the normalized log-likelihood

# Usage

```
cauchy_ldd(x, v1, d1, v2, fd2)
```

# Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-

eter

# Value

Square scalar matrix

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

# Usage

```
cauchy_ldda(x, v1, v2)
```

# **Arguments**

x a vector of training data values
------------------------------------

v1 first parameter v2 second parameter

#### Value

Matrix

cauchy\_lddd 37

cauchy_lddd	Third derivative tensor of the normalized log-likelihood

## Description

Third derivative tensor of the normalized log-likelihood

# Usage

```
cauchy_lddd(x, v1, d1, v2, fd2)
```

# Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

### Value

Cubic scalar array

cauchy_lddda	ive of the normalized log-likelihood
--------------	--------------------------------------

# Description

The third derivative of the normalized log-likelihood

# Usage

```
cauchy_lddda(x, v1, v2)
```

# Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter

## Value

3d array

38 cauchy\_lmnp

cauchy_lmn	One component of the second derivative of the normalized log-likelihood
------------	---

# Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
cauchy_lmn(x, v1, d1, v2, fd2, mm, nn)
```

# Arguments

x	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

### Value

Scalar value

cauchy_lmnp	One component of the third derivative of the normalized log-likelihood

# Description

One component of the third derivative of the normalized log-likelihood

# Usage

```
cauchy_lmnp(x, v1, d1, v2, fd2, mm, nn, rr)
```

cauchy\_logf 39

# Arguments

x	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

## Value

Scalar value

cauchy_logf $Logf for RUST$
-----------------------------

# Description

Logf for RUST

## Usage

```
cauchy_logf(params, x)
```

# Arguments

params model parameters for calculating logf
x a vector of training data values

# Value

Scalar value.

40 cauchy\_logfddd

cauchy_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
cauchy_logfdd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

## Value

Matrix

cauchy_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
cauchy_logfddd(x, v1, v2)
```

## **Arguments**

X	a vector of training data values
Λ	a vector or training data variets

v1 first parameter v2 second parameter

### Value

3d array

cauchy\_loglik 41

cauchy_loglik	log-likelihood function	

# Description

log-likelihood function

# Usage

```
cauchy_loglik(vv, x)
```

## Arguments

vv parameters

x a vector of training data values

### Value

Scalar

cauchy_logscores Log scores for MLE and RHP out	predictions calculated using leave-one-
---	---

# Description

Log scores for MLE and RHP predictions calculated using leave-one-out

## Usage

```
cauchy_logscores(logscores, x, d1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

# Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
х	a vector of training data values
d1	the delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

Two scalars

42 cauchy\_mu2f

cauchy_mu1f DM	IGS equation 3.3, mu1 term
----------------	----------------------------

# Description

DMGS equation 3.3, mu1 term

## Usage

```
cauchy_mu1f(alpha, v1, d1, v2, fd2)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

## Value

Matrix

cauchy_mu2f DMGS equation 3.3, mu2 term
---

# Description

DMGS equation 3.3, mu2 term

# Usage

```
cauchy_mu2f(alpha, v1, d1, v2, fd2)
```

### **Arguments**

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

cauchy\_p1f 43

### Value

3d array

cauchy\_p1f

DMGS equation 3.3, p1 term

## Description

DMGS equation 3.3, p1 term

# Usage

```
cauchy_p1f(y, v1, d1, v2, fd2)
```

### **Arguments**

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

Matrix

cauchy_p1_cp	Cauchy Distribution with a Predictor, Predictions Based on a Cali-
	brating Prior

### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

• q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.

- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qcauchy_p1_cp(
  Х,
  t,
  t0 = NA,
 n0 = NA,
 p = seq(0.1, 0.9, 0.1),
 d1 = 0.01,
  d2 = 0.01,
  fd3 = 0.01,
 means = FALSE,
 waicscores = FALSE,
  logscores = FALSE,
  dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 predictordata = TRUE,
  centering = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
rcauchy_p1_cp(
  n,
  Х,
  t,
  t0 = NA,
 n0 = NA,
 d1 = 0.01,
  d2 = 0.01,
  fd3 = 0.01,
  rust = FALSE,
 mlcp = TRUE,
  debug = FALSE,
  aderivs = TRUE
```

```
cauchy_p1_cp 45
```

```
dcauchy_p1_cp(
 Х,
 t,
 t0 = NA,
 n0 = NA,
 y = x,
 d1 = 0.01,
 d2 = 0.01,
 fd3 = 0.01,
  rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
pcauchy_p1_cp(
 Х,
 t,
 t0 = NA,
 n0 = NA,
 y = x,
 d1 = 0.01,
 d2 = 0.01,
 fd3 = 0.01,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
tcauchy_p1_cp(n, x, t, d1 = 0.01, d2 = 0.01, fd3 = 0.01, debug = FALSE)
```

## Arguments

X	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
р	a vector of probabilities at which to generate predictive quantiles
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter
d2	if aderivs=FALSE, the delta used for numerical derivatives with respect to the second parameter

fd3	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the third parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
predictordata	logical that indicates whether predictordata should be calculated
centering	logical that indicates whether the predictor should be centered
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- ullet adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

• ml\_params: maximum likelihood estimates for the parameters.

- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Cauchy distribution with a predictor has probability density function

$$f(x; a, b, \sigma) = \frac{1}{\pi \sigma} \left( 1 + \left( \frac{x - \mu(a, b)}{\sigma} \right)^2 \right)^{-1}$$

where x is the random variable,  $\mu = a + bt$  is the location parameter as a function of parameters a, b, and  $\sigma > 0$  is the scale parameter.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

#### If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

#### If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),

- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

cauchy\_p1\_f1f 51

### **Examples**

```
# # example 1
x=fitdistcp::d064cauchy_p1_example_data_v1_x
tt=fitdistcp::d064cauchy_p1_example_data_v1_t
p=c(1:9)/10
n0=10
q=qcauchy_p1_cp(x,tt,n0=n0,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qcauchy_p1_cp)",
main="Cauchy w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

cauchy\_p1\_f1f

DMGS equation 2.1, f1 term

## Description

DMGS equation 2.1, f1 term

### Usage

```
cauchy_p1_f1f(y, t0, v1, d1, v2, d2, v3, fd3)
```

### **Arguments**

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

### Value

Matrix

52 cauchy\_p1\_f1fw

cauchy\_p1\_f1fa

The first derivative of the density for DMGS

### **Description**

The first derivative of the density for DMGS

### Usage

```
cauchy_p1_f1fa(x, t0, v1, v2, v3)
```

### **Arguments**

x a vector of training data values

t0 a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameterv2 second parameterv3 third parameter

#### Value

Vector

cauchy\_p1\_f1fw

The first derivative of the density for WAIC

### Description

The first derivative of the density for WAIC

### Usage

```
cauchy_p1_f1fw(x, t, v1, v2, v3)
```

## Arguments

x a vector of training data valuest a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

### Value

Vector

cauchy\_p1\_f2f 53

cauchy_p1_f2f	cauchy	n1	f2f	
---------------	--------	----	-----	--

DMGS equation 2.1, f2 term

## **Description**

DMGS equation 2.1, f2 term

## Usage

```
cauchy_p1_f2f(y, t0, v1, d1, v2, d2, v3, fd3)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

## Value

3d array

cauchy	р1	f2fa

The second derivative of the density for DMGS

## Description

The second derivative of the density for DMGS

# Usage

```
cauchy_p1_f2fa(x, t0, v1, v2, v3)
```

## **Arguments**

X	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
v2	second parameter
v3	third parameter

54 cauchy\_p1\_fd

### Value

Matrix

cauchy\_p1\_f2fw

The second derivative of the density for WAIC

### **Description**

The second derivative of the density for WAIC

### Usage

```
cauchy_p1_f2fw(x, t, v1, v2, v3)
```

### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Matrix

cauchy\_p1\_fd First derivative of the den

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### **Description**

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
cauchy_p1_fd(x, t, v1, v2, v3)
```

## **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

cauchy\_p1\_fdd 55

## Value

Vector

cauchy_p1_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
cauchy_p1_fdd(x, t, v1, v2, v3)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Matrix

Matrix

cauc	hy_	p1.	_ldd
------	-----	-----	------

Second derivative matrix of the normalized log-likelihood

# Description

Second derivative matrix of the normalized log-likelihood

# Usage

```
cauchy_p1_ldd(x, t, v1, d1, v2, d2, v3, fd3)
```

56 cauchy\_p1\_ldda

# Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

## Value

Square scalar matrix

cauchy_p1_ldda Th	ne second derivative of the normalized log-likelihood
-------------------	---

# Description

The second derivative of the normalized log-likelihood

# Usage

```
cauchy_p1_ldda(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

# Value

Matrix

cauchy\_p1\_lddd 57

cauchy_p1_lddd	Third derivative tensor of the normalized log-likelihood
----------------	--

## Description

Third derivative tensor of the normalized log-likelihood

# Usage

```
cauchy_p1_1ddd(x, t, v1, d1, v2, d2, v3, fd3)
```

# Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

### Value

Cubic scalar array

cauchy_p1_lddda	The third derivative of the normalized log-likelihood
-----------------	---

## Description

The third derivative of the normalized log-likelihood

# Usage

```
cauchy_p1_1ddda(x, t, v1, v2, v3)
```

# **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

58 cauchy\_p1\_lmn

## Value

3d array

• •	One component of the second likelihood	derivative of the normalized log-
-----	--	-----------------------------------

# Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
cauchy_p1_lmn(x, t, v1, d1, v2, d2, v3, fd3, mm, nn)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

## Value

Scalar value

cauchy\_p1\_lmnp 59

cauchy_p1_lmnp	One component of the second derivative of the normalized log-likelihood
----------------	---

# Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
cauchy_p1_lmnp(x, t, v1, d1, v2, d2, v3, fd3, mm, nn, rr)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

### Value

Scalar value

cauchy_p1_logf	Logf for RUST
----------------	---------------

# Description

Logf for RUST

# Usage

```
cauchy_p1_logf(params, x, t)
```

60 cauchy\_p1\_logfdd

# Arguments

params	model parameters for calculating logf
x	a vector of training data values
t.	a vector or matrix of predictors

### Value

Scalar value.

cauchy_p1_logfdd Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol	cauchy_p1_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
--	------------------	--

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
cauchy_p1_logfdd(x, t, v1, v2, v3)
```

# Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Matrix

cauchy\_p1\_logfddd 61

cauchy_p1_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
cauchy_p1_logfddd(x, t, v1, v2, v3)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
	_

v1 first parameterv2 second parameterv3 third parameter

### Value

3d array

cauchy\_p1\_loglik

Cauchy-with-p1 observed log-likelihood function

## Description

Cauchy-with-p1 observed log-likelihood function

# Usage

```
cauchy_p1_loglik(vv, x, t)
```

## Arguments

vv para

x a vector of training data valuest a vector or matrix of predictors

#### Value

Scalar

62 cauchy\_p1\_means

3-1 - 6	Log scores for MLE and RHP predictions calculated using leave-one- out
	oui

# Description

Log scores for MLE and RHP predictions calculated using leave-one-out

## Usage

```
cauchy_p1_logscores(logscores, x, t, d1, d2, fd3, aderivs = TRUE)
```

## Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
x	a vector of training data values
t	a vector or matrix of predictors
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

Two scalars

_p1_means	Cauchy distribution: RHP mean

# Description

Cauchy distribution: RHP mean

# Usage

```
cauchy_p1_means(t0, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 2)
```

cauchy\_p1\_mu1f 63

### **Arguments**

t0 a single value of the predictor (specify either t0 or n0 but not both)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

### Value

Two scalars

cauchy\_p1\_mu1f

DMGS equation 3.3, mu1 term

## Description

DMGS equation 3.3, mu1 term

### Usage

```
cauchy_p1_mu1f(alpha, t0, v1, d1, v2, d2, v3, fd3)
```

### **Arguments**

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

### Value

Matrix

cauchy\_p1\_p1f

DMGS equation 3.3, mu2 term

# Description

DMGS equation 3.3, mu2 term

# Usage

```
cauchy_p1_mu2f(alpha, t0, v1, d1, v2, d2, v3, fd3)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

# Value

3d array

cauchy_	ք1	p1f
caaciiy_	- 12	_P ' '

DMGS equation 2.1, p1 term

# Description

```
DMGS equation 2.1, p1 term
```

# Usage

```
cauchy_p1_p1f(y, t0, v1, d1, v2, d2, v3, fd3)
```

cauchy\_p1\_p2f 65

# Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

## Value

Matrix

o1_p2f DMGS equation 2.1, p2 term
-----------------------------------

# Description

DMGS equation 2.1, p2 term

# Usage

```
cauchy_p1_p2f(y, t0, v1, d1, v2, d2, v3, fd3)
```

# Arguments

у	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

## Value

3d array

66 cauchy\_p1\_waic

```
cauchy_p1_predictordata
```

Predicted Parameter and Generalized Residuals

### **Description**

Predicted Parameter and Generalized Residuals

## Usage

```
cauchy_p1_predictordata(predictordata, x, t, t0, params)
```

## Arguments

predictordata logical that indicates whether to calculate and return predictordata x a vector of training data values t a vector or matrix of predictors to a single value of the predictor (specify either to or no but not both) params model parameters for calculating logf

### Value

Two vectors

cauchy\_p1\_waic Waic

# Description

Waic

## Usage

```
cauchy_p1_waic(
  waicscores,
  X,
  t,
  v1hat,
  d1,
  v2hat,
  d2,
  v3hat,
  fd3,
  lddi,
```

cauchy\_p2f 67

```
lddd,
  lambdad,
  aderivs
)
```

## Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
t	a vector or matrix of predictors
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3hat	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter

lddi inverse observed information matrix lddd third derivative of log-likelihood lambdad derivative of the log prior

logical for whether to use analytic derivatives (instead of numerical) aderivs

# Value

Two numeric values.

cauchy_p2f DMGS equation 3.3, p2 term
---------------------------------------

## Description

DMGS equation 3.3, p2 term

## Usage

```
cauchy_p2f(y, v1, d1, v2, fd2)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

68 cauchy\_waic

## Value

3d array

|--|--|--|

# Description

Waic

# Usage

```
cauchy_waic(waicscores, x, v1hat, d1, v2hat, fd2, lddi, lddd, lambdad, aderivs)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
x	a vector of training data values
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior
aderivs	logical for whether to use analytic derivatives (instead of numerical)

# Value

Two numeric values.

```
crhpflat_dmgs_cpmethod
```

Generates a comment about the method

## Description

Generates a comment about the method

### Usage

```
crhpflat_dmgs_cpmethod()
```

### Value

String

```
d010exp_example_data_v1
```

This is data to be included in my package

## Description

This is data to be included in my package

```
d011pareto_k2_example_data_v1
```

This is data to be included in my package

# Description

This is data to be included in my package

```
d020halfnorm_example_data_v1
```

This is data to be included in my package

# Description

d025unif\_example\_data\_v1

This is data to be included in my package

## Description

This is data to be included in my package

d030norm\_example\_data\_v1

This is data to be included in my package

## Description

This is data to be included in my package

 ${\tt d031norm\_dmgs\_example\_data\_v1}$ 

This is data to be included in my package

### **Description**

This is data to be included in my package

d032gnorm\_k3\_example\_data\_v1

This is data to be included in my package

# Description

This is data to be included in my package

d035lnorm\_example\_data\_v1

This is data to be included in my package

### **Description**

d036lnorm\_dmgs\_example\_data\_v1

This is data to be included in my package

### Description

This is data to be included in my package

d040logis\_example\_data\_v1

This is data to be included in my package

## Description

This is data to be included in my package

d041lst\_k3\_example\_data\_v1

This is data to be included in my package

## Description

This is data to be included in my package

d042cauchy\_example\_data\_v1

This is data to be included in my package

# Description

This is data to be included in my package

d050gumbel\_example\_data\_v1

This is data to be included in my package

### **Description**

d051frechet\_k1\_example\_data\_v1

This is data to be included in my package

## Description

This is data to be included in my package

d052weibull\_example\_data\_v1

This is data to be included in my package

## Description

This is data to be included in my package

 $d053gev_k3_example_data_v1$ 

This is data to be included in my package

### **Description**

This is data to be included in my package

d055exp\_p1\_example\_data\_v1\_t

This is data to be included in my package

# Description

This is data to be included in my package

 ${\tt d055exp\_p1\_example\_data\_v1\_x}$ 

This is data to be included in my package

# Description

d056pareto\_p1k2\_example\_data\_v1\_t

This is data to be included in my package

### **Description**

This is data to be included in my package

d056pareto\_p1k2\_example\_data\_v1\_x

This is data to be included in my package

## Description

This is data to be included in my package

d060norm\_p1\_example\_data\_v1\_t

This is data to be included in my package

## Description

This is data to be included in my package

d060norm\_p1\_example\_data\_v1\_x

This is data to be included in my package

## Description

This is data to be included in my package

d061lnorm\_p1\_example\_data\_v1\_t

This is data to be included in my package

### **Description**

```
d061lnorm_p1_example_data_v1_x
```

### **Description**

This is data to be included in my package

```
d062logis_p1_example_data_v1_t
```

This is data to be included in my package

## Description

This is data to be included in my package

```
d062logis_p1_example_data_v1_x
```

This is data to be included in my package

## Description

This is data to be included in my package

```
d063lst_p1k3_example_data_v1_t
```

This is data to be included in my package

## Description

This is data to be included in my package

```
d0631st\_p1k3\_example\_data\_v1\_x
```

This is data to be included in my package

# Description

```
d064cauchy_p1_example_data_v1_t
```

### **Description**

This is data to be included in my package

```
d064cauchy_p1_example_data_v1_x
```

This is data to be included in my package

### **Description**

This is data to be included in my package

```
d070gumbel_p1_example_data_v1_t
```

This is data to be included in my package

## Description

This is data to be included in my package

```
d070gumbel_p1_example_data_v1_x
```

This is data to be included in my package

## Description

This is data to be included in my package

```
\verb|d071frechet_p2k1_example_data_v1_t|\\
```

This is data to be included in my package

### **Description**

```
d071frechet_p2k1_example_data_v1_x
```

## Description

This is data to be included in my package

```
d072weibull_p1_example_data_v1_t
```

This is data to be included in my package

## Description

This is data to be included in my package

```
\verb|d072| we ibull_p1_example_data_v1_x|
```

This is data to be included in my package

### **Description**

This is data to be included in my package

```
d073weibull_p2_example_data_v1_t
```

This is data to be included in my package

## Description

This is data to be included in my package

```
\verb|d073weibull_p2_example_data_v1_x|\\
```

This is data to be included in my package

# Description

d074gev\_p1k3\_example\_data\_v1\_t

This is data to be included in my package

### **Description**

This is data to be included in my package

d074gev\_p1k3\_example\_data\_v1\_x

This is data to be included in my package

## Description

This is data to be included in my package

d080norm\_p12\_example\_data\_v1\_t1

This is data to be included in my package

## Description

This is data to be included in my package

d080norm\_p12\_example\_data\_v1\_t2

This is data to be included in my package

## Description

This is data to be included in my package

d080norm\_p12\_example\_data\_v1\_x

This is data to be included in my package

### **Description**

d081lst\_p12k3\_example\_data\_v1\_t1

This is data to be included in my package

### **Description**

This is data to be included in my package

d081lst\_p12k3\_example\_data\_v1\_t2

This is data to be included in my package

## Description

This is data to be included in my package

d081lst\_p12k3\_example\_data\_v1\_x

This is data to be included in my package

## Description

This is data to be included in my package

d082weibull\_p12\_example\_data\_v1\_t1

This is data to be included in my package

## Description

This is data to be included in my package

d082weibull\_p12\_example\_data\_v1\_t2

This is data to be included in my package

### **Description**

d082weibull\_p12\_example\_data\_v1\_x

This is data to be included in my package

### **Description**

This is data to be included in my package

d100gamma\_example\_data\_v1

This is data to be included in my package

## Description

This is data to be included in my package

d101invgamma\_example\_data\_v1

This is data to be included in my package

## Description

This is data to be included in my package

d102invgauss\_example\_data\_v1

This is data to be included in my package

## Description

This is data to be included in my package

d105burr\_example\_data\_v1

This is data to be included in my package

### **Description**

d110gev\_example\_data\_v1

This is data to be included in my package

### **Description**

This is data to be included in my package

d120gpd\_k1\_example\_data\_v1

This is data to be included in my package

## Description

This is data to be included in my package

d150gev\_p1\_example\_data\_v1\_t

This is data to be included in my package

## Description

This is data to be included in my package

d150gev\_p1\_example\_data\_v1\_x

This is data to be included in my package

## Description

This is data to be included in my package

d151gev\_p12\_example\_data\_v1\_t

This is data to be included in my package

# Description

```
d151gev_p12_example_data_v1_x
```

## Description

This is data to be included in my package

```
d152gev_p123_example_data_v1_t
```

This is data to be included in my package

## **Description**

This is data to be included in my package

```
d152gev_p123_example_data_v1_x
```

This is data to be included in my package

### **Description**

This is data to be included in my package

dcauchysub

Densities from MLE and RHP

### **Description**

Densities from MLE and RHP

### Usage

```
dcauchysub(x, y, d1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

## Arguments

X	a vector of training data values
у	a vector of values at which to calculate the density and distribution functions
d1	the delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

82 dcauchy\_p1sub

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

		_
dca	uchy.	pΊ

Cauchy-with-p1 density function

## Description

Cauchy-with-p1 density function

## Usage

```
dcauchy_p1(x, t0, ymn, slope, scale, log = FALSE)
```

### **Arguments**

X	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
ymn	the location parameter of the function of the predictor
,	

slope the slope of the function of the predictor
scale the scale parameter of the distribution
log logical for the density evaluation

### Value

Vector

dcauchy	/ n1suh

Densities from MLE and RHP

## Description

Densities from MLE and RHP

### Usage

```
dcauchy_p1sub(x, t, y, t0, d1, d2, fd3, aderivs = TRUE)
```

deriv\_copyfdd 83

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
у	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

deriv_copyfdd	Extract the results from derivatives and put them into f2	

# Description

Extract the results from derivatives and put them into f2

# Usage

```
deriv_copyfdd(temp1, nx, dim)
```

## Arguments

temp1 output from derivative calculations

 $\begin{array}{ll} \text{nx} & \text{number of x values} \\ \\ \text{dim} & \text{number of parameters} \end{array}$ 

## Value

3d array

84 deriv\_copyldd

deriv\_copyld2

Extract the results from derivatives and put them into ldd

## Description

Extract the results from derivatives and put them into ldd

### Usage

```
deriv_copyld2(temp1, nx, dim)
```

### **Arguments**

temp1 output from derivative calculations

nx number of x values
dim number of parameters

### Value

3d array

deriv\_copyldd

Extract the results from derivatives and put them into ldd

## Description

Extract the results from derivatives and put them into ldd

## Usage

```
deriv_copyldd(temp1, nx, dim)
```

## Arguments

temp1 output from derivative calculations

nx number of x values dim number of parameters

### Value

Matrix

deriv\_copylddd 85

deriv\_copylddd

Extract the results from derivatives and put them into lddd

### **Description**

Extract the results from derivatives and put them into lddd

## Usage

```
deriv_copylddd(temp1, nx, dim)
```

### Arguments

temp1 output from derivative calculations

nx number of x values
dim number of parameters

#### Value

3d array

dexpsub

Densities from MLE and RHP

### **Description**

Densities from MLE and RHP

## Usage

```
dexpsub(x, y)
```

## Arguments

x a vector of training data values

y a vector of values at which to calculate the density and distribution functions

#### Value

86 dexp\_p1sub

1		-
dex	n	n I
uca	$\nu_{-}$	$\boldsymbol{\nu}$ ,

Exponential-with-p1 density function

#### **Description**

Exponential-with-p1 density function

## Usage

```
dexp_p1(x, t0, ymn, slope, log = FALSE)
```

### **Arguments**

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor

log logical for the density evaluation

#### Value

Vector

dexp\_p1sub

Densities from MLE and RHP

### **Description**

Densities from MLE and RHP

## Usage

```
dexp_p1sub(x, t, y, t0)
```

### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors

y a vector of values at which to calculate the density and distribution functions

to a single value of the predictor (specify either to or no but not both)

#### Value

dfrechetsub 87

10			
αt	rec	nei	tsub

Densities from MLE and RHP

### **Description**

Densities from MLE and RHP

### Usage

```
dfrechetsub(x, y, kloc)
```

### **Arguments**

x a vector of training data values

y a vector of values at which to calculate the density and distribution functions

kloc the known location parameter

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dfrechet\_p2k1

Frechet\_k1-with-p2 density function

### **Description**

Frechet\_k1-with-p2 density function

## Usage

```
dfrechet_p2k1(x, t0, ymn, slope, lambda, log = FALSE, kloc)
```

#### **Arguments**

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor the lambda parameter of the distribution

log logical for the density evaluation kloc the known location parameter

#### Value

Vector

88 dgammasub

dfrechet_p2k1sub	Densities from MLE and RHP
a coco_pca.s	2 c. s. t. c. j. c. t. 1.122 c. t. c. 1.111

# Description

Densities from MLE and RHP

## Usage

```
dfrechet_p2k1sub(x, t, y, t0, kloc)
```

## Arguments

x	a vector of training data values
t	a vector or matrix of predictors
У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
kloc	the known location parameter

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dgammasub Densities from MLE and RHP
--------------------------------------

# Description

Densities from MLE and RHP

# Usage

```
dgammasub(x, y, fd1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

# Arguments

X	a vector of training data values
У	a vector of values at which to calculate the density and distribution functions
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

dgevsub 89

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dgevsub

Densities for 5 predictions

### **Description**

Densities for 5 predictions

#### Usage

```
dgevsub(x, y, ics, minxi, maxxi)
```

#### **Arguments**

x a vector of training data values

y a vector of values at which to calculate the density and distribution functions

ics initial conditions for the maximum likelihood search

minxi minimum value of shape parameter xi maxxi maximum value of shape parameter xi

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dgev\_k3sub

Densities from MLE and RHP

## Description

Densities from MLE and RHP

### Usage

```
dgev_k3sub(x, y, kshape)
```

### **Arguments**

x a vector of training data values

y a vector of values at which to calculate the density and distribution functions

kshape the known shape parameter

#### Value

90 dgev\_p12

dgev\_p1

GEVD-with-p1: Density function

# Description

GEVD-with-p1: Density function

## Usage

```
dgev_p1(x, t0, ymn, slope, sigma, xi, log = FALSE)
```

## Arguments

X	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma	the sigma parameter of the distribution
xi	the shape parameter of the distribution
log	logical for the density evaluation

### Value

Vector

dgev\_p12

GEVD-with-p1: Density function

# Description

GEVD-with-p1: Density function

# Usage

```
dgev_p12(x, t1, t2, ymn, slope, sigma1, sigma2, xi, log = FALSE)
```

dgev\_p123 91

### Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor

sigma1 first coefficient for the sigma parameter of the distribution sigma2 second coefficient for the sigma parameter of the distribution

xi the shape parameter of the distribution log logical for the density evaluation

### Value

Vector

dgev\_p123 GEVD-with-p1: Density function

#### **Description**

GEVD-with-p1: Density function

### Usage

```
dgev_p123(x, t1, t2, t3, ymn, slope, sigma1, sigma2, xi1, xi2, log = FALSE)
```

### Arguments

Х	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor

sigma1 first coefficient for the sigma parameter of the distribution
sigma2 second coefficient for the sigma parameter of the distribution
xi1 first coefficient for the shape parameter of the distribution
xi2 second coefficient for the shape parameter of the distribution

log logical for the density evaluation

#### Value

Vector

92 dgev\_p12sub

dgev_p123sub Densities for 5 predictions
--

# Description

Densities for 5 predictions

## Usage

```
dgev_p123sub(x, t1, t2, t3, y, t01, t02, t03, ics, extramodels, debug)
```

# Arguments

x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
у	a vector of values at which to calculate the density and distribution functions
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
ics	initial conditions for the maximum likelihood search
extramodels	logical that indicates whether to add three additional prediction models
debug	debug flag

## Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

Densities for 5 predictions	dgev_p12sub
Je - President	

# Description

Densities for 5 predictions

dgev\_p1k3 93

# Usage

```
dgev_p12sub(
    x,
    t1,
    t2,
    y,
    t01,
    t02,
    ics,
    minxi,
    maxxi,
    debug,
    extramodels = FALSE
)
```

# Arguments

X	a vector of training data values	
t1	a vector of predictors for the mean	
t2	a vector of predictors for the sd	
У	a vector of values at which to calculate the density and distribution functions	
t01	a single value of the predictor (specify either t01 or n01 but not both)	
t02	a single value of the predictor (specify either t02 or n02 but not both)	
ics	initial conditions for the maximum likelihood search	
minxi	minimum value of shape parameter xi	
maxxi	maximum value of shape parameter xi	
debug	debug flag	
extramodels	logical that indicates whether to add three additional prediction models	

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dgev_p1k3	GEV-with-known-shape-with-p1 density function

# Description

GEV-with-known-shape-with-p1 density function

## Usage

```
dgev_p1k3(x, t0, ymn, slope, sigma, log = FALSE, kshape)
```

94 dgev\_p1k3sub

#### **Arguments**

t0 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution

logical for the density evaluation

kshape the known shape parameter

### Value

Vector

dgev\_p1k3sub

Densities from MLE and RHP

### **Description**

Densities from MLE and RHP

### Usage

```
dgev_p1k3sub(x, t, y, t0, kshape)
```

### **Arguments**

x a vector of training data valuest a vector or matrix of predictors

y a vector of values at which to calculate the density and distribution functions

to a single value of the predictor (specify either to or no but not both)

kshape the known shape parameter

### Value

dgev\_p1n 95

d	ge	/	n1	n
u,	ರ್ಷ	v _	μı	11

GEVD-with-p1: Density function

### **Description**

```
GEVD-with-p1: Density function
```

### Usage

```
dgev_p1n(x, t0, params, log = FALSE)
```

#### **Arguments**

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

params model parameters for calculating logf logical for the density evaluation

#### Value

Vector

dgev_	n1	nsuk	1
ugev_	י ק.	Hou	J

Densities for 5 predictions

### **Description**

Densities for 5 predictions

### Usage

```
dgev_p1nsub(x, t, y, t0, ics, minxi, maxxi, extramodels = FALSE)
```

## Arguments

Х	a vector of training data values
t	a vector or matrix of predictors

y a vector of values at which to calculate the density and distribution functions

t0 a single value of the predictor (specify either t0 or n0 but not both)

ics initial conditions for the maximum likelihood search

minxi minimum value of shape parameter xi maxxi maximum value of shape parameter xi

extramodels logical that indicates whether to add three additional prediction models

96 dgnorm\_k3sub

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dgev_p1sub	Densities for 5 predictions	
------------	-----------------------------	--

# Description

Densities for 5 predictions

## Usage

```
dgev_p1sub(x, t, y, t0, ics, minxi, maxxi, extramodels = FALSE)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
у	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
ics	initial conditions for the maximum likelihood search
minxi	minimum value of shape parameter xi
maxxi	maximum value of shape parameter xi
extramodels	logical that indicates whether to add three additional prediction models

#### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dgnorm_k3sub Densities from MLE and RHP
---

# Description

Densities from MLE and RHP

## Usage

```
dgnorm_k3sub(x, y, d1 = 0.01, fd2 = 0.01, kbeta, aderivs = TRUE)
```

dgpdsub 97

# Arguments

X	a vector of training data values
у	a vector of values at which to calculate the density and distribution functions
d1	the delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dgpdsub	Densities for 5 predictions	

# Description

Densities for 5 predictions

## Usage

```
dgpdsub(x, y, ics, kloc = 0, dlogpi = 0, minxi, maxxi, extramodels = FALSE)
```

# Arguments

X	a vector of training data values
у	a vector of values at which to calculate the density and distribution functions
ics	initial conditions for the maximum likelihood search
kloc	the known location parameter
dlogpi	gradient of the log prior
minxi	minimum value of shape parameter xi
maxxi	maximum value of shape parameter xi
extramodels	logical that indicates whether to add three additional prediction models

## Value

98 dgumbel\_p1

da	umbo	1 cub
ug	umbe	TSUD

Densities from MLE and RHP

### **Description**

Densities from MLE and RHP

### Usage

```
dgumbelsub(x, y)
```

### **Arguments**

x a vector of training data values

y a vector of values at which to calculate the density and distribution functions

#### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dgumbel\_p1

Gumbel-with-p1 density function

## Description

Gumbel-with-p1 density function

## Usage

```
dgumbel_p1(x, t0, ymn, slope, sigma, log = FALSE)
```

#### **Arguments**

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution log logical for the density evaluation

### Value

Vector

dgumbel\_p1sub 99

dgumbel_p1sub	Densities from MLE and RHP	
---------------	----------------------------	--

# Description

Densities from MLE and RHP

## Usage

```
dgumbel_p1sub(x, t, y, t0)
```

## Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
у	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)

#### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dhalfnormsub	Densities from MLE and RHP	
--------------	----------------------------	--

# Description

Densities from MLE and RHP

## Usage

```
dhalfnormsub(x, y, fd1 = 0.01, aderivs = TRUE)
```

# Arguments

X	a vector of training data values
У	a vector of values at which to calculate the density and distribution functions
fd1	the fractional delta used in the numerical derivatives with respect to the param-
	eter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

100 dinvgausssub

|--|

# Description

Densities from MLE and cp

## Usage

```
dinvgammasub(x, y, fd1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

# Arguments

Χ	a vector of training data values
у	a vector of values at which to calculate the density and distribution functions
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dinvgausssub	Densities from MLE and RHP	

# Description

Densities from MLE and RHP

# Usage

```
dinvgausssub(x, y, prior, fd1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

## Arguments

Х	a vector of training data values
У	a vector of values at which to calculate the density and distribution functions
prior	logical indicating which prior to use
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

dlnormsub 101

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dlnormsub

Densities from MLE and RHP

## Description

Densities from MLE and RHP

## Usage

```
dlnormsub(x, y)
```

### **Arguments**

x a vector of training data values

y a vector of values at which to calculate the density and distribution functions

#### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

 ${\tt dlnorm\_dmgssub}$ 

Densities from MLE and RHP

## Description

Densities from MLE and RHP

### Usage

```
dlnorm_dmgssub(x, y)
```

## Arguments

x a vector of training data values

y a vector of values at which to calculate the density and distribution functions

### Value

102 dlnorm\_p1sub

41	m a 10m	m 1
$a_{\perp}$	norm_	_p i

Normal-with-p1 density function

## Description

Normal-with-p1 density function

### Usage

```
dlnorm_p1(x, t0, ymn, slope, sigma, log = FALSE)
```

### **Arguments**

Χ	a vector of training data values
---	----------------------------------

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution log logical for the density evaluation

#### Value

Vector

d٦	norm	n1	lsuh

Densities from MLE and RHP

### **Description**

Densities from MLE and RHP

### Usage

```
dlnorm_p1sub(x, t, y, t0, debug = FALSE)
```

## Arguments

x a vector of training data valuest a vector or matrix of predictors

y a vector of values at which to calculate the density and distribution functions

t0 a single value of the predictor (specify either t0 or n0 but not both)

debug debug flag

#### Value

dlogis2sub

41	~~	ic	. 2 .	ub
uт	.Og	. I S	~~	นม

Densities from MLE and RHP

### **Description**

Densities from MLE and RHP

### Usage

```
dlogis2sub(x, y)
```

### **Arguments**

x a vector of training data values

y a vector of values at which to calculate the density and distribution functions

#### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dlogis\_p1

Logistic-with-p1 density function

## Description

Logistic-with-p1 density function

#### Usage

```
dlogis_p1(x, t0, ymn, slope, scale, log = FALSE)
```

#### **Arguments**

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor scale the scale parameter of the distribution log logical for the density evaluation

### Value

Vector

104 dlst\_k3sub

		_	
МI	ogis.	n I	SUID
u ı	.ugij.	י א_	Sub

Densities from MLE and RHP

# Description

Densities from MLE and RHP

## Usage

```
dlogis_p1sub(x, t, y, t0)
```

## Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors
у	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dlst_k3sub	Densities from MLE and RHP

## Description

Densities from MLE and RHP

# Usage

```
dlst_k3sub(x, y, d1 = 0.01, fd2 = 0.01, kdf, aderivs = TRUE)
```

# Arguments

X	a vector of training data values
у	a vector of values at which to calculate the density and distribution functions
d1	the delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

dlst\_p1k3

LST-with-p1 density function

# Description

LST-with-p1 density function

# Usage

```
dlst_p1k3(x, t0, ymn, slope, sigma, log = FALSE, kdf)
```

## Arguments

x	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma	the sigma parameter of the distribution
log	logical for the density evaluation
kdf	the known degrees of freedom parameter

### Value

Vector

dlst_	р1	k3sub
GIOC_		NOOUD

Densities from MLE and RHP

# Description

Densities from MLE and RHP

# Usage

```
dlst_p1k3sub(x, t, y, t0, d1, d2, fd3, kdf, aderivs = TRUE)
```

106 dmgs

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
у	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

# Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

|--|

# Description

Evaluate DMGS equation 3.3

## Usage

```
dmgs(lddi, lddd, mu1, pidopi, mu2, dim)
```

## Arguments

lddi	inverse of second derivative of observed log-likelihood
lddd	third derivative of observed log-likelihood
mu1	DMGS mu1 vector
pidopi	derivative of log prior
mu2	DMGS mu2 matrix
dim	number of parameters

### Value

Vector

dnormsub 107

dnormsub

Densities from MLE and RHP

## Description

Densities from MLE and RHP

# Usage

```
dnormsub(x, y)
```

### **Arguments**

x a vector of training data values

y a vector of values at which to calculate the density and distribution functions

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dnorm\_dmgssub

Densities from MLE and RHP

## Description

Densities from MLE and RHP

### Usage

```
dnorm_dmgssub(x, y)
```

### **Arguments**

x a vector of training data values

y a vector of values at which to calculate the density and distribution functions

### Value

108 dnorm\_p12

## Description

Normal-with-p1 density function

### Usage

```
dnorm_p1(x, t0, ymn, slope, sigma, log = FALSE)
```

## Arguments

X	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma	the sigma parameter of the distribution

log logical for the density evaluation

### Value

Vector

dnorm_p12	Normal-with-p12: Density function	

## Description

Normal-with-p12: Density function

# Usage

```
dnorm_p12(x, t01, t02, ymn, slope, sigma1, sigma2, log = FALSE)
```

## **Arguments**

X	a vector of training data values
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma1	first coefficient for the sigma parameter of the distribution
sigma2	second coefficient for the sigma parameter of the distribution
log	logical for the density evaluation

dnorm\_p12dmgs 109

## Value

Vector

dnorm_p12dmgs	Densities for 5 predictions	
---------------	-----------------------------	--

# Description

Densities for 5 predictions

## Usage

```
dnorm_p12dmgs(x, t1, t2, y, t01, t02, ics)
```

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
У	a vector of values at which to calculate the density and distribution functions
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
ics	initial conditions for the maximum likelihood search

## Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dnorm_p1sub Densities from MLE and RHP
--

# Description

Densities from MLE and RHP

# Usage

```
dnorm_p1sub(x, t, y, t0)
```

## Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors
У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)

dpareto\_k2\_sub

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dnorm\_p1\_formula

Linear regression formula, densities

#### **Description**

Linear regression formula, densities

#### Usage

```
dnorm_p1_formula(y, tresid, tresid0, nx, muhat0, v3hat)
```

#### **Arguments**

y a vector of values at which to calculate the density and distribution functions

tresid predictor residuals

tresid0 predictor residual at the point being predicted

nx length of training data

muhat at the point being predicted

v3hat third parameter

#### Value

Vector

dpareto\_k2\_sub

Densities from MLE and RHP

### **Description**

Densities from MLE and RHP

## Usage

```
dpareto_k2_sub(x, y, kscale)
```

#### **Arguments**

x a vector of training data values

y a vector of values at which to calculate the density and distribution functions

kscale the known scale parameter

#### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dpareto\_p1k2

dpareto_p1k2	pareto_k1-with-p2 density function
apar c co_p m2	pareto_ki with p2 density function

### **Description**

```
pareto_k1-with-p2 density function
```

### Usage

```
dpareto_p1k2(x, t0, ymn, slope, kscale, log = FALSE)
```

#### **Arguments**

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor

kscale the known scale parameter

log logical for the density evaluation

#### Value

Vector

dpareto_p1k2sub	Densities from MLE and RHP	

## Description

Densities from MLE and RHP

### Usage

```
dpareto_p1k2sub(x, t, y, t0, kscale, debug = FALSE)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors

y a vector of values at which to calculate the density and distribution functions

to a single value of the predictor (specify either to or no but not both)

kscale the known scale parameter

debug debug flag

dweibullsub

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dunif\_formula

Predictive PDFs

## Description

Predictive PDFs

## Usage

```
dunif_formula(x, y)
```

## Arguments

x a vector of training data values

y a vector of values at which to calculate the density and distribution functions

#### Value

Two vectors

dweibullsub

Densities from MLE and RHP

## Description

Densities from MLE and RHP

### Usage

```
dweibullsub(x, y)
```

# Arguments

x a vector of training data values

y a vector of values at which to calculate the density and distribution functions

# Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

dweibull\_p2 113

### **Description**

Weibull-with-p1 density function

## Usage

```
dweibull_p2(x, t0, shape, ymn, slope, log = FALSE)
```

### **Arguments**

X	a vec	ctor of	training	data values		
				4	 	 _

a single value of the predictor (specify either t0 or n0 but not both) t0

the shape parameter of the distribution shape

the location parameter of the function of the predictor ymn

the slope of the function of the predictor slope

log logical for the density evaluation

### Value

Vector

dweibull_p2sub Densities from MLE and RHP
---

## Description

Densities from MLE and RHP

## Usage

```
dweibull_p2sub(x, t, y, t0)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
у	a vector of values at which to calculate the density and distribution functions

t0 a single value of the predictor (specify either t0 or n0 but not both)

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

exp\_cp

Exponential Distribution Predictions Based on a Calibrating Prior

### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below. Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algo-

#### Usage

rithm.

```
qexp_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    rust = FALSE,
    nrust = 1e+05,
    debug = FALSE
)

rexp_cp(n, x, rust = FALSE, mlcp = TRUE, debug = FALSE)

dexp_cp(x, y = x, rust = FALSE, nrust = 1000, debug = FALSE)

pexp_cp(x, y = x, rust = FALSE, nrust = 1000, debug = FALSE)

texp_cp(n, x, debug = FALSE)
```

#### **Arguments**

X	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- $\bullet$  ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The exponential distribution has exceedance distribution function

$$S(x; \lambda) = \exp(-\lambda x)$$

where  $x \ge 0$  is the random variable and  $\lambda > 0$  is the rate parameter.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\lambda) \propto \frac{1}{\lambda}$$

as given in Jewson et al. (2025).

## **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

• ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)

 cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (analytic integration)**

For this model, the Bayesian prediction equation is integrated analytically.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),

- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

### **Examples**

```
#
# example 1
x=fitdistcp::d010exp_example_data_v1
p=c(1:9)/10
q=qexp_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qexp_cp)",
main="Exponential: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

 $exp_f2fa$ 

exp\_f1fa

The first derivative of the density

## Description

The first derivative of the density

The first derivative of the density

## Usage

```
exp_f1fa(x, v1)
exp_f1fa(x, v1)
```

## Arguments

x a vector of training data values

v1 first parameter

## Value

Vector

Vector

exp\_f2fa

The second derivative of the density

## Description

The second derivative of the density

The second derivative of the density

## Usage

```
exp_f2fa(x, v1)
exp_f2fa(x, v1)
```

### **Arguments**

x a vector of training data values

v1 first parameter

exp\_fd 121

### Value

Matrix

Matrix

exp\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
exp_fd(x, v1)
exp_fd(x, v1)
```

# Arguments

v1

x a vector of training data values

first parameter

## Value

Vector

Vector

exp\_fdd

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

122 exp\_ldda

### Usage

```
exp_fdd(x, v1)
exp_fdd(x, v1)
```

## Arguments

x a vector of training data values

v1 first parameter

## Value

Matrix

Matrix

exp\_ldda

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

The second derivative of the normalized log-likelihood

## Usage

```
exp_ldda(x, v1)
exp_ldda(x, v1)
```

# Arguments

x a vector of training data values

v1 first parameter

#### Value

Matrix

Matrix

exp\_lddda 123

exp\_lddda

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood The third derivative of the normalized log-likelihood

### Usage

```
exp_lddda(x, v1)
exp_lddda(x, v1)
```

### **Arguments**

x a vector of training data values

v1 first parameter

### Value

3d array 3d array

exp\_logf

Logf for RUST

# Description

Logf for RUST

## Usage

```
exp_logf(params, x)
```

### **Arguments**

params model parameters for calculating logf x a vector of training data values

## Value

Scalar value.

124 exp\_logfddd

exp_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### **Description**

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
exp_logfdd(x, v1)
exp_logfdd(x, v1)
```

### **Arguments**

x a vector of training data values v1 first parameter

#### Value

Matrix

Matrix

exp_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
exp_logfddd(x, v1)
exp_logfddd(x, v1)
```

exp\_logscores 125

#### **Arguments**

x a vector of training data values

v1 first parameter

#### Value

3d array 3d array

exp\_logscores

Log scores for MLE and RHP predictions calculated using leave-one-

out

## Description

Log scores for MLE and RHP predictions calculated using leave-one-out

## Usage

```
exp_logscores(logscores, x)
```

# **Arguments**

logiscores logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data values

#### Value

Two scalars

exp\_p1fa

The first derivative of the cdf

## Description

The first derivative of the cdf

The first derivative of the cdf

```
exp_p1fa(x, v1)
```

$$exp_p1fa(x, v1)$$

#### **Arguments**

x a vector of training data values v1 first parameter

#### Value

Vector

Vector

exp\_p1\_cp

Exponential Distribution with a Predictor, Predictions Based on a Calibrating Prior

### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qexp_p1_cp(
    x,
    t,
    t0 = NA,
    n0 = NA,
```

```
p = seq(0.1, 0.9, 0.1),
 means = FALSE,
 waicscores = FALSE,
 logscores = FALSE,
 dmgs = TRUE,
 rust = FALSE,
 nrust = 1e+05,
 predictordata = TRUE,
 centering = TRUE,
 debug = FALSE
)
rexp_p1_cp(n, x, t, t0 = NA, n0 = NA, rust = FALSE, mlcp = TRUE, debug = FALSE)
dexp_p1_cp(
 х,
  t,
  t0 = NA,
 n0 = NA,
 y = x,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
  debug = FALSE
)
pexp_p1_cp(
 х,
  t,
  t0 = NA,
 n0 = NA,
 y = x,
  rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE
)
texp_p1_cp(n, x, t, debug = FALSE)
```

# **Arguments**

```
a vector of training data values
Χ
t
                   a vector of predictors, such that length(t)=length(x)
                   a single value of the predictor (specify either t0 or n0 but not both)
t0
                   an index for the predictor (specify either t0 or n0 but not both)
n0
                   a vector of probabilities at which to generate predictive quantiles
р
```

1 11.

means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
predictordata	logical that indicates whether predictordata should be calculated
centering	logical that indicates whether the predictor should be centered
debug	logical for turning on debug messages
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- $\bullet$  ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### Details of the Model

The exponential distribution with a predictor has exceedance distribution function

$$S(x; a, b) = \exp(-x\lambda(a, b))$$

where  $x \ge 0$  is the random variable and  $\lambda(a,b) = e^{-a-bt}$  is the rate parameter, modelled as a function of the parameters a,b and a predictor t.

The calibrating prior is given by the right Haar prior, which is

$$\pi(a,b) \propto 1$$

. as given in Jewson et al. (2025).

### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

• ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)

 cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

 $\exp_p 1_{cp}$ 

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),

- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

exp\_p1\_f1fa 133

### **Examples**

```
#
# example 1
x=fitdistcp::d055exp_p1_example_data_v1_x
tt=fitdistcp::d055exp_p1_example_data_v1_t
p=c(1:9)/10
n0=10
q=qexp_p1_cp(x,tt,n0=n0,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qexp_p1_cp)",
main="Exponential w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

exp\_p1\_f1fa

The first derivative of the density for DMGS

## Description

The first derivative of the density for DMGS

### Usage

```
exp_p1_f1fa(x, t0, v1, v2)
```

### **Arguments**

X	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
v2	second parameter

## Value

Vector

 $exp_p1_f2fa$ 

	_	C4 C
exp	рΊ	f1fw

The first derivative of the density for WAIC

## Description

The first derivative of the density for WAIC

### Usage

```
exp_p1_f1fw(x, t, v1, v2)
```

### **Arguments**

x a vector of training data valuest a vector or matrix of predictors

v1 first parameterv2 second parameter

#### Value

Vector

exp\_p1\_f2fa

The second derivative of the density for DMGS

## Description

The second derivative of the density for DMGS

## Usage

```
exp_p1_f2fa(x, t0, v1, v2)
```

# **Arguments**

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

v1 first parameter v2 second parameter

### Value

Matrix

 $\exp_p 1_f 2f w$  135

	1	£2£.
exp	DΙ	_f2fw

The second derivative of the density for WAIC

#### **Description**

The second derivative of the density for WAIC

#### Usage

```
exp_p1_f2fw(x, t, v1, v2)
```

### **Arguments**

x a vector of training data values
 t a vector or matrix of predictors
 v1 first parameter

v1 first parameter v2 second parameter

### Value

Matrix

exp\_p1\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### **Description**

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
exp_p1_fd(x, t, v1, v2)
```

### **Arguments**

x a vector of training data valuest a vector or matrix of predictors

v1 first parameter v2 second parameter

#### Value

Vector

exp\_p1\_ldda

exp_p1_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
exp_p1_fdd(x, t, v1, v2)
```

## **Arguments**

Х	a vector of training data values
t	a vector or matrix of predictors

v1 first parameter v2 second parameter

#### Value

Matrix

avn	n1	ldda

The second derivative of the normalized log-likelihood

## Description

The second derivative of the normalized log-likelihood

## Usage

```
exp_p1_1dda(x, t, v1, v2)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter

### Value

Matrix

exp\_p1\_lddda 137

exp	р1	lddda

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

## Usage

```
exp_p1_1ddda(x, t, v1, v2)
```

## **Arguments**

x a vector of training data valuest a vector or matrix of predictors

v1 first parameter v2 second parameter

# Value

3d array

exp\_p1\_logf

 $Log f for \, RUST$ 

## Description

Logf for RUST

### Usage

```
exp_p1_logf(params, x, t)
```

## Arguments

params model parameters for calculating logf
x a vector of training data values
t a vector or matrix of predictors

### Value

Scalar value.

138 exp\_p1\_logfddd

exp_p1_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
exp_p1_logfdd(x, t, v1, v2)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
_	C .

v1 first parameter v2 second parameter

### Value

### Matrix

exp_p1_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
exp_p1_logfddd(x, t, v1, v2)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter

v2 second parameter

### Value

3d array

exp\_p1\_loglik 139

exp_p1_loglik	observed log-likelihood function
---------------	----------------------------------

# Description

observed log-likelihood function

## Usage

```
exp_p1_loglik(vv, x, t)
```

## Arguments

VV	parameters
• •	parameters

x a vector of training data valuest a vector or matrix of predictors

## Value

Scalar

exp_p1_logscores	Log scores for MLE and RHP predictions calculated using leave-one-
	out

# Description

Log scores for MLE and RHP predictions calculated using leave-one-out

# Usage

```
exp_p1_logscores(logscores, x, t)
```

## Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the
	log-score (much longer runtime)
x	a vector of training data values
t	a vector or matrix of predictors

## Value

Two scalars

140 exp\_p1\_mu1fa

exp\_p1\_means exp distribution: RHP means

#### **Description**

exp distribution: RHP means

### Usage

```
exp_p1_means(means, t0, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 2)
```

## **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

to a single value of the predictor (specify either to or no but not both)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

#### Value

Two scalars

exp\_p1\_mu1fa Minus the first derivative of the cdf, at alpha

## Description

Minus the first derivative of the cdf, at alpha

## Usage

```
exp_p1_mu1fa(alpha, t0, v1, v2)
```

### **Arguments**

alpha a vector of values of alpha (one minus probability)

to a single value of the predictor (specify either to or no but not both)

v1 first parameter v2 second parameter 

### Value

Vector

exp\_p1\_mu2fa

Minus the second derivative of the cdf, at alpha

### Description

Minus the second derivative of the cdf, at alpha

#### Usage

```
exp_p1_mu2fa(alpha, t0, v1, v2)
```

### **Arguments**

alpha a vector of values of alpha (one minus probability)

t0 a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameter v2 second parameter

#### Value

Matrix

exp\_p1\_p1fa

The first derivative of the cdf

## Description

The first derivative of the cdf

# Usage

```
exp_p1_p1fa(x, t0, v1, v2)
```

### **Arguments**

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

v1 first parameter v2 second parameter

#### Value

Vector

142 exp\_p1\_pd

exn	n1	_p2fa

The second derivative of the cdf

## Description

The second derivative of the cdf

## Usage

```
exp_p1_p2fa(x, t0, v1, v2)
```

## Arguments

×	a vector of traini	ng data values
^	a vector or training	iig data varues

t0 a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameter v2 second parameter

### Value

Matrix

exp_p1_pd
-----------

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
exp_p1_pd(x, t, v1, v2)
```

# Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors

v1 first parameter v2 second parameter

#### Value

Vector

exp\_p1\_pdd 143

exp_p1_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### **Description**

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
exp_p1_pdd(x, t, v1, v2)
```

### **Arguments**

x a vector of training data valuest a vector or matrix of predictors

v1 first parameter v2 second parameter

#### Value

Matrix

### **Description**

Predicted Parameter and Generalized Residuals

### Usage

```
exp_p1_predictordata(predictordata, x, t, t0, params)
```

## Arguments

predictordata logical that indicates whether to calculate and return predictordata x a vector of training data values t a vector or matrix of predictors t0 a single value of the predictor (specify either t0 or n0 but not both) params model parameters for calculating logf

#### Value

Two vectors

144 exp\_p2fa

# Description

Waic

## Usage

```
exp_p1_waic(waicscores, x, t, v1hat, v2hat, lddi, lddd, lambdad)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
t	a vector or matrix of predictors
v1hat	first parameter
v2hat	second parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior

## Value

Two numeric values.

exp_p2fa The second derivative of the cdf	exp_p2fa	The second derivative of the cdf	
---	----------	----------------------------------	--

# Description

The second derivative of the cdf

The second derivative of the cdf

```
exp_p2fa(x, v1)
exp_p2fa(x, v1)
```

exp\_pd 145

# Arguments

x a vector of training data values

v1 first parameter

#### Value

Matrix

Matrix

exp\_pd

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
exp_pd(x, v1)
exp_pd(x, v1)
```

# Arguments

x a vector of training data values

v1 first parameter

## Value

Vector

146 exp\_waic

exp_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol
	•

## **Description**

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
exp_pdd(x, v1)
exp_pdd(x, v1)
```

# Arguments

x a vector of training data values

v1 first parameter

#### Value

Matrix

Matrix

exp_waic	Waicscores

# Description

Waicscores

## Usage

```
exp_waic(waicscores, x, v1hat)
```

## **Arguments**

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data values

v1hat first parameter

findnt 147

# Value

Two numeric values.

findnt

Find the number of predictors in the predictor

# Description

Find the number of predictors in the predictor

## Usage

```
findnt(t)
```

# Arguments

t

a vector or matrix of predictors

#### Value

Vector

fixgevrange

Deal with situations in which the user wants d or p outside the GEV range

# Description

Deal with situations in which the user wants d or p outside the GEV range

# Usage

```
fixgevrange(y, v1, v2, v3)
```

# Arguments

,	v a ·	vector of values at w	hich to calculate the	e density and dist	ribution functions

v1 first parameter v2 second parameter v3 third parameter

## Value

fixgpdrange	Deal with situations in which the user wants d or p outside the GPD
	range

#### **Description**

Deal with situations in which the user wants d or p outside the GPD range

#### Usage

```
fixgpdrange(y, v1, v2, v3)
```

#### **Arguments**

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter

v2 second parameter v3 third parameter

#### Value

Vector

frechet\_k1\_cp

Frechet Distribution Predictions Based on a Calibrating Prior

## Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qfrechet_k1_cp(
 х,
 p = seq(0.1, 0.9, 0.1),
 kloc = 0,
 means = FALSE,
 waicscores = FALSE,
  logscores = FALSE,
 dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 debug = FALSE
)
rfrechet_k1_cp(n, x, kloc = 0, rust = FALSE, mlcp = TRUE, debug = FALSE)
dfrechet_k1_cp(x, y = x, kloc = 0, rust = FALSE, nrust = 1000, debug = FALSE)
pfrechet_k1_cp(x, y = x, kloc = 0, rust = FALSE, nrust = 1000, debug = FALSE)
tfrechet_k1_cp(n, x, kloc = 0, debug = FALSE)
```

#### Arguments

X	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
kloc	the known location parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations

debug	logical for turning on debug messages
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Frechet distribution has distribution function

$$F(x; \sigma, \lambda) = \exp\left(-\left(\frac{x-\mu}{\sigma}\right)^{-\lambda}\right)$$

where  $x > \mu$  is the random variable,  $\sigma > 0, \lambda > 0$  are the parameters and we consider  $\mu$  to be known (hence the k1 in the name).

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma,\lambda) \propto \frac{1}{\sigma\lambda}$$

as given in Jewson et al. (2025).

## **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

## **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),

154 frechet\_k1\_f1fa

• t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),

- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
#
# example 1
x=fitdistcp::d051frechet_k1_example_data_v1
p=c(1:9)/10
q=qfrechet_k1_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qfrechet_k1_cp)",
main="Frechet: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

frechet\_k1\_f1fa

The first derivative of the density

#### Description

The first derivative of the density

frechet\_k1\_f2fa

## Usage

```
frechet_k1_f1fa(x, v1, v2, kloc)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

kloc the known location parameter

## Value

Vector

frechet\_k1\_f2fa

The second derivative of the density

# Description

The second derivative of the density

# Usage

```
frechet_k1_f2fa(x, v1, v2, kloc)
```

# Arguments

x a vector of training data values

v1 first parameter

v2 second parameter

kloc the known location parameter

#### Value

Matrix

frechet\_k1\_fdd

frechet_k1_fd First derivative of the density Created by Stephen Jewson using D riv() by Andrew Clausen and Serguei Sokol	De-
---	-----

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
frechet_k1_fd(x, v1, v2, v3)
```

# **Arguments**

X	a vector of training data values
v1	first parameter

v2 second parameter v3 third parameter

# Value

Vector

frechet_k1_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
frechet_k1_fdd(x, v1, v2, v3)
```

# Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Matrix

frechet\_k1\_ldda 157

frechet	1.1	1 44~
Trechet	ΚI	100a

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

# Usage

```
frechet_k1_ldda(x, v1, v2, kloc)
```

# Arguments

x a vector of train	ning data values
---------------------	------------------

v1 first parameter v2 second parameter

kloc the known location parameter

#### Value

Matrix

frechet\_k1\_lddda

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

## Usage

```
frechet_k1_lddda(x, v1, v2, kloc)
```

# **Arguments**

X	a vector of	training	data values

v1 first parameter v2 second parameter

kloc the known location parameter

## Value

3d array

158 frechet\_k1\_logfdd

frechet\_k1\_logf

Logf for RUST

# Description

Logf for RUST

## Usage

```
frechet_k1_logf(params, x, kloc)
```

## **Arguments**

model parameters for calculating logf params Х a vector of training data values kloc the known location parameter

#### Value

Scalar value.

frechet\_k1\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
frechet_k1_logfdd(x, v1, v2, v3)
```

## **Arguments**

x a vector of training data valu
----------------------------------

first parameter v1 v2 second parameter third parameter v3

# Value

Matrix

frechet\_k1\_logfddd 159

frechet_k1_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
frechet_k1_logfddd(x, v1, v2, v3)
```

# **Arguments**

X	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

3d array

frechet\_k1\_mu1fa

Minus the first derivative of the cdf, at alpha

# Description

Minus the first derivative of the cdf, at alpha

# Usage

```
frechet_k1_mu1fa(alpha, v1, v2, kloc)
```

# Arguments

alpha	a vector of va	alues of alpha	one minus	probability)

v1 first parameter v2 second parameter

kloc the known location parameter

## Value

frechet\_k1\_p1fa

frechet\_k1\_mu2fa

Minus the second derivative of the cdf, at alpha

## **Description**

Minus the second derivative of the cdf, at alpha

## Usage

```
frechet_k1_mu2fa(alpha, v1, v2, kloc)
```

## **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

kloc the known location parameter

#### Value

Matrix

frechet\_k1\_p1fa

The first derivative of the cdf

# Description

The first derivative of the cdf

# Usage

```
frechet_k1_p1fa(x, v1, v2, kloc)
```

# **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

kloc the known location parameter

## Value

frechet\_k1\_p2fa

_		1 4	2.0
tre	chet	kΙ	p2fa

The second derivative of the cdf

# Description

The second derivative of the cdf

## Usage

```
frechet_k1_p2fa(x, v1, v2, kloc)
```

# Arguments

Χ	a vector of training data values
---	----------------------------------

v1 first parameter v2 second parameter

kloc the known location parameter

## Value

Matrix

frechet\_k1\_pd

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
frechet_k1_pd(x, v1, v2, v3)
```

# Arguments

		- C 4 : :	4-41
X	a vector	or training	data values

v1 first parameterv2 second parameterv3 third parameter

#### Value

162 frechet\_k1\_waic

frechet_k1_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
frechet_k1_pdd(x, v1, v2, v3)
```

# Arguments

Х	a vector of training data values
v1	first parameter
v2	second parameter

third parameter

# v3

## Value

Matrix

# Description

Waic

# Usage

```
frechet_k1_waic(waicscores, x, v1hat, v2hat, kloc, lddi, lddd, lambdad)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
v1hat	first parameter
v2hat	second parameter
kloc	the known location parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior

frechet\_loglik 163

## Value

Two numeric values.

frechet\_loglik

log-likelihood function

## **Description**

log-likelihood function

## Usage

```
frechet_loglik(vv, x, kloc)
```

#### **Arguments**

vv parameters

x a vector of training data valueskloc the known location parameter

#### Value

Scalar

frechet\_logscores

 $Log\ scores\ for\ MLE\ and\ RHP\ predictions\ calculated\ using\ leave-one-$ 

out

#### **Description**

Log scores for MLE and RHP predictions calculated using leave-one-out

#### Usage

```
frechet_logscores(logscores, x, kloc)
```

## Arguments

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data valueskloc the known location parameter

#### Value

Two scalars

frechet_means	MLE and RHP	predictive	means
i i cerie e_illeario	mee and mar	predictive	means

#### **Description**

MLE and RHP predictive means

#### Usage

```
frechet_means(means, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 2, kloc)
```

## **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

1ddi inverse observed information matrix1ddd third derivative of log-likelihood1ambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

kloc the known location parameter

#### Value

Two scalars

frechet_p2k1_cp	Frechet Distribution with Predictor, Predictions Based on a Calibrat-
	ing Prior

## Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.

- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qfrechet_p2k1_cp(
  х,
  t,
  t0 = NA,
 n0 = NA,
 p = seq(0.1, 0.9, 0.1),
 means = FALSE,
 waicscores = FALSE,
  logscores = FALSE,
  kloc = 0,
 dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 predictordata = TRUE,
 centering = TRUE,
  debug = FALSE
)
rfrechet_p2k1_cp(
  n,
  х,
  t,
  t0 = NA,
  n0 = NA,
 kloc = 0,
  rust = FALSE,
 mlcp = TRUE,
 centering = TRUE,
  debug = FALSE
)
dfrechet_p2k1_cp(
  х,
  t,
```

```
t0 = NA,
 n0 = NA,
 y = x,
 kloc = 0,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE
)
pfrechet_p2k1_cp(
 х,
  t,
 t0 = NA,
 n0 = NA,
 y = x,
 kloc = 0,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE
)
tfrechet_p2k1_cp(n, x, t, kloc = 0, debug = FALSE)
```

# Arguments

x	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
р	a vector of probabilities at which to generate predictive quantiles
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
kloc	the known location parameter
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
predictordata	logical that indicates whether predictordata should be calculated

centering	logical that indicates whether the predictor should be centered
debug	logical for turning on debug messages
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.

• cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Frechet distribution with predictor has distribution function

$$F(x; a, b, \lambda) = \exp\left(-\left(\frac{x-\mu}{\sigma(a, b)}\right)^{-\lambda}\right)$$

where  $x>\mu$  is the random variable,  $\sigma=e^{a+bt}$  is the scale parameter, modelled as a function of parameters a,b and predictor t, and  $\lambda>0$  is the shape parameter. We consider  $\mu$  to be known (hence the k1 in the name).

The calibrating prior is given by the right Haar prior, which is

$$\pi(a,b) \propto 1$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using
posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),

- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
#
# example 1
x=fitdistcp::d071frechet_p2k1_example_data_v1_x
tt=fitdistcp::d071frechet_p2k1_example_data_v1_t
p=c(1:9)/10
n0=10
q=qfrechet_p2k1_cp(x,tt,n0=n0,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qfrechet_p2k1_cp)",
main="Frechet w/ p2: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

frechet\_p2k1\_f1fw

The first derivative of the density for DMGS

# Description

The first derivative of the density for DMGS

# Usage

```
frechet_p2k1_f1fa(x, t0, v1, v2, v3, kloc)
```

## **Arguments**

x	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
v2	second parameter

v3 third parameter

kloc the known location parameter

## Value

Vector

frechet\_p2k1\_f1fw

The first derivative of the density for WAIC

# Description

The first derivative of the density for WAIC

## Usage

```
frechet_p2k1_f1fw(x, t, v1, v2, v3, kloc)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

kloc the known location parameter

frechet\_p2k1\_f2fa 173

## Value

Vector

frechet\_p2k1\_f2fa

The second derivative of the density for DMGS

## **Description**

The second derivative of the density for DMGS

# Usage

```
frechet_p2k1_f2fa(x, t0, v1, v2, v3, kloc)
```

## **Arguments**

x a vector of training data valu
----------------------------------

t0 a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameter v2 second parameter v3 third parameter

kloc the known location parameter

#### Value

Matrix

frechet\_p2k1\_f2fw

The second derivative of the density for WAIC

# Description

The second derivative of the density for WAIC

## Usage

```
frechet_p2k1_f2fw(x, t, v1, v2, v3, kloc)
```

# Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

kloc the known location parameter

174 frechet\_p2k1\_fdd

## Value

Matrix

frechet\_p2k1\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## **Description**

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
frechet_p2k1_fd(x, t, v1, v2, v3, v4)
```

## **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Vector

frechet\_p2k1\_fdd

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
frechet_p2k1_fdd(x, t, v1, v2, v3, v4)
```

frechet\_p2k1\_ldda 175

# Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

# Value

Matrix

frechet\_p2k1\_ldda

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

# Usage

```
frechet_p2k1_ldda(x, t, v1, v2, v3, kloc)
```

# Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
kloc	the known location parameter

## Value

Matrix

176 frechet\_p2k1\_logf

frechet\_p2k1\_lddda

The third derivative of the normalized log-likelihood

## **Description**

The third derivative of the normalized log-likelihood

## Usage

```
frechet_p2k1_lddda(x, t, v1, v2, v3, kloc)
```

## **Arguments**

x a vector of training data values
 t a vector or matrix of predictors
 v1 first parameter
 v2 second parameter
 v3 third parameter

kloc the known location parameter

## Value

3d array

 $frechet_p2k1_logf$ 

Logf for RUST

# Description

```
Logf for RUST
```

## Usage

```
frechet_p2k1_logf(params, x, t, kloc)
```

## **Arguments**

params model parameters for calculating logf
x a vector of training data values
t a vector or matrix of predictors
kloc the known location parameter

#### Value

Scalar value.

frechet\_p2k1\_logfdd 177

frechet_p2k1_logfdd	Second derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
frechet_p2k1_logfdd(x, t, v1, v2, v3, v4)
```

# Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Matrix

frechet_p2k1_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
frechet_p2k1_logfddd(x, t, v1, v2, v3, v4)
```

178 frechet\_p2k1\_loglik

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

# Value

3d array

# Description

observed log-likelihood function

# Usage

```
frechet_p2k1_loglik(vv, x, t, kloc)
```

# Arguments

VV	parameters
Х	a vector of training data values
t	a vector or matrix of predictors
kloc	the known location parameter

## Value

Scalar

```
frechet_p2k1_logscores
```

Log scores for MLE and RHP predictions calculated using leave-one-out

## **Description**

Log scores for MLE and RHP predictions calculated using leave-one-out

## Usage

```
frechet_p2k1_logscores(logscores, x, t, kloc)
```

## **Arguments**

logscores logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)

x a vector of training data values
t a vector or matrix of predictors
kloc the known location parameter

#### Value

Two scalars

frechet\_p2k1\_means

frechet\_k1 distribution: RHP mean

## Description

frechet\_k1 distribution: RHP mean

# Usage

```
frechet_p2k1_means(
  means,
  t0,
  ml_params,
  lddi,
  lddd,
  lambdad_rhp,
  nx,
  dim,
  kloc
)
```

180 frechet\_p2k1\_mu1fa

## Arguments

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

to a single value of the predictor (specify either to or no but not both)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

kloc the known location parameter

## Value

Two scalars

frechet\_p2k1\_mu1fa

Minus the first derivative of the cdf, at alpha

# Description

Minus the first derivative of the cdf, at alpha

## Usage

```
frechet_p2k1_mu1fa(alpha, t0, v1, v2, v3, kloc)
```

## **Arguments**

alpha a vector of values of alpha (one minus probability)

t0 a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameterv2 second parameterv3 third parameter

kloc the known location parameter

## Value

frechet\_p2k1\_mu2fa 181

frechet\_p2k1\_mu2fa

Minus the second derivative of the cdf, at alpha

# Description

Minus the second derivative of the cdf, at alpha

# Usage

```
frechet_p2k1_mu2fa(alpha, t0, v1, v2, v3, kloc)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
kloc	the known location parameter

# Value

Matrix

frechet\_p2k1\_p1fa

The first derivative of the cdf

# Description

The first derivative of the cdf

# Usage

```
frechet_p2k1_p1fa(x, t0, v1, v2, v3, kloc)
```

# Arguments

X	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
kloc	the known location parameter

frechet\_p2k1\_pd

# Value

Vector

frechet\_p2k1\_p2fa
The second derivative of the cdf

# Description

The second derivative of the cdf

# Usage

```
frechet_p2k1_p2fa(x, t0, v1, v2, v3, kloc)
```

# **Arguments**

X	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
v2	second parameter

v2 second parameter v3 third parameter

kloc the known location parameter

# Value

Matrix

frechet\_p2k1\_pd First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
frechet_p2k1_pd(x, t, v1, v2, v3, v4)
```

frechet\_p2k1\_pdd 183

# Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

# Value

Vector

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
frechet_p2k1_pdd(x, t, v1, v2, v3, v4)
```

# **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

# Value

Matrix

184 frechet\_p2k1\_waic

```
frechet_p2k1_predictordata
```

Predicted Parameter and Generalized Residuals

# **Description**

Predicted Parameter and Generalized Residuals

# Usage

```
frechet_p2k1_predictordata(predictordata, x, t, t0, params, kloc)
```

# **Arguments**

```
predictordata logical that indicates whether to calculate and return predictordata x a vector of training data values t a vector or matrix of predictors t0 a single value of the predictor (specify either t0 or n0 but not both) params model parameters for calculating logf kloc the known location parameter
```

### Value

Two vectors

```
frechet_p2k1_waic
Waic
```

# **Description**

Waic

```
frechet_p2k1_waic(
  waicscores,
  x,
  t,
  v1hat,
  v2hat,
  v3hat,
  kloc,
  lddi,
  lddd,
  lambdad
)
```

#### **Arguments**

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
t	a vector or matrix of predictors
v1hat	first parameter
v2hat	second parameter
v3hat	third parameter
kloc	the known location parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood

# Value

lambdad

Two numeric values.

gamma\_cp

Gamma Distribution Predictions Based on a Calibrating Prior

# Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.

derivative of the log prior

- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qgamma_cp(
  Х,
  p = seq(0.1, 0.9, 0.1),
  fd1 = 0.01,
  fd2 = 0.01,
 means = FALSE,
 waicscores = FALSE,
  logscores = FALSE,
  dmgs = TRUE,
  rust = FALSE,
  nrust = 1e+05,
  prior = "type 1",
  debug = FALSE,
  aderivs = TRUE
)
rgamma_cp(
  n,
 х,
  fd1 = 0.01,
  fd2 = 0.01,
  rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
dgamma_cp(
 х,
 y = x,
  fd1 = 0.01,
  fd2 = 0.01,
  rust = FALSE,
  nrust = 1000,
 debug = FALSE,
  aderivs = TRUE
)
pgamma_cp(
 Х,
 y = x,
  fd1 = 0.01,
  fd2 = 0.01,
  rust = FALSE,
  nrust = 1000,
  debug = FALSE,
  aderivs = TRUE
```

```
)
tgamma_cp(n, x, fd1 = 0.01, fd2 = 0.01, debug = FALSE)
```

# Arguments

X	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
fd1	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the first parameter
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
prior	logical indicating which prior to use
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.

• cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

### **Details of the Model**

The Gamma distribution has probability density function

$$f(x; \alpha, \sigma) = \frac{1}{\sigma^{\alpha} \Gamma(\alpha)} x^{\alpha - 1} e^{-x/\sigma}$$

where  $x \ge 0$  is the random variable and  $\alpha > 0, \sigma > 0$  are the parameters.

The calibrating prior we use is

$$\pi(\alpha,\sigma)\propto \frac{1}{\alpha\sigma}$$

# **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

# Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

• Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),

192 gamma\_f1f

- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

### **Examples**

```
#
# example 1
x=fitdistcp::d100gamma_example_data_v1
p=c(1:9)/10
q=qgamma_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),sub="(from qgamma_cp)",
main="Gamma: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

gamma\_f1f

DMGS equation 3.3, f1 term

### **Description**

DMGS equation 3.3, f1 term

# Usage

```
gamma_f1f(y, v1, fd1, v2, fd2)
```

# **Arguments**

у	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

# Value

Matrix

gamma\_f1fa 193

	C 4	_
gamma_	+ 1	t a
gaiiiiia_	_ ' '	ı u

The first derivative of the density

# **Description**

The first derivative of the density

# Usage

```
gamma_f1fa(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

### Value

Vector

~~~~	モンモ
gamma	T/T

DMGS equation 3.3, f2 term

# Description

DMGS equation 3.3, f2 term

# Usage

```
gamma_f2f(y, v1, fd1, v2, fd2)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-

eter

# Value

3d array

194 gamma\_fd

~~mm~	fafa
gamma	t∠ta

The second derivative of the density

# Description

The second derivative of the density

# Usage

```
gamma_f2fa(x, v1, v2)
```

# **Arguments**

x a vector of training data values

v1 first parameterv2 second parameter

# Value

Matrix

gamma\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gamma_fd(x, v1, v2)
```

# **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

# Value

Vector

gamma\_fdd 195

gamma_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gamma_fdd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

# Value

Matrix

gamma_gg Second derivative matrix of the expected log-likelihood	
------------------------------------------------------------------	--

# Description

Second derivative matrix of the expected log-likelihood

# Usage

```
gamma_gg(v1, fd1, v2, fd2)
```

# **Arguments**

v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

# Value

Square scalar matrix

196 gamma\_ldd

gamma_	gmn

One component of the second derivative of the expected log-likelihood

# Description

One component of the second derivative of the expected log-likelihood

# Usage

```
gamma_gmn(alpha, v1, fd1, v2, fd2, mm, nn)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

# Value

Scalar value

gamma	ldd	

Second derivative matrix of the normalized log-likelihood

# Description

Second derivative matrix of the normalized log-likelihood

```
gamma_1dd(x, v1, fd1, v2, fd2)
```

gamma\_ldda 197

# Arguments

X	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

# Value

Square scalar matrix

gamma_ldda	elihood
------------	---------

# Description

The second derivative of the normalized log-likelihood

# Usage

```
gamma_ldda(x, v1, v2)
```

# Arguments

X	a vector of training d	lata values

v1 first parameter

v2 second parameter

# Value

Matrix

198 gamma\_lddda

gamma_	1	d	d	d

Third derivative tensor of the normalized log-likelihood

# **Description**

Third derivative tensor of the normalized log-likelihood

# Usage

```
gamma_lddd(x, v1, fd1, v2, fd2)
```

# Arguments

X	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second perometer

v2 second parameter

fd2 the fractional delta used in the numerical derivatives with respect to the param-

eter

# Value

Cubic scalar array

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

# Usage

```
gamma_lddda(x, v1, v2)
```

# Arguments

X	a vector of	f training	data val	ues
---	-------------	------------	----------	-----

v1 first parameter v2 second parameter

# Value

3d array

gamma\_lmn 199

gamma_lmn One component of the second derivative of the normalized log- likelihood
---------------------------------------------------------------------------------------

# Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
gamma_lmn(x, v1, fd1, v2, fd2, mm, nn)
```

# Arguments

X	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

# Value

Scalar value

gamma_lmnp	One component of the second derivative of the normalized log-likelihood
------------	-------------------------------------------------------------------------

# Description

One component of the second derivative of the normalized log-likelihood

```
gamma_lmnp(x, v1, fd1, v2, fd2, mm, nn, rr)
```

200 gamma\_logf

# Arguments

x	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

# Value

Scalar value

gf Logf for RUST
------------------

# Description

Logf for RUST

# Usage

```
gamma_logf(params, x)
```

# Arguments

params model parameters for calculating logf x a vector of training data values

# Value

Scalar value.

gamma\_logfdd 201

gamma_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gamma_logfdd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

# Value

Matrix

gamma_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	3

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gamma_logfddd(x, v1, v2)
```

# Arguments

X	a vector of training data values
Λ	a vector or training data variets

v1 first parameter v2 second parameter

# Value

3d array

202 gamma\_logscores

# Description

log-likelihood function

# Usage

```
gamma_loglik(vv, x)
```

# Arguments

vv parameters

x a vector of training data values

# Value

Scalar

gamma_logscores	Log scores for MLE and RHP predictions calculated using leave-one- out
-----------------	---------------------------------------------------------------------------

# Description

Log scores for MLE and RHP predictions calculated using leave-one-out

# Usage

```
gamma_logscores(logscores, x, fd1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

# Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
X	a vector of training data values
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

# Value

Two scalars

gamma\_means 203

gamma_means	MLE and RHP predictive means
8aa	THE COLOR THAT PROGRESS AND COLORS

# Description

MLE and RHP predictive means

# Usage

```
gamma_means(means, ml_params, lddi, lddd, lambdad_cp, nx, dim = 2)
```

# Arguments

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

lddi inverse observed information matrix
 lddd third derivative of log-likelihood
 lambdad\_cp derivative of the log prior
 length of training data
 dim number of parameters

### Value

Two scalars

gamma_mu1f DMGS equation 3.3, mu1 term
----------------------------------------

# Description

DMGS equation 3.3, mu1 term

# Usage

```
gamma_mu1f(alpha, v1, fd1, v2, fd2)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

204 gamma\_p1f

# Value

Matrix

 ${\tt gamma\_mu2f}$ 

DMGS equation 3.3, mu2 term

# Description

DMGS equation 3.3, mu2 term

# Usage

```
gamma_mu2f(alpha, v1, fd1, v2, fd2)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)	
	7)	

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

v2 second parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

# Value

3d array

gamma\_p1f

DMGS equation 3.3, p1 term

# Description

DMGS equation 3.3, p1 term

```
gamma_p1f(y, v1, fd1, v2, fd2)
```

gamma\_p2f 205

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

# Value

Matrix

gamma_p2f	DMGS equation 3.3, p2 term

# Description

DMGS equation 3.3, p2 term

# Usage

```
gamma_p2f(y, v1, fd1, v2, fd2)
```

# Arguments

у	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

# Value

3d array

206 gev\_boot

# Description

Waic

# Usage

```
gamma_waic(waicscores, x, v1hat, fd1, v2hat, fd2, lddi, lddd, lambdad, aderivs)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
v1hat	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior

logical for whether to use analytic derivatives (instead of numerical)

# Value

aderivs

Two numeric values.

|--|--|--|

# Description

Bootstrap

```
gev_boot(x, n)
```

gev\_checkmle 207

# Arguments

x a vector of training data values

n number of random samples required

### Value

A list containing a matrix of simulated parameter values

gev\_checkmle

Check MLE

# **Description**

Check MLE

# Usage

```
gev_checkmle(ml_params, minxi = -1, maxxi = 1)
```

### **Arguments**

ml\_params parameters

minxi minimum value of shape parameter xi maxxi maximum value of shape parameter xi

# Value

No return value (just a message to the screen).

gev\_cp Generalized Extreme Value Distribution, Predictions Based on a Calibrating Prior

# Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.

- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qgev_cp(
  Х,
 p = seq(0.1, 0.9, 0.1),
  ics = c(0, 0, 0),
  fdalpha = 0.01,
 minxi = -1,
 \max x i = 1,
 means = FALSE,
 waicscores = FALSE,
  extramodels = FALSE,
 pdf = FALSE,
  customprior = 0.
  dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 pwm = FALSE,
 debug = FALSE
)
rgev_cp(
 n,
 Х,
 ics = c(0, 0, 0),
 minxi = -1,
 maxxi = 1,
 method = "rust",
 extramodels = FALSE,
  rust = FALSE,
 mlcp = TRUE,
 debug = FALSE
)
dgev_cp(
 х,
```

```
y = x,
 ics = c(0, 0, 0),
 minxi = -1,
 maxxi = 1,
 extramodels = FALSE,
 rust = FALSE,
 nrust = 1000,
 boot = FALSE,
 nboot = 1000,
 debug = FALSE
)
pgev_cp(
 Х,
 y = x,
 ics = c(0, 0, 0),
 minxi = -1,
 maxxi = 1,
 extramodels = FALSE,
 rust = FALSE,
 nrust = 1000,
 boot = FALSE,
 nboot = 1000,
 debug = FALSE
tgev_cp(method, n, x, ics = c(0, 0, 0), extramodels = FALSE, debug = FALSE)
```

# Arguments x

	C
p	a vector of probabilities at which to generate predictive quantiles
ics	initial conditions for the maximum likelihood search
fdalpha	if pdf=TRUE, the fractional delta used for numerical derivatives with respect to probability, for calculating the pdf as a function of quantiles
minxi	the minimum allowed value of the shape parameter (decrease with caution)
maxxi	the maximum allowed value of the shape parameter (increase with caution)
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
extramodels	logical that indicates whether to run additional calculations and add three additional prediction models (longer runtime)
pdf	logical that indicates whether to run additional calculations and return density functions evaluated at quantiles specified by the input probabilities (longer runtime)

a vector of training data values

customprior	a custom value for the slope of the log prior at the maxlik estimate
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
pwm	logical for whether to include PWM results (longer runtime)
debug	logical for turning on debug messages
n	the number of random samples required
method	character string that indicates whether to use rust method=rust or bootstrap method=boot
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions
boot	logical that indicates whether bootstrap-based posterior sampling calculations should be run or not (longer run time)
nboot	the number of posterior samples used in the bootstrap calculations

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

211

• cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The GEV distribution has distribution function

$$F(x; \mu, \sigma, \xi) = \exp\left(-t(x; \mu, \sigma, \xi)\right)$$

where

$$t(x; \mu, \sigma, \xi) = \begin{cases} \left[1 + \xi \left(\frac{x - \mu}{\sigma}\right)\right]^{-1/\xi} & \text{if } \xi \neq 0\\ \exp\left(-\frac{x - \mu}{\sigma}\right) & \text{if } \xi = 0 \end{cases}$$

where x is the random variable and  $\mu, \sigma > 0, \xi$  are the parameters.

The calibrating prior we use is given by

$$\pi(\mu, \sigma, \xi) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

The code will stop with an error if the input data gives a maximum likelihood value for the shape parameter that lies outside the range (minxi, maxxi), since outside this range there may be numerical problems. Such values seldom occur in real observed data for maxima.

# **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

• waic1: the WAIC1 score for the calibrating prior model.

• waic2: the WAIC2 score for the calibrating prior model.

### If logscores=TRUE:

• ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)

• cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where
  mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

# If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

### **Optional Return Values (EVT models only)**

q\*\*\*\* optionally returns the following, for EVT models only:

cp\_pdf: the density function at quantiles corresponding to input probabilities p. We provide this for EVD models, because direct estimation of the density function using the DMGS density equation is not possible.

### Optional Return Values (some EVT models only)

q\*\*\*\* optionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_quantiles: predictive quantiles calculated from Bayesian integration with a flat prior.
- rh\_ml\_quantiles: predictive quantiles calculated from Bayesian integration with the calibrating prior, and the maximmum likelihood estimate for the shape parameter.
- jp\_quantiles: predictive quantiles calculated from Bayesian integration with Jeffreys' prior.

r\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_deviates: predictive random deviates calculated using a Bayesian analysis with a flat prior.
- rh\_ml\_deviates: predictive random deviates calculated using a Bayesian analysis with the RHP-MLE prior.
- jp\_deviates: predictive random deviates calculated using a Bayesian analysis with the JP.

d\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_pdf: predictive density function from a Bayesian analysis with the flat prior.
- rh\_ml\_pdf: predictive density function from a Bayesian analysis with the RHP-MLE prior.
- jp\_pdf: predictive density function from a Bayesian analysis with the JP.

p\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_cdf: predictive distribution function from a Bayesian analysis with the flat prior.
- rh\_ml\_cdf: predictive distribution function from a Bayesian analysis with the RHP-MLE prior.
- jp\_cdf: predictive distribution function from a Bayesian analysis with the JP.

These additional predictive distributions are included for comparison with the calibrating prior model. They generally give less good reliability than the calibrating prior.

# **Details (non-homogeneous models)**

This model is not homogeneous, i.e. it does not have a transitive transformation group, and so there is no right Haar prior and no method for generating exactly reliable predictions. The cp outputs are generated using a prior that has been shown in tests to give reasonable reliability. See Jewson et al. (2025a) for discussion of the prior and test results. For non-homogeneous models, reliability is generally poor for small sample sizes (<20), but is still much better than maximum likelihood. For small sample sizes, it is advisable to check the level of reliability using the routine reltest.

### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),

- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

216 gev\_f1fa

# **Examples**

```
#
# example 1
shape=-0.4
x=fitdistcp::d110gev_example_data_v1
p=c(1:9)/10
q=qgev_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgev_cp)",
main="GEVD: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue",lwd=2)
cat(" ml_params=",q$ml_params,"\n")
```

gev\_f1fa

The first derivative of the density

# Description

The first derivative of the density

# Usage

```
gev_f1fa(x, v1, v2, v3)
```

# Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

# Value

Vector

gev\_f2fa 217

gev	f2fa
KCV_	1410

The second derivative of the density

# Description

The second derivative of the density

### Usage

```
gev_f2fa(x, v1, v2, v3)
```

# Arguments

X	a vector of training data values
1	C 4

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

gev	fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### **Description**

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_fd(x, v1, v2, v3)
```

# Arguments

a vector of training data varies	X	a vector of training data values	
----------------------------------	---	----------------------------------	--

v1 first parameterv2 second parameterv3 third parameter

#### Value

Vector

gev_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gev_fdd(x, v1, v2, v3)
```

# Arguments

x a vector of training data	values
-----------------------------	--------

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

```
gev_k12_ppm_minusloglik
```

Temporary dummy for one of the ppm models

# Description

Temporary dummy for one of the ppm models

### Usage

```
gev_k12_ppm_minusloglik(x)
```

#### **Arguments**

x a vector of training data values

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

gev\_k3\_cp

Generalized Extreme Value Distribution with Known Shape, Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics
- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qgev_k3_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    fdalpha = 0.01,
    kshape = 0,
    means = FALSE,
    waicscores = FALSE,
    pdf = FALSE,
    dmgs = TRUE,
    rust = FALSE,
    nrust = 1e+05,
    debug = FALSE
)

rgev_k3_cp(n, x, kshape = 0, rust = FALSE, mlcp = TRUE, debug = FALSE)
```

```
dgev_k3_cp(x, y = x, kshape = 0, rust = FALSE, nrust = 1000, debug = FALSE)
pgev_k3_cp(x, y = x, kshape = 0, rust = FALSE, nrust = 1000, debug = FALSE)
tgev_k3_cp(n, x, kshape = 0, debug = FALSE)
```

#### **Arguments**

X	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
fdalpha	if pdf=TRUE, the fractional delta used for numerical derivatives with respect to probability, for calculating the pdf as a function of quantiles
kshape	the known shape parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
pdf	logical that indicates whether to run additional calculations and return density functions evaluated at quantiles specified by the input probabilities (longer runtime)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.

• cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The GEV distribution with known shape has distribution function

$$F(x; \mu, \sigma) = \exp(-t(x; \mu, \sigma))$$

where

$$t(x; \mu, \sigma) = \begin{cases} \left[1 + \xi \left(\frac{x - \mu}{\sigma}\right)\right]^{-1/\xi} & \text{if } \xi \neq 0 \\ \exp\left(-\frac{x - \mu}{\sigma}\right) & \text{if } \xi = 0 \end{cases}$$

where x is the random variable,  $\mu, \sigma > 0$  are the parameters and  $\xi$  is known (hence the k3 in the name).

The calibrating prior we use is given by

$$\pi(\mu,\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Optional Return Values (EVT models only)**

q\*\*\* optionally returns the following, for EVT models only:

cp\_pdf: the density function at quantiles corresponding to input probabilities p. We provide this for EVD models, because direct estimation of the density function using the DMGS density equation is not possible.

### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (1st\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),

226 gev\_k3\_f1fa

- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
# example 1
kshape=-0.4
x=fitdistcp::d053gev_k3_example_data_v1
p=c(1:9)/10
q=qgev_k3_cp(x,p,kshape=kshape,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgev_k3_cp)",
main="GEV: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
muhat=q$ml_params[1]
sghat=q$ml_params[2]
xi=kshape
qmax=ifelse(xi<0,muhat-sghat/xi,Inf)</pre>
cat(" ml_params=",q$ml_params,",")
cat(" qmax=",qmax,"\n")
```

gev\_k3\_f1fa

The first derivative of the density

# Description

The first derivative of the density

gev\_k3\_f2fa 227

# Usage

```
gev_k3_f1fa(x, v1, v2, kshape)
```

# Arguments

x a vector of training data values

v1 first parameter

v2 second parameter

kshape the known shape parameter

#### Value

Vector

gev\_k3\_f2fa

The second derivative of the density

# Description

The second derivative of the density

# Usage

# Arguments

x a vector of training data values

v1 first parameter

v2 second parameter

kshape the known shape parameter

#### Value

Matrix

228 gev\_k3\_fdd

gev_k3_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gev_k3_fd(x, v1, v2, v3)
```

# Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

# Value

Vector

gev_k3_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_k3_fdd(x, v1, v2, v3)
```

# Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

# Value

Matrix

gev\_k3\_ldda 229

COV	ト3	ldda
gev	ĸs	Tuua

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

# Usage

```
gev_k3_ldda(x, v1, v2, kshape)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

kshape the known shape parameter

#### Value

Matrix

gev\_k3\_lddda

The third derivative of the normalized log-likelihood

### **Description**

The third derivative of the normalized log-likelihood

#### Usage

```
gev_k3_lddda(x, v1, v2, kshape)
```

#### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

kshape the known shape parameter

### Value

3d array

230 gev\_k3\_logfdd

gev\_k3\_logf

Logf for RUST

# Description

Logf for RUST

#### Usage

```
gev_k3_logf(params, x, kshape)
```

#### **Arguments**

params model parameters for calculating logf
x a vector of training data values
kshape the known shape parameter

#### Value

Scalar value.

gev\_k3\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gev_k3_logfdd(x, v1, v2, v3)
```

#### **Arguments**

a vector or training data value	X	a vector of trainir	1g data values
---------------------------------	---	---------------------	----------------

v1 first parameterv2 second parameterv3 third parameter

# Value

Matrix

gev\_k3\_logfddd 231

gev_k3_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gev_k3_logfddd(x, v1, v2, v3)
```

# Arguments

Χ	a vector of training data values
---	----------------------------------

v1 first parameterv2 second parameterv3 third parameter

#### Value

3d array

# Description

log-likelihood function

# Usage

```
gev_k3_loglik(vv, x, kshape)
```

# **Arguments**

vv parameters

x a vector of training data values kshape the known shape parameter

# Value

Scalar

gev\_k3\_mu1fa

gev_k3_means	MLE and RHP means
--------------	-------------------

#### **Description**

MLE and RHP means

#### Usage

```
gev_k3_means(means, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 2, kshape)
```

# **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data
dim number of parameters

kshape the known shape parameter

#### Value

Two scalars

gev_k3_mu1fa	Minus the first derivative of the cdf, at alpha

# Description

Minus the first derivative of the cdf, at alpha

### Usage

```
gev_k3_mu1fa(alpha, v1, v2, kshape)
```

# Arguments

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

kshape the known shape parameter

gev\_k3\_mu2fa 233

#### Value

Vector

gev\_k3\_mu2fa

Minus the second derivative of the cdf, at alpha

### Description

Minus the second derivative of the cdf, at alpha

### Usage

```
gev_k3_mu2fa(alpha, v1, v2, kshape)
```

#### **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameterv2 second parameter

kshape the known shape parameter

#### Value

Matrix

gev\_k3\_pd First derivative of the cdf Created by Stephen Jewson using Deriv() by
Andrew Clausen and Serguei Sokol

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gev_k3_pd(x, v1, v2, v3)
```

# Arguments

x a vector of training data values

v1 first parameterv2 second parameterv3 third parameter

gev\_k3\_waic

#### Value

Vector

gev\_k3\_pdd Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gev_k3_pdd(x, v1, v2, v3)
```

#### **Arguments**

x a vector of training data values

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

gev\_k3\_waic Waic

# Description

Waic

#### Usage

```
gev_k3_waic(waicscores, x, v1hat, v2hat, kshape, lddi, lddd, lambdad)
```

gev\_ld12a 235

### **Arguments**

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data values

v1hat first parameter v2hat second parameter

kshape the known shape parameter

lddi inverse observed information matrix lddd third derivative of log-likelihood

lambdad derivative of the log prior

# Value

Two numeric values.

gev\_ld12a

The combined derivative of the normalized log-likelihood

# Description

The combined derivative of the normalized log-likelihood

# Usage

```
gev_ld12a(x, v1, v2, v3)
```

#### **Arguments**

x a vector of training data values

v1 first parameterv2 second parameterv3 third parameter

#### Value

3d array

236 gev\_ldda

gev\_lda

The first derivative of the normalized log-likelihood

# Description

The first derivative of the normalized log-likelihood

# Usage

```
gev_lda(x, v1, v2, v3)
```

# Arguments

x a vector of training data values

v1 first parameterv2 second parameterv3 third parameter

#### Value

Vector

gev\_ldda

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

# Usage

```
gev_ldda(x, v1, v2, v3)
```

# Arguments

x a vector of training data val	ues
---------------------------------	-----

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

gev\_lddda 237

gev\_lddda

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

# Usage

```
gev_lddda(x, v1, v2, v3)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

v3 third parameter

#### Value

3d array

gev\_logf

 $Log f for \, RUST$ 

# Description

Logf for RUST

# Usage

```
gev_logf(params, x)
```

# Arguments

params model parameters for calculating logf x a vector of training data values

### Value

Scalar value.

238 gev\_logfdd

gev_logfd	First derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gev_logfd(x, v1, v2, v3)
```

# Arguments

х	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

# Value

Vector

gev_logfdd	Second derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_logfdd(x, v1, v2, v3)
```

# Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

# Value

Matrix

gev\_logfddd 239

gev_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gev_logfddd(x, v1, v2, v3)
```

# Arguments

x a vector of training data values

v1 first parameterv2 second parameterv3 third parameter

#### Value

3d array

gev\_loglik

log-likelihood function

# Description

log-likelihood function

# Usage

```
gev_loglik(vv, x)
```

#### **Arguments**

vv parameters

x a vector of training data values

# Value

Scalar

240 gev\_means

gev\_means

Analytical Expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

# Description

Analytical Expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

# Usage

```
gev_means(
  means,
  ml_params,
  lddi,
  lddd,
  lambdad_rh_flat,
  lambdad_custom,
  nx,
  dim = 3
)
```

#### **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihood

lambdad\_rh\_flat

derivative of the log CRHP-FLAT prior

lambdad\_custom custom value of the derivative of the log prior

nx length of training data dim number of parameters

### Value

Two scalars

gev\_mu1fa 241

gev	mıı1	fa
REV	IIIU I	ıα

Minus the first derivative of the cdf, at alpha

# Description

Minus the first derivative of the cdf, at alpha

# Usage

```
gev_mu1fa(alpha, v1, v2, v3)
```

# Arguments

alpha	a vector of values	of alpha (one	e minus probability)

v1 first parameterv2 second parameterv3 third parameter

#### Value

Vector

Minus the second derivative of the cdf, at alpha

# Description

Minus the second derivative of the cdf, at alpha

# Usage

```
gev_mu2fa(alpha, v1, v2, v3)
```

# Arguments

alpha	a vector	of values	of alpha	one minus	probability)
атрна	a vector	or varues	or arpira i	One minus	probability,

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

gev_p123_checkmle	Check MLE
-------------------	-----------

#### **Description**

Check MLE

#### Usage

```
gev_p123_checkmle(ml_params, minxi = -1, maxxi = 1, t1, t2, t3)
```

#### **Arguments**

m1_params	parameters
minxi	minimum value of shape parameter xi
maxxi	maximum value of shape parameter xi
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape

#### Value

No return value (just a message to the screen).

gev_p123_cp	Generalized Extreme Value Distribution with Three Predictors, Pre-
	dictions based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y

• t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qgev_p123_cp(
 х,
  t1,
  t2,
  t3,
  t01 = NA,
  t02 = NA
  t03 = NA,
 n01 = NA,
 n02 = NA
 n03 = NA,
 p = seq(0.1, 0.9, 0.1),
  ics = c(0, 0, 0, 0, 0, 0),
  fdalpha = 0.01,
 minxi = -1,
 maxxi = 1,
 means = FALSE,
 waicscores = FALSE,
  extramodels = FALSE,
 pdf = FALSE,
 dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 centering = TRUE,
  debug = FALSE
)
rgev_p123_cp(
 n,
 Х,
  t1,
  t2,
  t3,
  t01 = NA,
  t02 = NA
  t03 = NA,
  n01 = NA,
 n02 = NA,
```

```
n03 = NA,
  ics = c(0, 0, 0, 0, 0, 0),
 minxi = -1,
 \max x i = 1,
  extramodels = FALSE,
  rust = FALSE,
 mlcp = TRUE,
 centering = TRUE,
 debug = FALSE
)
dgev_p123_cp(
 Х,
  t1,
  t2,
  t3,
  t01 = NA,
  t02 = NA,
  t03 = NA,
 n01 = NA
 n02 = NA,
 n03 = NA,
 y = x,
  ics = c(0, 0, 0, 0, 0, 0),
 minxi = -1,
 \max x i = 1,
 extramodels = FALSE,
 rust = FALSE,
 nrust = 10,
 centering = TRUE,
  debug = FALSE
)
pgev_p123_cp(
 х,
  t1,
  t2,
  t3,
  t01 = NA,
 t02 = NA,
 t03 = NA,
 n01 = NA,
 n02 = NA,
 n03 = NA,
 y = x,
  ics = c(0, 0, 0, 0, 0, 0),
 minxi = -1,
 maxxi = 1,
```

```
extramodels = FALSE,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE
)
tgev_p123_cp(
 n,
 Х,
 t1,
 t2,
 t3,
 ics = c(0, 0, 0, 0, 0, 0),
 extramodels = FALSE,
 debug = FALSE
)
```

# Arguments

x	a vector of training data values
t1	a vector of predictors for the mean, such that $length(t1)=length(x)$
t2	a vector of predictors for the sd, such that length(t2)=length(x)
t3	a vector of predictors for the shape, such that $length(t3)=length(x)$
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
n01	an index for the predictor (specify either t01 or n01 but not both)
n02	an index for the predictor (specify either t02 or n02 but not both)
n03	an index for the predictor (specify either t03 or n03 but not both)
р	a vector of probabilities at which to generate predictive quantiles
ics	initial conditions for the maximum likelihood search
fdalpha	if pdf=TRUE, the fractional delta used for numerical derivatives with respect to probability, for calculating the pdf as a function of quantiles
minxi	the minimum allowed value of the shape parameter (decrease with caution)
maxxi	the maximum allowed value of the shape parameter (increase with caution)
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
extramodels	logical that indicates whether to run additional calculations and add three additional prediction models (longer runtime)

pdf	logical that indicates whether to run additional calculations and return density functions evaluated at quantiles specified by the input probabilities (longer runtime)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
centering	logical that indicates whether the predictor should be centered
debug	logical for turning on debug messages
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.

• cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The GEV distribution with three predictors has distribution function

$$F(x; a_1, b_1, a_2, b_2, a_3, b_3) = \exp(-t(x; \mu(a_1, b_1), \sigma(a_2, b_2), \xi(a_3, b_3)))$$

where

$$t(x; \mu(a_1, b_1), \sigma(a_2, b_2), \xi(a_3, b_3)) = \begin{cases} \left[1 + \xi(a_3, b_3) \left(\frac{x - \mu(a_1, b_1)}{\sigma(a_2, b_2)}\right)\right]^{-1/\xi(a_3, b_3)} & \text{if } \xi(a_3, b_3) \neq 0 \\ \exp\left(-\frac{x - \mu(a_1, b_1)}{\sigma(a_2, b_2)}\right) & \text{if } \xi(a_3, b_3) = 0 \end{cases}$$

where x is the random variable,  $\mu=a_1+b_1t_1$  is the location parameter, modelled as a function of parameters  $a_1,b_1$  and predictor  $t_1,\,\sigma=e^{a_2+b_2t_2}$  is the scale parameter, modelled as a function of parameters  $a_2,b_2$  and predictor  $t_2$ , and  $\xi=a_3+b_3t_3$  is the shape parameter, modelled as a function of parameters  $a_3,b_3$  and predictor  $t_3$ .

The calibrating prior we use is given by

$$\pi(a_1,b_1,a_2,b_2,a_3,b_3) \propto 1$$

as given in Jewson et al. (2025).

The code will switch to maximum likelihood prediction if the input data gives a maximum likelihood value for the shape parameter that lies outside the range (minxi,maxxi), since outside this range there may be numerical problems. If this happens, it is reported in the revert2ml flag. Such values seldom occur in real observed data for maxima.

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

#### If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

#### If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

# Optional Return Values (EVT models only)

q\*\*\*\* optionally returns the following, for EVT models only:

• cp\_pdf: the density function at quantiles corresponding to input probabilities p. We provide this for EVD models, because direct estimation of the density function using the DMGS density equation is not possible.

#### **Details (non-homogeneous models)**

This model is not homogeneous, i.e. it does not have a transitive transformation group, and so there is no right Haar prior and no method for generating exactly reliable predictions. The cp outputs are generated using a prior that has been shown in tests to give reasonable reliability. See Jewson et al. (2025a) for discussion of the prior and test results. For non-homogeneous models, reliability is generally poor for small sample sizes (<20), but is still much better than maximum likelihood. For small sample sizes, it is advisable to check the level of reliability using the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

· Cauchy (cauchy),

- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

gev\_p123\_f1fa 251

#### **Examples**

```
# example 1
x=fitdistcp::d152gev_p123_example_data_v1_x
tt=fitdistcp::d152gev_p123_example_data_v1_t
t1=tt[,1]
t2=tt[,2]
t3=tt[,3]
p=c(1:9)/10
n01=10
n02=10
n03=10
q = qgev_p 123_cp(x=x,t1=t1,t2=t2,t3=t3,n01=n01,n02=n02,n03=n03,t01=NA,t02=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA,t03=NA
p=p,rust=FALSE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgev_p123_cp)",
main="GEVD w/ p123: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
cat(" ml_params=",q$ml_params,"\n")
```

gev\_p123\_f1fa

The first derivative of the density for DMGS

### Description

The first derivative of the density for DMGS

#### Usage

```
gev_p123_f1fa(x, t01, t02, t03, v1, v2, v3, v4, v5, v6)
```

#### **Arguments**

x	a vector of training data values
t01	a single value of the predictor (specify either $t01$ or $n01$ but not both)
t02	a single value of the predictor (specify either $t02$ or $n02$ but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

252 gev\_p123\_f1fw

# Value

Vector

gev\_p123\_f1fw

The first derivative of the density for WAIC

# Description

The first derivative of the density for WAIC

# Usage

```
gev_p123_f1fw(x, t1, t2, t3, v1, v2, v3, v4, v5, v6)
```

# Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

# Value

Vector

gev\_p123\_f2fa 253

σ <sub>E</sub> V	n1	23	f2fa

The second derivative of the density for DMGS

# Description

The second derivative of the density for DMGS

## Usage

```
gev_p123_f2fa(x, t01, t02, t03, v1, v2, v3, v4, v5, v6)
```

## Arguments

X	a vector of training data values
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

#### Value

Matrix

gev\_p123\_f2fw

The second derivative of the density for WAIC

# Description

The second derivative of the density for WAIC

```
gev_p123_f2fw(x, t1, t2, t3, v1, v2, v3, v4, v5, v6)
```

254 gev\_p123\_fd

# Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

## Value

Matrix

gev_p123_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	Tit () by Thaten Clausen and Serguet Sonot

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p123_fd(x, t1, t2, t3, v1, v2, v3, v4, v5, v6)
```

# Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

## Value

Vector

gev\_p123\_fdd 255

gev_p123_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gev_p123_fdd(x, t1, t2, t3, v1, v2, v3, v4, v5, v6)
```

# Arguments

Χ	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

## Value

Matrix

gev_p123_ldda	The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

```
gev_p123_ldda(x, t1, t2, t3, v1, v2, v3, v4, v5, v6)
```

256 gev\_p123\_lddda

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

## Value

Matrix

# Description

The third derivative of the normalized log-likelihood

# Usage

```
gev_p123_lddda(x, t1, t2, t3, v1, v2, v3, v4, v5, v6)
```

# Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

## Value

3d array

gev\_p123\_logf 257

	100	1 C	
gev	_p123_	logt	

Logf for RUST

## Description

Logf for RUST

## Usage

```
gev_p123_logf(params, x, t1, t2, t3)
```

#### **Arguments**

params	model parameters for calculating logf
X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape

#### Value

Scalar value.

gev	p123	logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p123_logfdd(x, t1, t2, t3, v1, v2, v3, v4, v5, v6)
```

# Arguments

Χ	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter

258 gev\_p123\_logfddd

v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

## Value

Matrix

gev_p123_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p123_logfddd(x, t1, t2, t3, v1, v2, v3, v4, v5, v6)
```

# Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

## Value

3d array

gev\_p123\_loglik 259

gev_p123_loglik observed log-likelihood function	
--------------------------------------------------	--

## Description

observed log-likelihood function

## Usage

```
gev_p123_loglik(vv, x, t1, t2, t3)
```

# Arguments

VV	parameters
X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape

#### Value

Scalar

gev_p123_means	Analytical expressions for Predictive Means RHP mean based on the
	expectation of DMGS equation 2.1

# Description

Analytical expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

# Usage

```
gev_p123_means(means, t01, t02, t03, ml_params, nx)
```

# Arguments

means	logical that indicates whether to return analytical estimates for the distribution means (longer runtime)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
ml_params	parameters
nx	length of training data

260 gev\_p123\_mu1fa

## Value

Two scalars

gev_p'	123	mu1	fa
--------	-----	-----	----

Minus the first derivative of the cdf, at alpha

# Description

Minus the first derivative of the cdf, at alpha

# Usage

```
gev_p123_mu1fa(alpha, t01, t02, t03, v1, v2, v3, v4, v5, v6)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

## Value

Vector

gev\_p123\_mu2fa 261

gev_p123_mu2fa	Minus the second derivative of the cdf, at alpha

# Description

Minus the second derivative of the cdf, at alpha

## Usage

```
gev_p123_mu2fa(alpha, t01, t02, t03, v1, v2, v3, v4, v5, v6)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either $t02$ or $n02$ but not both)
t03	a single value of the predictor (specify either $t03$ or $n03$ but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

## Value

Matrix

gev_p123_pd First derivative of the cdf Created by Stephen Jewson using Deriv()  Andrew Clausen and Serguei Sokol	) by
-------------------------------------------------------------------------------------------------------------------	------

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gev_p123_pd(x, t1, t2, t3, v1, v2, v3, v4, v5, v6)
```

262 gev\_p123\_pdd

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

## Value

Vector

gev_p123_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p123_pdd(x, t1, t2, t3, v1, v2, v3, v4, v5, v6)
```

## Arguments

x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

#### Value

Matrix

```
gev_p123_predictordata
```

Predicted Parameter and Generalized Residuals

## Description

Predicted Parameter and Generalized Residuals

## Usage

```
gev_p123_predictordata(x, t1, t2, t3, t01, t02, t03, params)
```

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
params	model parameters for calculating logf

#### Value

Two vectors

```
gev_p123_setics Set initial conditions
```

## Description

Set initial conditions

# Usage

```
gev_p123_setics(x, t1, t2, t3, ics)
```

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
ics	initial conditions for the maximum likelihood search

264 *gev\_p123\_waic* 

# Value

Vector

gev\_p123\_waic Waic

# Description

Waic

# Usage

```
gev_p123_waic(
  waicscores,
  Х,
  t1,
  t2,
  t3,
  v1h,
  v2h,
  v3h,
  v4h,
  v5h,
  v6h,
  lddi,
  lddd,
  lambdad
)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores
	(longer runtime)
X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
v1h	first parameter
v2h	second parameter
v3h	third parameter
v4h	fourth parameter
v5h	fifth parameter
v6h	sixth parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior

gev\_p12k3\_f1fa 265

## Value

Two numeric values.

gev	n1	21/3	£1	fa
267	υı	ZNJ	- 1 1	ıa

The first derivative of the density for DMGS

# Description

The first derivative of the density for DMGS

#### Usage

```
gev_p12k3_f1fa(x, t01, t02, v1, v2, v3, v4, kshape)
```

## Arguments

X	a vector of training data values
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
kshape	the known shape parameter

#### Value

Vector

gev\_p12k3\_f1fw

The first derivative of the density for WAIC

# Description

The first derivative of the density for WAIC

```
gev_p12k3_f1fw(x, t1, t2, v1, v2, v3, v4, kshape)
```

266 gev\_p12k3\_f2fa

# Arguments

x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
kshape	the known shape parameter

## Value

Vector

# Description

The second derivative of the density for DMGS

# Usage

```
gev_p12k3_f2fa(x, t01, t02, v1, v2, v3, v4, kshape)
```

# Arguments

х	a vector of training data values
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
kshape	the known shape parameter

## Value

Matrix

gev\_p12k3\_f2fw 267

	gev	p12k3	_f2fw
--	-----	-------	-------

The second derivative of the density for WAIC

## Description

The second derivative of the density for WAIC

# Usage

```
gev_p12k3_f2fw(x, t1, t2, v1, v2, v3, v4, kshape)
```

## Arguments

x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
kshape	the known shape parameter

#### Value

Matrix

gev_p12k3_fd	First derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gev_p12k3_fd(x, t1, t2, v1, v2, v3, v4, v5)
```

268 gev\_p12k3\_fdd

# Arguments

x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

## Value

Vector

gev_p12k3_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p12k3_fdd(x, t1, t2, v1, v2, v3, v4, v5)
```

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

## Value

Matrix

gev\_p12k3\_ldda 269

σeν	n1	2k3	ldda
200	$\nu$	2NJ_	_±uua

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

## Usage

```
gev_p12k3_ldda(x, t1, t2, v1, v2, v3, v4, kshape)
```

#### **Arguments**

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
kshape	the known shape parameter

## Value

Matrix

	4 0		111
$\alpha = v$	กเป	k	.ddda
5 C V .	_ ( )   ( )	$\sim$ $\sim$ $\sim$	.uuuu

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

```
gev_p12k3_lddda(x, t1, t2, v1, v2, v3, v4, kshape)
```

270 gev\_p12k3\_logfdd

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
kshape	the known shape parameter

#### Value

3d array

gev_p12k3_logfdd	Second derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gev_p12k3_logfdd(x, t1, t2, v1, v2, v3, v4, v5)
```

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

## Value

Matrix

gev\_p12k3\_logfddd 271

gev_p12k3_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p12k3_logfddd(x, t1, t2, v1, v2, v3, v4, v5)
```

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

## Value

3d array

gev_p12k3_mu1fa	Minus the first derivative of the cdf, at alpha	
gev_p12k3_md11a	minus ine jirsi derivative oj ine caj, di dipila	

# Description

Minus the first derivative of the cdf, at alpha

```
gev_p12k3_mu1fa(alpha, t01, t02, v1, v2, v3, v4, kshape)
```

272 gev\_p12k3\_mu2fa

# Arguments

alpha	a vector of values of alpha (one minus probability)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
kshape	the known shape parameter

## Value

Vector

gev_p12k3_mu2fa	Minus the second derivative of the cdf, at alpha	

# Description

Minus the second derivative of the cdf, at alpha

# Usage

```
gev_p12k3_mu2fa(alpha, t01, t02, v1, v2, v3, v4, kshape)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
t01	a single value of the predictor (specify either $t01$ or $n01$ but not both)
t02	a single value of the predictor (specify either $t02$ or $n02$ but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
kshape	the known shape parameter

## Value

Matrix

gev\_p12k3\_pd 273

gev_p12k3_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p12k3_pd(x, t1, t2, v1, v2, v3, v4, v5)
```

## Arguments

Χ	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

#### Value

Vector

gev_p12k3_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gev_p12k3_pdd(x, t1, t2, v1, v2, v3, v4, v5)
```

274 gev\_p12\_checkmle

## Arguments

x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
	t1 t2 v1 v2 v3 v4

#### Value

Matrix

# Description

Check MLE

## Usage

```
gev_p12_checkmle(ml_params, minxi = -1, maxxi = 1)
```

# Arguments

ml\_params parameters

minxi minimum value of shape parameter xi maxxi maximum value of shape parameter xi

## Value

No return value (just a message to the screen).

gev\_p12\_cp

Generalized Extreme Value Distribution with Two Predictors, Predictions based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qgev_p12_cp(
    x,
    t1,
    t2,
    t01 = NA,
    t02 = NA,
    n01 = NA,
    n02 = NA,
    p = seq(0.1, 0.9, 0.1),
    ics = c(0, 0, 0, 0, 0),
    fdalpha = 0.01,
    minxi = -1,
    maxxi = 1,
    means = FALSE,
    waicscores = FALSE,
```

```
extramodels = FALSE,
  pdf = FALSE,
  dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 predictordata = TRUE,
 centering = TRUE,
 debug = FALSE
)
rgev_p12_cp(
 n,
 х,
  t1,
  t2,
  t01 = NA,
  t02 = NA,
 n01 = NA,
 n02 = NA,
 ics = c(0, 0, 0, 0, 0),
 minxi = -1,
 maxxi = 1,
 extramodels = FALSE,
 rust = FALSE,
 mlcp = TRUE,
 centering = TRUE,
 debug = FALSE
)
dgev_p12_cp(
 Χ,
  t1,
 t2,
  t01 = NA,
  t02 = NA,
 n01 = NA,
 n02 = NA,
 y = x,
  ics = c(0, 0, 0, 0, 0),
 minxi = -1,
 maxxi = 1,
 extramodels = FALSE,
 rust = FALSE,
 nrust = 10,
 centering = TRUE,
 debug = FALSE
)
```

```
pgev_p12_cp(
 Χ,
 t1,
 t2,
 t01 = NA,
 t02 = NA,
 n01 = NA,
 n02 = NA,
 y = x,
 ics = c(0, 0, 0, 0, 0),
 minxi = -1,
 maxxi = 1,
 extramodels = FALSE,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE
)
tgev_p12_cp(
 n,
 Х,
 t1,
  t2,
 ics = c(0, 0, 0, 0, 0),
 extramodels = FALSE,
 debug = FALSE
)
```

# Arguments

Х	a vector of training data values
t1	a vector of predictors for the mean, such that length(t1)=length(x)
t2	a vector of predictors for the sd, such that length(t2)=length(x)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
n01	an index for the predictor (specify either t01 or n01 but not both)
n02	an index for the predictor (specify either t02 or n02 but not both)
р	a vector of probabilities at which to generate predictive quantiles
ics	initial conditions for the maximum likelihood search
fdalpha	if pdf=TRUE, the fractional delta used for numerical derivatives with respect to probability, for calculating the pdf as a function of quantiles
minxi	the minimum allowed value of the shape parameter (decrease with caution)
maxxi	the maximum allowed value of the shape parameter (increase with caution)
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)

logical that indicates whether to run additional calculations and return estimates waicscores for the WAIC1 and WAIC2 scores (longer runtime) extramodels logical that indicates whether to run additional calculations and add three additional prediction models (longer runtime) pdf logical that indicates whether to run additional calculations and return density functions evaluated at quantiles specified by the input probabilities (longer rundmgs logical that indicates whether DMGS calculations should be run or not (longer run time) logical that indicates whether RUST-based posterior sampling calculations should rust be run or not (longer run time) the number of posterior samples used in the RUST calculations nrust predictordata logical that indicates whether predictordata should be calculated logical that indicates whether the predictor should be centered centering debug logical for turning on debug messages n the number of random samples required mlcp logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST) a vector of values at which to calculate the density and distribution functions У

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The GEV distribution with two predictors has distribution function

$$F(x; a_1, b_1, a_2, b_2, \xi) = \exp(-t(x; \mu(a_1, b_1), \sigma(a_2, b_2), \xi))$$

where

$$t(x; \mu(a_1, b_1), \sigma(a_2, b_2), \xi) = \begin{cases} \left[ 1 + \xi \left( \frac{x - \mu(a_1, b_1)}{\sigma(a_2, b_2)} \right) \right]^{-1/\xi} & \text{if } \xi \neq 0 \\ \exp\left( -\frac{x - \mu(a_1, b_1)}{\sigma(a_2, b_2)} \right) & \text{if } \xi = 0 \end{cases}$$

where x is the random variable,  $\mu = a_1 + b_1 t_1$  is the location parameter, modelled as a function of parameters  $a_1, b_1$  and predictor  $t_1$ ,  $\sigma = e^{a_2 + b_2 t_2}$  is the scale parameter, modelled as a function of parameters  $a_2, b_2$  and predictor  $t_2$ , and  $\xi$  is the shape parameter.

The calibrating prior we use is given by

$$\pi(a_1, b_1, a_2, b_2, \xi) \propto 1$$

as given in Jewson et al. (2025).

The code will switch to maximum likelihood prediction if the input data gives a maximum likelihood value for the shape parameter that lies outside the range (minxi,maxxi), since outside this range there may be numerical problems. If this happens, it is reported in the revert2ml flag. Such values seldom occur in real observed data for maxima.

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Optional Return Values (EVT models only)**

q\*\*\*\* optionally returns the following, for EVT models only:

cp\_pdf: the density function at quantiles corresponding to input probabilities p. We provide this for EVD models, because direct estimation of the density function using the DMGS density equation is not possible.

#### **Details (non-homogeneous models)**

This model is not homogeneous, i.e. it does not have a transitive transformation group, and so there is no right Haar prior and no method for generating exactly reliable predictions. The cp outputs are generated using a prior that has been shown in tests to give reasonable reliability. See Jewson et al. (2025a) for discussion of the prior and test results. For non-homogeneous models, reliability is generally poor for small sample sizes (<20), but is still much better than maximum likelihood. For small sample sizes, it is advisable to check the level of reliability using the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

• Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),

gev\_p12\_f1fa 283

- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
# example 1
x=fitdistcp::d151gev_p12_example_data_v1_x
tt=fitdistcp::d151gev_p12_example_data_v1_t
t1=tt[,1]
t2=tt[,2]
p=c(1:9)/10
n01=10
n02=10
 \\ q = qgev_p \\ 12\_cp(x = x, t1 = t1, t2 = t2, n01 = n01, n02 = n02, t01 = NA, t02 = NA, p = p, rust = TRUE, nrust = 1000) \\ \\ q = qgev_p \\ 12\_cp(x = x, t1 = t1, t2 = t2, n01 = n01, n02 = n02, t01 = NA, t02 = NA, p = p, rust = TRUE, nrust = 1000) \\ \\ q = qgev_p \\ 12\_cp(x = x, t1 = t1, t2 = t2, n01 = n01, n02 = n02, t01 = NA, t02 = NA, p = p, rust = TRUE, nrust = 1000) \\ \\ q = qgev_p \\ 12\_cp(x = x, t1 = t1, t2 = t2, n01 = n01, n02 = n02, t01 = NA, t02 = NA, p = p, rust = TRUE, nrust = 1000) \\ \\ q = qgev_p \\ 12\_cp(x = x, t1 = t1, t2 = t2, n01 = n01, n02 = n02, t01 = NA, t02 = NA, p = p, rust = TRUE, nrust = 1000) \\ \\ q = qgev_p \\ 12\_cp(x = x, t1 = t1, t2 = t2, n01 = n02, t01 = NA, t02 = NA, p = p, rust = TRUE, nrust = 1000) \\ \\ q = qgev_p \\ 12\_cp(x = x, t1 = t1, t2 = t2, t01 = NA, t02 
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgev_p12_cp)",
main="GEVD w/ p12: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue",lwd=2)
cat(" ml_params=",q$ml_params,"\n")
```

gev\_p12\_f1fa

The first derivative of the density for DMGS

#### **Description**

The first derivative of the density for DMGS

#### Usage

```
gev_p12_f1fa(x, t01, t02, v1, v2, v3, v4, v5)
```

# Arguments

X	a vector of training data values
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either $t02$ or $n02$ but not both)
v1	first parameter

284 gev\_p12\_f1fw

v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

#### Value

Vector

The first derivative of the density for WAIC

# Description

The first derivative of the density for WAIC

# Usage

```
gev_p12_f1fw(x, t1, t2, v1, v2, v3, v4, v5)
```

# Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

## Value

Vector

gev\_p12\_f2fa 285

	1	$\sim$	$C \cap C$	_
gev_	ı q.	۷_	.T Z T	а

The second derivative of the density for DMGS

# Description

The second derivative of the density for DMGS

## Usage

```
gev_p12_f2fa(x, t01, t02, v1, v2, v3, v4, v5)
```

# Arguments

X	a vector of training data values
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

## Value

Matrix

	- 4	$\sim$	COC
gev	nι	/	_f2fw

The second derivative of the density for WAIC

# Description

The second derivative of the density for WAIC

```
gev_p12_f2fw(x, t1, t2, v1, v2, v3, v4, v5)
```

286 gev\_p12\_fd

# Arguments

Х	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

## Value

Matrix

gev_p12_fd	First derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p12_fd(x, t1, t2, v1, v2, v3, v4, v5)
```

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

## Value

Vector

gev\_p12\_fdd 287

gev_p12_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p12_fdd(x, t1, t2, v1, v2, v3, v4, v5)
```

# Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

## Value

Matrix

gev_p12_ldda	The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

```
gev_p12_ldda(x, t1, t2, v1, v2, v3, v4, v5)
```

288 gev\_p12\_lddda

# Arguments

Х	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

#### Value

Matrix

# Description

The third derivative of the normalized log-likelihood

# Usage

```
gev_p12_lddda(x, t1, t2, v1, v2, v3, v4, v5)
```

# **Arguments**

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

## Value

3d array

gev\_p12\_logf 289

gev_	p1	2 1	ogf
5°'-	м.		· ob ·

Logf for RUST

### Description

Logf for RUST

## Usage

```
gev_p12_logf(params, x, t1, t2)
```

### Arguments

params	model parameters for calculating logf
Х	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd

#### Value

Scalar value.

gev_p12_logfdd	Second derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_p12_logfdd(x, t1, t2, v1, v2, v3, v4, v5)
```

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

290 gev\_p12\_loglik

### Value

Matrix

gev_p12_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gev_p12_logfddd(x, t1, t2, v1, v2, v3, v4, v5)
```

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

### Value

3d array

gev_p12_loglik observed log-likelihood function
-------------------------------------------------

## Description

observed log-likelihood function

```
gev_p12_loglik(vv, x, t1, t2)
```

gev\_p12\_means 291

## Arguments

VV	parameters
X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd

### Value

Scalar

# Description

Analytical expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

## Usage

```
gev_p12_means(means, t01, t02, ml_params, nx)
```

## Arguments

means	logical that indicates whether to return analytical estimates for the distribution means (longer runtime)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
ml_params	parameters
nx	length of training data

#### Value

Two scalars

292 gev\_p12\_mu2fa

	1	$\sim$	1	C -
gev	рι	2	mu i	та

Minus the first derivative of the cdf, at alpha

## Description

Minus the first derivative of the cdf, at alpha

### Usage

```
gev_p12_mu1fa(alpha, t01, t02, v1, v2, v3, v4, v5)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

### Value

Vector

gev p	12	_mu2fa
-------	----	--------

Minus the second derivative of the cdf, at alpha

## Description

Minus the second derivative of the cdf, at alpha

```
gev_p12_mu2fa(alpha, t01, t02, v1, v2, v3, v4, v5)
```

gev\_p12\_pd 293

## Arguments

alpha	a vector of values of alpha (one minus probability)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

## Value

Matrix

gev_p12_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p12_pd(x, t1, t2, v1, v2, v3, v4, v5)
```

## Arguments

x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

### Value

Vector

gev_p12_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_p12_pdd(x, t1, t2, v1, v2, v3, v4, v5)
```

## Arguments

Х	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

#### Value

Matrix

```
gev_p12_predictordata Predicted Parameter and Generalized Residuals
```

## Description

Predicted Parameter and Generalized Residuals

```
gev_p12_predictordata(predictordata, x, t1, t2, t01, t02, params)
```

gev\_p12\_setics 295

## Arguments

predictordata	logical that indicates whether to calculate and return predictordata
x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
params	model parameters for calculating logf

### Value

Two vectors

gev_p12_setics	Set initial conditions	
----------------	------------------------	--

# Description

Set initial conditions

# Usage

```
gev_p12_setics(x, t1, t2, ics)
```

### **Arguments**

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
ics	initial conditions for the maximum likelihood search

#### Value

Vector

296 *gev\_p12\_waic* 

gev\_p12\_waic

Waic

## Description

Waic

# Usage

```
gev_p12_waic(
  waicscores,
  x,
  t1,
  t2,
  v1hat,
  v2hat,
  v3hat,
  v4hat,
  v5hat,
  lddi,
  lddd,
  lambdad
)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1hat	first parameter
v2hat	second parameter
v3hat	third parameter
v4hat	fourth parameter
v5hat	fifth parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior

## Value

Two numeric values.

gev\_p1a\_f1fa 297

				_
gev	n'l	а	+ 1	t a

The first derivative of the density for DMGS

### Description

The first derivative of the density for DMGS

### Usage

```
gev_p1a_f1fa(x, t0, v1, v2, v3, v4)
```

### Arguments

X	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
v2	second parameter
v3	third parameter

v4 fourth parameter

#### Value

Vector

	-		~	_
gev_	n	<b>a</b>	+ 1	+ \\
5 C V _	.Р	· u_		1 44

The first derivative of the density for WAIC

## Description

The first derivative of the density for WAIC

### Usage

```
gev_p1a_f1fw(x, t, v1, v2, v3, v4)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

298 gev\_p1a\_f2fw

#### Value

Vector

gev\_p1a\_f2fa

The second derivative of the density for DMGS

### Description

The second derivative of the density for DMGS

### Usage

```
gev_p1a_f2fa(x, t0, v1, v2, v3, v4)
```

#### **Arguments**

Χ	a vector of training data values
---	----------------------------------

to a single value of the predictor (specify either to or no but not both)

v1 first parameter
v2 second parameter
v3 third parameter
v4 fourth parameter

#### Value

Matrix

gev\_p1a\_f2fw

The second derivative of the density for WAIC

### Description

The second derivative of the density for WAIC

#### Usage

```
gev_p1a_f2fw(x, t, v1, v2, v3, v4)
```

### Arguments

v4

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

fourth parameter

gev\_p1a\_fd 299

### Value

Matrix

### Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_p1a_fd(x, t, v1, v2, v3, v4)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Vector

gev_p1a_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gev_p1a_fdd(x, t, v1, v2, v3, v4)
```

300 gev\_p1a\_ldda

### Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

### Value

Matrix

gev\_p1a\_ldda

 $The \ second \ derivative \ of \ the \ normalized \ log-like lihood$ 

# Description

The second derivative of the normalized log-likelihood

### Usage

```
gev_p1a_ldda(x, t, v1, v2, v3, v4)
```

## Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Matrix

gev\_p1a\_lddda 301

	-	
Q P V	nla	lddda

The third derivative of the normalized log-likelihood

#### **Description**

The third derivative of the normalized log-likelihood

### Usage

```
gev_p1a_lddda(x, t, v1, v2, v3, v4)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

3d array

gev_p1a_logfdd	Seco
----------------	------

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gev_p1a_logfdd(x, t, v1, v2, v3, v4)
```

#### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

302 gev\_p1a\_mu1fa

### Value

Matrix

gev_p1a_logfddd Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol		
	gev_p1a_logfddd	

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p1a_logfddd(x, t, v1, v2, v3, v4)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

3d array

gev_p1a_mu1fa	Minus the first derivative of the cdf, at alpha

## Description

Minus the first derivative of the cdf, at alpha

```
gev_p1a_mu1fa(alpha, t0, v1, v2, v3, v4)
```

gev\_p1a\_mu2fa 303

## Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

### Value

Vector

# Description

Minus the second derivative of the cdf, at alpha

### Usage

```
gev_p1a_mu2fa(alpha, t0, v1, v2, v3, v4)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Matrix

304 gev\_p1a\_pdd

gev_p1a_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_p1a_pd(x, t, v1, v2, v3, v4)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Vector

gev_p1a_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gev_p1a_pdd(x, t, v1, v2, v3, v4)
```

gev\_p1b\_f1fa 305

# Arguments

U	
Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

### Value

Matrix

gev_p1b_f1fa	The first derivative of the density for DMGS

# Description

The first derivative of the density for DMGS

## Usage

```
gev_p1b_f1fa(x, t0a, t0b, v1, v2, v3, v4, v5)
```

## Arguments

x	a vector of training data values
t0a	a single value of the predictor, for the first column of the predictor (specify either $t0a$ or $t0a$ but not both)
t0b	a single value of the predictor, for the second column of the predictor (specify either $t\theta b$ or $n\theta b$ but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

## Value

Vector

306 gev\_p1b\_f2fa

	1	I.	C	C
gev_	ŊΙ	n	ΤI	TW

The first derivative of the density for WAIC

# Description

The first derivative of the density for WAIC

### Usage

```
gev_p1b_f1fw(x, ta, tb, v1, v2, v3, v4, v5)
```

## Arguments

X	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

### Value

Vector

gev_	р1	b	f2	fa

The second derivative of the density for DMGS

## Description

The second derivative of the density for DMGS

```
gev_p1b_f2fa(x, t0a, t0b, v1, v2, v3, v4, v5)
```

gev\_p1b\_f2fw 307

## Arguments

X	a vector of training data values
t0a	a single value of the predictor, for the first column of the predictor (specify either $t0a$ or $t0a$ but not both)
t0b	a single value of the predictor, for the second column of the predictor (specify either $t\theta b$ or $n\theta b$ but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

### Value

Matrix

# Description

The second derivative of the density for WAIC

## Usage

```
gev_p1b_f2fw(x, ta, tb, v1, v2, v3, v4, v5)
```

# Arguments

X	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

### Value

Matrix

308 gev\_p1b\_fdd

gev_p1b_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	riv() by Anarew Clausen and Serguei Sokoi

### Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_p1b_fd(x, ta, tb, v1, v2, v3, v4, v5)
```

### Arguments

Χ	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

#### Value

Vector

gev_p1b_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gev_p1b_fdd(x, ta, tb, v1, v2, v3, v4, v5)
```

gev\_p1b\_ldda 309

## Arguments

x	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

### Value

Matrix

## Description

The second derivative of the normalized log-likelihood

## Usage

```
gev_p1b_ldda(x, ta, tb, v1, v2, v3, v4, v5)
```

## Arguments

X	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

### Value

Matrix

310 gev\_p1b\_logfdd

gev_p1b_lddda	The third derivative of the normalized log-likelihood
---------------	-------------------------------------------------------

## Description

The third derivative of the normalized log-likelihood

## Usage

```
gev_p1b_lddda(x, ta, tb, v1, v2, v3, v4, v5)
```

### Arguments

Х	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

#### Value

3d array

gev_p1b_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	Denv() by Andrew Clausen and Serguet Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gev_p1b_logfdd(x, ta, tb, v1, v2, v3, v4, v5)
```

gev\_p1b\_logfddd 311

### Arguments

X	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

### Value

Matrix

gev_p1b_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	Deriv() by Anarew Ciausen and Serguei Sokoi

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_p1b_logfddd(x, ta, tb, v1, v2, v3, v4, v5)
```

### Arguments

x	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

#### Value

3d array

312 gev\_p1b\_mu2fa

gev_p1b_mu1fa	Minus the first derivative of the cdf, at alpha	
---------------	-------------------------------------------------	--

## Description

Minus the first derivative of the cdf, at alpha

## Usage

```
gev_p1b_mu1fa(alpha, t0a, t0b, v1, v2, v3, v4, v5)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
t0a	a single value of the predictor, for the first column of the predictor (specify either $t0a$ or $t0a$ but not both)
t0b	a single value of the predictor, for the second column of the predictor (specify either $t0b$ or $n0b$ but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

## Value

Vector

gev_p1b_mu2fa	Minus the second derivative of the cdf, at alpha
	• • •

## Description

Minus the second derivative of the cdf, at alpha

```
gev_p1b_mu2fa(alpha, t0a, t0b, v1, v2, v3, v4, v5)
```

gev\_p1b\_pd 313

## Arguments

alpha	a vector of values of alpha (one minus probability)
t0a	a single value of the predictor, for the first column of the predictor (specify either $t0a$ or $t0a$ but not both)
t0b	a single value of the predictor, for the second column of the predictor (specify either $t\theta b$ or $n\theta b$ but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

### Value

Matrix

gev_p1b_pd First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol	ry
--------------------------------------------------------------------------------------------------------------------	----

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p1b_pd(x, ta, tb, v1, v2, v3, v4, v5)
```

# Arguments

X	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

### Value

Vector

314 gev\_p1c\_f1fa

gev_p1b_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_p1b_pdd(x, ta, tb, v1, v2, v3, v4, v5)
```

## Arguments

X	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter

## Value

Matrix

gev_p1c_f1fa	
--------------	--

## Description

The first derivative of the density for DMGS

```
gev_p1c_f1fa(x, t0a, t0b, t0c, v1, v2, v3, v4, v5, v6)
```

gev\_p1c\_f1fw 315

## Arguments

X	a vector of training data values
t0a	a single value of the predictor, for the first column of the predictor (specify either $t0a$ or $n0a$ but not both)
t0b	a single value of the predictor, for the second column of the predictor (specify either t0b or n0b but not both)
t0c	a single value of the predictor, for the third column of the predictor (specify either t0c or n0c but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

## Value

Vector

gev_p1c_f1fw	The first derivative of the density for WAIC
--------------	----------------------------------------------

## Description

The first derivative of the density for WAIC

## Usage

```
gev_p1c_f1fw(x, ta, tb, tc, v1, v2, v3, v4, v5, v6)
```

# Arguments

X	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
tc	a vector of predictors for the mean (third column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

316 gev\_p1c\_f2fa

### Value

Vector

gev	n1	C	f2fa	a

The second derivative of the density for DMGS

## Description

The second derivative of the density for DMGS

## Usage

```
gev_p1c_f2fa(x, t0a, t0b, t0c, v1, v2, v3, v4, v5, v6)
```

## Arguments

х	a vector of training data values
t0a	a single value of the predictor, for the first column of the predictor (specify either $t0a$ or $t0a$ but not both)
t0b	a single value of the predictor, for the second column of the predictor (specify either $t\theta b$ or $n\theta b$ but not both)
t0c	a single value of the predictor, for the third column of the predictor (specify either $t0c$ or $t0c$ but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

#### Value

Matrix

gev\_p1c\_f2fw 317

	- 4		COC
gev	nΙ	$\sim$	_f2fw

The second derivative of the density for WAIC

## Description

The second derivative of the density for WAIC

### Usage

```
gev_p1c_f2fw(x, ta, tb, tc, v1, v2, v3, v4, v5, v6)
```

### Arguments

x	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
tc	a vector of predictors for the mean (third column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

### Value

Matrix

gev_p1c_fd	First derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gev_p1c_fd(x, ta, tb, tc, v1, v2, v3, v4, v5, v6)
```

318 gev\_p1c\_fdd

### Arguments

X	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
tc	a vector of predictors for the mean (third column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

### Value

Vector

gev_p1c_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_p1c_fdd(x, ta, tb, tc, v1, v2, v3, v4, v5, v6)
```

## Arguments

x	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
tc	a vector of predictors for the mean (third column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

### Value

Matrix

gev\_p1c\_ldda 319

gev_p1c_ldda	The second derivative of the normalized log-likelihood
--------------	--------------------------------------------------------

# Description

The second derivative of the normalized log-likelihood

### Usage

```
gev_p1c_ldda(x, ta, tb, tc, v1, v2, v3, v4, v5, v6)
```

## Arguments

x	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
tc	a vector of predictors for the mean (third column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

#### Value

Matrix

gev_p1c_lddda	The third derivative of the normalized log-likelihood
---------------	-------------------------------------------------------

## Description

The third derivative of the normalized log-likelihood

```
gev_p1c_lddda(x, ta, tb, tc, v1, v2, v3, v4, v5, v6)
```

320 gev\_p1c\_logfdd

### Arguments

x	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
tc	a vector of predictors for the mean (third column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

### Value

3d array

gev_p1c_logfdd	Second derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_p1c_logfdd(x, ta, tb, tc, v1, v2, v3, v4, v5, v6)
```

## Arguments

X	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
tc	a vector of predictors for the mean (third column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

### Value

Matrix

gev\_p1c\_logfddd 321

gev_p1c_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p1c_logfddd(x, ta, tb, tc, v1, v2, v3, v4, v5, v6)
```

## Arguments

X	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
tc	a vector of predictors for the mean (third column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

### Value

3d array

gev_p1c_mu1fa	Minus the first derivative of the cdf, at alpha

## Description

Minus the first derivative of the cdf, at alpha

```
gev_p1c_mu1fa(alpha, t0a, t0b, t0c, v1, v2, v3, v4, v5, v6)
```

322 gev\_p1c\_mu2fa

## Arguments

alpha	a vector of values of alpha (one minus probability)
t0a	a single value of the predictor, for the first column of the predictor (specify either $t0a$ or $t0a$ but not both)
t0b	a single value of the predictor, for the second column of the predictor (specify either t0b or n0b but not both)
t0c	a single value of the predictor, for the third column of the predictor (specify either t0c or n0c but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

## Value

Vector

gev_p1c_mu2fa	Minus the second derivative of the cdf, at alpha
	, and the second

## Description

Minus the second derivative of the cdf, at alpha

### Usage

```
gev_p1c_mu2fa(alpha, t0a, t0b, t0c, v1, v2, v3, v4, v5, v6)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
t0a	a single value of the predictor, for the first column of the predictor (specify either $t0a$ or $t0a$ but not both)
t0b	a single value of the predictor, for the second column of the predictor (specify either t0b or n0b but not both)
t0c	a single value of the predictor, for the third column of the predictor (specify either t0c or n0c but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

gev\_p1c\_pd 323

### Value

Matrix

gev_p1c_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	Andrew Chausen and Berguet Bokol

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_p1c_pd(x, ta, tb, tc, v1, v2, v3, v4, v5, v6)
```

## Arguments

х	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
tc	a vector of predictors for the mean (third column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

#### Value

Vector

324 gev\_p1k3\_cp

gev_p1c_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gev_p1c_pdd(x, ta, tb, tc, v1, v2, v3, v4, v5, v6)
```

#### Arguments

X	a vector of training data values
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
tc	a vector of predictors for the mean (third column)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter
v5	fifth parameter
v6	sixth parameter

#### Value

Matrix

gev_p1k3_cp	GEV Distribution with Known Shape with a Predictor, Predictions
	Based on a Calibrating Prior

### Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.

- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qgev_p1k3_cp(
 х,
  t,
  t0 = NA,
  n0 = NA,
 p = seq(0.1, 0.9, 0.1),
  fdalpha = 0.01,
  kshape = 0,
 means = FALSE,
 waicscores = FALSE,
  pdf = FALSE,
 dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 predictordata = TRUE,
  centering = TRUE,
  debug = FALSE
)
rgev_p1k3_cp(
 n,
  Х,
  t,
  t0 = NA,
  n0 = NA,
 kshape = 0,
  rust = FALSE,
 mlcp = TRUE,
 centering = TRUE,
  debug = FALSE
)
```

```
dgev_p1k3_cp(
 Х,
  t,
 t0 = NA,
 n0 = NA,
 y = x,
 kshape = 0,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE
pgev_p1k3_cp(
 Х,
  t,
 t0 = NA,
 n0 = NA,
 y = x,
 kshape = 0,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE
)
tgev_p1k3_cp(n, x, t, kshape = 0, debug = FALSE)
```

a vector of training data values

# **Arguments** x

	8
t	a vector of predictors, such that length(t)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
p	a vector of probabilities at which to generate predictive quantiles
fdalpha	if pdf=TRUE, the fractional delta used for numerical derivatives with respect to probability, for calculating the pdf as a function of quantiles
kshape	the known shape parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
pdf	logical that indicates whether to run additional calculations and return density functions evaluated at quantiles specified by the input probabilities (longer runtime)

logical that indicates whether DMGS calculations should be run or not (longer dmgs run time) logical that indicates whether RUST-based posterior sampling calculations should rust be run or not (longer run time) nrust the number of posterior samples used in the RUST calculations predictordata logical that indicates whether predictordata should be calculated centering logical that indicates whether the predictor should be centered logical for turning on debug messages debug the number of random samples required mlcp logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST) a vector of values at which to calculate the density and distribution functions У

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The GEV distribution with known shape with a predictor has distribution function

$$F(x; a, b, \sigma) = \exp\left(-t(x; \mu(a, b), \sigma)\right)$$

where

$$t(x; a, b, \sigma) = \begin{cases} \left[1 + \xi \left(\frac{x - \mu(a, b)}{\sigma}\right)\right]^{-1/\xi} & \text{if } \xi \neq 0\\ \exp\left(-\frac{x - \mu(a, b)}{\sigma}\right) & \text{if } \xi = 0 \end{cases}$$

where x is the random variable,  $\mu = a + bt$  is the location parameter,  $\sigma > 0$  is the shape parameter and  $\xi$  is known (hence the k3 in the name).

The calibrating prior we use is given by

$$\pi(\mu,\sigma)\propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

• ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)

• cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),

- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

332 gev\_p1k3\_f1fa

#### **Examples**

```
#
# example 1
x=fitdistcp::d150gev_p1_example_data_v1_x #use data for 150
tt=fitdistcp::d150gev_p1_example_data_v1_t
p=c(1:9)/10
n0=10
q=qgev_p1k3_cp(x=x,t=tt,n0=n0,t0=NA,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgev_p1k3_cp)",
main="GEVD w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue",lwd=2)
cat(" ml_params=",q$ml_params,"\n")
```

gev\_p1k3\_f1fa

The first derivative of the density for DMGS

#### **Description**

The first derivative of the density for DMGS

#### Usage

```
gev_p1k3_f1fa(x, t0, v1, v2, v3, kshape)
```

#### **Arguments**

Χ	a vector of training data values
t0	a single value of the predictor (specify either $t0$ or $n0$ but not both)
v1	first parameter
v2	second parameter
v3	third parameter
kshape	the known shape parameter

#### Value

Vector

gev\_p1k3\_f1fw 333

	- 4		C 4	_
gev	n I	kХ	+ 1	† w

The first derivative of the density for WAIC

### Description

The first derivative of the density for WAIC

### Usage

```
gev_p1k3_f1fw(x, t, v1, v2, v3, kshape)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter

v2 second parameter v3 third parameter

kshape the known shape parameter

### Value

Vector

	. 1	1.5	C 2 C -
σeν	nΙ	ĸК	_f2fa

The second derivative of the density for DMGS

### Description

The second derivative of the density for DMGS

### Usage

```
gev_p1k3_f2fa(x, t0, v1, v2, v3, kshape)
```

### Arguments

Х	a vector of training data values
+0	a cingle value of the predictor (eneci

to a single value of the predictor (	(specify either t0 or n0 but not both)
--------------------------------------	----------------------------------------

v1 first parameterv2 second parameterv3 third parameter

kshape the known shape parameter

334 gev\_p1k3\_fd

#### Value

Matrix

gev\_p1k3\_f2fw

The second derivative of the density for WAIC

### Description

The second derivative of the density for WAIC

### Usage

```
gev_p1k3_f2fw(x, t, v1, v2, v3, kshape)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

kshape the known shape parameter

#### Value

Matrix

gev_p1k3_fd	First derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_p1k3_fd(x, t, v1, v2, v3, v4)
```

gev\_p1k3\_fdd 335

### Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Vector

gev\_p1k3\_fdd Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_p1k3_fdd(x, t, v1, v2, v3, v4)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

### Value

Matrix

336 gev\_p1k3\_lddda

	1	1. つ	1dd:	_
O P V	nı	ĸК	1 (1(1)	4

The second derivative of the normalized log-likelihood

### Description

The second derivative of the normalized log-likelihood

#### Usage

```
gev_p1k3_ldda(x, t, v1, v2, v3, kshape)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

kshape the known shape parameter

#### Value

Matrix

gev_	n1	ト3	1,	44	42
gev_	_レ ו	NJ_		uu	ıа

The third derivative of the normalized log-likelihood

### Description

The third derivative of the normalized log-likelihood

#### Usage

```
gev_p1k3_lddda(x, t, v1, v2, v3, kshape)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

kshape the known shape parameter

gev\_p1k3\_logf 337

#### Value

3d array

gev\_p1k3\_logf

Logf for RUST

#### **Description**

Logf for RUST

#### Usage

```
gev_p1k3_logf(params, x, t, kshape)
```

#### **Arguments**

params model parameters for calculating logf
x a vector of training data values
t a vector or matrix of predictors
kshape the known shape parameter

#### Value

Scalar value.

gev\_p1k3\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gev_p1k3_logfdd(x, t, v1, v2, v3, v4)
```

#### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

338 gev\_p1k3\_loglik

### Value

Matrix

gev_p1k3_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
------------------	-----------------------------------------------------------------------------------------------------------------

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_p1k3_logfddd(x, t, v1, v2, v3, v4)
```

### Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

### Value

3d array

gev_p1	k3_1	ogl	il	<
--------	------	-----	----	---

GEV-with-known-shape-with-p1 observed log-likelihood function

### Description

GEV-with-known-shape-with-p1 observed log-likelihood function

### Usage

```
gev_p1k3_loglik(vv, x, t, kshape)
```

gev\_p1k3\_means 339

#### **Arguments**

٧٧

x	a vector of training data values
t	a vector or matrix of predictors

parameters

kshape the known shape parameter

#### Value

Scalar

gev_p1k3_means	Analytical expressions for Predictive Means RHP mean based on the
	expectation of DMGS equation 2.1

### Description

Analytical expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

### Usage

```
gev_p1k3_means(means, t0, ml_params, kshape, nx)
```

# Arguments

means	logical that indicates	whether to return anal	vtical estimates for	the distribution
IIICaris	logical that mulcates	whichici to ictuili aliai	ytical confinates for	the distribution

means (longer runtime)

t0 a single value of the predictor (specify either t0 or n0 but not both)

ml\_params parameters

kshape the known shape parameter

nx length of training data

#### Value

Two scalars

340 *gev\_p1k3\_mu2fa* 

# Description

Minus the first derivative of the cdf, at alpha

# Usage

```
gev_p1k3_mu1fa(alpha, t0, v1, v2, v3, kshape)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
kshape	the known shape parameter

#### Value

Vector

gev_p1k3_mu2fa	Minus the second derivative of the cdf, at alpha

# Description

Minus the second derivative of the cdf, at alpha

# Usage

```
gev_p1k3_mu2fa(alpha, t0, v1, v2, v3, kshape)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
kshape	the known shape parameter

### Value

Matrix

gev_p1k3_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gev_p1k3_pd(x, t, v1, v2, v3, v4)
```

#### **Arguments**

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Vector

gev_p1k3_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gev_p1k3_pdd(x, t, v1, v2, v3, v4)
```

#### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Matrix

```
gev_p1k3_predictordata
```

Predicted Parameter and Generalized Residuals

# Description

Predicted Parameter and Generalized Residuals

### Usage

```
gev_p1k3_predictordata(predictordata, x, t, t0, params, kshape)
```

### Arguments

predictordata	logical that indicates whether to calculate and return predictordata
x	a vector of training data values
t	a vector or matrix of predictors
t0	a single value of the predictor (specify either t0 or n0 but not both)
params	model parameters for calculating logf
kshape	the known shape parameter

#### Value

Two vectors

gev\_p1k3\_waic 343

gev\_p1k3\_waic

Waic

# Description

Waic

# Usage

```
gev_p1k3_waic(
   waicscores,
   x,
   t,
   v1hat,
   v2hat,
   v3hat,
   kshape,
   lddi,
   lddd,
   lambdad
)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
x	a vector of training data values
t	a vector or matrix of predictors
v1hat	first parameter
v2hat	second parameter
v3hat	third parameter
kshape	the known shape parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior

### Value

Two numeric values.

gev_p1n_checkmle	Check MLE
------------------	-----------

#### Description

Check MLE

#### Usage

```
gev_p1n_checkmle(ml_params, minxi = -1, maxxi = 1)
```

#### **Arguments**

ml\_params parameters

minxi minimum value of shape parameter xi maxxi maximum value of shape parameter xi

#### Value

No return value (just a message to the screen).

gev_p1n_cp	Generalized Extreme Value Distribution with Multiple Predictors on
	the Location, Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qgev_p1n_cp(
 х,
  t,
  t0 = NA,
 n0 = NA,
 p = seq(0.1, 0.9, 0.1),
  fdalpha = 0.01,
 minxi = -1,
 maxxi = 1,
 means = FALSE,
 waicscores = FALSE,
 extramodels = FALSE,
 pdf = FALSE,
 dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 predictordata = TRUE,
 centering = TRUE,
  debug = FALSE
)
rgev_p1n_cp(
 n,
 х,
  t,
  t0 = NA,
 n0 = NA,
 minxi = -1,
 maxxi = 1,
 extramodels = FALSE,
  rust = FALSE,
 mlcp = TRUE,
 debug = FALSE
)
dgev_p1n_cp(
 х,
  t,
  t0 = NA,
```

```
n0 = NA,
 y = x,
 minxi = -1,
 maxxi = 1,
 extramodels = FALSE,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE
)
pgev_p1n_cp(
 х,
 t,
 t0 = NA,
 n0 = NA,
 y = x,
 minxi = -1,
 maxxi = 1,
 extramodels = FALSE,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE
)
tgev_p1n_cp(n, x, t, extramodels = FALSE, debug = FALSE)
```

# Arguments

x	a vector of training data values
t	predictors, which can be a vector, or a matrix with 1, 2 or 3 columns
t0	a single value for each predictor, as 1, 2 or 3 scalars (specify $t0$ or $n0$ but not both)
n0	an index for the each predictor, as 1, 2 or 3 integers (specify $t0$ or $n0$ but not both)
р	a vector of probabilities at which to generate predictive quantiles
fdalpha	if pdf=TRUE, the fractional delta used for numerical derivatives with respect to probability, for calculating the pdf as a function of quantiles
minxi	the minimum allowed value of the shape parameter (decrease with caution)
maxxi	the maximum allowed value of the shape parameter (increase with caution)
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)

extramodels logical that indicates whether to run additional calculations and add three additional prediction models (longer runtime) pdf logical that indicates whether to run additional calculations and return density functions evaluated at quantiles specified by the input probabilities (longer runtime) logical that indicates whether DMGS calculations should be run or not (longer dmgs run time) logical that indicates whether RUST-based posterior sampling calculations should rust be run or not (longer run time) the number of posterior samples used in the RUST calculations nrust predictordata logical that indicates whether predictordata should be calculated logical that indicates whether the predictor should be centered centering logical for turning on debug messages debug the number of random samples required mlcp logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST) a vector of values at which to calculate the density and distribution functions у

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

1, 2 or 3 predictors on the location parameter are supported. For instance, the GEV distribution with 2 predictors has distribution function

$$F(x; \alpha, \beta_1, \beta_2, \sigma, \xi) = \exp\left(-t(x; \mu(\alpha, \beta_1, \beta_2), \sigma, \xi)\right)$$

where

$$t(x; \mu(\alpha, \beta_1, \beta_2), \sigma, \xi) = \begin{cases} \left[ 1 + \xi \left( \frac{x - \mu(\alpha, \beta_1, \beta_2)}{\sigma} \right) \right]^{-1/\xi} & \text{if } \xi \neq 0 \\ \exp \left( -\frac{x - \mu(\alpha, \beta_1, \beta_2)}{\sigma} \right) & \text{if } \xi = 0 \end{cases}$$

where x is the random variable,  $\mu = \alpha + \beta_1 t_1 + \beta_2 t_2$  is the location parameter, modelled as a function of parameters  $\alpha, \beta_1, \beta_2$  and predictor  $t_1, t_2$ , and  $\sigma > 0, \xi$  are the scale and shape parameters.

The calibrating prior we use is given by

$$\pi(\alpha, \beta_1, \beta_2, \sigma, \xi) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

The code will switch to maximum likelihood prediction if the input data gives a maximum likelihood value for the shape parameter that lies outside the range (minxi,maxxi), since outside this range there may be numerical problems. If this happens, it is reported in the revert2ml flag. Such values seldom occur in real observed data for maxima.

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

#### If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

### Optional Return Values (EVT models only)

q\*\*\*\* optionally returns the following, for EVT models only:

• cp\_pdf: the density function at quantiles corresponding to input probabilities p. We provide this for EVD models, because direct estimation of the density function using the DMGS density equation is not possible.

#### Optional Return Values (some EVT models only)

q\*\*\*\* optionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_quantiles: predictive quantiles calculated from Bayesian integration with a flat prior.
- rh\_ml\_quantiles: predictive quantiles calculated from Bayesian integration with the calibrating prior, and the maximmum likelihood estimate for the shape parameter.
- jp\_quantiles: predictive quantiles calculated from Bayesian integration with Jeffreys' prior.

r\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_deviates: predictive random deviates calculated using a Bayesian analysis with a flat prior.
- rh\_ml\_deviates: predictive random deviates calculated using a Bayesian analysis with the RHP-MLE prior.
- jp\_deviates: predictive random deviates calculated using a Bayesian analysis with the JP.

d\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_pdf: predictive density function from a Bayesian analysis with the flat prior.
- rh\_ml\_pdf: predictive density function from a Bayesian analysis with the RHP-MLE prior.
- jp\_pdf: predictive density function from a Bayesian analysis with the JP.

p\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_cdf: predictive distribution function from a Bayesian analysis with the flat prior.
- rh\_ml\_cdf: predictive distribution function from a Bayesian analysis with the RHP-MLE prior.
- jp\_cdf: predictive distribution function from a Bayesian analysis with the JP.

These additional predictive distributions are included for comparison with the calibrating prior model. They generally give less good reliability than the calibrating prior.

#### **Details (non-homogeneous models)**

This model is not homogeneous, i.e. it does not have a transitive transformation group, and so there is no right Haar prior and no method for generating exactly reliable predictions. The cp outputs are generated using a prior that has been shown in tests to give reasonable reliability. See Jewson et al. (2025a) for discussion of the prior and test results. For non-homogeneous models, reliability is generally poor for small sample sizes (<20), but is still much better than maximum likelihood. For small sample sizes, it is advisable to check the level of reliability using the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),

- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

gev\_p1n\_logf 353

#### **Examples**

```
# example 1
x=fitdistcp::d150gev_p1_example_data_v1_x
t1=fitdistcp::d150gev_p1_example_data_v1_t
t2=sample(t1)
t=cbind(t1,t2)
p=c(1:9)/10
n0=c(10,10)
q=qgev_p1n_cp(x=x,t=t,n0=n0,t0=NA,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgev_p1n_cp)",
main="GEVD w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue",lwd=2)
cat(" ml_params=",q$ml_params,"\n")
```

gev\_p1n\_logf

Logf for RUST

### Description

Logf for RUST

### Usage

```
gev_p1n_logf(params, x, t)
```

#### **Arguments**

params model parameters for calculating logf
x a vector of training data values
t a vector or matrix of predictors

#### Value

Scalar value.

354 gev\_p1n\_means

gev\_p1n\_loglik

observed log-likelihood function

### Description

observed log-likelihood function

### Usage

```
gev_p1n_loglik(vv, x, t)
```

# Arguments

t

vv parametersx a vector of training data values

a vector or matrix of predictors

### Value

Scalar

gev\_p1n\_means

Analytical expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

# Description

Analytical expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

### Usage

```
gev_p1n_means(
  means,
  t0,
  ml_params,
  lddi,
  lddd,
  lambdad_rh_flat,
  nx,
  dim = (nt + 3)
)
```

#### **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

t0 a single value of the predictor (specify either t0 or n0 but not both)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihood

 $lambdad_rh_flat$ 

derivative of the log CRHP-FLAT prior

nx length of training data dim number of parameters

#### Value

Two scalars

gev\_p1n\_n1\_exampledata

 $GEV\_p1n \ n=1 \ example \ data$ 

### Description

GEV\_p1n n=1 example data

#### Usage

```
gev_p1n_n1_exampledata(iseed)
```

### Arguments

iseed The random seed

### Value

A list containing data to run an example

```
gev_p1n_n2_exampledata
```

 $GEV_p1n n=2$  example data

#### **Description**

GEV\_p1n n=2 example data

### Usage

```
gev_p1n_n2_exampledata(iseed)
```

### Arguments

iseed The random seed

#### Value

A list containing data to run an example

gev\_p1n\_predictordata Predicted Parameter and Generalized Residuals

### Description

Predicted Parameter and Generalized Residuals

### Usage

```
gev_p1n_predictordata(predictordata, x, t, t0, params)
```

### Arguments

predictordata logical that indicates whether to calculate and return predictordata

x a vector of training data valuest a vector or matrix of predictors

to a single value of the predictor (specify either to or no but not both)

params model parameters for calculating logf

#### Value

Two vectors

gev\_p1n\_setics 357

gev_p1n_setics Set initial conditions
---------------------------------------

### Description

Set initial conditions

### Usage

```
gev_p1n_setics(x, t)
```

### Arguments

x a vector of training data valuest a vector or matrix of predictors

#### Value

Vector

gev_p1n_waic Waic
-------------------

### Description

Waic

### Usage

```
gev_p1n_waic(waicscores, x, t0, vhat, lddi, lddd, lambdad)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
x	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
vhat	vector of all parameters
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior

### Value

Two numeric values.

ev_p1	_checkmle	Check MLE
ev_pi	_cneckiiite	Checi

#### **Description**

Check MLE

#### Usage

```
gev_p1_checkmle(ml_params, minxi = -1, maxxi = 1)
```

#### **Arguments**

ml\_params parameters

minxi minimum value of shape parameter xi maxxi maximum value of shape parameter xi

#### Value

No return value (just a message to the screen).

gev_p1_cp	Generalized Extreme Value Distribution with a Single Predictor on the
	Location, Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qgev_p1_cp(
 Х,
  t,
  t0 = NA,
 n0 = NA,
 p = seq(0.1, 0.9, 0.1),
  ics = c(0, 0, 0, 0),
  fdalpha = 0.01,
 minxi = -1,
 maxxi = 1,
 means = FALSE,
 waicscores = FALSE,
 extramodels = FALSE,
 pdf = FALSE,
  dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 predictordata = TRUE,
  centering = TRUE,
 debug = FALSE
)
rgev_p1_cp(
 n,
 х,
  t,
  t0 = NA,
 n0 = NA,
 ics = c(0, 0, 0, 0),
 minxi = -1,
 maxxi = 1,
 extramodels = FALSE,
 rust = FALSE,
 mlcp = TRUE,
 debug = FALSE
)
dgev_p1_cp(
 х,
```

```
t,
  t0 = NA,
 n0 = NA,
 y = x,
  ics = c(0, 0, 0, 0),
 minxi = -1,
 \max x i = 1,
 extramodels = FALSE,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
  debug = FALSE
pgev_p1_cp(
 х,
 t,
 t0 = NA,
 n0 = NA,
 y = x,
 ics = c(0, 0, 0, 0),
 minxi = -1,
 maxxi = 1,
 extramodels = FALSE,
 rust = FALSE,
 nrust = 1000,
  centering = TRUE,
 debug = FALSE
)
tgev_p1_cp(n, x, t, ics = c(0, 0, 0, 0), extramodels = FALSE, debug = FALSE)
```

### Arguments

X	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
р	a vector of probabilities at which to generate predictive quantiles
ics	initial conditions for the maximum likelihood search
fdalpha	if pdf=TRUE, the fractional delta used for numerical derivatives with respect to probability, for calculating the pdf as a function of quantiles
minxi	the minimum allowed value of the shape parameter (decrease with caution)
maxxi	the maximum allowed value of the shape parameter (increase with caution)
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)

waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
extramodels	logical that indicates whether to run additional calculations and add three additional prediction models (longer runtime)
pdf	logical that indicates whether to run additional calculations and return density functions evaluated at quantiles specified by the input probabilities (longer runtime)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
predictordata	logical that indicates whether predictordata should be calculated
centering	logical that indicates whether the predictor should be centered
debug	logical for turning on debug messages
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- ullet adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

### **Details of the Model**

The GEV distribution with a predictor has distribution function

$$F(x; a, b, \sigma, \xi) = \exp\left(-t(x; \mu(a, b), \sigma, \xi)\right)$$

where

$$t(x; \mu(a, b), \sigma, \xi) = \begin{cases} \left[1 + \xi \left(\frac{x - \mu(a, b)}{\sigma}\right)\right]^{-1/\xi} & \text{if } \xi \neq 0\\ \exp\left(-\frac{x - \mu(a, b)}{\sigma}\right) & \text{if } \xi = 0 \end{cases}$$

where x is the random variable,  $\mu = a + bt$  is the location parameter, modelled as a function of parameters a, b and predictor t, and  $\sigma > 0$ ,  $\xi$  are the scale and shape parameters.

The calibrating prior we use is given by

$$\pi(a,b,\sigma,\xi) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

The code will switch to maximum likelihood prediction if the input data gives a maximum likelihood value for the shape parameter that lies outside the range (minxi,maxxi), since outside this range there may be numerical problems. If this happens, it is reported in the revert2ml flag. Such values seldom occur in real observed data for maxima.

### **Optional Return Values**

q\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

### Optional Return Values (EVT models only)

q\*\*\*\* optionally returns the following, for EVT models only:

• cp\_pdf: the density function at quantiles corresponding to input probabilities p. We provide this for EVD models, because direct estimation of the density function using the DMGS density equation is not possible.

### Optional Return Values (some EVT models only)

q\*\*\*\* optionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_quantiles: predictive quantiles calculated from Bayesian integration with a flat prior.
- rh\_ml\_quantiles: predictive quantiles calculated from Bayesian integration with the calibrating prior, and the maximmum likelihood estimate for the shape parameter.
- jp\_quantiles: predictive quantiles calculated from Bayesian integration with Jeffreys' prior.

r\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_deviates: predictive random deviates calculated using a Bayesian analysis with a flat prior.
- rh\_ml\_deviates: predictive random deviates calculated using a Bayesian analysis with the RHP-MLE prior.
- jp\_deviates: predictive random deviates calculated using a Bayesian analysis with the JP.

d\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_pdf: predictive density function from a Bayesian analysis with the flat prior.
- rh\_ml\_pdf: predictive density function from a Bayesian analysis with the RHP-MLE prior.
- jp\_pdf: predictive density function from a Bayesian analysis with the JP.

p\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_cdf: predictive distribution function from a Bayesian analysis with the flat prior.
- rh\_ml\_cdf: predictive distribution function from a Bayesian analysis with the RHP-MLE prior.
- jp\_cdf: predictive distribution function from a Bayesian analysis with the JP.

These additional predictive distributions are included for comparison with the calibrating prior model. They generally give less good reliability than the calibrating prior.

### **Details (non-homogeneous models)**

This model is not homogeneous, i.e. it does not have a transitive transformation group, and so there is no right Haar prior and no method for generating exactly reliable predictions. The cp outputs are generated using a prior that has been shown in tests to give reasonable reliability. See Jewson et al. (2025a) for discussion of the prior and test results. For non-homogeneous models, reliability is generally poor for small sample sizes (<20), but is still much better than maximum likelihood. For small sample sizes, it is advisable to check the level of reliability using the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

· Cauchy (cauchy),

- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

gev\_p1\_logf 367

### **Examples**

```
# example 1
x=fitdistcp::d150gev_p1_example_data_v1_x
tt=fitdistcp::d150gev_p1_example_data_v1_t
p=c(1:9)/10
n0=10
q=qgev_p1_cp(x=x,t=tt,n0=n0,t0=NA,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgev_p1_cp)",
main="GEVD w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue",lwd=2)
cat(" ml_params=",q$ml_params,"\n")
```

gev\_p1\_logf

Logf for RUST

### **Description**

Logf for RUST

### Usage

```
gev_p1_logf(params, x, t)
```

### **Arguments**

params model parameters for calculating logf

x a vector of training data values

t a vector or matrix of predictors

### Value

Scalar value.

368 gev\_p1\_means

	-	-	٠.	
gev_	n1	Inc	ווכ	k
5 C V _	. P ' -	_+0,	~ + +	

observed log-likelihood function

## Description

observed log-likelihood function

### Usage

```
gev_p1_loglik(vv, x, t)
```

### **Arguments**

vv parameters

x a vector of training data valuest a vector or matrix of predictors

#### Value

Scalar

gev_p1_me	ans
-----------	-----

Analytical expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

### **Description**

Analytical expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

### Usage

```
gev_p1_means(means, t0, ml_params, lddi, lddd, lambdad_rh_flat, nx, dim = 4)
```

### **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

to a single value of the predictor (specify either to or no but not both)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihood

lambdad\_rh\_flat

derivative of the log CRHP-FLAT prior

nx length of training data dim number of parameters gev\_p1\_predictordata 369

### Value

Two scalars

gev\_p1\_predictordata Predicted Parameter and Generalized Residuals

### **Description**

Predicted Parameter and Generalized Residuals

#### **Usage**

```
gev_p1_predictordata(predictordata, x, t, t0, params)
```

#### **Arguments**

predictordata logical that indicates whether to calculate and return predictordata

x a vector of training data valuest a vector or matrix of predictors

to a single value of the predictor (specify either to or no but not both)

params model parameters for calculating logf

### Value

Two vectors

gev\_p1\_setics

Set initial conditions

### **Description**

Set initial conditions

## Usage

```
gev_p1_setics(x, t, ics)
```

### **Arguments**

x a vector of training data values t a vector or matrix of predictors

ics initial conditions for the maximum likelihood search

### Value

Vector

gev\_pd

# Description

Waic

# Usage

```
gev_p1_waic(waicscores, x, t0, v1hat, v2hat, v3hat, v4hat, lddi, lddd, lambdad)
```

## Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1hat	first parameter
v2hat	second parameter
v3hat	third parameter
v4hat	fourth parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior

### Value

Two numeric values.

gev_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	Anarew Clausen and Serguei Sokoi

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gev_pd(x, v1, v2, v3)
```

gev\_pdd 371

# Arguments

X	a vector of training data values
---	----------------------------------

v1 first parameter

v2 second parameter

v3 third parameter

## Value

Vector

gev_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gev_pdd(x, v1, v2, v3)
```

# Arguments

x a vector of traini	ing data values
----------------------	-----------------

v1 first parameter

v2 second parameter

v3 third parameter

### Value

Matrix

372 gev\_setics

gev\_pwm\_params

PWM parameter estimation

# Description

PWM parameter estimation

## Usage

```
gev_pwm_params(x)
```

## Arguments

Χ

a vector of training data values

### Value

Vector

gev\_setics

Set initial conditions

# Description

Set initial conditions

# Usage

```
gev_setics(x, ics)
```

### **Arguments**

x a vector of training data values

ics initial conditions for the maximum likelihood search

## Value

Vector

gev\_waic 373

|--|

## Description

Waic

### Usage

```
gev_waic(waicscores, x, v1hat, v2hat, v3hat, lddi, lddd, lambdad)
```

## Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
x	a vector of training data values
v1hat	first parameter
v2hat	second parameter
v3hat	third parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior

#### Value

Two numeric values.

gnorm_k3_cp	Generalized Normal Distribution Predictions Based on a Calibrating Prior

### Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.

- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

### Usage

```
qgnorm_k3_cp(
  х,
 p = seq(0.1, 0.9, 0.1),
 kbeta = 4,
 d1 = 0.01,
  fd2 = 0.01,
 means = FALSE,
 waicscores = FALSE,
 logscores = FALSE,
  dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 debug = FALSE,
  aderivs = TRUE
)
rgnorm_k3_cp(
 n,
 d1 = 0.01,
  fd2 = 0.01,
 kbeta = 4,
  rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
dgnorm_k3_cp(
 х,
 y = x,
 d1 = 0.01,
  fd2 = 0.01,
 kbeta = 4,
```

```
rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
pgnorm_k3_cp(
 х,
 y = x,
 d1 = 0.01,
 fd2 = 0.01,
 kbeta = 4,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
tgnorm_k3_cp(n, x, d1 = 0.01, fd2 = 0.01, kbeta = 4, debug = FALSE)
```

# Arguments

Х	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
kbeta	the known beta parameter
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required

mlcp	logical that indicates whether maxlik and parameter uncertainty calculations
	should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

### **Details of the Model**

The generalized normal distribution has probability density function

$$f(x; \mu, \alpha) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(|x-\mu|/\alpha)^{\beta}}$$

where x is the random variable,  $\mu, \alpha > 0$  are the parameters and we consider  $\beta$  to be known (hence the k3 in the name).

The calibrating prior is given by the right Haar prior, which is

$$\pi(\alpha) \propto \frac{1}{\alpha}$$

as given in Jewson et al. (2025).

### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),

380 gnorm\_k3\_f1f

- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

### **Examples**

```
# example 1
x=fitdistcp::d032gnorm_k3_example_data_v1
p=c(1:9)/10
q=qgnorm_k3_cp(x,p,kbeta=4,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgnorm_k3_cp)",
main="gnorm: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

gnorm\_k3\_f1f

DMGS equation 3.3, f1 term

## Description

```
DMGS equation 3.3, f1 term
```

## Usage

```
gnorm_k3_f1f(y, v1, d1, v2, fd2, kbeta)
```

gnorm\_k3\_f1fa 381

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter

## Value

Matrix

gnorm_k3_f1fa	The first derivative of the density	
---------------	-------------------------------------	--

# Description

The first derivative of the density

# Usage

```
gnorm_k3_f1fa(x, v1, v2, kbeta)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

kbeta the known beta parameter

## Value

Vector

382 gnorm\_k3\_f2fa

gnorm	k3	f2f

DMGS equation 3.3, f2 term

## Description

DMGS equation 3.3, f2 term

## Usage

```
gnorm_k3_f2f(y, v1, d1, v2, fd2, kbeta)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-

eter

kbeta the known beta parameter

### Value

3d array

gnorm k3 f2fa
---------------

The second derivative of the density

## Description

The second derivative of the density

## Usage

```
gnorm_k3_f2fa(x, v1, v2, kbeta)
```

## Arguments

X	a vector of	training	data values

v1 first parameterv2 second parameter

kbeta the known beta parameter

gnorm\_k3\_fd 383

### Value

Matrix

Matrix

gnorm\_k3\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gnorm_k3_fd(x, v1, v2, v3)
```

## Arguments

x a vector of training data values

v1 first parameter
v2 second parameter

v3 third parameter

### Value

Vector

 $gnorm_k3_fdd$ 

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### **Description**

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gnorm_k3_fdd(x, v1, v2, v3)
```

384 gnorm\_k3\_ldd

# Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter

v3 third parameter

# Value

Matrix

gnorm_k3_ldd	Second derivative matrix of the normalized log-likelihood

# Description

Second derivative matrix of the normalized log-likelihood

# Usage

```
gnorm_k3_ldd(x, v1, d1, v2, fd2, kbeta)
```

# Arguments

x	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter

## Value

Square scalar matrix

gnorm\_k3\_ldda 385

gnorm	k3	1dda
ELIOT III	NJ	Tuua

The second derivative of the normalized log-likelihood

### **Description**

The second derivative of the normalized log-likelihood

### Usage

```
gnorm_k3_ldda(x, v1, v2, kbeta)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

kbeta the known beta parameter

### Value

Matrix

gnorm\_k3\_lddd

Third derivative tensor of the normalized log-likelihood

### **Description**

Third derivative tensor of the normalized log-likelihood

### Usage

```
gnorm_k3_lddd(x, v1, d1, v2, fd2, kbeta)
```

## Arguments

x a vector of training data value	es
-----------------------------------	----

v1 first parameter

d1 the delta used in the numerical derivatives with respect to the parameter

v2 second parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

kbeta the known beta parameter

### Value

Cubic scalar array

386 gnorm\_k3\_lmn

		_		
gnorm	k3	-10	dd	lda

The third derivative of the normalized log-likelihood

### **Description**

The third derivative of the normalized log-likelihood

### Usage

```
gnorm_k3_lddda(x, v1, v2, kbeta)
```

## Arguments

a vector of training data values Χ

v1 first parameter second parameter v2

the known beta parameter kbeta

### Value

3d array

gnorm_k3_lmn	One component of the second derivative of the normalized log-
	likelihood

## Description

One component of the second derivative of the normalized log-likelihood

### Usage

```
gnorm_k3_lmn(x, v1, d1, v2, fd2, kbeta, mm, nn)
```

## Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

gnorm\_k3\_logf 387

### Value

Scalar value

gnorm\_k3\_logf

Logf for RUST

### **Description**

Logf for RUST

# Usage

```
gnorm_k3_logf(params, x, kbeta)
```

### **Arguments**

params model parameters for calculating logf x a vector of training data values kbeta the known beta parameter

### Value

Scalar value.

gnorm\_k3\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gnorm_k3_logfdd(x, v1, v2, v3)
```

### Arguments

v1 first parameterv2 second parameterv3 third parameter

### Value

Matrix

388 gnorm\_k3\_loglik

gnorm_k3_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gnorm_k3_logfddd(x, v1, v2, v3)
```

## Arguments

	X	a vector of training	g data values
--	---	----------------------	---------------

v1 first parameterv2 second parameterv3 third parameter

### Value

3d array

gnorm_k3_loglik	log-likelihood function
-----------------	-------------------------

## Description

log-likelihood function

# Usage

```
gnorm_k3_loglik(vv, x, kbeta)
```

## **Arguments**

vv parameters

x a vector of training data values kbeta the known beta parameter

## Value

Scalar

gnorm\_k3\_logscores 389

gnorm_k3_logscores	Log scores for MLE and RHP predictions calculated using leave-one-out

# Description

Log scores for MLE and RHP predictions calculated using leave-one-out

## Usage

```
gnorm_k3_logscores(logscores, x, d1 = 0.01, fd2 = 0.01, kbeta, aderivs)
```

## Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
x	a vector of training data values
d1	the delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

Two scalars

# Description

DMGS equation 3.3, mu1 term

# Usage

```
gnorm_k3_mu1f(alpha, v1, d1, v2, fd2, kbeta)
```

390 gnorm\_k3\_mu2f

## Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

kbeta the known beta parameter

## Value

Matrix

gnorm_k3_mu2f	DMGS equation 3.3, mu2 term	

# Description

DMGS equation 3.3, mu2 term

# Usage

```
gnorm_k3_mu2f(alpha, v1, d1, v2, fd2, kbeta)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter

## Value

3d array

gnorm\_k3\_p1f 391

gnorm_k3_p1f DMGS equation 3.3, p1 term
-----------------------------------------

# Description

DMGS equation 3.3, p1 term

# Usage

```
gnorm_k3_p1f(y, v1, d1, v2, fd2, kbeta)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter
	1

## Value

Matrix

|--|

# Description

DMGS equation 3.3, p2 term

# Usage

```
gnorm_k3_p2f(y, v1, d1, v2, fd2, kbeta)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter

392 gnorm\_lmnp

## Value

3d array

gnorm_lmnp	One component of the second derivative of the normalized log-likelihood
------------	-------------------------------------------------------------------------

# Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
gnorm_lmnp(x, v1, d1, v2, fd2, kbeta, mm, nn, rr)
```

# Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

## Value

Scalar value

gnorm\_waic 393

gnorm\_waic

Waic for RUST

# Description

Waic for RUST

# Usage

```
gnorm_waic(
  waicscores,
  x,
  v1hat,
  d1,
  v2hat,
  fd2,
  kbeta,
  lddi,
  lddd,
  lambdad,
  aderivs
)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
Х	a vector of training data values
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kbeta	the known beta parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

Two numeric values.

394 gpd\_k13\_f2fa

gpd\_k13\_f1fa

The first derivative of the density

### **Description**

The first derivative of the density

### Usage

```
gpd_k13_f1fa(x, v1, v2, kloc)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

kloc the known location parameter

### Value

Vector

gpd\_k13\_f2fa

The second derivative of the density

## Description

The second derivative of the density

### Usage

$$gpd_k13_f2fa(x, v1, v2, kloc)$$

## **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

kloc the known location parameter

### Value

Matrix

gpd\_k13\_fd 395

gpd_k13_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gpd_k13_fd(x, v1, v2, v3)
```

## **Arguments**

Χ	a vector of training data values

v1 first parameterv2 second parameterv3 third parameter

# Value

Vector

gpd_k13_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gpd_k13_fdd(x, v1, v2, v3)
```

## Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

## Value

Matrix

396 gpd\_k13\_lddda

gpd	1.1	2	٦.	٦	٦,
ยมน	ΚI	2		u١	ua

The second derivative of the normalized log-likelihood

## Description

The second derivative of the normalized log-likelihood

## Usage

```
gpd_k13_ldda(x, v1, v2, kloc)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

kloc the known location parameter

#### Value

Matrix

	lddda

The third derivative of the normalized log-likelihood

### **Description**

The third derivative of the normalized log-likelihood

### Usage

```
gpd_k13_lddda(x, v1, v2, kloc)
```

## **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

kloc the known location parameter

### Value

3d array

gpd\_k13\_logfdd 397

gpd_k13_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gpd_k13_logfdd(x, v1, v2, v3)
```

# Arguments

		c	1 . 1
Y	a vector o	it training	data values
^	a vector o	n uaning	uata varues

v1 first parameterv2 second parameterv3 third parameter

### Value

Matrix

gpd_k13_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gpd_k13_logfddd(x, v1, v2, v3)
```

# Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

### Value

3d array

398 gpd\_k13\_mu2fa

and	レイコ	mıı1	f~
gpd_	_K I O	_IIIU I	ıα

Minus the first derivative of the cdf, at alpha

### **Description**

Minus the first derivative of the cdf, at alpha

### Usage

```
gpd_k13_mu1fa(alpha, v1, v2, kloc)
```

### Arguments

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

kloc the known location parameter

#### Value

Vector

gpd\_k13\_mu2fa

Minus the second derivative of the cdf, at alpha

# Description

Minus the second derivative of the cdf, at alpha

### Usage

```
gpd_k13_mu2fa(alpha, v1, v2, kloc)
```

# **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

kloc the known location parameter

### Value

Matrix

gpd\_k13\_pd 399

gpd_k13_pd First derivative of the cdf Created by Stephen Jewson using Der Andrew Clausen and Serguei Sokol	riv() by
----------------------------------------------------------------------------------------------------------------	----------

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gpd_k13_pd(x, v1, v2, v3)
```

# Arguments

<b>v</b>	a vector of training	data values
X	a vector of training	data varues

v1 first parameter v2 second parameter v3 third parameter

# Value

Vector

gpd_k13_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gpd_k13_pdd(x, v1, v2, v3)
```

# Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

# Value

Matrix

<pre>gpd_k1_checkmle</pre>	Check MLE
Spa_K I_clicckiiiiic	CITCUL MILL

#### **Description**

Check MLE

#### Usage

```
gpd_k1_checkmle(ml_params, kloc, minxi = -1, maxxi = 2)
```

#### **Arguments**

ml\_params parameters

kloc the known location parameter

minxi minimum value of shape parameter xi maxxi maximum value of shape parameter xi

#### Value

No return value (just a message to the screen).

gpd_k1_cp	Generalized Pareto Distribution with Known Location Parameter, Pre-
	dictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qgpd_k1_cp(
 Х,
 p = seq(0.1, 0.9, 0.1),
 kloc = 0,
 ics = c(0, 0),
  fdalpha = 0.01,
  customprior = 0,
 minxi = -1,
 maxxi = 2,
 means = FALSE,
 waicscores = FALSE,
 extramodels = FALSE,
 pdf = FALSE,
 dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 debug = FALSE
)
rgpd_k1_cp(
 n,
 х,
 kloc = 0,
 ics = c(0, 0),
 minxi = -1,
 maxxi = 2,
 extramodels = FALSE,
 rust = FALSE,
 mlcp = TRUE,
  debug = FALSE
)
dgpd_k1_cp(
 х,
 y = x,
 kloc = 0,
 ics = c(0, 0),
  customprior = 0,
 minxi = -1,
```

```
\max x i = 2,
 extramodels = FALSE,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE
)
pgpd_k1_cp(
 Х,
 y = x,
 kloc = 0,
 ics = c(0, 0),
  customprior = 0,
 minxi = -1,
 \max x i = 2,
 extramodels = FALSE,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE
tgpd_k1_cp(n, x, kloc = 0, ics = c(0, 0), extramodels = FALSE, debug = FALSE)
```

# Arguments

x	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
kloc	the known location parameter
ics	initial conditions for the maximum likelihood search
fdalpha	if pdf=TRUE, the fractional delta used for numerical derivatives with respect to probability, for calculating the pdf as a function of quantiles
customprior	a custom value for the slope of the log prior at the maxlik estimate
minxi	the minimum allowed value of the shape parameter (decrease with caution)
maxxi	the maximum allowed value of the shape parameter (increase with caution)
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
extramodels	logical that indicates whether to run additional calculations and add three additional prediction models (longer runtime)
pdf	logical that indicates whether to run additional calculations and return density functions evaluated at quantiles specified by the input probabilities (longer runtime)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)

rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

### **Details**

The GP distribution has exceedcance distribution function

$$S(x; \mu, \sigma, \xi) = \begin{cases} \left[1 + \xi \left(\frac{x - \mu}{\sigma}\right)\right]^{-1/\xi} & \text{if } \xi \neq 0\\ \exp\left(-\frac{x - \mu}{\sigma}\right) & \text{if } \xi = 0 \end{cases}$$

where x is the random variable and  $\mu, \sigma > 0, \xi$  are the parameters.

The calibrating prior we use is given by

$$\pi(\mu, \sigma, \xi) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

The code will stop with an error if the input data gives a maximum likelihood value for the shape parameter that lies outside the range (minxi, maxxi), since outside this range there may be numerical problems. Such values seldom occur in real observed data for maxima.

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

• ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible

• cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

### Optional Return Values (EVT models only)

q\*\*\*\* optionally returns the following, for EVT models only:

cp\_pdf: the density function at quantiles corresponding to input probabilities p. We provide this for EVD models, because direct estimation of the density function using the DMGS density equation is not possible.

#### Optional Return Values (some EVT models only)

q\*\*\*\* optionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_quantiles: predictive quantiles calculated from Bayesian integration with a flat prior.
- rh\_ml\_quantiles: predictive quantiles calculated from Bayesian integration with the calibrating prior, and the maximmum likelihood estimate for the shape parameter.
- jp\_quantiles: predictive quantiles calculated from Bayesian integration with Jeffreys' prior.

r\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_deviates: predictive random deviates calculated using a Bayesian analysis with a flat prior.
- rh\_ml\_deviates: predictive random deviates calculated using a Bayesian analysis with the RHP-MLE prior.

jp\_deviates: predictive random deviates calculated using a Bayesian analysis with the JP.

d\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_pdf: predictive density function from a Bayesian analysis with the flat prior.
- rh\_ml\_pdf: predictive density function from a Bayesian analysis with the RHP-MLE prior.
- jp\_pdf: predictive density function from a Bayesian analysis with the JP.

p\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_cdf: predictive distribution function from a Bayesian analysis with the flat prior.
- rh\_ml\_cdf: predictive distribution function from a Bayesian analysis with the RHP-MLE prior.
- jp\_cdf: predictive distribution function from a Bayesian analysis with the JP.

These additional predictive distributions are included for comparison with the calibrating prior model. They generally give less good reliability than the calibrating prior.

#### **Details (non-homogeneous models)**

This model is not homogeneous, i.e. it does not have a transitive transformation group, and so there is no right Haar prior and no method for generating exactly reliable predictions. The cp outputs are generated using a prior that has been shown in tests to give reasonable reliability. See Jewson et al. (2025a) for discussion of the prior and test results. For non-homogeneous models, reliability is generally poor for small sample sizes (<20), but is still much better than maximum likelihood. For small sample sizes, it is advisable to check the level of reliability using the routine reltest.

### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),

408 gpd\_k1\_f1fa

• t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),

- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

# Examples

```
#
# example 1
x=fitdistcp::d120gpd_k1_example_data_v1
p=c(1:9)/10
q=qgpd_k1_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgpd_k1_cp)",
main="GPD: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue",lwd=2)
cat(" ml_params=",q$ml_params,"\n")
```

gpd\_k1\_f1fa

The first derivative of the density

### **Description**

The first derivative of the density

gpd\_k1\_f2fa 409

### Usage

```
gpd_k1_f1fa(x, v1, v2, kloc)
```

# Arguments

x a vector of training data values

v1 first parameter

v2 second parameter

kloc the known location parameter

### Value

Vector

gpd\_k1\_f2fa

The second derivative of the density

# Description

The second derivative of the density

# Usage

$$gpd_k1_f2fa(x, v1, v2, kloc)$$

# Arguments

x a vector of training data values

v1 first parameter

v2 second parameter

kloc the known location parameter

#### Value

Matrix

410 gpd\_k1\_fdd

and 1.1 fd	First deminative of the density Created by Stanhan Laugan using De
gpd_k1_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gpd_k1_fd(x, v1, v2, v3)
```

# **Arguments**

X	a vector of training data values	
	_	

v1 first parameterv2 second parameterv3 third parameter

# Value

Vector

gpd_k1_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gpd_k1_fdd(x, v1, v2, v3)
```

# Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter

v3 third parameter

#### Value

Matrix

gpd\_k1\_ldda 411

gnd	k1	_ldda

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

# Usage

```
gpd_k1_ldda(x, v1, v2, kloc)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

kloc the known location parameter

#### Value

Matrix

gpd	l-1	14	ムムっ
ะมน	NΙ	тu	uua

The third derivative of the normalized log-likelihood

## Description

The third derivative of the normalized log-likelihood

### Usage

```
gpd_k1_lddda(x, v1, v2, kloc)
```

# **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

kloc the known location parameter

#### Value

3d array

412 gpd\_k1\_logfdd

 $gpd_k1_logf$ 

Logf for RUST

# Description

Logf for RUST

### Usage

```
gpd_k1_logf(params, x, kloc)
```

### **Arguments**

params model parameters for calculating logf
x a vector of training data values
kloc the known location parameter

#### Value

Scalar value.

gpd\_k1\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gpd_k1_logfdd(x, v1, v2, v3)
```

### **Arguments**

X	a vector	of traini	ng data	values
X	a vector	oi uaiiii	ng uata	varue

v1 first parameterv2 second parameterv3 third parameter

# Value

Matrix

gpd\_k1\_logfddd 413

gpd_k1_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
gpd_k1_logfddd(x, v1, v2, v3)
```

# Arguments

	X	a vector of training	g data values
--	---	----------------------	---------------

v1 first parameterv2 second parameterv3 third parameter

#### Value

3d array

# Description

log-likelihood function

# Usage

```
gpd_k1_loglik(vv, x, kloc)
```

# Arguments

VV	parameters

x a vector of training data valueskloc the known location parameter

## Value

Scalar

414 gpd\_k1\_means

gpd_k1_means	Analytical Expressions for Predictive Means RHP mean based on the
	expectation of DMGS equation 2.1

# Description

Analytical Expressions for Predictive Means RHP mean based on the expectation of DMGS equation 2.1

# Usage

```
gpd_k1_means(
   means,
   ml_params,
   lddi,
   lddd,
   lambdad_rh_flat,
   lambdad_jp,
   nx,
   dim = 2,
   kloc = 0
)
```

# Arguments

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihood

lambdad\_rh\_flat

derivative of the log CRHP-FLAT prior

lambdad\_jp derivative of the log JP prior

nx length of training data dim number of parameters

kloc the known location parameter

#### Value

Two scalars

gpd\_k1\_mu1fa 415

- 1	1 4	4.0	
gpd	ΚI	mu1fa	1

Minus the first derivative of the cdf, at alpha

### **Description**

Minus the first derivative of the cdf, at alpha

### Usage

```
gpd_k1_mu1fa(alpha, v1, v2, kloc)
```

### Arguments

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

kloc the known location parameter

#### Value

Vector

gpd\_k1\_mu2fa

Minus the second derivative of the cdf, at alpha

# Description

Minus the second derivative of the cdf, at alpha

### Usage

```
gpd_k1_mu2fa(alpha, v1, v2, kloc)
```

# **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

kloc the known location parameter

### Value

Matrix

416 gpd\_k1\_pdd

gpd_k1_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gpd_k1_pd(x, v1, v2, v3)
```

# **Arguments**

v1 first parameter v2 second parameter v3 third parameter

# Value

Vector

gpd_k1_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gpd_k1_pdd(x, v1, v2, v3)
```

# Arguments

,umemes	
Х	a vector of training data values
v1	first parameter
v2	second parameter
v3	third parameter

# Value

Matrix

gpd\_k1\_setics 417

gpd_k1_setics Set initial conditions
--------------------------------------

# Description

Set initial conditions

# Usage

```
gpd_k1_setics(x, ics)
```

# Arguments

x a vector of training data values

ics initial conditions for the maximum likelihood search

### Value

Vector

gpd_k1_waic Waic
------------------

# Description

Waic

# Usage

```
gpd_k1_waic(waicscores, x, v1hat, v2hat, kloc, lddi, lddd, lambdad)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
v1hat	first parameter
v2hat	second parameter
kloc	the known location parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior

## Value

Two numeric values.

gumbel\_cp

Gumbel Distribution Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qgumbel_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    rust = FALSE,
    nrust = 1e+05,
    debug = FALSE
)

rgumbel_cp(n, x, rust = FALSE, mlcp = TRUE, debug = FALSE)

dgumbel_cp(x, y = x, rust = FALSE, nrust = 1000, debug = FALSE)
```

```
pgumbel_cp(x, y = x, rust = FALSE, nrust = 1000, debug = FALSE)
tgumbel_cp(n, x, debug = FALSE)
```

#### Arguments

X	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Gumbel distribution has distribution function

$$F(x; \mu, \sigma) = \exp\left(-\exp\left(-\frac{x - \mu}{\sigma}\right)\right)$$

where x is the random variable and  $\mu, \sigma > 0$  are the parameters.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

#### If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUF:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),

- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

424 gumbel\_f2fa

#### **Examples**

```
#
# example 1
x=fitdistcp::d050gumbel_example_data_v1
p=c(1:9)/10
q=qgumbel_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),sub="(from qgumbel_cp)",
main="Gumbel: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

gumbel\_f1fa

The first derivative of the density

### **Description**

The first derivative of the density

### Usage

```
gumbel_f1fa(x, v1, v2)
```

# **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Vector

gumbel\_f2fa

The second derivative of the density

### **Description**

The second derivative of the density

```
gumbel_f2fa(x, v1, v2)
```

gumbel\_fd 425

## **Arguments**

Χ	a vector of training	g data values
---	----------------------	---------------

v1 first parameterv2 second parameter

#### Value

Matrix

gumbel\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gumbel_fd(x, v1, v2)
```

### **Arguments**

x a vector of training data values

v1 first parameterv2 second parameter

### Value

Vector

 $gumbel_fdd$ 

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### **Description**

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gumbel_fdd(x, v1, v2)
```

426 gumbel\_lddda

## **Arguments**

x a vector of training data values

v1 first parameter

v2 second parameter

### Value

Matrix

gumbel\_ldda

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

# Usage

```
gumbel_ldda(x, v1, v2)
```

### **Arguments**

x a vector of training data values

v1 first parameter

v2 second parameter

## Value

Matrix

gumbel\_lddda

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

```
gumbel_lddda(x, v1, v2)
```

gumbel\_logf 427

# Arguments

x a vector of training data values

v1 first parameter

v2 second parameter

### Value

3d array

gumbel\_logf

Logf for RUST

# Description

Logf for RUST

## Usage

```
gumbel_logf(params, x)
```

### **Arguments**

params model parameters for calculating logf x a vector of training data values

#### Value

Scalar value.

gumbel\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
gumbel_logfdd(x, v1, v2)
```

428 gumbel\_loglik

## **Arguments**

Χ	a vector of training data values
---	----------------------------------

v1 first parameterv2 second parameter

#### Value

Matrix

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gumbel_logfddd(x, v1, v2)
```

### **Arguments**

x a vector of training data values

v1 first parameterv2 second parameter

#### Value

3d array

# Description

log-likelihood function

```
gumbel_loglik(vv, x)
```

gumbel\_logscores 429

#### **Arguments**

vv parameters

x a vector of training data values

#### Value

Scalar

gumbel\_logscores Log scores for MLE and F

Log scores for MLE and RHP predictions calculated using leave-one-out

# Description

Log scores for MLE and RHP predictions calculated using leave-one-out

#### Usage

```
gumbel_logscores(logscores, x)
```

### **Arguments**

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data values

#### Value

Two scalars

 ${\tt gumbel\_means}$ 

MLE and RHP predictive means

# Description

MLE and RHP predictive means

```
gumbel_means(means, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 2)
```

gumbel\_mu1fa

## **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

#### Value

Two scalars

 $gumbel\_mu1fa$ 

Minus the first derivative of the cdf, at alpha

# Description

Minus the first derivative of the cdf, at alpha

### Usage

```
gumbel_mu1fa(alpha, v1, v2)
```

# Arguments

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

### Value

Vector

gumbel\_mu2fa 431

7	A C -
gumber	_mu2fa

Minus the second derivative of the cdf, at alpha

# Description

Minus the second derivative of the cdf, at alpha

# Usage

```
gumbel_mu2fa(alpha, v1, v2)
```

# **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameterv2 second parameter

# Value

Matrix

gumbel\_p1fa

The first derivative of the cdf

# Description

The first derivative of the cdf

# Usage

```
gumbel_p1fa(x, v1, v2)
```

### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

## Value

Vector

432 gumbel\_p1\_cp

gumbel\_p1\_cp

Gumbel Distribution with a Predictor, Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qgumbel_p1_cp(
    x,
    t,
    t0 = NA,
    n0 = NA,
    p = seq(0.1, 0.9, 0.1),
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    rust = FALSE,
    nrust = 1e+05,
    predictordata = TRUE,
    centering = TRUE,
    debug = FALSE
```

```
rgumbel_p1_cp(
 n,
 Х,
 t,
 t0 = NA,
 n0 = NA,
 rust = FALSE,
 mlcp = TRUE,
 debug = FALSE
)
dgumbel_p1_cp(
 Х,
  t,
 t0 = NA,
 n0 = NA,
 y = x,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE
)
pgumbel_p1_cp(
 х,
  t,
 t0 = NA,
 n0 = NA,
 y = x,
  rust = FALSE,
 nrust = 1000,
  centering = TRUE,
 debug = FALSE
)
tgumbel_p1_cp(n, x, t, debug = FALSE)
```

# Arguments

X	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
р	a vector of probabilities at which to generate predictive quantiles
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)

waicscores logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime) logical that indicates whether to run additional calculations and return leavelogscores one-out estimates of the log-score (much longer runtime, non-EVT models only) logical that indicates whether DMGS calculations should be run or not (longer dmgs run time) logical that indicates whether RUST-based posterior sampling calculations should rust be run or not (longer run time) the number of posterior samples used in the RUST calculations nrust logical that indicates whether predictordata should be calculated predictordata centering logical that indicates whether the predictor should be centered logical for turning on debug messages debug the number of random samples required logical that indicates whether maxlik and parameter uncertainty calculations mlcp should be performed (turn off to speed up RUST) a vector of values at which to calculate the density and distribution functions У

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Gumbel distribution with a predictor has distribution function

$$F(x; a, b, \sigma) = \exp\left(-\exp\left(-\frac{x - \mu(a, b)}{\sigma}\right)\right)$$

where x is the random variable,  $\mu = a + bt$  is the shape parameter as a function of parameters a, b and predictor t, and  $\sigma > 0$  is the scale parameter.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

## **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

• ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)

• cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),

- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

gumbel\_p1\_f1fa 439

#### **Examples**

```
#
# example 1
x=fitdistcp::d070gumbel_p1_example_data_v1_x
tt=fitdistcp::d070gumbel_p1_example_data_v1_t
p=c(1:9)/10
n0=10
q=qgumbel_p1_cp(x,tt,n0=n0,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qgumbel_p1_cp)",
main="Gumbel w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

gumbel\_p1\_f1fa

The first derivative of the density for DMGS

## **Description**

The first derivative of the density for DMGS

#### Usage

```
gumbel_p1_f1fa(x, t0, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t0	a single value of the predictor (specify either $t0$ or $n0$ but not both)
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Vector

gumbel\_p1\_f2fa

gumbel\_p1\_f1fw

The first derivative of the density for WAIC

## **Description**

The first derivative of the density for WAIC

## Usage

```
gumbel_p1_f1fw(x, t, v1, v2, v3)
```

# Arguments

x a vector of training data values
 t a vector or matrix of predictors
 v1 first parameter
 v2 second parameter
 v3 third parameter

#### Value

Vector

gumbel\_p1\_f2fa

The second derivative of the density for DMGS

## **Description**

The second derivative of the density for DMGS

## Usage

```
gumbel_p1_f2fa(x, t0, v1, v2, v3)
```

# Arguments

Χ	a vector of training data values
---	----------------------------------

to a single value of the predictor (specify either to or no but not both)

v1 first parameterv2 second parameterv3 third parameter

## Value

Matrix

gumbel\_p1\_f2fw 441

gumbel	n1	f2fw
Sallinet	υı	$I \angle I W$

The second derivative of the density for WAIC

# Description

The second derivative of the density for WAIC

# Usage

```
gumbel_p1_f2fw(x, t, v1, v2, v3)
```

# Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter

third parameter

#### Value

Matrix

v3

<pre>gumbel_p1_fd</pre>	First derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gumbel_p1_fd(x, t, v1, v2, v3)
```

# Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

gumbel\_p1\_ldda

# Value

Vector

gumbel_p1_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gumbel_p1_fdd(x, t, v1, v2, v3)
```

# Arguments

Suments	
х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

## Value

Matrix

<pre>gumbel_p1_ldda</pre>	The second derivative of the normalized log-likelihood
---------------------------	--------------------------------------------------------

# Description

The second derivative of the normalized log-likelihood

# Usage

```
gumbel_p1_ldda(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

gumbel\_p1\_lddda 443

## Value

Matrix

gumbel\_p1\_lddda

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

#### Usage

```
gumbel_p1_lddda(x, t, v1, v2, v3)
```

#### **Arguments**

x a vector of training data values t a vector or matrix of predictors v1 first parameter

v2 second parameter v3 third parameter

#### Value

3d array

gumbel\_p1\_logf

Logf for RUST

## **Description**

Logf for RUST

# Usage

```
gumbel_p1_logf(params, x, t)
```

# **Arguments**

params model parameters for calculating logf
x a vector of training data values
t a vector or matrix of predictors

#### Value

Scalar value.

gumbel\_p1\_logfddd

gumbel_p1_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
S =, = S	

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gumbel_p1_logfdd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Matrix

<pre>gumbel_p1_logfddd</pre>	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gumbel_p1_logfddd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

gumbel\_p1\_loglik 445

## Value

3d array

gumbel_p1_loglik	observed log-likelihood function
8asc=_b=s8==	coserred to a time time out function

# Description

observed log-likelihood function

# Usage

```
gumbel_p1_loglik(vv, x, t)
```

# Arguments

vv parameters

x a vector of training data valuest a vector or matrix of predictors

#### Value

Scalar

#### **Description**

Log scores for MLE and RHP predictions calculated using leave-one-out

#### Usage

```
gumbel_p1_logscores(logscores, x, t)
```

# Arguments

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data valuest a vector or matrix of predictors

## Value

Two scalars

gumbel\_p1\_mu1fa

gumbel\_p1\_means Gumbel distribution: RHP mean

#### **Description**

Gumbel distribution: RHP mean

#### Usage

```
gumbel_p1_means(means, t0, ml_params, lddi, lddd, lambdad_rhp, nx, dim)
```

# **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

to a single value of the predictor (specify either to or no but not both)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

#### Value

Two scalars

## **Description**

Minus the first derivative of the cdf, at alpha

# Usage

```
gumbel_p1_mu1fa(alpha, t0, v1, v2, v3)
```

## **Arguments**

alpha	a vector of values of alpha (	(one minus probability)

to a single value of the predictor (specify either to or no but not both)

v1 first parameterv2 second parameterv3 third parameter

gumbel\_p1\_mu2fa 447

## Value

Vector

gumbel\_p1\_mu2fa  $\it M$ 

Minus the second derivative of the cdf, at alpha

# Description

Minus the second derivative of the cdf, at alpha

## Usage

```
gumbel_p1_mu2fa(alpha, t0, v1, v2, v3)
```

## **Arguments**

alpha a vector of values of alpha (one minus probability)

to a single value of the predictor (specify either to or no but not both)

v1 first parameterv2 second parameterv3 third parameter

## Value

Matrix

gumbel\_p1\_p1fa

The first derivative of the cdf

## **Description**

The first derivative of the cdf

## Usage

```
gumbel_p1_p1fa(x, t0, v1, v2, v3)
```

# Arguments

X	a vector of training data values

t0 a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameterv2 second parameterv3 third parameter

## Value

Vector

gumbel\_p1\_p2fa

The second derivative of the cdf

## **Description**

The second derivative of the cdf

## Usage

```
gumbel_p1_p2fa(x, t0, v1, v2, v3)
```

## **Arguments**

x a vector of training data value	X	a vector	of training	data value
-----------------------------------	---	----------	-------------	------------

t0 a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameter v2 second parameter v3 third parameter

## Value

Matrix

gumbel\_p1\_pd

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## **Description**

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gumbel_p1_pd(x, t, v1, v2, v3)
```

# **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameter v2 second parameter v3 third parameter gumbel\_p1\_pdd 449

# Value

Vector

gumbel_p1_pdd Second derivative of the cdf Created by Stephen Jewson using Deby Andrew Clausen and Serguei Sokol
------------------------------------------------------------------------------------------------------------------

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
gumbel_p1_pdd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

# Value

Matrix

```
gumbel_p1_predictordata
```

Predicted Parameter and Generalized Residuals

# Description

Predicted Parameter and Generalized Residuals

```
gumbel_p1_predictordata(predictordata, x, t, t0, params)
```

gumbel\_p1\_waic

# Arguments

predictordata logical that indicates whether to calculate and return predictordata

x a vector of training data valuest a vector or matrix of predictors

to a single value of the predictor (specify either to or no but not both)

params model parameters for calculating logf

#### Value

Two vectors

gumbel\_p1\_waic Waic

## **Description**

Waic

## Usage

```
gumbel_p1_waic(waicscores, x, t, v1hat, v2hat, v3hat, lddi, lddd, lambdad)
```

## **Arguments**

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data valuest a vector or matrix of predictors

v1hat first parameter v2hat second parameter v3hat third parameter

lddi inverse observed information matrixlddd third derivative of log-likelihood

lambdad derivative of the log prior

#### Value

Two numeric values.

gumbel\_p2fa 451

		_
αι im	hal	p2fa
٤um	NCT	$\nu z i a$

The second derivative of the cdf

# Description

The second derivative of the cdf

## Usage

```
gumbel_p2fa(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameterv2 second parameter

## Value

Matrix

gumbel_	pd

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
gumbel_pd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter

v2 second parameter

# Value

Vector

452 gumbel\_waic

gumbel_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
gumbel_pdd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

## Value

Matrix

gumbel_waic
-------------

# Description

Waic

## Usage

```
gumbel_waic(waicscores, x, v1hat, v2hat, lddi, lddd, lambdad)
```

# Arguments

waicscores	logical that indicates whether to	return ectimates for the w	aic1 and waic2 scores
Waltstores	logical that mulcates whether to	ficturin estimates for the w	aici aiiu waicz scores

(longer runtime)

x a vector of training data values

v1hat first parameter v2hat second parameter

1ddi inverse observed information matrix1ddd third derivative of log-likelihood

lambdad derivative of the log prior

#### Value

Two numeric values.

halfnorm\_cp

Half-Normal Distribution Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qhalfnorm_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    fd1 = 0.01,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    rust = FALSE,
    nrust = 1e+05,
    debug = FALSE,
    aderivs = TRUE
)
```

```
rhalfnorm_cp(
 n,
 х,
 fd1 = 0.01,
 rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
dhalfnorm_cp(
 Х,
 y = x,
 fd1 = 0.01,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
phalfnorm_cp(
 х,
 y = x,
 fd1 = 0.01,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
thalfnorm_cp(n, x, fd1 = 0.01, debug = FALSE)
```

# **Arguments** x

X	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
fd1	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the first parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)

rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The half-normal distribution has probability density function

$$f(x;\theta) = \frac{2\theta}{\pi} e^{-\theta^2 x^2/\pi}$$

where  $x \ge 0$  is the random variable and  $\theta > 0$  is the parameter.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\theta) \propto \frac{1}{\theta}$$

as given in Jewson et al. (2025). Some other authors may parametrize the half-normal differently.

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using
posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),

- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

## **Examples**

```
#
# example 1
x=fitdistcp::d020halfnorm_example_data_v1
p=c(1:9)/10
q=qhalfnorm_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles)
xmax=max(q$ml_quantiles,q$cp_quantiles)
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qhalfnorm_cp)",
main="Halfnorm: quantile estimates")
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

460 halfnorm\_f1fa

halfnorm\_f1f

DMGS equation 2.1, f1 term

#### **Description**

DMGS equation 2.1, f1 term

# Usage

```
halfnorm_f1f(y, v1, fd1)
```

# Arguments

y a vector of values at which to calculate the density and distribution functions

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

## Value

Matrix

halfnorm\_f1fa

The first derivative of the density

# Description

The first derivative of the density

The first derivative of the density

# Usage

```
halfnorm_f1fa(x, v1)
halfnorm_f1fa(x, v1)
```

## **Arguments**

x a vector of training data values

v1 first parameter

## Value

Vector

Vector

halfnorm\_f2f 461

halfnorm\_f2f

DMGS equation 2.1, f2 term

#### **Description**

DMGS equation 2.1, f2 term

# Usage

```
halfnorm_f2f(y, v1, fd1)
```

# Arguments

y a vector of values at which to calculate the density and distribution functions

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

## Value

3d array

halfnorm\_f2fa

The second derivative of the density

# Description

The second derivative of the density

The second derivative of the density

# Usage

```
halfnorm_f2fa(x, v1)
halfnorm_f2fa(x, v1)
```

# **Arguments**

x a vector of training data values

v1 first parameter

#### Value

Matrix

Matrix

462 halfnorm\_fdd

halfnorm_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## **Description**

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
halfnorm_fd(x, v1)
halfnorm_fd(x, v1)
```

## **Arguments**

x a vector of training data values

v1 first parameter

#### Value

Vector

Vector

halfnorm_fdd	Second derivative of the density (
···	account the control of the decimally a

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## **Description**

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
halfnorm_fdd(x, v1)
halfnorm_fdd(x, v1)
```

halfnorm\_gg 463

# **Arguments**

x a vector of training data values

v1 first parameter

## Value

Matrix

Matrix

halfnorm\_gg

Expected information matrix

# Description

Expected information matrix

# Usage

```
halfnorm_gg(v1, fd1)
```

# Arguments

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

Square scalar matrix

halfnorm\_gg11

Second derivative of the expected log-likelihood

# Description

Second derivative of the expected log-likelihood

```
halfnorm_gg11(alpha, v1, fd1)
```

464 halfnorm\_ldd

#### **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

Scalar value

halfnorm\_l111

Third derivative of the normalized log-likelihood

## **Description**

Third derivative of the normalized log-likelihood

# Usage

```
halfnorm_l111(x, v1, fd1)
```

## **Arguments**

x a vector of training data values

v1 first parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

Scalar value

halfnorm\_ldd

The second derivative of the normalized log-likelihood

## Description

The second derivative of the normalized log-likelihood

```
halfnorm_ldd(x, v1, fd1)
```

halfnorm\_ldda 465

# Arguments

x a vector of training data values

v1 first parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

## Value

Square scalar matrix

halfnorm\_ldda

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

The second derivative of the normalized log-likelihood

## Usage

```
halfnorm_ldda(x, v1)
halfnorm_ldda(x, v1)
```

# Arguments

x a vector of training data values

v1 first parameter

# Value

Matrix

Matrix

466 halfnorm\_lddda

halfnorm\_lddd

Third derivative tensor of the log-likelihood

## **Description**

Third derivative tensor of the log-likelihood

#### Usage

```
halfnorm_lddd(x, v1, fd1)
```

# **Arguments**

x a vector of training data values

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

Cubic scalar array

halfnorm\_lddda

The third derivative of the normalized log-likelihood

# **Description**

The third derivative of the normalized log-likelihood The third derivative of the normalized log-likelihood

## Usage

```
halfnorm_lddda(x, v1)
halfnorm_lddda(x, v1)
```

## **Arguments**

x a vector of training data values

v1 first parameter

## Value

3d array

3d array

halfnorm\_logf 467

halfnorm\_logf

Logf for RUST

# **Description**

Logf for RUST

## Usage

```
halfnorm_logf(params, x)
```

# **Arguments**

params model parameters for calculating logf x a vector of training data values

#### Value

Scalar value.

halfnorm\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
halfnorm_logfdd(x, v1)
halfnorm_logfdd(x, v1)
```

# Arguments

x a vector of training data values

v1 first parameter

#### Value

Matrix

Matrix

468 halfnorm\_loglik

halfnorm_logfddd Third derivative of the log density Created by Stephen Jewson usin Deriv() by Andrew Clausen and Serguei Sokol	ng
------------------------------------------------------------------------------------------------------------------------------------	----

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
halfnorm_logfddd(x, v1)
halfnorm_logfddd(x, v1)
```

## **Arguments**

x a vector of training data values

v1 first parameter

#### Value

3d array 3d array

halfnorm\_loglik

Log-likelihood function

# Description

Log-likelihood function

## Usage

```
halfnorm_loglik(vv, x)
```

## **Arguments**

vv parameters

x a vector of training data values

## Value

Scalar

halfnorm\_logscores 469

halfnorm_logscores	Log scores for MLE and RHP predictions calculated using leave-one-
	out

#### **Description**

Log scores for MLE and RHP predictions calculated using leave-one-out

#### Usage

```
halfnorm_logscores(logscores, x, fd1 = 0.01, aderivs = TRUE)
```

#### Arguments

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data values

the fractional delta used in the numerical derivatives with respect to the param-

eter

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two scalars

halfnorm\_means MLE and RHP predictive means RHP mean based on the expectation

of DMGS equation 2.1

#### **Description**

MLE and RHP predictive means RHP mean based on the expectation of DMGS equation 2.1

#### Usage

```
halfnorm_means(means, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 1)
```

#### **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

1ddi inverse observed information matrix1ddd third derivative of log-likelihood1ambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters 470 halfnorm\_mu2f

#### Value

Two scalars

halfnorm\_mu1f

DMGS equation 3.3, mul term

# Description

DMGS equation 3.3, mu1 term

# Usage

```
halfnorm_mu1f(alpha, v1, fd1)
```

# Arguments

alpha a vector of values of alpha (one minus probability)

v1 first parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

Matrix

halfnorm\_mu2f

DMGS equation 3.3, mu2 term

# Description

DMGS equation 3.3, mu2 term

# Usage

```
halfnorm_mu2f(alpha, v1, fd1)
```

#### **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

3d array

halfnorm\_p1f 471

	_	
hal	fnorm	n1f
пат	1 1101 111	UII

DMGS equation 2.1, p1 term

### **Description**

DMGS equation 2.1, p1 term

### Usage

```
halfnorm_p1f(y, v1, fd1)
```

# Arguments

y a vector of values at which to calculate the density and distribution functions

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

Matrix

	_	
ha⊥	fnorm	p2f

DMGS equation 2.1, p2 term

#### **Description**

DMGS equation 2.1, p2 term

### Usage

```
halfnorm_p2f(y, v1, fd1)
```

#### **Arguments**

y a vector of values at which to calculate the density and distribution functions

v1 first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

3d array

472 ifvectorthenmatrix

# Description

Waic

#### Usage

```
halfnorm_waic(waicscores, x, v1hat, fd1, lddi, lddd, lambdad, aderivs)
```

# Arguments

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data values

v1hat first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

1ddi inverse observed information matrix 1ddd third derivative of log-likelihood

lambdad derivative of the log prior

aderivs logical for whether to use analytic derivatives (instead of numerical)

# Value

Two numeric values.

# Description

If a variable is a vector, convert it to a matrix

#### Usage

ifvectorthenmatrix(t)

#### **Arguments**

t a vector or matrix of predictors

#### Value

Vector

invgamma\_cp

Inverse Gamma Distribution, Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

### Usage

```
qinvgamma_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    fd1 = 0.01,
    fd2 = 0.01,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    rust = FALSE,
    nrust = 1e+05,
    prior = "type 1",
    debug = FALSE,
    aderivs = TRUE
)
```

```
rinvgamma_cp(
 n,
 х,
 fd1 = 0.01,
 fd2 = 0.01,
 rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
dinvgamma_cp(
 Х,
 y = x,
 fd1 = 0.01,
 fd2 = 0.01,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
pinvgamma_cp(
 х,
 y = x,
 fd1 = 0.01,
 fd2 = 0.01,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
tinvgamma_cp(n, x, fd1 = 0.01, fd2 = 0.01, debug = FALSE)
```

### **Arguments**

X	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
fd1	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the first parameter $\frac{1}{2}$
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)

logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
prior	logical indicating which prior to use
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- $\bullet$  predicted parameter: the estimated value for parameter, as a function of the predictor.
- ullet adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

• ml\_params: maximum likelihood estimates for the parameters.

- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Inverse Gamma distribution has probability density function

$$f(x; \alpha, \sigma) = \frac{1}{x\Gamma(\alpha)} \left(\frac{\sigma}{x}\right)^{\alpha} e^{-\sigma/x}$$

where  $x \ge 0$  is the random variable and  $\alpha > 0, \sigma > 0$  are the parameters.

The calibrating prior we use is

$$\pi(\alpha, \sigma) \propto \frac{1}{\alpha \sigma}$$

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),

- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

### **Examples**

```
# # example 1
x=fitdistcp::d101invgamma_example_data_v1
p=c(1:9)/10
q=qinvgamma_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qinvgamma_cp)",
main="Invgamma: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

480 invgamma\_f1fa

invgamma_f1	
	t

DMGS equation 3.3, f1 term

# Description

DMGS equation 3.3, f1 term

# Usage

```
invgamma_f1f(y, v1, fd1, v2, fd2)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-

eter

# Value

Matrix

invo	gamma_	f1	fa
TIIV	≾alllllla_	_	Ιd

The first derivative of the density

# Description

The first derivative of the density

# Usage

```
invgamma_f1fa(x, v1, v2)
```

# Arguments

x a vector of training	data values
------------------------	-------------

v1 first parameter v2 second parameter

#### Value

Vector

invgamma\_f2f 481

	000
invgamma_	+ 7+
TIIV E allillia	141

DMGS equation 3.3, f2 term

# Description

DMGS equation 3.3, f2 term

# Usage

```
invgamma_f2f(y, v1, fd1, v2, fd2)
```

# Arguments

у	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

# Value

3d array

i	nvgamma	f2fa
_	II v gaiiiii ia	_141a

The second derivative of the density

# Description

The second derivative of the density

# Usage

```
invgamma_f2fa(x, v1, v2)
```

# Arguments

X	a vector of	training	data values

v1 first parameter v2 second parameter

#### Value

Matrix

482 invgamma\_fdd

invgamma_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
invgamma_fd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Vector

invgamma_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
invgamma_fdd(x, v1, v2)
```

# Arguments

S
5

v1 first parameter v2 second parameter

#### Value

Matrix

invgamma\_ldd 483

invgamma_ldd Second derivative matrix of the normalized log-like	elihood
------------------------------------------------------------------	---------

# Description

Second derivative matrix of the normalized log-likelihood

# Usage

```
invgamma_ldd(x, v1, fd1, v2, fd2)
```

# Arguments

X	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

# Value

Square scalar matrix

invgamma_ldda	The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

# Usage

```
invgamma_ldda(x, v1, v2)
```

# Arguments

X	a vector of training data values
v1	first parameter
v2	second parameter

#### Value

Matrix

484 invgamma\_lddda

inv	gamma	1.4	MA
TIIV	gaillilla	_ T (	luu

Third derivative tensor of the normalized log-likelihood

#### **Description**

Third derivative tensor of the normalized log-likelihood

#### Usage

```
invgamma_lddd(x, v1, fd1, v2, fd2)
```

#### **Arguments**

x	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter

v2 second parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

Cubic scalar array

invgamma\_lddda

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

#### Usage

```
invgamma_lddda(x, v1, v2)
```

# Arguments

x a vector of training data	values
-----------------------------	--------

v1 first parameter v2 second parameter

### Value

3d array

invgamma\_lmn 485

invgamma_lmn One component of the second derivative of the normalized log- likelihood
------------------------------------------------------------------------------------------

# Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
invgamma_lmn(x, v1, fd1, v2, fd2, mm, nn)
```

# Arguments

x	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

# Value

Scalar value

invgamma_lmnp	One component of the second derivative of the normalized log-likelihood
---------------	-------------------------------------------------------------------------

# Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
invgamma_lmnp(x, v1, fd1, v2, fd2, mm, nn, rr)
```

486 invgamma\_logf

# Arguments

X	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

# Value

Scalar value

	Logf for RUST	invgamma_logf
--	---------------	---------------

# Description

Logf for RUST

# Usage

```
invgamma_logf(params, x)
```

# Arguments

params model parameters for calculating logf x a vector of training data values

# Value

Scalar value.

invgamma\_logfdd 487

invgamma_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
iiivgaliilia_iogi uu	

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
invgamma_logfdd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

# Value

Matrix

invgamma_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
invgamma_logfddd(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameterv2 second parameter

#### Value

3d array

488 invgamma\_logscores

invgamma_loglik	log-likelihood function
-----------------	-------------------------

#### **Description**

log-likelihood function

#### Usage

```
invgamma_loglik(vv, x)
```

# **Arguments**

vv parameters

x a vector of training data values

#### Value

Scalar

invgamma\_logscores

Log scores for MLE and cp predictions calculated using leave-one-out

#### **Description**

Log scores for MLE and cp predictions calculated using leave-one-out

#### Usage

```
invgamma_logscores(logscores, x, fd1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

# Arguments

1	ogscores [	logical	that ind	icates w	hether t	o return	leave-one-out	estimates	estimates of	the
---	------------	---------	----------	----------	----------	----------	---------------	-----------	--------------	-----

log-score (much longer runtime)

x a vector of training data values

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

fd2 the fractional delta used in the numerical derivatives with respect to the param-

eter

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two scalars

invgamma\_mu1f 489

invgamma_mu1f DMGS equation 3.3, mu1 term	
-------------------------------------------	--

# Description

DMGS equation 3.3, mu1 term

# Usage

```
invgamma_mu1f(alpha, v1, fd1, v2, fd2)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

Matrix

# Description

DMGS equation 3.3, mu2 term

# Usage

```
invgamma_mu2f(alpha, v1, fd1, v2, fd2)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

490 invgamma\_p2f

# Value

3d array

invgamma\_p1f

DMGS equation 3.3, p1 term

# Description

DMGS equation 3.3, p1 term

# Usage

```
invgamma_p1f(y, v1, fd1, v2, fd2)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

# Value

Matrix

invgamma\_p2f

DMGS equation 3.3, p2 term

# Description

DMGS equation 3.3, p2 term

# Usage

```
invgamma_p2f(y, v1, fd1, v2, fd2)
```

invgamma\_waic 491

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

3d array

invgamma\_waic Waic

# Description

Waic

# Usage

```
invgamma_waic(
  waicscores,
  x,
  v1hat,
  fd1,
  v2hat,
  fd2,
  lddi,
  lddd,
  lambdad,
  aderivs
)
```

# Arguments

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)

x a vector of training data values

v1hat first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the parameter

v2hat second parameter

fd2	the fractional	dalta usad in the	a numerical deri	ivotivac with rac	pect to the param-
TUZ	uic machonai c	icha uscu ili un	i iiuiiiciicai ucii	ivanivės wini ies	pect to the param-

eter

lddi inverse observed information matrix

third derivative of log-likelihood

lambdad derivative of the log prior

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

1ddd

Two numeric values.

invgauss\_cp

Inverse Gauss Distribution, Predictions Based on a Calibrating Prior

### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

### Usage

```
qinvgauss_cp(
  х,
  p = seq(0.1, 0.9, 0.1),
  fd1 = 0.01,
  fd2 = 0.01,
 means = FALSE,
 waicscores = FALSE,
  logscores = FALSE,
  dmgs = TRUE,
  rust = FALSE,
  nrust = 1e+05,
  prior = "type 1",
  debug = FALSE,
  aderivs = TRUE
)
rinvgauss_cp(
  n,
 Х,
  fd1 = 0.01,
  fd2 = 0.01,
  rust = FALSE,
  prior = "type 1",
 mlcp = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
dinvgauss_cp(
  х,
 y = x,
 fd1 = 0.01,
  fd2 = 0.01,
  rust = FALSE,
 nrust = 1000,
 prior = "type 1",
 debug = FALSE,
  aderivs = TRUE
)
pinvgauss_cp(
 х,
  y = x,
  fd1 = 0.01,
  fd2 = 0.01,
  rust = FALSE,
  nrust = 1000,
```

```
prior = "type 1",
  debug = FALSE,
  aderivs = TRUE
)

tinvgauss_cp(n, x, fd1 = 0.01, fd2 = 0.01, prior = "type 1", debug = FALSE)
```

#### **Arguments**

x	a vector of training data values
p	a vector of probabilities at which to generate predictive quantiles
fd1	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the first parameter
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
prior	logical indicating which prior to use
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.

- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Inverse Gaussian distribution has probability density function

$$f(x; \mu, \phi) = \left(\frac{1}{2\pi\phi x^3}\right)^{1/2} \exp\left(-\frac{(x-\mu)^2}{2\mu^2\phi x}\right)$$

where  $x \ge 0$  is the random variable and  $\mu > 0, \phi > 0$  are the parameters.

The calibrating prior we use by default is

$$\pi(\alpha,\sigma) \propto \frac{1}{\phi}$$

The prior

$$\pi(\alpha, \sigma) \propto \frac{1}{\mu \phi}$$

is also available as an option with prior="type 2".

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),

invgauss\_f1f 499

- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

### **Examples**

```
debug=FALSE
# example 1 can go wrong for small sample sizes, so I've increased to 50
#
# example 1
if(debug)cat("example 1\n")
x=fitdistcp::d102invgauss_example_data_v1
if(debug)cat("x=",x,"\n")
p=c(1:9)/10
q=qinvgauss_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qinvgauss_cp)",
main="Invgauss: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

invgauss\_f1f

DMGS equation 3.3, f1 term

# Description

DMGS equation 3.3, f1 term

#### Usage

```
invgauss_f1f(y, v1, fd1, v2, fd2)
```

#### **Arguments**

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter

500 invgauss\_f2f

# Value

Matrix

invgauss\_f1fa

The first derivative of the density

# Description

The first derivative of the density

# Usage

```
invgauss_f1fa(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Vector

invgauss\_f2f

DMGS equation 3.3, f2 term

# Description

DMGS equation 3.3, f2 term

# Usage

```
invgauss_f2f(y, v1, fd1, v2, fd2)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

invgauss\_f2fa 501

# Value

3d array

invgauss\_f2fa

The second derivative of the density

# Description

The second derivative of the density

### Usage

```
invgauss_f2fa(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Matrix

invgauss\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
invgauss_fd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Vector

502 invgauss\_ldd

invgauss_fdd Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol	De-
---------------------------------------------------------------------------------------------------------------------------	-----

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
invgauss_fdd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Matrix

invgauss_ldd Second derivative matrix of the normalized log-likelihood
------------------------------------------------------------------------

# Description

Second derivative matrix of the normalized log-likelihood

# Usage

```
invgauss_ldd(x, v1, fd1, v2, fd2)
```

# Arguments

X	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

Square scalar matrix

invgauss\_ldda 503

i	nvg	าลเม	SS	1	dd	a
_	11 V S	,uu	33		uu	u

The second derivative of the normalized log-likelihood

#### **Description**

The second derivative of the normalized log-likelihood

### Usage

```
invgauss_ldda(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Matrix

invgauss\_lddd

Third derivative tensor of the normalized log-likelihood

# Description

Third derivative tensor of the normalized log-likelihood

# Usage

```
invgauss_lddd(x, v1, fd1, v2, fd2)
```

# Arguments

X	a vector	of training	data values

v1 first parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

v2 second parameter

the fractional delta used in the numerical derivatives with respect to the param-

eter

#### Value

Cubic scalar array

504 invgauss\_lmn

invgauss_	1ddda
IIIVgauss_	_tuuua

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

# Usage

```
invgauss_lddda(x, v1, v2)
```

# **Arguments**

		c . · ·	1 . 1
Y	a vector of	of fraining	data values
^	a vector o	n uaning	uata varues

v1 first parameter v2 second parameter

#### Value

3d array

invgauss_lmn	One component of the second derivative of the normalized log
	likelihood

# Description

One component of the second derivative of the normalized log-likelihood

# Usage

```
invgauss_lmn(x, v1, fd1, v2, fd2, mm, nn)
```

# Arguments

X	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

invgauss\_lmnp 505

### Value

Scalar value

invgauss_lmnp	One component of the second derivative of the normalized log-likelihood
---------------	-------------------------------------------------------------------------

## Description

One component of the second derivative of the normalized log-likelihood

### Usage

```
invgauss_lmnp(x, v1, fd1, v2, fd2, mm, nn, rr)
```

## Arguments

X	a vector of training data values
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

### Value

Scalar value

invgauss_logf	Logf for RUST	

# Description

Logf for RUST

```
invgauss_logf(params, x, prior)
```

506 invgauss\_logfddd

#### **Arguments**

params model parameters for calculating logf

x a vector of training data values

prior logical indicating which prior to use

#### Value

Scalar value.

invgauss\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
invgauss_logfdd(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameterv2 second parameter

#### Value

Matrix

invgauss\_logfddd

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### **Description**

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
invgauss_logfddd(x, v1, v2)
```

invgauss\_loglik 507

## Arguments

x a vector of training data values

v1 first parameter

v2 second parameter

#### Value

3d array

invgauss\_loglik

log-likelihood function

# Description

log-likelihood function

### Usage

```
invgauss_loglik(vv, x)
```

### Arguments

vv parameters

x a vector of training data values

#### Value

Scalar

invgauss\_logscores

Log scores for MLE and RHP predictions calculated using leave-one-out

## Description

Log scores for MLE and RHP predictions calculated using leave-one-out

```
invgauss_logscores(logscores, x, prior, fd1 = 0.01, fd2 = 0.01, aderivs = TRUE)
```

508 invgauss\_means

#### **Arguments**

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data values

prior logical indicating which prior to use

fd1 the fractional delta used in the numerical derivatives with respect to the param-

eter

the fractional delta used in the numerical derivatives with respect to the param-

eter

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two scalars

invgauss\_means

MLE and RHP predictive means

#### Description

MLE and RHP predictive means

#### Usage

```
invgauss_means(means, ml_params, lddi, lddd, lambdad_cp, nx, dim = 2)
```

#### Arguments

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihood

lambdad\_cpderivative of the log priornxlength of training datadimnumber of parameters

#### Value

Two scalars

invgauss\_mu1f 509

invgauss_mu1f	DMGS equation 3.3, mu1 term
---------------	-----------------------------

## Description

DMGS equation 3.3, mu1 term

## Usage

```
invgauss_mu1f(alpha, v1, fd1, v2, fd2)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

Matrix

invgauss_mu2f	DMGS equation 3.3, mu2 term	

## Description

DMGS equation 3.3, mu2 term

## Usage

```
invgauss_mu2f(alpha, v1, fd1, v2, fd2)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

510 invgauss\_p2f

### Value

3d array

invgauss\_p1f

DMGS equation 3.3, p1 term

## Description

DMGS equation 3.3, p1 term

## Usage

```
invgauss_p1f(y, v1, fd1, v2, fd2)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

## Value

Matrix

invgauss\_p2f

DMGS equation 3.3, p2 term

## Description

DMGS equation 3.3, p2 term

```
invgauss_p2f(y, v1, fd1, v2, fd2)
```

invgauss\_waic 511

## Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
fd1	the fractional delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter

#### Value

3d array

invgauss\_waic Waic

## Description

Waic

### Usage

```
invgauss_waic(
  waicscores,
  x,
  v1hat,
  fd1,
  v2hat,
  fd2,
  lddi,
  lddd,
  lambdad,
  aderivs
)
```

## Arguments

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)

x a vector of training data values

v1hat first parameter

fd1 the fractional delta used in the numerical derivatives with respect to the parameter

v2hat second parameter

*jpf3p* 

the fractional delta used in the numerical derivatives with respect to the param-

eter

lddi inverse observed information matrix lddd third derivative of log-likelihood

lambdad derivative of the log prior

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two numeric values.

jpf2p

Jeffreys' Prior with two parameters

## Description

Jeffreys' Prior with two parameters

#### Usage

```
jpf2p(ggd, detg, ggi)
```

### Arguments

ggd gradient of the expected information matrix
detg determinant of the expected information matrix
ggi inverse of the expected information matrix

#### Value

Vector of 2 values

jpf3p Jeffreys' Prior with three parameters

## Description

Jeffreys' Prior with three parameters

```
jpf3p(ggd, detg, ggi)
```

*jpf4p* 513

#### Arguments

ggd gradient of the expected information matrix
detg determinant of the expected information matrix
ggi inverse of the expected information matrix

#### Value

Vector of 3 values

jpf4p Jeffreys' Prior with four parameters

### Description

Jeffreys' Prior with four parameters

#### Usage

```
jpf4p(ggd, detg, ggi)
```

#### **Arguments**

ggd gradient of the expected information matrix
detg determinant of the expected information matrix
ggi inverse of the expected information matrix

#### Value

Vector of 4 values

Inorm\_cp Log-normal Distribution Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.

- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qlnorm_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    rust = FALSE,
    nrust = 1e+05,
    debug = FALSE
)

rlnorm_cp(n, x, rust = FALSE, mlcp = TRUE, debug = FALSE)

dlnorm_cp(x, y = x, rust = FALSE, nrust = 1000, debug = FALSE)

plnorm_cp(x, y = x, rust = FALSE, nrust = 1000, debug = FALSE)

tlnorm_cp(n, x, debug = FALSE)
```

#### **Arguments**

X	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)

nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.

• cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The log normal distribution has probability density function

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma x} e^{-(\log(x) - \mu)^2/(2\sigma^2)}$$

where x is the random variable and  $\mu, \sigma > 0$  are the parameters.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (analytic integration)**

For this model, the Bayesian prediction equation is integrated analytically.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),

 t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),

- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
#
# example 1
x=fitdistcp::d035lnorm_example_data_v1
p=c(1:9)/10
q=qlnorm_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qlnorm_cp)",
main="Log-normal: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

lnorm\_dmgs\_cp

Log-normal Distribution Predictions Based on a Calibrating Prior, using DMGS (for testing only)

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qlnorm_dmgs_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    debug = FALSE
)

rlnorm_dmgs_cp(n, x, mlcp = TRUE, debug = FALSE)

dlnorm_dmgs_cp(x, y = x, debug = FALSE)

plnorm_dmgs_cp(x, y, debug = FALSE)
```

#### **Arguments**

x a vector of training data values

p a vector of probabilities at which to generate predictive quantiles

means logical that indicates whether to run additional calculations and return analytical

estimates for the distribution means (longer runtime)

waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
debug	logical for turning on debug messages
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The log normal distribution has probability density function

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma x} e^{-(\log(x) - \mu)^2/(2\sigma^2)}$$

where x is the random variable and  $\mu, \sigma > 0$  are the parameters.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

• ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible

• cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

 ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),

- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

## **Examples**

```
#
# example 1
x=fitdistcp::d035lnorm_example_data_v1
p=c(1:9)/10
q=qlnorm_dmgs_cp(x,p)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qlnorm_dmgs_cp)",
main="Log-normal_DMGS: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
```

 ${\tt lnorm\_dmgs\_loglik}$ 

log-likelihood function

### Description

log-likelihood function

#### Usage

```
lnorm_dmgs_loglik(vv, x)
```

#### **Arguments**

vv parameters

x a vector of training data values

#### Value

Scalar value.

lnorm\_dmgs\_logscores La

Log scores for MLE and RHP predictions calculated using leave-one-out

### Description

Log scores for MLE and RHP predictions calculated using leave-one-out

#### Usage

```
lnorm_dmgs_logscores(logscores, x)
```

## Arguments

logiscores logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data values

#### Value

Two scalars

lnorm\_dmgs\_means 527

lnorm\_dmgs\_means

MLE and RHP predictive means

### Description

MLE and RHP predictive means

### Usage

```
lnorm_dmgs_means(means, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 2)
```

### Arguments

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

#### Value

Two scalars

lnorm\_dmgs\_waic
Waic

### Description

Waic

```
lnorm_dmgs_waic(waicscores, x, v1hat, v2hat, lddi, lddd, lambdad)
```

528 lnorm\_f1fa

### Arguments

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data values

v1hat first parameter v2hat second parameter

lddi inverse observed information matrixlddd third derivative of log-likelihood

lambdad derivative of the log prior

#### Value

Two numeric values.

lnorm\_f1fa

The first derivative of the density

### Description

The first derivative of the density

### Usage

```
lnorm_f1fa(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Vector

lnorm\_f2fa 529

7			COC-
	ΠO	r III	f2fa

The second derivative of the density

### Description

The second derivative of the density

#### Usage

```
lnorm_f2fa(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameterv2 second parameter

#### Value

Matrix

lnorm\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
lnorm_fd(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameterv2 second parameter

## Value

Vector

lnorm\_ldda

lnorm_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
lnorm_fdd(x, v1, v2)
```

### Arguments

x a vector of training data values

v1 first parameterv2 second parameter

#### Value

Matrix

lnorm\_ldda

The second derivative of the normalized log-likelihood

### Description

The second derivative of the normalized log-likelihood

#### Usage

```
lnorm_ldda(x, v1, v2)
```

### Arguments

x a vector of training data values

v1 first parameterv2 second parameter

## Value

Matrix

lnorm\_lddda 531

lnorm\_lddda

The third derivative of the normalized log-likelihood

### Description

The third derivative of the normalized log-likelihood

## Usage

```
lnorm_lddda(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameter

v2 second parameter

#### Value

3d array

lnorm\_logf

Logf for RUST

### Description

```
Logf for RUST
```

### Usage

```
lnorm_logf(params, x)
```

## **Arguments**

params model parameters for calculating logf

x a vector of training data values

#### Value

Scalar value.

lnorm\_logfddd

lnorm_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
lnorm_logfdd(x, v1, v2)
```

### Arguments

x a vector of training data values

v1 first parameter v2 second parameter

### Value

Matrix

lnorm_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

### Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
lnorm_logfddd(x, v1, v2)
```

### **Arguments**

X	a vector of training data values
X	a vector of training data values

v1 first parameter v2 second parameter

#### Value

3d array

lnorm\_logscores 533

Inorm_logscores Log scores for MLE and RHP predictions calculated using leave-one- out	or MLE and RHP predictions calculated using lea	e-one-
-------------------------------------------------------------------------------------------	-------------------------------------------------	--------

### Description

Log scores for MLE and RHP predictions calculated using leave-one-out

#### Usage

```
lnorm_logscores(logscores, x)
```

### Arguments

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data values

#### Value

Two scalars

lnorm_mu1fa	Minus the first derivative of the cdf, at alpha
-------------	-------------------------------------------------

### Description

Minus the first derivative of the cdf, at alpha

### Usage

```
lnorm_mu1fa(alpha, v1, v2)
```

## Arguments

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

#### Value

Vector

534 Inorm\_p1fa

lnorm\_mu2fa

Minus the second derivative of the cdf, at alpha

### Description

Minus the second derivative of the cdf, at alpha

### Usage

```
lnorm_mu2fa(alpha, v1, v2)
```

## Arguments

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

#### Value

Matrix

lnorm\_p1fa

The first derivative of the cdf

## Description

The first derivative of the cdf

## Usage

```
lnorm_p1fa(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Vector

lnorm\_p1\_cp

Log-normal Distribution with a Predictor, Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qlnorm_p1_cp(
    x,
    t,
    t0 = NA,
    n0 = NA,
    p = seq(0.1, 0.9, 0.1),
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    rust = FALSE,
    nrust = 1e+05,
    centering = TRUE,
    debug = FALSE
)
```

```
rlnorm_p1_cp(
 n,
 Х,
  t,
 t0 = NA,
 n0 = NA,
 rust = FALSE,
 mlcp = TRUE,
 debug = FALSE
)
dlnorm_p1_cp(
 Х,
  t,
  t0 = NA,
 n0 = NA,
 y = x,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE
)
plnorm_p1_cp(
 Χ,
  t,
 t0 = NA,
 n0 = NA,
 y = x,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE
)
tlnorm_p1_cp(n, x, t, debug = FALSE)
```

### Arguments

Х	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
р	a vector of probabilities at which to generate predictive quantiles
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)

logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
centering	logical that indicates whether the predictor should be centered
debug	logical for turning on debug messages
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
y	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The log normal distribution with a predictor has probability density function

$$f(x; a, b, \sigma) = \frac{1}{\sqrt{2\pi}x\sigma} e^{-(\log(x) - \mu(a, b))^2/(2\sigma^2)}$$

where x is the random variable,  $\mu = a + bt$  is the location parameter of the log of the random variable, modelled as a function of parameters a, b and predictor t, and  $\sigma > 0$  is the scale parameter of the log of the random variable.

The calibrating prior is given by the right Haar prior, which is

$$\pi(a,b,\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\* optionally returns the following:

If rust=TRUE:

 ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (analytic integration)**

For this model, the Bayesian prediction equation is integrated analytically.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),

lnorm\_p1\_cp 541

- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

### **Examples**

```
#
# example 1
x=fitdistcp::d061lnorm_p1_example_data_v1_x
tt=fitdistcp::d061lnorm_p1_example_data_v1_t
p=c(1:9)/10
n0=10
q=qlnorm_p1_cp(x,tt,n0=n0,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qlnorm_p1_cp)",
main="Log-Normal w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

542 lnorm\_p1\_f1fw

lnorm\_p1\_f1fa

The first derivative of the density for DMGS

## **Description**

The first derivative of the density for DMGS

## Usage

```
lnorm_p1_f1fa(x, t0, v1, v2, v3)
```

# Arguments

x a vector of training data values

t0 a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameterv2 second parameterv3 third parameter

## Value

Vector

lnorm\_p1\_f1fw

The first derivative of the density for WAIC

# Description

The first derivative of the density for WAIC

## Usage

```
lnorm_p1_f1fw(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameter v2 second parameter v3 third parameter

## Value

Vector

lnorm\_p1\_f2fa 543

lnorm\_p1\_f2fa

The second derivative of the density for DMGS

## **Description**

The second derivative of the density for DMGS

## Usage

```
lnorm_p1_f2fa(x, t0, v1, v2, v3)
```

# Arguments

x a vector of training data values

t0 a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameterv2 second parameterv3 third parameter

## Value

Matrix

lnorm\_p1\_f2fw

The second derivative of the density for WAIC

## Description

The second derivative of the density for WAIC

## Usage

```
lnorm_p1_f2fw(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameter v2 second parameter v3 third parameter

### Value

Matrix

544 lnorm\_p1\_fdd

lnorm_p1_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
lnorm_p1_fd(x, t, v1, v2, v3)
```

# **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Vector

lnorm_p1_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
lnorm_p1_fdd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

lnorm\_p1\_ldda 545

## Value

Matrix

lnorm\_p1\_ldda

The second derivative of the normalized log-likelihood

## **Description**

The second derivative of the normalized log-likelihood

# Usage

```
lnorm_p1_ldda(x, t, v1, v2, v3)
```

## **Arguments**

x a vector of training data values
 t a vector or matrix of predictors
 v1 first parameter
 v2 second parameter

v2 second parameterv3 third parameter

## Value

Matrix

lnorm\_p1\_lddda

The third derivative of the normalized log-likelihood

## **Description**

The third derivative of the normalized log-likelihood

## Usage

```
lnorm_p1_lddda(x, t, v1, v2, v3)
```

# Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors

546 lnorm\_p1\_logfdd

## Value

3d array

lnorm\_p1\_logf

Logf for RUST

## **Description**

Logf for RUST

## Usage

```
lnorm_p1_logf(params, x, t)
```

## **Arguments**

params model parameters for calculating logf
x a vector of training data values
t a vector or matrix of predictors

## Value

Scalar value.

lnorm\_p1\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
lnorm_p1_logfdd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors

lnorm\_p1\_logfddd 547

## Value

Matrix

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
lnorm_p1_logfddd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

### Value

3d array

 $lnorm\_p1\_loglik \qquad \qquad Log-normal-with-p1\ observed\ log-likelihood\ function$ 

# Description

Log-normal-with-p1 observed log-likelihood function

## Usage

```
lnorm_p1_loglik(vv, x, t)
```

# Arguments

rameters

x a vector of training data valuest a vector or matrix of predictors

lnorm\_p1\_mu1fa

## Value

Scalar

## **Description**

Log scores for MLE and RHP predictions calculated using leave-one-out

## Usage

```
lnorm_p1_logscores(logscores, x, t)
```

### **Arguments**

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data valuest a vector or matrix of predictors

### Value

Two scalars

# Description

Minus the first derivative of the cdf, at alpha

## Usage

```
lnorm_p1_mu1fa(alpha, t0, v1, v2, v3)
```

# Arguments

alph	a a	vector of va	lues of al	lpha (one 1	minus prol	bability)
------	-----	--------------	------------	-------------	------------	-----------

t0 a single value of the predictor (specify either t0 or n0 but not both)

lnorm\_p1\_mu2fa 549

## Value

Vector

lnorm\_p1\_mu2fa

Minus the second derivative of the cdf, at alpha

## Description

Minus the second derivative of the cdf, at alpha

## Usage

```
lnorm_p1_mu2fa(alpha, t0, v1, v2, v3)
```

## **Arguments**

alpha a vector of values of alpha (one minus probability)

to a single value of the predictor (specify either to or no but not both)

v1 first parameterv2 second parameterv3 third parameter

## Value

Matrix

lnorm\_p1\_p1fa

The first derivative of the cdf

## **Description**

The first derivative of the cdf

# Usage

```
lnorm_p1_p1fa(x, t0, v1, v2, v3)
```

# Arguments

X	a vector	of	training	data	values
•		-			

t0 a single value of the predictor (specify either t0 or n0 but not both)

lnorm\_p1\_pd

## Value

Vector

lnorm\_p1\_p2fa

The second derivative of the cdf

## **Description**

The second derivative of the cdf

## Usage

```
lnorm_p1_p2fa(x, t0, v1, v2, v3)
```

## **Arguments**

	X	a vector of training data value
--	---	---------------------------------

to a single value of the predictor (specify either to or no but not both)

v1 first parameter v2 second parameter v3 third parameter

## Value

Matrix

lnorm\_p1\_pd

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## **Description**

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
lnorm_p1_pd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors

lnorm\_p1\_pdd 551

# Value

Vector

lnorm_p1_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
lnorm_p1_pdd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

# Value

Matrix

lnorm\_p1\_predictordata

Predicted Parameter and Generalized Residuals

# Description

Predicted Parameter and Generalized Residuals

```
lnorm_p1_predictordata(x, t, t0, params)
```

552 Inorm\_p1\_waic

# Arguments

x a vector of training data valuest a vector or matrix of predictors

t0 a single value of the predictor (specify either t0 or n0 but not both)

params model parameters for calculating logf

# Value

Two vectors

# Description

Waic

# Usage

```
lnorm_p1_waic(waicscores, x, t, v1hat, v2hat, v3hat)
```

third parameter

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
x	a vector of training data values
t	a vector or matrix of predictors
v1hat	first parameter
v2hat	second parameter

## Value

v3hat

Two numeric values.

lnorm\_p2fa 553

٦.			- 2 C -
- 11	าดเ	rm	p2fa

The second derivative of the cdf

# Description

The second derivative of the cdf

## Usage

```
lnorm_p2fa(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

## Value

Matrix

lnorm\_pd

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
lnorm_pd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter

v2 second parameter

# Value

Vector

lnorm\_waic

lnorm_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
lnorm_pdd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

## Value

Matrix

lnorm_waic	Waic for RUST

# Description

Waic for RUST

## Usage

```
lnorm_waic(waicscores, x, v1hat, v2hat)
```

# Arguments

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data values

v1hat first parameter v2hat second parameter

### Value

Two numeric values.

logis\_cp

Logistic Distribution Predictions Based on a Calibrating Prior

### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qlogis_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    rust = FALSE,
    nrust = 1e+05,
    debug = FALSE
)

rlogis_cp(n, x, rust = FALSE, mlcp = TRUE, debug = FALSE)

dlogis_cp(x, y = x, rust = FALSE, nrust = 1000, debug = FALSE)
```

```
plogis_cp(x, y = x, rust = FALSE, nrust = 1000, debug = FALSE)
tlogis_cp(n, x, debug = FALSE)
```

### **Arguments**

X	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

### **Details of the Model**

The logistic distribution has distribution function

$$f(x; \mu, \sigma) = \frac{1}{1 + e^{-(x-\mu)/\sigma}}$$

where x is the random variable and  $\mu, \sigma > 0$  are the parameters.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

### If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

#### If rust=TRUE:

 ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

### **Details (non-homogeneous models)**

This model is not homogeneous, i.e. it does not have a transitive transformation group, and so there is no right Haar prior and no method for generating exactly reliable predictions. The cp outputs are generated using a prior that has been shown in tests to give reasonable reliability. See Jewson et al. (2025a) for discussion of the prior and test results. For non-homogeneous models, reliability is generally poor for small sample sizes (<20), but is still much better than maximum likelihood. For small sample sizes, it is advisable to check the level of reliability using the routine reltest.

### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),

- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

logis\_f1fa 561

## **Examples**

```
# # example 1
x=fitdistcp::d040logis_example_data_v1
p=c(1:9)/10
q=qlogis_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qlogis_cp)",
main="Logistic: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

logis\_f1fa

The first derivative of the density

## **Description**

The first derivative of the density

## Usage

```
logis_f1fa(x, v1, v2)
```

# **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

### Value

Vector

logis\_f2fa

The second derivative of the density

## **Description**

The second derivative of the density

```
logis_f2fa(x, v1, v2)
```

562 logis\_fdd

### **Arguments**

x a	vector of	training	data va	lues
-----	-----------	----------	---------	------

v1 first parameterv2 second parameter

### Value

Matrix

logis\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
logis_fd(x, v1, v2)
```

## **Arguments**

x a vector of training data values

v1 first parameterv2 second parameter

## Value

Vector

logis\_fdd Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## **Description**

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
logis_fdd(x, v1, v2)
```

logis\_ldda 563

# Arguments

x a vector of training data values

v1 first parameter

v2 second parameter

## Value

Matrix

logis\_ldda

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

# Usage

```
logis_ldda(x, v1, v2)
```

# **Arguments**

x a vector of training data values

v1 first parameter

v2 second parameter

## Value

Matrix

logis\_lddda

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

```
logis_lddda(x, v1, v2)
```

564 logis\_logfdd

### **Arguments**

x a vector of training data values

v1 first parameter

v2 second parameter

## Value

3d array

logis\_logf

Logf for RUST

# Description

Logf for RUST

## Usage

```
logis_logf(params, x)
```

## **Arguments**

params model parameters for calculating logf x a vector of training data values

### Value

Scalar value.

logis\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
logis_logfdd(x, v1, v2)
```

logis\_logfddd 565

## **Arguments**

x a	vector of	f training	data va	lues
-----	-----------	------------	---------	------

v1 first parameterv2 second parameter

### Value

Matrix

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
logis_logfddd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameterv2 second parameter

### Value

3d array

# Description

log-likelihood function

```
logis_loglik(vv, x)
```

logis\_mu1fa

### **Arguments**

vv parameters

x a vector of training data values

## Value

Scalar

logis\_logscores Log scores for MLE and RHP predictions calculated using leave-oneout

# Description

Log scores for MLE and RHP predictions calculated using leave-one-out

## Usage

```
logis_logscores(logscores, x)
```

# **Arguments**

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data values

### Value

Two scalars

logis\_mu1fa Minus the first derivative of the cdf, at alpha

## **Description**

Minus the first derivative of the cdf, at alpha

# Usage

```
logis_mu1fa(alpha, v1, v2)
```

### **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter logis\_mu2fa 567

# Value

Vector

logis\_mu2fa

Minus the second derivative of the cdf, at alpha

## **Description**

Minus the second derivative of the cdf, at alpha

## Usage

```
logis_mu2fa(alpha, v1, v2)
```

# Arguments

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

### Value

Matrix

logis\_p1fa

The first derivative of the cdf

# Description

The first derivative of the cdf

# Usage

```
logis_p1fa(x, v1, v2)
```

# **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

# Value

Vector

logis\_p1\_cp

Logistic Distribution with a Predictor, Predictions Based on a Calibrating Prior

### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics
- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qlogis_p1_cp(
    x,
    t,
    t0 = NA,
    n0 = NA,
    p = seq(0.1, 0.9, 0.1),
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    rust = FALSE,
    nrust = 1e+05,
    predictordata = TRUE,
    centering = TRUE,
    debug = FALSE
```

```
logis_p1_cp
                                                                                  569
   rlogis_p1_cp(
     n,
     Х,
     t,
     t0 = NA,
     n0 = NA,
     rust = FALSE,
     mlcp = TRUE,
     debug = FALSE
   )
   dlogis_p1_cp(
     Х,
     t,
     t0 = NA,
     n0 = NA,
     y = x,
     rust = FALSE,
     nrust = 1000,
     centering = TRUE,
     debug = FALSE
   )
   plogis_p1_cp(
     х,
     t,
     t0 = NA,
     n0 = NA,
     y = x,
     rust = FALSE,
     nrust = 1000,
     centering = TRUE,
     debug = FALSE
   )
```

# Arguments

 $tlogis_p1_cp(n, x, t, debug = FALSE)$ 

X	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
р	a vector of probabilities at which to generate predictive quantiles
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)

waicscores logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime) logical that indicates whether to run additional calculations and return leavelogscores one-out estimates of the log-score (much longer runtime, non-EVT models only) logical that indicates whether DMGS calculations should be run or not (longer dmgs run time) logical that indicates whether RUST-based posterior sampling calculations should rust be run or not (longer run time) the number of posterior samples used in the RUST calculations nrust logical that indicates whether predictordata should be calculated predictordata centering logical that indicates whether the predictor should be centered logical for turning on debug messages debug the number of random samples required logical that indicates whether maxlik and parameter uncertainty calculations mlcp should be performed (turn off to speed up RUST) a vector of values at which to calculate the density and distribution functions У

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The logistic distribution with a predictor has distribution function

$$f(x; a, b, \sigma) = \frac{1}{1 + e^{-(x - \mu(a, b))/\sigma}}$$

where x is the random variable,  $\mu = a + bt$  is the location paramter, and  $\sigma > 0$  is the scale parameter. The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

## **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

• ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)

• cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),

- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

logis\_p1\_f1fa 575

## **Examples**

```
#
# example 1
x=fitdistcp::d062logis_p1_example_data_v1_x
tt=fitdistcp::d062logis_p1_example_data_v1_t
p=c(1:9)/10
n0=10
q=qlogis_p1_cp(x,tt,n0=n0,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qlogis_p1_cp)",
main="Logistic w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

logis\_p1\_f1fa

The first derivative of the density for DMGS

## **Description**

The first derivative of the density for DMGS

## Usage

```
logis_p1_f1fa(x, t0, v1, v2, v3)
```

## Arguments

X	a vector of training data values
t0	a single value of the predictor (specify either $t0$ or $n0$ but not both)
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Vector

576 logis\_p1\_f2fa

logis\_p1\_f1fw

The first derivative of the density for WAIC

## **Description**

The first derivative of the density for WAIC

## Usage

```
logis_p1_f1fw(x, t, v1, v2, v3)
```

# Arguments

x a vector of training data values
 t a vector or matrix of predictors
 v1 first parameter
 v2 second parameter
 v3 third parameter

## Value

Vector

logis\_p1\_f2fa

The second derivative of the density for DMGS

# Description

The second derivative of the density for DMGS

## Usage

```
logis_p1_f2fa(x, t0, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
v2	second parameter
v3	third parameter

## Value

Matrix

logis\_p1\_f2fw 577

logis	n1	f2fw
TORIZ	υı	$1 \le 1 \le W$

The second derivative of the density for WAIC

### Description

The second derivative of the density for WAIC

### Usage

```
logis_p1_f2fw(x, t, v1, v2, v3)
```

### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictor
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Matrix

logis_p1_fd	First derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
logis_p1_fd(x, t, v1, v2, v3)
```

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

578 logis\_p1\_ldda

### Value

Vector

logis_p1_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

### Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
logis_p1_fdd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Matrix

logis_p1_ldda	The second derivative of the normalized log-likelihood
---------------	--------------------------------------------------------

# Description

The second derivative of the normalized log-likelihood

# Usage

```
logis_p1_ldda(x, t, v1, v2, v3)
```

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

logis\_p1\_lddda 579

#### Value

Matrix

logis\_p1\_lddda

The third derivative of the normalized log-likelihood

### Description

The third derivative of the normalized log-likelihood

#### Usage

```
logis_p1_lddda(x, t, v1, v2, v3)
```

#### **Arguments**

x a vector of training data valuest a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

#### Value

3d array

logis\_p1\_logf

Logf for RUST

#### **Description**

Logf for RUST

# Usage

```
logis_p1_logf(params, x, t)
```

### **Arguments**

params model parameters for calculating logf
x a vector of training data values
t a vector or matrix of predictors

#### Value

Scalar value.

logis\_p1\_logfddd

logis_p1_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
logis_p1_logfdd(x, t, v1, v2, v3)
```

### Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Matrix

logis_p1_logfddd Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
----------------------------------------------------------------------------------------------------------------------------------

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
logis_p1_logfddd(x, t, v1, v2, v3)
```

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

logis\_p1\_loglik 581

#### Value

3d array

logis\_p1\_loglik

Logistic-with-p1 observed log-likelihood function

#### **Description**

Logistic-with-p1 observed log-likelihood function

### Usage

```
logis_p1_loglik(vv, x, t)
```

#### **Arguments**

vv parameters

x a vector of training data valuest a vector or matrix of predictors

#### Value

Scalar

logis\_p1\_logscores

Log scores for MLE and RHP predictions calculated using leave-one-

#### **Description**

Log scores for MLE and RHP predictions calculated using leave-one-out

#### Usage

```
logis_p1_logscores(logscores, x, t)
```

#### Arguments

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data valuest a vector or matrix of predictors

#### Value

Two scalars

logis\_p1\_mu1fa

logis\_p1\_means

Logistic distribution: RHP mean

#### **Description**

Logistic distribution: RHP mean

#### Usage

```
logis_p1_means(t0, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 2)
```

#### **Arguments**

t0 a single value of the predictor (specify either t0 or n0 but not both)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

#### Value

Two scalars

logis\_p1\_mu1fa

Minus the first derivative of the cdf, at alpha

# Description

Minus the first derivative of the cdf, at alpha

#### Usage

```
logis_p1_mu1fa(alpha, t0, v1, v2, v3)
```

### Arguments

alpha	a vector o	f va	lues of	alpl	ha (on	e minus	proba	ıbil	ity	)
-------	------------	------	---------	------	--------	---------	-------	------	-----	---

to a single value of the predictor (specify either to or no but not both)

v1 first parameterv2 second parameterv3 third parameter

logis\_p1\_mu2fa 583

#### Value

Vector

logis\_p1\_mu2fa

Minus the second derivative of the cdf, at alpha

#### Description

Minus the second derivative of the cdf, at alpha

#### Usage

```
logis_p1_mu2fa(alpha, t0, v1, v2, v3)
```

#### **Arguments**

alpha a vector of values of alpha (one minus probability)

t0 a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

logis\_p1\_p1fa

The first derivative of the cdf

#### **Description**

The first derivative of the cdf

# Usage

```
logis_p1_p1fa(x, t0, v1, v2, v3)
```

### Arguments

X	a vector of training	data values
^	a rector or training	autu turuos

t0 a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameterv2 second parameterv3 third parameter

logis\_p1\_pd

#### Value

Vector

logis\_p1\_p2fa

The second derivative of the cdf

#### **Description**

The second derivative of the cdf

#### Usage

```
logis_p1_p2fa(x, t0, v1, v2, v3)
```

#### **Arguments**

x a vector of training d	data values
--------------------------	-------------

t0 a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameter v2 second parameter v3 third parameter

#### Value

Matrix

 $logis_p1_pd$ 

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### **Description**

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
logis_p1_pd(x, t, v1, v2, v3)
```

### Arguments

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

logis\_p1\_pdd 585

### Value

Vector

logis_p1_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
logis_p1_pdd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Matrix

```
logis_p1_predictordata
```

Predicted Parameter and Generalized Residuals

### Description

Predicted Parameter and Generalized Residuals

# Usage

```
logis_p1_predictordata(predictordata, x, t, t0, params)
```

logis\_p1\_waic

#### **Arguments**

predictordata logical that indicates whether to calculate and return predictordata

x a vector of training data valuest a vector or matrix of predictors

t0 a single value of the predictor (specify either t0 or n0 but not both)

params model parameters for calculating logf

#### Value

Two vectors

logis\_p1\_waic Waic

#### **Description**

Waic

#### Usage

```
logis_p1_waic(waicscores, x, t, v1hat, v2hat, v3hat, lddi, lddd, lambdad)
```

#### **Arguments**

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data valuest a vector or matrix of predictors

v1hat first parameter v2hat second parameter v3hat third parameter

lddi inverse observed information matrixlddd third derivative of log-likelihood

lambdad derivative of the log prior

#### Value

Two numeric values.

logis\_p2fa 587

- ·	~ ^	
Log1	s_p2fa	

The second derivative of the cdf

### Description

The second derivative of the cdf

#### Usage

```
logis_p2fa(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Vector

logis\_pd

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
logis_pd(x, v1, v2)
```

### Arguments

x a vector of training data values

v1 first parameter

v2 second parameter

### Value

Vector

588 logis\_waic

logis_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
logis_pdd(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Matrix

logis_waic	Waic			
------------	------	--	--	--

# Description

Waic

#### Usage

```
logis_waic(waicscores, x, v1hat, v2hat, lddi, lddd, lambdad)
```

derivative of the log prior

### **Arguments**

lambdad

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
X	a vector of training data values
v1hat	first parameter
v2hat	second parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood

#### Value

Two numeric values.

1st\_k3\_cp

t Distribution Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qlst_k3_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    kdf = 5,
    d1 = 0.01,
    fd2 = 0.01,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    rust = FALSE,
    nrust = 1e+05,
    debug = FALSE,
```

```
aderivs = TRUE
rlst_k3_cp(
 n,
 х,
 d1 = 0.01,
 fd2 = 0.01,
 kdf = 5,
 rust = FALSE,
 mlcp = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
dlst_k3_cp(
 Х,
 y = x,
 d1 = 0.01,
 fd2 = 0.01,
 kdf = 5,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
plst_k3_cp(
 Х,
 y = x,
 d1 = 0.01,
 fd2 = 0.01,
 kdf = 5,
 rust = FALSE,
 nrust = 1000,
 debug = FALSE,
 aderivs = TRUE
)
tlst_k3_{cp}(n, x, d1 = 0.01, fd2 = 0.01, kdf = 5, debug = FALSE)
```

x	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
kdf	the known degrees of freedom parameter
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter

fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.

• cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The t distribution (also known as the location-scale t distribution, hence the name 1st), has probability density function

$$f(x;\mu,\sigma) = \frac{\Gamma((\nu+1)/2)}{\sqrt{\pi\nu}\sigma\Gamma(\nu/2)} \left(1 + \frac{(x-\mu)^2}{\sigma^2\nu}\right)^{(\nu+1)/2}$$

where x is the random variable,  $\mu, \sigma > 0$  are the parameters, and we consider the degrees of freedom  $\nu$  to be known (hence the k3 in the name).

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

• waic1: the WAIC1 score for the calibrating prior model.

• waic2: the WAIC2 score for the calibrating prior model.

#### If logscores=TRUE:

• ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)

• cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

#### If rust=TRUE:

 ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),

- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

596 lst\_k3\_f1f

#### **Examples**

```
#
# example 1
x=fitdistcp::d041lst_k3_example_data_v1
p=c(1:9)/10
q=qlst_k3_cp(x,p,kdf=5,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qlst_k3_cp)",
main="t: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

lst\_k3\_f1f

DMGS equation 3.3, f1 term

### Description

DMGS equation 3.3, f1 term

#### Usage

```
lst_k3_f1f(y, v1, d1, v2, fd2, kdf)
```

### **Arguments**

у	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

#### Value

Matrix

lst\_k3\_f1fa 597

lst	k3	f1	fa

The first derivative of the density

# Description

The first derivative of the density

### Usage

```
lst_k3_f1fa(x, v1, v2, kdf)
```

### Arguments

Χ	a vector of training data values
---	----------------------------------

v1 first parameter v2 second parameter

kdf the known degrees of freedom parameter

#### Value

Vector

lst\_k3\_f2f

DMGS equation 3.3, f2 term

# Description

DMGS equation 3.3, f2 term

# Usage

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter
kdf	the known degrees of freedom parameter

#### Value

3d array

598 lst\_k3\_fd

1st	k3	f2fa

The second derivative of the density

#### **Description**

The second derivative of the density

#### Usage

```
lst_k3_f2fa(x, v1, v2, kdf)
```

### Arguments

x a vector of training data val	ues
---------------------------------	-----

v1 first parameter v2 second parameter

kdf the known degrees of freedom parameter

#### Value

Matrix

lst_k3_fd	
-----------	--

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### **Description**

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

# Arguments

a vector of training data varies	X	a vector of training data values	
----------------------------------	---	----------------------------------	--

v1 first parameterv2 second parameterv3 third parameter

#### Value

Vector

lst\_k3\_fdd 599

lst_k3_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	Title of Thinker Changer and Set Soller Soller

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
lst_k3_fdd(x, v1, v2, v3)
```

# Arguments

x a vector of training data value	ues
-----------------------------------	-----

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

lst_k3_ldd	Second derivative matrix oj	f the normalized log-likelihood
------------	-----------------------------	---------------------------------

### Description

Second derivative matrix of the normalized log-likelihood

#### Usage

```
lst_k3_ldd(x, v1, d1, v2, fd2, kdf)
```

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

lst\_k3\_lddd

#### Value

Square scalar matrix

lst\_k3\_ldda

The second derivative of the normalized log-likelihood

### Description

The second derivative of the normalized log-likelihood

#### Usage

```
lst_k3_1dda(x, v1, v2, kdf)
```

### Arguments

Χ	a vector of training data values
---	----------------------------------

v1 first parameter v2 second parameter

kdf the known degrees of freedom parameter

#### Value

Matrix

lst\_k3\_lddd

Third derivative tensor of the normalized log-likelihood

#### **Description**

Third derivative tensor of the normalized log-likelihood

#### Usage

```
lst_k3_lddd(x, v1, d1, v2, fd2, kdf)
```

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

lst\_k3\_lddda 601

### Value

Cubic scalar array

lst\_k3\_lddda

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

#### Usage

```
lst_k3_lddda(x, v1, v2, kdf)
```

### Arguments

x a vector of training data values

v1 first parameter

v2 second parameter

kdf the known degrees of freedom parameter

#### Value

3d array

lst_k3_lmn	One component of the second derivative of the normalized log-
	likelihood

### Description

One component of the second derivative of the normalized log-likelihood

### Usage

```
lst_k3_lmn(x, v1, d1, v2, fd2, kdf, mm, nn)
```

602 lst\_k3\_lmnp

# Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

### Value

Scalar value

lst_k3_lmnp	One component of the third derivative of the normalized log-likelihood

# Description

One component of the third derivative of the normalized log-likelihood

# Usage

```
lst_k3_lmnp(x, v1, d1, v2, fd2, kdf, mm, nn, rr)
```

# Arguments

X	a vector of training data values
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

### Value

Scalar value

lst\_k3\_logf 603

 $lst_k3_logf$ 

Logf for RUST

#### **Description**

Logf for RUST

#### Usage

```
lst_k3_logf(params, x, kdf)
```

#### **Arguments**

params model parameters for calculating logf x a vector of training data values

kdf the known degrees of freedom parameter

#### Value

Scalar value.

1st\_k3\_logfdd Second derivative of the log density Created by Stephen Jewson using

Deriv() by Andrew Clausen and Serguei Sokol

### Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
lst_k3_logfdd(x, v1, v2, v3)
```

#### **Arguments**

x a vector of training data values

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

lst\_k3\_loglik

lst_k3_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
lst_k3_logfddd(x, v1, v2, v3)
```

### Arguments

x a vector of train	ning data values
---------------------	------------------

v1 first parameterv2 second parameterv3 third parameter

#### Value

3d array

lst\_k3\_loglik

log-likelihood function

### Description

log-likelihood function

# Usage

```
lst_k3_loglik(vv, x, kdf)
```

### **Arguments**

VV	parameters
v v	parameters

x a vector of training data values

kdf the known degrees of freedom parameter

### Value

Scalar

lst\_k3\_logscores 605

lst_k3_logscores Log scores for MLE and RHP predictions calculated using leave-one- out	
--------------------------------------------------------------------------------------------	--

# Description

Log scores for MLE and RHP predictions calculated using leave-one-out

# Usage

```
lst_k3_logscores(logscores, x, d1 = 0.01, fd2 = 0.01, kdf, aderivs = TRUE)
```

### Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
X	a vector of training data values
d1	the delta used in the numerical derivatives with respect to the parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two scalars

lst_k3_mu1f	DMGS equation 3.3, mu1 term	

# Description

DMGS equation 3.3, mu1 term

# Usage

```
lst_k3_mu1f(alpha, v1, d1, v2, fd2, kdf)
```

lst\_k3\_mu2f

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

# Value

Matrix

lst_k3_mu2f	DMGS equation 3.3, mu2 term	

# Description

DMGS equation 3.3, mu2 term

# Usage

```
lst_k3_mu2f(alpha, v1, d1, v2, fd2, kdf)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

### Value

3d array

lst\_k3\_p1f 607

lst_k3_p1f	DMGS equation 3.3, p1 term

# Description

DMGS equation 3.3, p1 term

# Usage

```
lst_k3_p1f(y, v1, d1, v2, fd2, kdf)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

### Value

Matrix

	lst_k3_p2f	DMGS equation 3.3, p2 term	
--	------------	----------------------------	--

# Description

DMGS equation 3.3, p2 term

# Usage

```
lst_k3_p2f(y, v1, d1, v2, fd2, kdf)
```

У	a vector of values at which to calculate the density and distribution functions
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

lst\_k3\_waic

### Value

3d array

lst\_k3\_waic Waic

# Description

Waic

# Usage

```
lst_k3_waic(
  waicscores,
  x,
  v1hat,
  d1,
  v2hat,
  fd2,
  kdf,
  lddi,
  lddd,
  lambdad,
  aderivs
)
```

### Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores
	(longer runtime)
X	a vector of training data values
v1hat	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2hat	second parameter
fd2	the fractional delta used in the numerical derivatives with respect to the param-
	eter
kdf	the known degrees of freedom parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

Two numeric values.

t Distribution with a Predictor, Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qlst_p1k3_cp(
    x,
    t,
    t0 = NA,
    n0 = NA,
    p = seq(0.1, 0.9, 0.1),
    d1 = 0.01,
    d2 = 0.01,
    fd3 = 0.01,
    kdf = 10,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    rust = FALSE,
```

```
nrust = 1e+05,
  predictordata = TRUE,
  centering = TRUE,
  debug = FALSE,
  aderivs = TRUE
)
rlst_p1k3_cp(
 n,
 х,
  t,
  t0 = NA,
 n0 = NA,
 d1 = 0.01,
 d2 = 0.01,
  fd3 = 0.01,
  kdf = 10,
  rust = FALSE,
 mlcp = TRUE,
  centering = TRUE,
 debug = FALSE,
  aderivs = TRUE
)
dlst_p1k3_cp(
 Х,
 t,
 t0 = NA,
 n0 = NA,
 y = x,
  d1 = 0.01,
  d2 = 0.01,
  fd3 = 0.01,
  kdf = 10,
  rust = FALSE,
 nrust = 1000,
  centering = TRUE,
  debug = FALSE,
  aderivs = TRUE
)
plst_p1k3_cp(
 Х,
  t,
 t0 = NA,
 n0 = NA,
 y = x,
 d1 = 0.01,
```

```
d2 = 0.01,
 fd3 = 0.01,
 kdf = 10,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
 debug = FALSE,
 aderivs = TRUE
)
tlst_p1k3_cp(
 n,
 Х,
 t,
 d1 = 0.01,
 d2 = 0.01,
 fd3 = 0.01,
 kdf = 10,
 debug = FALSE
```

	and the second s
Х	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
р	a vector of probabilities at which to generate predictive quantiles
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter
d2	if aderivs=FALSE, the delta used for numerical derivatives with respect to the second parameter
fd3	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the third parameter
kdf	the known degrees of freedom parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)

nrust	the number of posterior samples used in the RUST calculations
predictordata	logical that indicates whether predictordata should be calculated
centering	logical that indicates whether the predictor should be centered
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The t distribution with a predictor (also known as the location-scale t distribution with a predictor, hence the name lst), has probability density function

$$f(x;a,b,\sigma) = \frac{\Gamma((\nu+1)/2)}{\sqrt{\pi\nu}\sigma\Gamma(\nu/2)} \left(1 + \frac{(x-\mu(a,b))^2}{\sigma^2\nu}\right)^{(\nu+1)/2}$$

where x is the random variable,  $\mu = a + bt$  is the location parameter, and  $\sigma > 0$  is the scale parameter. We consider the degrees of freedom  $\nu$  to be known (hence the k3 in the name).

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),

- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
# example 1
x=fitdistcp::d063lst_p1k3_example_data_v1_x
tt=fitdistcp::d063lst_p1k3_example_data_v1_t
p=c(1:9)/10
n0=10
q=qlst_p1k3_cp(x,tt,n0=n0,p=p,kdf=5,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qlst_p1k3_cp)",
main="t w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

lst\_p1k3\_f1f 617

_				_
lst_	n1	kЗ	f1	f
10 L_	$\mathbf{p}$	NJ_		

DMGS equation 2.1, f1 term

# Description

DMGS equation 2.1, f1 term

## Usage

```
lst_p1k3_f1f(y, t0, v1, d1, v2, d2, v3, fd3, kdf)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

#### Value

Matrix

lst\_p1k3\_f1fa

The first derivative of the density for DMGS

# Description

The first derivative of the density for DMGS

```
lst_p1k3_f1fa(x, t0, v1, v2, v3, kdf)
```

# Arguments

X	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
kdf	the known degrees of freedom parameter

## Value

Vector

# Description

The first derivative of the density for WAIC

## Usage

```
lst_p1k3_f1fw(x, t, v1, v2, v3, kdf)
```

# Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
kdf	the known degrees of freedom parameter

#### Value

Vector

lst\_p1k3\_f2f 619

lst_	n1	k٦	f2f
13 L_	.vı	ヘン_	_   _

DMGS equation 2.1, f2 term

## Description

DMGS equation 2.1, f2 term

## Usage

```
lst_p1k3_f2f(y, t0, v1, d1, v2, d2, v3, fd3, kdf)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

#### Value

3d array

lst\_p1k3\_f2fa

The second derivative of the density for DMGS

# Description

The second derivative of the density for DMGS

```
lst_p1k3_f2fa(x, t0, v1, v2, v3, kdf)
```

620 lst\_p1k3\_f2fw

## Arguments

x	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
v2	second parameter
v3	third parameter

kdf the known degrees of freedom parameter

## Value

Matrix

# Description

The second derivative of the density for WAIC

## Usage

```
lst_p1k3_f2fw(x, t, v1, v2, v3, kdf)
```

# Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
kdf	the known degrees of freedom parameter

#### Value

Matrix

lst\_p1k3\_fd 621

lst_p1k3_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
lst_p1k3_fd(x, t, v1, v2, v3, v4)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Vector

lst_p1k3_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
lst_p1k3_fdd(x, t, v1, v2, v3, v4)
```

lst\_p1k3\_ldd

# Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Matrix

lst\_p1k3\_ldd Second derivative matrix of the normalized log-likelihood

# Description

Second derivative matrix of the normalized log-likelihood

# Usage

```
lst_p1k3_ldd(x, t, v1, d1, v2, d2, v3, fd3, kdf)
```

# Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

#### Value

Square scalar matrix

lst\_p1k3\_ldda 623

_		
16+	ก1レ2	ldda
TOL	DIKO	Tuua

The second derivative of the normalized log-likelihood

## Description

The second derivative of the normalized log-likelihood

## Usage

```
lst_p1k3_ldda(x, t, v1, v2, v3, kdf)
```

## Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
kdf	the known degrees of freedom parameter

#### Value

Matrix

lst_	n1	k3	1	hbb
13 t_	יש	·\J_		uuu

Third derivative tensor of the normalized log-likelihood

# Description

Third derivative tensor of the normalized log-likelihood

### Usage

```
lst_p1k3_lddd(x, t, v1, d1, v2, d2, v3, fd3, kdf)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter

lst\_p1k3\_lddda

v3	third parameter
V 3	unita parametei

the fractional delta used in the numerical derivatives with respect to the param-

eter

kdf the known degrees of freedom parameter

#### Value

Cubic scalar array

lst\_p1k3\_lddda

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

## Usage

```
lst_p1k3_lddda(x, t, v1, v2, v3, kdf)
```

# Arguments

X	a vector of training data values

t a vector or matrix of predictors

v1 first parameter

v2 second parameter

v3 third parameter

kdf the known degrees of freedom parameter

## Value

3d array

lst\_p1k3\_lmn 625

likelihood	lst_p1k3_lmn	One component of the second derivative of the normalized log-likelihood
------------	--------------	-------------------------------------------------------------------------

# Description

One component of the second derivative of the normalized log-likelihood

## Usage

```
lst_p1k3_lmn(x, t, v1, d1, v2, d2, v3, fd3, kdf, mm, nn)
```

## Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate

## Value

Scalar value

lst_p1k3_lmnp	One component of the second derivative of the normalized log-likelihood

# Description

One component of the second derivative of the normalized log-likelihood

```
lst_p1k3_lmnp(x, t, v1, d1, v2, d2, v3, fd3, kdf, mm, nn, rr)
```

lst\_p1k3\_logf

## Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter
mm	an index for which derivative to calculate
nn	an index for which derivative to calculate
rr	an index for which derivative to calculate

## Value

Scalar value

lst_p1k3_logf	Logf for RUST	
---------------	---------------	--

# Description

Logf for RUST

## Usage

```
lst_p1k3_logf(params, x, t, kdf)
```

# Arguments

params	model parameters for calculating logf
X	a vector of training data values
t	a vector or matrix of predictors
kdf	the known degrees of freedom parameter

#### Value

Scalar value.

lst\_p1k3\_logfdd 627

lst_p1k3_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
lst_p1k3_logfdd(x, t, v1, v2, v3, v4)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Matrix

lst_p1k3_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
lst_p1k3_logfddd(x, t, v1, v2, v3, v4)
```

lst\_p1k3\_loglik

# Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

## Value

3d array

lst\_p1k3\_loglik

LST-with-p1 observed log-likelihood function

# Description

LST-with-p1 observed log-likelihood function

# Usage

```
lst_p1k3_loglik(vv, x, t, kdf)
```

# Arguments

VV	parameters
x	a vector of training data values
t	a vector or matrix of predictors
kdf	the known degrees of freedom parameter

#### Value

Scalar

lst\_p1k3\_logscores 629

lst_p1k3_logscores	Log scores for MLE and RHP predictions calculated using leave-one-out
--------------------	-----------------------------------------------------------------------

# Description

Log scores for MLE and RHP predictions calculated using leave-one-out

# Usage

```
lst_p1k3_logscores(logscores, x, t, d1, d2, fd3, kdf, aderivs)
```

## Arguments

logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)
x	a vector of training data values
t	a vector or matrix of predictors
d1	the delta used in the numerical derivatives with respect to the parameter
d2	the delta used in the numerical derivatives with respect to the parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter
aderivs	logical for whether to use analytic derivatives (instead of numerical)

### Value

Two scalars

lst_p1k3_mu1f	DMGS equation 3.3, mu1 term

# Description

DMGS equation 3.3, mu1 term

```
lst_p1k3_mu1f(alpha, t0, v1, d1, v2, d2, v3, fd3, kdf)
```

lst\_p1k3\_mu2f

## Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the param-
	eter
kdf	the known degrees of freedom parameter

# Value

Matrix

DMGS equation 3.3, mu2 term
-----------------------------

# Description

DMGS equation 3.3, mu2 term

# Usage

```
lst_p1k3_mu2f(alpha, t0, v1, d1, v2, d2, v3, fd3, kdf)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the param-
	eter
kdf	the known degrees of freedom parameter

## Value

3d array

lst\_p1k3\_p1f 631

lst\_p1k3\_p1f

DMGS equation 2.1, p1 term

# Description

DMGS equation 2.1, p1 term

# Usage

```
lst_p1k3_p1f(y, t0, v1, d1, v2, d2, v3, fd3, kdf)
```

# Arguments

У	value of random variable
t0	value of predictor
v1	first parameter
d1	delta for numerical derivative
v2	second parameter
d2	delta for numerical derivative
v3	third parameter
fd3	fractional delta for numerical derivative
kdf	the known number of degrees of freedom

## Value

Matrix

 $lst_p1k3_p2f$ 

DMGS equation 2.1, p2 term

## Description

DMGS equation 2.1, p2 term

```
lst_p1k3_p2f(y, t0, v1, d1, v2, d2, v3, fd3, kdf)
```

## Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
v1	first parameter
d1	the delta used in the numerical derivatives with respect to the parameter
v2	second parameter
d2	the delta used in the numerical derivatives with respect to the parameter
v3	third parameter
fd3	the fractional delta used in the numerical derivatives with respect to the parameter
kdf	the known degrees of freedom parameter

## Value

3d array

lst\_p1k3\_predictordata

Predicted Parameter and Generalized Residuals

# Description

Predicted Parameter and Generalized Residuals

# Usage

```
lst_p1k3_predictordata(predictordata, x, t, t0, params, kdf)
```

# **Arguments**

predictordata	logical that indicates whether to calculate and return predictordata
X	a vector of training data values
t	a vector or matrix of predictors
t0	a single value of the predictor (specify either t0 or n0 but not both)
params	model parameters for calculating logf
kdf	the known degrees of freedom parameter

#### Value

Two vectors

lst\_p1k3\_setics 633

lst\_p1k3\_setics

Set initial conditions

# Description

Set initial conditions

## Usage

```
lst_p1k3_setics(x, t, ics)
```

# Arguments

x a vector of training data values
t a vector or matrix of predictors

ics initial conditions for the maximum likelihood search

#### Value

Vector

lst\_p1k3\_waic

Waic

# Description

Waic

```
lst_p1k3_waic(
 waicscores,
  х,
  t,
  v1hat,
 d1,
  v2hat,
  d2,
  v3hat,
  fd3,
  kdf,
  lddi,
  lddd,
 lambdad,
  aderivs
)
```

634 makebetat0

#### **Arguments**

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data values t a vector or matrix of predictors

v1hat first parameter

d1 the delta used in the numerical derivatives with respect to the parameter

v2hat second parameter

d2 the delta used in the numerical derivatives with respect to the parameter

v3hat third parameter

fd3 the fractional delta used in the numerical derivatives with respect to the param-

eter

kdf the known degrees of freedom parameter
lddi inverse observed information matrix
lddd third derivative of log-likelihood

lambdad derivative of the log prior

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

Two numeric values.

makebetat0 Calculate the location parameter when there are predictors (single

time point)

#### **Description**

Calculate the location parameter when there are predictors (single time point)

#### Usage

```
makebetat0(nt, params, t0)
```

#### **Arguments**

nt the number of columns in t

params model parameters for calculating logf

to a single value of the predictor (specify either to or no but not both)

#### Value

Vector

makebetatm 635

makebetatm	Calculate the location parameter when there are predictors (multiple time points)

# Description

Calculate the location parameter when there are predictors (multiple time points)

## Usage

```
makebetatm(nt, params, t)
```

### **Arguments**

nt the number of columns in t

params model parameters for calculating logf t a vector or matrix of predictors

#### Value

Vector

akemuhat0 Make muhat0	makemuhat0 Make muhat0
-----------------------	------------------------

# Description

Make muhat0

# Usage

```
makemuhat0(t0, n0, t, mle_params)
```

# Arguments

t0	the value of the predictor vector at which to make the prediction (if n0 not specified)
n0	the position in the predictor vector at which to make the prediction (positive integer less than or equal to the length of $x$ ) (if t0 not specified)
t	predictor
mle_params	MLE params

#### Value

Scalar

636 maket0

makeq

Calculates quantiles from simulations by inverting the Hazen CDF

## Description

Calculates quantiles from simulations by inverting the Hazen CDF

## Usage

```
makeq(yy, pp)
```

## Arguments

yy vector of samples
pp vector of probabilities

#### Value

Vector

maket0

Determine t0

## Description

Determine t0

## Usage

```
maket0(t0, n0, t)
```

# Arguments

a single value of the predictor (specify either t0 or n0 but not both)
 an index for the predictor (specify either t0 or n0 but not both)
 a vector or matrix of predictors

#### Value

Scalar

maketresid0 637

maketresid0	Make ta0		

## Description

Make ta0

#### Usage

```
maketresid0(t0, n0, t)
```

## Arguments

to the value of the predictor vector at which to make the prediction (if n0 not spec-

ified)

no the position in the predictor vector at which to make the prediction (positive

integer less than or equal to the length of x) (if t0 not specified)

t predictor

## Value

Scalar

Make WAIC
-----------

## Description

Make WAIC

# Usage

```
make_cwaic(x, fhatx, lddi, lddd, f1f, lambdad, f2f, dim)
```

## Arguments

dim

the training data
density of x at the maximum likelihood parameters
inverse of the second derivative log-likelihood matrix
the third derivative log-likelihood tensor
the f1 term from DMGS equation 2.1
the slope of the log prior
the f2 term from DMGS equation 2.1

number of free parameters

make\_se

#### Value

Two scalars

make\_maic

Calculate MAIC

# Description

Calculate MAIC

## Usage

```
make_maic(ml_value, nparams)
```

## Arguments

ml\_value maximum of the likelihood nparams number of parameters

#### Value

Vector of 3 values Returns the two components of MAIC, and their sum

make\_se

Make Standard Errors from lddi

# Description

Make Standard Errors from Iddi

## Usage

```
make_se(nx, lddi)
```

## Arguments

nx length of training data

1ddi the inverse log-likelihood matrix

#### Value

Vector

make\_waic 639

make_waic	Make WAIC

#### **Description**

Make WAIC

#### Usage

```
make_waic(x, fhatx, lddi, lddd, f1f, lambdad, f2f, dim)
```

### **Arguments**

X	the training data
fhatx	density of x at the maximum likelihood parameters
lddi	inverse of the second derivative log-likelihood matrix
lddd	the third derivative log-likelihood tensor
f1f	the f1 term from DMGS equation 2.1
lambdad	the slope of the log prior
f2f	the f2 term from DMGS equation 2.1
dim	number of free parameters

### Value

Two scalars

man	A blank function I use for setting up the man page information

## Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y

- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

## Usage

man( х, t, t1, t2, t3, t0, t01, t02, t03, t10, t20, n0, n01, n02, n03, n10, n20, р, n, у, ics, kloc, kscale, kshape, kdf, kbeta, d1, fd1, d2, fd2, d3, fd3, d4, fd4,

d5,

```
fd5,
  d6,
  fd6,
  fdalpha,
 minxi,
 maxxi,
 dlogpi,
 means,
 waicscores,
 logscores,
 extramodels,
 pdf,
  customprior,
 dmgs,
 mlcp,
 predictordata,
  centering,
 method,
 nonnegslopesonly,
  rnonnegslopesonly,
 prior,
 debug,
  rust,
 nrust,
 boot,
 nboot,
 pwm,
 unbiasedv,
 aderivs
)
```

# Arguments

Х	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
t1	a vector of predictors for the mean, such that $length(t1)=length(x)$
t2	a vector of predictors for the sd, such that length(t2)=length(x)
t3	a vector of predictors for the shape, such that length(t3)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
t10	a single value of the predictor for the mean (specify either $t10$ or $n10$ but not both)
t20	a single value of the predictor for the sd (specify either t20 or n20 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)

n01	an index for the predictor (specify either t01 or n01 but not both)
n02	an index for the predictor (specify either t02 or n02 but not both)
n03	an index for the predictor (specify either t03 or n03 but not both)
n10	an index for the predictor for the mean (specify either t10 or n10 but not both)
n20	an index for the predictor for the sd (specify either t20 or n20 but not both)
p	a vector of probabilities at which to generate predictive quantiles
n	the number of random samples required
У	a vector of values at which to calculate the density and distribution functions
ics	initial conditions for the maximum likelihood search
kloc	the known location parameter
kscale	the known scale parameter
kshape	the known shape parameter
kdf	the known degrees of freedom parameter
kbeta	the known beta parameter
d1	if aderivs=FALSE, the delta used for numerical derivatives with respect to the first parameter ${\sf res}$
fd1	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the first parameter $$
d2	if aderivs=FALSE, the delta used for numerical derivatives with respect to the second parameter $% \left( 1\right) =\left( 1\right) \left( 1\right)$
fd2	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the second parameter
d3	if aderivs=FALSE, the delta used for numerical derivatives with respect to the third parameter $% \left( 1\right) =\left( 1\right) \left( 1\right) $
fd3	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the third parameter $\frac{1}{2}$
d4	if aderivs=FALSE, the delta used for numerical derivatives with respect to the fourth parameter $% \left( 1\right) =\left( 1\right) \left( 1\right)$
fd4	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the fourth parameter
d5	if aderivs=FALSE, the delta used for numerical derivatives with respect to the fifth parameter ${\sf rest}$
fd5	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the fourth parameter $\frac{1}{2}$
d6	if aderivs=FALSE, the delta used for numerical derivatives with respect to the sixth parameter $% \left( 1\right) =\left( 1\right) \left( 1\right) $
fd6	if aderivs=FALSE, the fractional delta used for numerical derivatives with respect to the fourth parameter
fdalpha	if pdf=TRUE, the fractional delta used for numerical derivatives with respect to probability, for calculating the pdf as a function of quantiles

minxi the minimum allowed value of the shape parameter (decrease with caution) the maximum allowed value of the shape parameter (increase with caution) maxxi

dlogpi gradient of the log prior

means logical that indicates whether to run additional calculations and return analytical

estimates for the distribution means (longer runtime)

waicscores logical that indicates whether to run additional calculations and return estimates

for the WAIC1 and WAIC2 scores (longer runtime)

logscores logical that indicates whether to run additional calculations and return leave-

one-out estimates of the log-score (much longer runtime, non-EVT models only)

extramodels logical that indicates whether to run additional calculations and add three addi-

tional prediction models (longer runtime)

pdf logical that indicates whether to run additional calculations and return density

functions evaluated at quantiles specified by the input probabilities (longer run-

time)

a custom value for the slope of the log prior at the maxlik estimate customprior

logical that indicates whether DMGS calculations should be run or not (longer dmgs

run time)

mlcp logical that indicates whether maxlik and parameter uncertainty calculations

should be performed (turn off to speed up RUST)

logical that indicates whether predictordata should be calculated predictordata centering logical that indicates whether the predictor should be centered

method character string that indicates whether to use rust method=rust or bootstrap

method=boot

nonnegslopesonly

logical that indicates whether to disallow non-negative slopes

rnonnegslopesonly

logical that indicates whether to disallow non-negative slopes

logical indicating which prior to use prior logical for turning on debug messages debug

logical that indicates whether RUST-based posterior sampling calculations should rust

be run or not (longer run time)

the number of posterior samples used in the RUST calculations nrust

logical that indicates whether bootstrap-based posterior sampling calculations boot

should be run or not (longer run time)

nboot the number of posterior samples used in the bootstrap calculations

pwm logical for whether to include PWM results (longer runtime) unbiasedv

logical for whether to include unbiased variance results in norm

aderivs (for code testing only) logical for whether to use analytic derivatives (instead of

numerical). By default almost all models now use analytical derivatives.

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Optional Return Values (EVT models only)**

q\*\*\*\* optionally returns the following, for EVT models only:

• cp\_pdf: the density function at quantiles corresponding to input probabilities p. We provide this for EVD models, because direct estimation of the density function using the DMGS density equation is not possible.

#### Optional Return Values (some EVT models only)

q\*\*\*\* optionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_quantiles: predictive quantiles calculated from Bayesian integration with a flat prior.
- rh\_ml\_quantiles: predictive quantiles calculated from Bayesian integration with the calibrating prior, and the maximmum likelihood estimate for the shape parameter.
- jp\_quantiles: predictive quantiles calculated from Bayesian integration with Jeffreys' prior.

r\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_deviates: predictive random deviates calculated using a Bayesian analysis with a flat prior.
- rh\_ml\_deviates: predictive random deviates calculated using a Bayesian analysis with the RHP-MLE prior.
- jp\_deviates: predictive random deviates calculated using a Bayesian analysis with the JP.

d\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_pdf: predictive density function from a Bayesian analysis with the flat prior.
- rh\_ml\_pdf: predictive density function from a Bayesian analysis with the RHP-MLE prior.
- jp\_pdf: predictive density function from a Bayesian analysis with the JP.

p\*\*\*\* additionally returns the following, for some EVT models only:

If extramodels=TRUE:

- flat\_cdf: predictive distribution function from a Bayesian analysis with the flat prior.
- rh\_ml\_cdf: predictive distribution function from a Bayesian analysis with the RHP-MLE prior.
- jp\_cdf: predictive distribution function from a Bayesian analysis with the JP.

These additional predictive distributions are included for comparison with the calibrating prior model. They generally give less good reliability than the calibrating prior.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (non-homogeneous models)**

This model is not homogeneous, i.e. it does not have a transitive transformation group, and so there is no right Haar prior and no method for generating exactly reliable predictions. The cp outputs are generated using a prior that has been shown in tests to give reasonable reliability. See Jewson et al. (2025a) for discussion of the prior and test results. For non-homogeneous models, reliability is generally poor for small sample sizes (<20), but is still much better than maximum likelihood. For small sample sizes, it is advisable to check the level of reliability using the routine reltest.

#### **Details (analytic integration)**

For this model, the Bayesian prediction equation is integrated analytically.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),

man1f 649

• t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),

- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

man1f

Return message for flf, plf, mulf

### **Description**

Return message for f1f, p1f, mu1f

### Usage

man1f()

#### Value

Matrix

650 mancheckmle

man2f

Return message for f2f, p2f, mu2f

## Description

Return message for f2f, p2f, mu2f

## Usage

man2f()

### Value

3d array

manboot

Return message for boot

# Description

Return message for boot

## Usage

manboot()

## Value

A list containing a matrix of simulated parameter values

mancheckmle

Return message for checkmle

## Description

Return message for checkmle

## Usage

mancheckmle()

#### Value

No return value (just a message to the screen).

mandsub 651

mandsub

Return message for dsub

## Description

Return message for dsub

## Usage

```
mandsub()
```

### Value

A vector of parameter estimates, two pdf vectors, two cdf vectors

manf

Blank function I use for setting up the man page information for the functions

# Description

Blank function I use for setting up the man page information for the functions

## Usage

```
manf(
  dim,
  νv,
  ml_params,
  nx,
  nxx,
  Х,
  ХХ,
  t,
  nt,
  ta,
  tb,
  tc,
  t1,
  t2,
  t3,
  tt,
  tt1,
  tt2,
```

tt3,

```
tt2d,
tt3d,
t0,
t0a,
t0b,
t0c,
t01,
t02,
t03,
t10,
t20,
t30,
n0,
n10,
n20,
p,
n,
у,
ics,
tresid,
tresid0,
muhat0,
vhat,
v1,
v1hat,
v1h,
d1,
fd1,
v2,
v2hat,
v2h,
d2,
fd2,
v3,
v3hat,
v3h,
d3,
fd3,
v4,
v4hat,
v4h,
d4,
fd4,
v5,
v5hat,
v5h,
d5,
```

v6,

```
v6hat,
v6h,
d6,
minxi,
maxxi,
ximin,
ximax,
fdalpha,
kscale,
kloc,
kshape,
kdf,
kbeta,
alpha,
ymn,
slope,
mu,
sigma,
sigma1,
sigma2,
scale,
shape,
хi,
xi1,
xi2,
lambda,
log,
mm,
nn,
rr,
lddi,
lddi_k2,
lddi_k3,
lddi_k4,
lddd,
lddd_k2,
1ddd_k3,
lddd_k4,
lambdad,
lambdad_cp,
lambdad_rhp,
lambdad_flat,
lambdad_rh_mle,
lambdad_rh_flat,
lambdad_jp,
lambdad_custom,
means,
waicscores,
```

```
logscores,
  extramodels,
 pdf,
 predictordata,
 nonnegslopesonly,
 rnonnegslopesonly,
 customprior,
 prior,
 params,
 уу,
 pp,
 dlogpi,
 debug,
 centering,
 aderivs
)
```

## Arguments

dim	number of parameters
vv	parameters
ml_params	parameters
nx	length of training data
nxx	length of training data
x	a vector of training data values
xx	a vector of training data values
t	a vector or matrix of predictors
nt	the number of columns in t
ta	a vector of predictors for the mean (first column)
tb	a vector of predictors for the mean (second column)
tc	a vector of predictors for the mean (third column)
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
tt	a vector of predictors
tt1	a vector of predictors for the mean
tt2	a vector of predictors for the sd
tt3	a vector of predictors for the shape
tt2d	a matrix of predictors (nx by 2)
tt3d	a matrix of predictors (nx by 3)
t0	a single value of the predictor (specify either t0 or n0 but not both)

t0a	a single value of the predictor, for the first column of the predictor (specify either $t0a$ or $t0a$ but not both)
t0b	a single value of the predictor, for the second column of the predictor (specify either t0b or n0b but not both)
t0c	a single value of the predictor, for the third column of the predictor (specify either t0c or n0c but not both)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
t03	a single value of the predictor (specify either t03 or n03 but not both)
t10	a single value of the predictor for the mean (specify either $t10$ or $n10$ but not both)
t20	a single value of the predictor for the sd (specify either t20 or n20 but not both)
t30	a single value of the predictor for the shape (specify either t30 or n30 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
n10	an index for the predictor for the mean (specify either t10 or n10 but not both)
n20	an index for the predictor for the sd (specify either t10 or n10 but not both)
p	a vector of probabilities at which to generate predictive quantiles
n	number of random samples required
У	a vector of values at which to calculate the density and distribution functions
ics	initial conditions for the maximum likelihood search
tresid	predictor residuals
tresid0	predictor residual at the point being predicted
muhat0	muhat at the point being predicted
vhat	vector of all parameters
v1	first parameter
v1hat	first parameter
v1h	
V 111	first parameter
d1	first parameter the delta used in the numerical derivatives with respect to the parameter
	•
d1	the delta used in the numerical derivatives with respect to the parameter the fractional delta used in the numerical derivatives with respect to the param-
d1 fd1	the delta used in the numerical derivatives with respect to the parameter the fractional delta used in the numerical derivatives with respect to the parameter
d1 fd1 v2	the delta used in the numerical derivatives with respect to the parameter the fractional delta used in the numerical derivatives with respect to the parameter second parameter
d1 fd1 v2 v2hat	the delta used in the numerical derivatives with respect to the parameter the fractional delta used in the numerical derivatives with respect to the parameter second parameter second parameter
d1 fd1 v2 v2hat v2h	the delta used in the numerical derivatives with respect to the parameter the fractional delta used in the numerical derivatives with respect to the parameter second parameter second parameter second parameter
d1 fd1 v2 v2hat v2h d2	the delta used in the numerical derivatives with respect to the parameter the fractional delta used in the numerical derivatives with respect to the parameter second parameter second parameter second parameter the delta used in the numerical derivatives with respect to the parameter the fractional delta used in the numerical derivatives with respect to the parameter

v3h third parameter

d3 the delta used in the numerical derivatives with respect to the parameter

fd3 the fractional delta used in the numerical derivatives with respect to the param-

eter

v4 fourth parameter v4hat fourth parameter v4h fourth parameter

the delta used in the numerical derivatives with respect to the parameter

fd4 the fractional delta used in the numerical derivatives with respect to the param-

eter

v5 fifth parameter v5hat fifth parameter v5h fifth parameter

d5 the delta used in the numerical derivatives with respect to the parameter

v6 sixth parameter v6hat sixth parameter v6h sixth parameter

d6 the delta used in the numerical derivatives with respect to the parameter

minxi minimum value of shape parameter xi
maxxi maximum value of shape parameter xi
ximin minimum value of shape parameter xi
ximax maximum value of shape parameter xi

fdalpha the fractional delta used in the numerical derivatives with respect to probability,

for calculating the pdf as a function of quantiles

kscale the known scale parameter
kloc the known location parameter
kshape the known shape parameter

kdf the known degrees of freedom parameter

kbeta the known beta parameter

alpha a vector of values of alpha (one minus probability)
ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor mu the location parameter of the distribution sigma the sigma parameter of the distribution

sigma1 first coefficient for the sigma parameter of the distribution sigma2 second coefficient for the sigma parameter of the distribution

scale the scale parameter of the distribution shape the shape parameter of the distribution

xi	the shape parameter of the distribution	
xi1	first coefficient for the shape parameter of the distribution	
xi2	second coefficient for the shape parameter of the distribution	
lambda	the lambda parameter of the distribution	
log	logical for the density evaluation	
mm	an index for which derivative to calculate	
nn	an index for which derivative to calculate	
rr	an index for which derivative to calculate	
lddi	inverse observed information matrix	
lddi_k2	inverse observed information matrix, fixed shape parameter	
lddi_k3	inverse observed information matrix, fixed shape parameter	
lddi_k4	inverse observed information matrix, fixed shape parameter	
lddd	third derivative of log-likelihood	
lddd_k2	third derivative of log-likelihood, fixed shape parameter	
lddd_k3	third derivative of log-likelihood, fixed shape parameter	
lddd_k4	third derivative of log-likelihood, fixed shape parameter	
lambdad	derivative of the log prior	
lambdad_cp	derivative of the log prior	
lambdad_rhp	derivative of the log RHP prior	
lambdad_flat	derivative of the log flat prior	
lambdad_rh_mle	derivative of the log CRHP-MLE prior	
lambdad_rh_flat		
	derivative of the log CRHP-FLAT prior	
lambdad_jp	derivative of the log JP prior	
lambdad_custom	custom value of the derivative of the log prior	
means	logical that indicates whether to return analytical estimates for the distribution means (longer runtime)	
waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)	
logscores	logical that indicates whether to return leave-one-out estimates estimates of the log-score (much longer runtime)	
extramodels	logical that indicates whether to add three additional prediction models	
pdf	logical that indicates whether to return density functions evaluated at quantiles specified by input probabilities	
predictordata	logical that indicates whether to calculate and return predictordata	
nonnegslopesonly		
_	logical that indicates whether to disallow non-negative slopes	
rnonnegslopesor	nly	

logical that indicates whether to disallow non-negative slopes

658 manlddd

customprior a custom value for the slope of the log prior at the maxlik estimate

prior logical indicating which prior to use params model parameters for calculating logf

yy vector of samples
pp vector of probabilities
dlogpi gradient of the log prior

debug debug flag

centering indicates whether the routine should center the data or not

aderivs logical for whether to use analytic derivatives (instead of numerical)

#### Value

No return value

manldd Return message for ldd

# Description

Return message for ldd

# Usage

manldd()

#### Value

Square scalar matrix

manlddd Return message for lddd

## Description

Return message for lddd

### Usage

manlddd()

#### Value

Cubic scalar array

manlnn 659

manlnn

Return message for lnn

## Description

Return message for lnn

# Usage

manlnn()

### Value

Scalar value

manlnnn

Return message for lnnn

# Description

Return message for lnnn

# Usage

manlnnn()

## Value

Scalar value

 ${\tt manlogf}$ 

Return message for Logf

# Description

Return message for Logf

# Usage

manlogf()

### Value

Scalar value.

manmeans manmeans

manloglik

Return message for loglik

## Description

Return message for loglik

## Usage

manloglik()

### Value

Scalar

 ${\tt manlogscores}$ 

Return message for logscores

# Description

Return message for logscores

## Usage

manlogscores()

## Value

Two scalars

manmeans

Return message for means

# Description

Return message for means

# Usage

manmeans()

### Value

Two scalars

manpredictor 661

 ${\tt manpredictor}$ 

Return message for predictor.

## Description

Return message for predictor.

# Usage

manpredictor()

### Value

Two vectors

manvector

 $Return\ message\ for\ vector$ 

# Description

Return message for vector

## Usage

manvector()

## Value

Vector

manwaic

Return message for WAIC

# Description

Return message for WAIC

# Usage

manwaic()

### Value

Two numeric values.

ms\_flat\_1tail

movexiawayfromzero

Move xi away from zero a bit

## Description

Move xi away from zero a bit

### Usage

```
movexiawayfromzero(xi)
```

### **Arguments**

xi xi

#### Value

Scalar

ms\_flat\_1tail

Illustration of Model Selection Among 10 One Tail Distributions from the fitdistcp Package

### **Description**

Applies model selection using AIC, WAIC1, WAIC2 and leave-one-out logscore to the input data x, for 10 one tailed models in the fitdistcp package (although for the GPD, the logscore is NA for mathematical reasons).

The code is straightforward, and the point is to illustrate what is possible using the model selection outputs from the fitdistcp routines.

The input data may be automatically shifted so that the minimum value is positive.

For the Pareto, the data may be further shifted so that the minimum value is slightly greater than 1.

# Usage

```
ms_flat_1tail(
    x,
    index = 1,
    nyears = 10,
    plottype = "empirical",
    plottingposition = "Weibull",
    quiet = FALSE
)
```

ms\_flat\_1tail 663

## **Arguments**

x	data vector	
index	which data point to use for plotting positions	
nyears	number of years for frequency calculations	
plottype	What to plot? Possible values are 'both', 'empirical', 'cp'	
plottingposition		
	Weibull or Hazen	
quiet	logical for whether to print screen messages	

#### **Details**

The 10 models are: exp, pareto\_k2, halfnorm, lnorm, frechet\_k1, weibull, gamma, invgamma, invgauss and gpd\_k1.

#### Value

Plots QQ plots to the screen, for each of the models, and returns a data frame containing

- MLE parameter values
- AIC scores (times -0.5), AIC weights
- WAIC1 scores, WAIC1 weights
- WAIC2 scores, WAIC2 weights
- logscores, logscore weights
- maximum likelihood and calibrating prior means
- maximum likelihood and calibrating prior standard deviations

### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

## **Examples**

```
# because it's too slow for CRAN
set.seed(1)
nx=50
x=rlnorm(nx)
print(ms_flat_1tail(x))
```

ms\_flat\_2tail

ms\_flat\_2tail

Illustration of Model Selection Among 18 Distributions from the fitdistcp Package

#### Description

Applies model selection using AIC, WAIC1, WAIC2 and leave-one-out logscore to the input data x, for 7 two tailed models in the fitdistcp packages

The code is straightforward, and the point is to illustrate what is possible using the model selection outputs from the fitdistcp routines.

## Usage

```
ms_flat_2tail(x)
```

#### Arguments

Χ

data vector

#### **Details**

The 7 models are: norm, gnorm\_k3, gumbel, logis, lst\_k3, cauchy, gev

#### Value

Plots QQ plots to the screen, for each of the models, and returns a data frame containing

- AIC scores (times -0.5), AIC weights
- WAIC1 scores, WAIC1 weights
- WAIC2 scores, WAIC2 weights
- logscores, logscore weights
- · maximum likelihood and calibrating prior means
- maximum likelihood and calibrating prior standard deviations

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

### **Examples**

```
# because it's too slow for CRAN
set.seed(1)
nx=50
x=rnorm(nx)
print(ms_flat_2tail(x))
```

ms\_predictors\_1tail 665

### Description

Applies model selection using AIC, WAIC1, WAIC2 and leave-one-out logscore to the input data x, t, for 5 one tailed models with predictors in the fitdistcp package.

The code is straightforward, and the point is to illustrate what is possible using the model selection outputs from the fitdistcp routines.

The input data may be automatically shifted so that the minimum value is positive.

For the Pareto, the data is so that the minimum value is slightly greater than 1.

## Usage

```
ms_predictors_1tail(x, t)
```

#### **Arguments**

x data vector

t predictor vector

### **Details**

The 5 models are: exp\_p1, pareto\_p1k2, lnorm\_p1, frechet\_p2k1, weibull\_p2.

## Value

Plots QQ plots to the screen, for each of the 5 models, and returns a data frame containing

- AIC scores, AIC weights
- WAIC1 scores, WAIC1 weights
- WAIC2 scores, WAIC2 weights
- logscores and logscore weights

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

666 ms\_predictors\_2tail

### **Examples**

```
# because it's too slow for CRAN
set.seed(3)
nx=100
predictor=c(1:nx)/nx
x=rlnorm(nx,meanlog=predictor,sdlog=0.1)
print(ms_predictors_1tail(x,predictor))
```

### **Description**

Applies model selection using AIC, WAIC1, WAIC2 and leave-one-out logscore to the input data x, t, for 6 two tail models with predictors in the fitdistcp packages (although for the GEV, the logscore is NA for mathematical reasons).

The code is straightforward, and the point is to illustrate what is possible using the model selection outputs from the fitdistcp routines.

GEVD is temperamental in that it doesn't work if the shape parameter is extreme.

#### Usage

```
ms_predictors_2tail(x, t)
```

#### **Arguments**

- x data vector
  t predictor vector
- Details

The 11 models are: norm\_p1, gumbel\_p1, logis\_p1, lst\_k3\_p1, cauchy\_p1 and gev\_p1.

## Value

Plots QQ plots to the screen, for each of the 6 models, and returns a data frame containing

- AIC scores, AIC weights
- WAIC1 scores, WAIC1 weights
- WAIC2 scores, WAIC2 weights
- logscores and logscore weights

nopdfcdfmsg 667

### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

### **Examples**

```
# because it's too slow for CRAN
set.seed(2)
nx=100
predictor=c(1:nx)/nx
x=rnorm(nx,mean=predictor,sd=1)
print(ms_predictors_2tail(x,predictor))
```

nopdfcdfmsg

Message to explain why GEV and GPD d\*\*\* and p\*\*\* routines don't return DMGS pdfs and cdfs

## Description

Message to explain why GEV and GPD d\*\*\* and p\*\*\* routines don't return DMGS pdfs and cdfs

### Usage

```
nopdfcdfmsg(yy, pp)
```

# Arguments

yy vector of samples
pp vector of probabilities

### Value

String

norm\_boot

**Bootstrap** 

### Description

Bootstrap

#### Usage

```
norm_boot(x, n)
```

#### **Arguments**

x a vector of training data values

n number of random samples required

#### Value

A list containing a matrix of simulated parameter values

norm\_cp

Normal Distribution Predictions Based on a Calibrating Prior

## Description

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

## Usage

```
qnorm_cp(
 Х,
  p = seq(0.1, 0.9, 0.1),
 means = FALSE,
 waicscores = FALSE,
 logscores = FALSE,
  rust = FALSE,
 nrust = 1e+05,
 unbiasedv = FALSE,
  debug = FALSE
)
rnorm_cp(n, x, method = "rust", rust = FALSE, mlcp = TRUE, debug = FALSE)
dnorm_cp(
 Х,
 y = x,
  rust = FALSE,
 nrust = 1000,
 boot = FALSE,
 nboot = 1000,
 debug = FALSE
)
pnorm_cp(
 Х,
 y = x,
 rust = FALSE,
 nrust = 1000,
 boot = FALSE,
 nboot = 1000,
 debug = FALSE
)
tnorm\_cp(method, n, x, debug = FALSE)
```

### **Arguments**

X	a vector of training data values
р	a vector of probabilities at which to generate predictive quantiles
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)

rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
unbiasedv	logical for whether to include unbiased variance results in norm
debug	logical for turning on debug messages
n	the number of random samples required
method	character string that indicates whether to use rust $method=rust$ or $bootstrap$ $method=boot$
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions
boot	logical that indicates whether bootstrap-based posterior sampling calculations should be run or not (longer run time)
nboot	the number of posterior samples used in the bootstrap calculations

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.

• cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The normal distribution has probability density function

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/(2\sigma^2)}$$

where x is the random variable and  $\mu, \sigma > 0$  are the parameters.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

### **Details (analytic integration)**

For this model, the Bayesian prediction equation is integrated analytically.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),

- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

### **Examples**

```
#
# example 1
x=fitdistcp::d030norm_example_data_v1
p=c(1:9)/10
q=qnorm_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qnorm_cp)",
main="Normal: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

norm\_dmgs\_cp

Normal Distribution Predictions Based on a Calibrating Prior, using DMGS (for testing only)

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

### Usage

```
qnorm_dmgs_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    debug = FALSE
)

rnorm_dmgs_cp(n, x, mlcp = TRUE, debug = FALSE)

dnorm_dmgs_cp(x, y = x, debug = FALSE)

pnorm_dmgs_cp(x, y = x, debug = FALSE)
```

#### **Arguments**

X	a vector of training data values
p	a vector of probabilities at which to generate predictive quantiles
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
debug	logical for turning on debug messages
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- $\bullet$  adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- $\bullet \ \, {\tt ml\_deviates:} \ \, {\tt random} \ \, {\tt deviates} \ \, {\tt calculated} \ \, {\tt using} \ \, {\tt maximum} \ \, {\tt likelihood.}$
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### Details of the Model

The normal distribution has probability density function

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/(2\sigma^2)}$$

where x is the random variable and  $\mu, \sigma > 0$  are the parameters.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

• ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)

• cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),

- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

## **Examples**

```
# # example 1
x=fitdistcp::d030norm_example_data_v1
p=c(1:9)/10
q=qnorm_dmgs_cp(x,p)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qnorm_dmgs_cp)",
main="Normal_DMGS: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
```

norm\_dmgs\_loglik

log-likelihood function

### **Description**

log-likelihood function

## Usage

```
norm_dmgs_loglik(vv, x)
```

## Arguments

vv parameters

x a vector of training data values

#### Value

Scalar

norm\_dmgs\_logscores

Log scores for MLE and RHP predictions calculated using leave-one-out

## Description

Log scores for MLE and RHP predictions calculated using leave-one-out

## Usage

```
norm_dmgs_logscores(logscores, x)
```

norm\_dmgs\_means 681

### **Arguments**

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data values

#### Value

Two scalars

norm\_dmgs\_means

MLE and RHP predictive means

## Description

MLE and RHP predictive means

## Usage

```
norm_dmgs_means(means, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 2)
```

## **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

lddi inverse observed information matrix

1ddd third derivative of log-likelihood

lambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

## Value

Two scalars

682 norm\_f1fa

norm_dmgs_waic	Waic
----------------	------

# Description

Waic

## Usage

```
norm_dmgs_waic(waicscores, x, v1hat, v2hat, lddi, lddd, lambdad)
```

## Arguments

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data values

v1hat first parameter v2hat second parameter

1ddi inverse observed information matrix1ddd third derivative of log-likelihood1ambdad derivative of the log prior

#### Value

Two numeric values.

norm_f1fa	The first derivative of the density

# Description

The first derivative of the density

## Usage

```
norm_f1fa(x, v1, v2)
```

## Arguments

x a ve	ector of	training of	data values
--------	----------	-------------	-------------

v1 first parameter v2 second parameter

### Value

Vector

norm\_f2fa 683

norm	f2fa
HOLIN	1210

The second derivative of the density

## Description

The second derivative of the density

### Usage

```
norm_f2fa(x, v1, v2)
```

## **Arguments**

x a vector of training data values

v1 first parameterv2 second parameter

### Value

Matrix

norm\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
norm_fd(x, v1, v2)
```

### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

# Value

Vector

684 norm\_ldda

norm_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
norm_fdd(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameterv2 second parameter

### Value

Matrix

norm\_ldda

The second derivative of the normalized log-likelihood

## Description

The second derivative of the normalized log-likelihood

### Usage

```
norm_ldda(x, v1, v2)
```

### **Arguments**

x a vector of training data values

v1 first parameterv2 second parameter

## Value

Matrix

norm\_lddda 685

norm\_lddda

The third derivative of the normalized log-likelihood

## Description

The third derivative of the normalized log-likelihood

# Usage

```
norm_lddda(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

3d array

norm\_logf

Logf for RUST

## Description

Logf for RUST

## Usage

```
norm_logf(params, x)
```

# **Arguments**

params model parameters for calculating logf

x a vector of training data values

#### Value

Scalar value.

686 norm\_logfddd

norm_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
norm_logfdd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

## Value

Matrix

norm_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	V/ 3

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
norm_logfddd(x, v1, v2)
```

## **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

3d array

norm\_logscores 687

norm_logscores Log scores for MLE and RHP predictions calculated using leave-one- out	ve-one-
------------------------------------------------------------------------------------------	---------

## Description

Log scores for MLE and RHP predictions calculated using leave-one-out

## Usage

```
norm_logscores(logscores, x)
```

# Arguments

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data values

#### Value

Two scalars

norm\_ml\_params

Maximum likelihood estimator

# Description

Maximum likelihood estimator

# Usage

```
norm_ml_params(x)
```

# Arguments

x a vector of training data values

#### Value

Scalar

norm\_mu2fa

norm\_mu1fa

Minus the first derivative of the cdf, at alpha

## Description

Minus the first derivative of the cdf, at alpha

#### Usage

```
norm_mu1fa(alpha, v1, v2)
```

#### **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

#### Value

Vector

norm\_mu2fa

Minus the second derivative of the cdf, at alpha

## Description

Minus the second derivative of the cdf, at alpha

## Usage

```
norm_mu2fa(alpha, v1, v2)
```

#### **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

#### Value

Matrix

norm\_p12\_boot 689

norm\_p12\_boot

Bootstrap

# Description

Bootstrap

## Usage

```
norm_p12\_boot(x, t1, t2, n)
```

# Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
n	number of random samples required

#### Value

A list containing a matrix of simulated parameter values

norm\_p12\_checkmle

Check MLE

# Description

Check MLE

# Usage

```
norm_p12_checkmle(ml_params)
```

## Arguments

ml\_params

parameters

## Value

No return value (just a message to the screen).

norm\_p12\_cp

Normal Distribution with Predictors on both Mean and Standard Deviation, with Parameter Uncertainty

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qnorm_p12_cp(
    x,
    t1,
    t2,
    t01 = NA,
    t02 = NA,
    n01 = NA,
    n02 = NA,
    p = seq(0.1, 0.9, 0.1),
    ics = c(0, 0, 0, 0),
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    rust = FALSE,
```

```
nrust = 1e+05,
  extramodels = FALSE,
 predictordata = TRUE,
  centering = TRUE,
  debug = FALSE
)
rnorm_p12_cp(
 n,
 х,
  t1,
  t2,
 n01 = NA,
 n02 = NA
  t01 = NA,
  t02 = NA,
  ics = c(0, 0, 0, 0),
  rust = FALSE,
 mlcp = TRUE,
 debug = FALSE
)
dnorm_p12_cp(
 х,
  t1,
  t2,
  t01 = NA,
  t02 = NA,
 n01 = NA
 n02 = NA,
 y = x,
  ics = c(0, 0, 0, 0),
  rust = FALSE,
  nrust = 1000,
  boot = FALSE,
 nboot = 10,
  centering = TRUE,
  rnonnegslopesonly = FALSE,
  debug = FALSE
)
pnorm_p12_cp(
 Х,
  t1,
  t2,
  t01 = NA,
  t02 = NA,
  n01 = NA,
```

```
n02 = NA,
 y = x,
 ics = c(0, 0, 0, 0),
 rust = FALSE,
 nrust = 1000,
 boot = FALSE,
 nboot = 10,
 centering = TRUE,
 rnonnegslopesonly = FALSE,
 debug = FALSE
)
tnorm_p12_cp(
 method,
 n,
 х,
  t1,
  t2,
 nonnegslopesonly = FALSE,
 ics = c(0, 0, 0, 0),
 debug = FALSE
)
```

# **Arguments** ×

t1	a vector of predictors for the mean, such that $length(t1)=length(x)$
t2	a vector of predictors for the sd, such that length(t2)=length(x)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
n01	an index for the predictor (specify either t01 or n01 but not both)
n02	an index for the predictor (specify either t02 or n02 but not both)
р	a vector of probabilities at which to generate predictive quantiles
ics	initial conditions for the maximum likelihood search
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations

a vector of training data values

extramodels logical that indicates whether to run additional calculations and add three additional prediction models (longer runtime) logical that indicates whether predictordata should be calculated predictordata centering logical that indicates whether the predictor should be centered logical for turning on debug messages debug the number of random samples required mlcp logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST) a vector of values at which to calculate the density and distribution functions logical that indicates whether bootstrap-based posterior sampling calculations boot should be run or not (longer run time) the number of posterior samples used in the bootstrap calculations nboot rnonnegslopesonly logical that indicates whether to disallow non-negative slopes method character string that indicates whether to use rust method=rust or bootstrap

method=boot

nonnegslopesonly

logical that indicates whether to disallow non-negative slopes

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The normal distribution with predictors on both parameters has probability density function

$$f(x; \alpha, \beta, \gamma, \delta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x - \mu(\alpha, \beta))^2/(2\sigma(\gamma, \delta)^2)}$$

where x is the random variable,  $\mu = \alpha + \beta t_1$  is the location parameter, modelled as a function of parameters  $\alpha, \beta$  and predictor  $t_1$ , where  $t_1$  is typically the ensemble mean, and  $\sigma = \exp(\gamma + \delta \log(t_2))$  is the scale parameter, modelled as a function of parameters  $\gamma, \delta$  and predictor  $t_2$ , where  $t_2$  is typically the ensemble spread.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\alpha, \beta, \gamma, \delta) \propto \frac{1}{\sigma}$$

as given in the Jewson et al. (2025) reference given below.

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

#### If logscores=TRUE:

• ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)

• cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which introduces this model.

• Jewson S., Olivetti L., Messori G., Northop P., Sweeting T. (2025): An Objective Bayesian Method for Including Parameter Uncertainty in Ensemble Model Output Statistics; QJRMS (Quarterly Journal of the Royal Meteorological Society).

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),

- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

#### **Examples**

```
#
# example 1
x=fitdistcp::d060norm_p1_example_data_v1_x
tt=fitdistcp::d060norm_p1_example_data_v1_t
p=c(1:9)/10
n0=10
q=qnorm_p12_cp(x,t1=tt,t2=tt,n01=n0,n02=n0,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qnorm_p12_cp)",
main="Normal w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

norm\_p12\_f1fa

norm\_p12\_exampledata Norm\_p12 example data

## Description

Norm\_p12 example data

# Usage

```
norm_p12_exampledata(iseed)
```

# Arguments

iseed The random seed

## Value

A list containing data to run an example

norm\_p12\_f1fa

The first derivative of the density for DMGS

# Description

The first derivative of the density for DMGS

#### Usage

```
norm_p12_f1fa(x, t01, t02, v1, v2, v3, v4)
```

# Arguments

X	a vector of training data values
t01	a single value of the predictor (specify either $t01$ or $n01$ but not both)
t02	a single value of the predictor (specify either $t02$ or $n02$ but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Vector

norm\_p12\_f1fw 699

norm	p12	f1	fw

The first derivative of the density for WAIC

## Description

The first derivative of the density for WAIC

## Usage

```
norm_p12_f1fw(x, t1, t2, v1, v2, v3, v4)
```

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

## Value

Vector

norm\_p12\_f2fa

The second derivative of the density for DMGS

## Description

The second derivative of the density for DMGS

## Usage

```
norm_p12_f2fa(x, t01, t02, v1, v2, v3, v4)
```

# Arguments

X	a vector of training data values
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

700 norm\_p12\_fd

#### Value

Matrix

norm\_p12\_f2fw

The second derivative of the density for WAIC

# Description

The second derivative of the density for WAIC

# Usage

```
norm_p12_f2fw(x, t1, t2, v1, v2, v3, v4)
```

#### Arguments

Х	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Matrix

norm\_p12\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
norm_p12_fd(x, t1, t2, v1, v2, v3, v4)
```

norm\_p12\_fdd 701

# Arguments

x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

# Value

Vector

norm_p12_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
norm_p12_fdd(x, t1, t2, v1, v2, v3, v4)
```

# Arguments

V	a vector of training data values
X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

# Value

Matrix

702 norm\_p12\_lddda

10 0 10 m	<b>-1</b> 2	٦ ما.	٦,
norm	DIZ	Tu	ua

The second derivative of the normalized log-likelihood

#### **Description**

The second derivative of the normalized log-likelihood

#### Usage

```
norm_p12_ldda(x, t1, t2, v1, v2, v3, v4)
```

## Arguments

x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

## Value

Matrix

norm\_p12\_lddda

The third derivative of the normalized log-likelihood

## Description

The third derivative of the normalized log-likelihood

## Usage

```
norm_p12_lddda(x, t1, t2, v1, v2, v3, v4)
```

# Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

norm\_p12\_logf 703

#### Value

3d array

norm\_p12\_logf

Logf for RUST

#### Description

Logf for RUST

## Usage

```
norm_p12_logf(params, x, t1, t2, nonnegslopesonly = FALSE)
```

#### **Arguments**

params model parameters for calculating logf

x a vector of training data values

t1 a vector of predictors for the mean

t2 a vector of predictors for the sd

nonnegslopesonly

logical that indicates whether to disallow non-negative slopes

#### Value

Scalar value.

norm\_p12\_logfdd Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
norm_p12_logfdd(x, t1, t2, v1, v2, v3, v4)
```

704 norm\_p12\_logfddd

## Arguments

Χ	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Matrix

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
norm_p12_logfddd(x, t1, t2, v1, v2, v3, v4)
```

# Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

3d array

norm\_p12\_loglik 705

norm_	n12	1001	il
TIOT III_	_レ ၊ ∠_	TORT	ıκ

observed log-likelihood function

## Description

observed log-likelihood function

#### Usage

```
norm_p12_loglik(vv, x, t1, t2)
```

## Arguments

VV	parameters
v v	parameters

x a vector of training data values
 t1 a vector of predictors for the mean
 t2 a vector of predictors for the sd

#### Value

Scalar

norm\_p12\_logscores

Log scores for 5 predictions calculated using leave-one-out

# Description

Log scores for 5 predictions calculated using leave-one-out

# Usage

```
norm_p12_logscores(logscores, x, t1, t2, ics)
```

#### **Arguments**

logscores	logical that indicates whether to return leave-one-out estimates estimates of the
	log-score (much longer runtime)
X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
ics	initial conditions for the maximum likelihood search

#### Value

Two scalars

706 norm\_p12\_mu2fa

norm_	n1	2	mu1	fa
1101 111_	_ D I	<b>-</b>	_IIIIQ I	ı u

Minus the first derivative of the cdf, at alpha

## **Description**

Minus the first derivative of the cdf, at alpha

## Usage

```
norm_p12_mu1fa(alpha, t01, t02, v1, v2, v3, v4)
```

## Arguments

alpha	a vector of values of alpha (one minus probability)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

## Value

Vector

norm	n12	mu2fa
1101 111	012	_IIIU∠I a

Minus the second derivative of the cdf, at alpha

# Description

Minus the second derivative of the cdf, at alpha

## Usage

```
norm_p12_mu2fa(alpha, t01, t02, v1, v2, v3, v4)
```

# Arguments

alpha	a vector of values of alpha (one minus probability)
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

norm\_p12\_p1fa 707

## Value

Matrix

norm\_p12\_p1fa

The first derivative of the cdf

## Description

The first derivative of the cdf

# Usage

```
norm_p12_p1fa(x, t01, t02, v1, v2, v3, v4)
```

# Arguments

x	a vector of training data values
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either $t02$ or $n02$ but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

## Value

Vector

norm\_p12\_p2fa

The second derivative of the cdf

# Description

The second derivative of the cdf

# Usage

```
norm_p12_p2fa(x, t01, t02, v1, v2, v3, v4)
```

# Arguments

X	a vector of training data values
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

## Value

Matrix

norm_p12_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
norm_p12_pd(x, t1, t2, v1, v2, v3, v4)
```

# Arguments

x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

# Value

Vector

norm\_p12\_pdd 709

norm_p12_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	by Maren Causen and Serguer Solor

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
norm_p12_pdd(x, t1, t2, v1, v2, v3, v4)
```

# Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1	first parameter
v2	second parameter
v3	third parameter
v4	fourth parameter

#### Value

Matrix

```
norm_p12_predictordata
```

Predicted Parameter and Generalized Residuals

# Description

Predicted Parameter and Generalized Residuals

## Usage

```
norm_p12_predictordata(predictordata, x, t1, t2, t01, t02, params)
```

710 norm\_p12\_setics

# Arguments

predictordata	logical that indicates whether to calculate and return predictordata
x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
params	model parameters for calculating logf

## Value

Two vectors

norm\_p12\_setics Set initial conditions

# Description

Set initial conditions

# Usage

```
norm_p12_setics(x, t1, t2, ics)
```

## Arguments

X	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
ics	initial conditions for the maximum likelihood search

#### Value

Vector

norm\_p12\_waic 711

norm\_p12\_waic

Waic

# Description

Waic

# Usage

```
norm_p12_waic(
  waicscores,
  x,
  t1,
  t2,
  v1hat,
  v2hat,
  v3hat,
  v4hat,
  lddi,
  lddd,
  lambdad
)
```

# Arguments

waicscores	logical that indicates whether to return estimates for the waic1 and waic2 scores (longer runtime)
x	a vector of training data values
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
v1hat	first parameter
v2hat	second parameter
v3hat	third parameter
v4hat	fourth parameter
lddi	inverse observed information matrix
lddd	third derivative of log-likelihood
lambdad	derivative of the log prior

# Value

Two numeric values.

	- 4	_
norm	nΙ	tа

The first derivative of the cdf

## Description

The first derivative of the cdf

#### Usage

```
norm_p1fa(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameter

v2 second parameter

#### Value

Vector

norm_	<b>ը</b> 1	CD
	м.	_~~

Normal Distribution with a Predictor, Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

#### Usage

```
qnorm_p1_cp(
  х,
  t,
  t0 = NA,
 n0 = NA
 p = seq(0.1, 0.9, 0.1),
 means = FALSE,
 waicscores = FALSE,
  logscores = FALSE,
  rust = FALSE,
  nrust = 1e+05,
  centering = TRUE,
  debug = FALSE
)
rnorm_p1_cp(
  n,
  Х,
  t,
  t0 = NA,
  n0 = NA,
  rust = FALSE,
 mlcp = TRUE,
  debug = FALSE
)
dnorm_p1_cp(
  Х,
  t,
  t0 = NA,
  n0 = NA,
 y = x,
  rust = FALSE,
 nrust = 1000,
  centering = TRUE,
  debug = FALSE
)
pnorm_p1_cp(
```

```
x,
t,
t0 = NA,
n0 = NA,
y = x,
rust = FALSE,
nrust = 1000,
centering = TRUE,
debug = FALSE
)

tnorm_p1_cp(n, x, t, debug = FALSE)
```

## Arguments

X	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
р	a vector of probabilities at which to generate predictive quantiles
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
centering	logical that indicates whether the predictor should be centered
debug	logical for turning on debug messages
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.

- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The normal distribution with a predictor has probability density function

$$f(x; \alpha, \beta, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu(\alpha,\beta))^2/(2\sigma^2)}$$

where x is the random variable,  $\mu = \alpha + \beta t$  is the location parameter, modelled as a function of parameters  $\alpha, \beta$  and predictor t, and  $\sigma > 0$  is the scale parameter.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\alpha, \beta, \sigma) \propto \frac{1}{\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (analytic integration)**

For this model, the Bayesian prediction equation is integrated analytically.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

· Cauchy (cauchy),

- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (1st\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

norm\_p1\_f1fa 719

#### **Examples**

```
#
# example 1
x=fitdistcp::d060norm_p1_example_data_v1_x
tt=fitdistcp::d060norm_p1_example_data_v1_t
p=c(1:9)/10
n0=10
q=qnorm_p1_cp(x,tt,n0=n0,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qnorm_p1_cp)",
main="Normal w/ p1: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

norm\_p1\_f1fa

The first derivative of the density for DMGS

#### **Description**

The first derivative of the density for DMGS

The first derivative of the density

#### Usage

```
norm_p1_f1fa(x, t, v1, v2, v3)
norm_p1_f1fa(x, t, v1, v2, v3)
```

#### **Arguments**

х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Vector

720 norm\_p1\_f2fa

norm\_p1\_f1fw

The first derivative of the density for WAIC

## Description

The first derivative of the density for WAIC

## Usage

```
norm_p1_f1fw(x, t, v1, v2, v3)
```

#### **Arguments**

x a vector of training data values
 t a vector or matrix of predictors
 v1 first parameter

v2 second parameter v3 third parameter

#### Value

Vector

norm\_p1\_f2fa

The second derivative of the density for DMGS

#### **Description**

The second derivative of the density for DMGS

The second derivative of the density

#### Usage

```
norm_p1_f2fa(x, t, v1, v2, v3)
norm_p1_f2fa(x, t, v1, v2, v3)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

norm\_p1\_f2fw 721

### Value

Matrix

norm\_p1\_f2fw

The second derivative of the density for WAIC

## Description

The second derivative of the density for WAIC

## Usage

```
norm_p1_f2fw(x, t, v1, v2, v3)
```

# Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Matrix

norm_p1_fd	First derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
norm_p1_fd(x, t, v1, v2, v3)
norm_p1_fd(x, t, v1, v2, v3)
```

722 norm\_p1\_fdd

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Vector

norm\_p1\_fdd Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### **Description**

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
norm_p1_fdd(x, t, v1, v2, v3)
norm_p1_fdd(x, t, v1, v2, v3)
```

## Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Matrix

norm\_p1\_ldda 723

norm	n1	1dda
		Tuua

The second derivative of the normalized log-likelihood

## Description

The second derivative of the normalized log-likelihood The second derivative of the normalized log-likelihood

# Usage

```
norm_p1_ldda(x, t, v1, v2, v3)
norm_p1_ldda(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

## Value

Matrix

norm\_p1\_lddda

The third derivative of the normalized log-likelihood

## Description

The third derivative of the normalized log-likelihood The third derivative of the normalized log-likelihood

```
norm_p1_lddda(x, t, v1, v2, v3)
norm_p1_lddda(x, t, v1, v2, v3)
```

724 norm\_p1\_logf

## Arguments

x a vector of training data values
 t a vector or matrix of predictors
 v1 first parameter
 v2 second parameter

v3 third parameter

## Value

3d array

 $norm_p1_logf$ 

Logf for RUST

# Description

Logf for RUST

# Usage

```
norm_p1_logf(params, x, t)
```

# Arguments

params model parameters for calculating logf
x a vector of training data values
t a vector or matrix of predictors

## Value

Scalar value.

norm\_p1\_logfdd 725

(, , ,	norm_p1_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
--------	----------------	------------------------------------------------------------------------------------------------------------------

## Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
norm_p1_logfdd(x, t, v1, v2, v3)
norm_p1_logfdd(x, t, v1, v2, v3)
```

### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Matrix

norm_p1_logfddd	Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	Zerri() ey intaren etanzen arta zergatet zenet

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
norm_p1_logfddd(x, t, v1, v2, v3)
norm_p1_logfddd(x, t, v1, v2, v3)
```

726 norm\_p1\_loglik

# Arguments

Х	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

# Value

3d array

norm\_p1\_loglik

Normal-with-p1 observed log-likelihood function

# Description

Normal-with-p1 observed log-likelihood function

# Usage

```
norm_p1_loglik(vv, x, t)
```

# Arguments

VV	parameters
x	a vector of training data values
t	a vector or matrix of predictors

## Value

Scalar

norm\_p1\_logscores 727

norm_p1_logscores	Log scores for MLE and RHP predictions calculated using leave-one-
	out

## Description

Log scores for MLE and RHP predictions calculated using leave-one-out

### Usage

```
norm_p1_logscores(logscores, x, t)
```

## Arguments

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data valuest a vector or matrix of predictors

### Value

Two scalars

norm\_p1\_mlparams

Maximum likelihood estimator

# Description

Maximum likelihood estimator

# Usage

```
norm_p1_mlparams(x, t)
```

# Arguments

x a vector of training data valuest a vector or matrix of predictors

### Value

Vector

728 norm\_p1\_mu2fa

	- 4	4	_
norm_	nι	miil	l t a
1101 111_	_ P ' .		

Minus the first derivative of the cdf, at alpha

## Description

Minus the first derivative of the cdf, at alpha Minus the first derivative of the cdf, at alpha

# Usage

```
norm_p1_mu1fa(alpha, t, v1, v2, v3)
norm_p1_mu1fa(alpha, t, v1, v2, v3)
```

# Arguments

alpha	a vector of	of values	of alnha (	one minus	probability)
атрна	a vector (	n varues i	oi aipiia i	One minus	probability

t a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

## Value

Vector

norm\_p1\_mu2fa

Minus the second derivative of the cdf, at alpha

## Description

Minus the second derivative of the cdf, at alpha Minus the second derivative of the cdf, at alpha

```
norm_p1_mu2fa(alpha, t, v1, v2, v3)
norm_p1_mu2fa(alpha, t, v1, v2, v3)
```

norm\_p1\_p1fa 729

# Arguments

alpha a vector of values of alpha (one minus probability)

t a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

### Value

Matrix

norm\_p1\_p1fa

The first derivative of the cdf

# Description

The first derivative of the cdf

The first derivative of the cdf

### Usage

```
norm_p1_p1fa(x, t, v1, v2, v3)
norm_p1_p1fa(x, t, v1, v2, v3)
```

## Arguments

x a vector of training data values

t a vector or matrix of predictors v1 first parameter

v2 second parameter v3 third parameter

Value

Vector

730 norm\_p1\_pd

norm	n 1	nafa
norm	ום	р∠та

The second derivative of the cdf

## Description

The second derivative of the cdf

The second derivative of the cdf

## Usage

```
norm_p1_p2fa(x, t, v1, v2, v3)
norm_p1_p2fa(x, t, v1, v2, v3)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Matrix

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### **Description**

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
norm_p1_pd(x, t, v1, v2, v3)
norm_p1_pd(x, t, v1, v2, v3)
```

norm\_p1\_pdd 731

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Vector

norm_p1_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

## Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
norm_p1_pdd(x, t, v1, v2, v3)
norm_p1_pdd(x, t, v1, v2, v3)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Matrix

732 norm\_p1\_waic

# Description

Predicted Parameter and Generalized Residuals

## Usage

```
norm_p1_predictordata(x, t, t0, params)
```

### **Arguments**

x a vector of training data valuest a vector or matrix of predictors

t0 a single value of the predictor (specify either t0 or n0 but not both)

params model parameters for calculating logf

### Value

Two vectors

## Description

Waic

### Usage

```
norm_p1_waic(waicscores, x, t, v1hat, v2hat, v3hat)
```

## Arguments

waicscores	logical that indicates w	hether to return estimates i	for the waicl	and waic2 scores
------------	--------------------------	------------------------------	---------------	------------------

(longer runtime)

x a vector of training data valuest a vector or matrix of predictors

v1hat first parameter
v2hat second parameter
v3hat third parameter

### Value

Two numeric values.

norm\_p2fa 733

	- 2 C -
norm	DZTa

The second derivative of the cdf

## Description

The second derivative of the cdf

### Usage

```
norm_p2fa(x, v1, v2)
```

### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

### Value

Matrix

norm\_pd

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
norm_pd(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

# Value

Vector

norm\_pdd

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## **Description**

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
norm_pdd(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

### Value

Matrix

# Description

Method of moments estimator

## Usage

```
norm_unbiasedv_params(x)
```

### **Arguments**

x a vector of training data values

### Value

Vector

norm\_waic 735

### **Description**

Waic

#### **Usage**

```
norm_waic(waicscores, x, v1hat, v2hat)
```

### **Arguments**

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data values

v1hat first parameter v2hat second parameter

#### Value

Two numeric values.

pareto\_k2\_cp Pa

Pareto Distribution Predictions Based on a Calibrating Prior

### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

### Usage

```
qpareto_k2_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    kscale = 1,
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    rust = FALSE,
    nrust = 1e+05,
    debug = FALSE
)

rpareto_k2_cp(n, x, kscale = 1, rust = FALSE, mlcp = TRUE, debug = FALSE)

dpareto_k2_cp(x, y = x, kscale = 1, rust = FALSE, nrust = 1000, debug = FALSE)

ppareto_k2_cp(x, y = x, kscale = 1, rust = FALSE, nrust = 1000, debug = FALSE)

tpareto_k2_cp(n, x, kscale = 1, debug = FALSE)
```

### Arguments

p	a vector of probabilities at which to generate predictive quantiles
kscale	the known scale parameter
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
n	the number of random samples required

a vector of training data values

mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Pareto distribution has various forms. The form we are using has exceedance distribution function

 $S(x;\alpha) = \left(\frac{\sigma}{x}\right)^{\alpha}$ 

where  $x \ge \sigma$  is the random variable and  $\alpha > 0, \sigma > 0$  are the shape and scale parameters. We consider the scale parameter  $\sigma$  to be known (hence the k2 in the name).

The calibrating prior is given by the right Haar prior, which is

$$\pi(\alpha) \propto \frac{1}{\alpha}$$

as given in Jewson et al. (2025). Some others authors may refer to the shape and scale parameters as the scale and location parameters, respectively.

### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

### **Details (analytic integration)**

For this model, the Bayesian prediction equation is integrated analytically.

### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),

pareto\_k2\_f1fa 741

- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

### **Examples**

```
#
# example 1
x=fitdistcp::d011pareto_k2_example_data_v1
p=c(1:9)/10
q=qpareto_k2_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles)
xmax=max(q$ml_quantiles,q$cp_quantiles)
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qpareto_k2_cp)",
main="Pareto: quantile estimates")
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

pareto\_k2\_f1fa

The first derivative of the density

### Description

The first derivative of the density

The first derivative of the density

```
pareto_k2_f1fa(x, v1, kscale)
pareto_k2_f1fa(x, v1, kscale)
```

742 pareto\_k2\_f2fa

# Arguments

x a vector of training data values

v1 first parameter

kscale the known scale parameter

### Value

Vector

Vector

pareto\_k2\_f2fa

The second derivative of the density

# Description

The second derivative of the density

The second derivative of the density

## Usage

```
pareto_k2_f2fa(x, v1, kscale)
pareto_k2_f2fa(x, v1, kscale)
```

## Arguments

x a vector of training data values

v1 first parameter

kscale the known scale parameter

### Value

Matrix

Matrix

pareto\_k2\_fd 743

pareto_k2_fd First derivative of the density Created by Stephen Jewson using De- riv() by Andrew Clausen and Serguei Sokol	pareto_k2_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
-------------------------------------------------------------------------------------------------------------------------------	--------------	-------------------------------------------------------------------------------------------------------------

### **Description**

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
pareto_k2_fd(x, v1, v2)
pareto_k2_fd(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

### Value

Vector

Vector

pareto_k2_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
pareto_k2_fdd(x, v1, v2)
pareto_k2_fdd(x, v1, v2)
```

744 pareto\_k2\_ldda

## Arguments

x a vector of training data values

v1 first parameterv2 second parameter

## Value

Matrix

Matrix

pareto\_k2\_ldda

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

The second derivative of the normalized log-likelihood

## Usage

```
pareto_k2_ldda(x, v1, kscale)
pareto_k2_ldda(x, v1, kscale)
```

## Arguments

x a vector of training data values

v1 first parameter

kscale the known scale parameter

### Value

Matrix

Matrix

pareto\_k2\_lddda 745

pareto	レつ	1 4444
Dareto	K /	TUUUA

The third derivative of the normalized log-likelihood

### **Description**

The third derivative of the normalized log-likelihood The third derivative of the normalized log-likelihood

### Usage

```
pareto_k2_lddda(x, v1, kscale)
pareto_k2_lddda(x, v1, kscale)
```

## Arguments

x a vector of training data values

v1 first parameter

kscale the known scale parameter

#### Value

3d array 3d array

pareto\_k2\_logf

Logf for RUST

## Description

```
Logf for RUST
```

### Usage

```
pareto_k2_logf(params, x, kscale)
```

### **Arguments**

params model parameters for calculating logf
x a vector of training data values
kscale the known scale parameter

### Value

Scalar value.

746 pareto\_k2\_logfddd

2 et t () o) Thate it common and 2 et 8 ite 2 et 6 ite	pareto_k2_logfdd	Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
--------------------------------------------------------	------------------	------------------------------------------------------------------------------------------------------------------

### **Description**

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
pareto_k2_logfdd(x, v1, v2)
pareto_k2_logfdd(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Matrix

Matrix

pareto_k2_logfddd	Third derivative of the log density Created by Stephen Jewson using
	Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
pareto_k2_logfddd(x, v1, v2)
pareto_k2_logfddd(x, v1, v2)
```

pareto\_k2\_logscores 747

### **Arguments**

x a vector of training data values

v1 first parameter

v2 second parameter

### Value

3d array

3d array

pareto\_k2\_logscores

Log scores for MLE and RHP predictions calculated using leave-oneout

## Description

Log scores for MLE and RHP predictions calculated using leave-one-out

### Usage

```
pareto_k2_logscores(logscores, x, kscale)
```

## **Arguments**

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data values

kscale the known scale parameter

## Value

Two scalars

748 pareto\_k2\_mu1fa

pareto\_k2\_ml\_params Max

Maximum likelihood estimator

## Description

Maximum likelihood estimator

## Usage

```
pareto_k2_ml_params(x, kscale)
```

# **Arguments**

x a vector of training data values kscale the known scale parameter

### Value

Scalar

pareto\_k2\_mu1fa

Minus the first derivative of the cdf, at alpha

## Description

Minus the first derivative of the cdf, at alpha Minus the first derivative of the cdf, at alpha

### Usage

```
pareto_k2_mu1fa(alpha, v1, kscale)
pareto_k2_mu1fa(alpha, v1, kscale)
```

### **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter

kscale the known scale parameter

### Value

Vector

Vector

pareto\_k2\_mu2fa 749

pareto	1 k2	mu2fa
Dai e L	NZ	IIIUZI

Minus the second derivative of the cdf, at alpha

### **Description**

Minus the second derivative of the cdf, at alpha Minus the second derivative of the cdf, at alpha

### Usage

```
pareto_k2_mu2fa(alpha, v1, kscale)
pareto_k2_mu2fa(alpha, v1, kscale)
```

### **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter

kscale the known scale parameter

### Value

Matrix Matrix

pareto\_k2\_p1fa

The first derivative of the cdf

## Description

The first derivative of the cdf The first derivative of the cdf

### Usage

```
pareto_k2_p1fa(x, v1, kscale)
pareto_k2_p1fa(x, v1, kscale)
```

### **Arguments**

x a vector of training data values

v1 first parameter

kscale the known scale parameter

### Value

Vector

Vector

pareto\_k2\_p2fa

The second derivative of the cdf

## Description

The second derivative of the cdf

The second derivative of the cdf

## Usage

```
pareto_k2_p2fa(x, v1, kscale)
pareto_k2_p2fa(x, v1, kscale)
```

### **Arguments**

x a vector of training data values

v1 first parameter

kscale the known scale parameter

# Value

Matrix

Matrix

pareto\_k2\_pd

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
pareto_k2_pd(x, v1, v2)
pareto_k2_pd(x, v1, v2)
```

pareto\_k2\_pdd 751

### **Arguments**

X	a vector of training data values
---	----------------------------------

v1 first parameter

v2 second parameter

### Value

Vector

Vector

pareto\_k2\_pdd

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### **Description**

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
pareto_k2_pdd(x, v1, v2)
pareto_k2_pdd(x, v1, v2)
```

## Arguments

x a vector of training data values

v1 first parameter v2 second parameter

### Value

Matrix

Matrix

pareto\_k2\_waic Waic

### **Description**

Waic

### Usage

```
pareto_k2_waic(waicscores, x, v1hat, kscale)
```

### **Arguments**

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data values

v1hat first parameter

kscale the known scale parameter

#### Value

Two numeric values.

pareto\_p1k2\_cp Pareto Distribution with a Predictor, Predictions Based on a Calibrating Prior

### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qpareto_p1k2_cp(
 х,
  t,
  t0 = NA,
 n0 = NA,
 p = seq(0.1, 0.9, 0.1),
 kscale = 1,
 means = FALSE,
 waicscores = FALSE,
 logscores = FALSE,
 dmgs = TRUE,
  rust = FALSE,
 nrust = 1e+05,
 predictordata = TRUE,
  centering = TRUE,
  debug = FALSE
)
rpareto_p1k2_cp(
 n,
 Х,
  t,
  t0 = NA,
 n0 = NA,
 kscale = 1,
 rust = FALSE,
 mlcp = TRUE,
 centering = TRUE,
 debug = FALSE
)
dpareto_p1k2_cp(
 х,
  t.
  t0 = NA,
 n0 = NA,
 y = x,
 kscale = 1,
  rust = FALSE,
```

```
nrust = 1000,
  centering = TRUE,
  debug = FALSE
)
ppareto_p1k2_cp(
  х,
  t,
  t0 = NA.
  n0 = NA,
  y = x,
  kscale = 1,
  rust = FALSE,
  nrust = 1000,
  centering = TRUE,
  debug = FALSE
)
tpareto_p1k2_cp(n, x, t, kscale = 1, debug = FALSE)
```

### **Arguments**

Y	a vector of training data values

t a vector of predictors, such that length(t)=length(x)

a single value of the predictor (specify either t0 or n0 but not both)
 an index for the predictor (specify either t0 or n0 but not both)
 a vector of probabilities at which to generate predictive quantiles

kscale the known scale parameter

means logical that indicates whether to run additional calculations and return analytical

estimates for the distribution means (longer runtime)

waicscores logical that indicates whether to run additional calculations and return estimates

for the WAIC1 and WAIC2 scores (longer runtime)

logical that indicates whether to run additional calculations and return leave-

one-out estimates of the log-score (much longer runtime, non-EVT models only)

dmgs logical that indicates whether DMGS calculations should be run or not (longer

run time)

rust logical that indicates whether RUST-based posterior sampling calculations should

be run or not (longer run time)

nrust the number of posterior samples used in the RUST calculations predictordata logical that indicates whether predictordata should be calculated logical that indicates whether the predictor should be centered

debug logical for turning on debug messages
n the number of random samples required

mlcp logical that indicates whether maxlik and parameter uncertainty calculations

should be performed (turn off to speed up RUST)

y a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

### **Details of the Model**

The Pareto distribution with a predictor has various forms. The form we are using has exceedance distribution function

 $S(x; a, b) = \left(\frac{\sigma}{x}\right)^{\alpha(a,b)}$ 

where  $x \ge \sigma$  is the random variable,  $\alpha = \exp(-a - bt)$  is the shape parameter, modelled as a function of parameters a, b, and  $\sigma$  is the scale parameter. We consider the scale parameter  $\sigma$  to be known (hence the k2 in the name).

The calibrating prior is given by the right Haar prior, which is

$$\pi(a,b) \propto 1$$

as given in Jewson et al. (2025). Note that others authors have referred to the shape and scale parameters as the scale and location parameters, respectively.

### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUF:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

pareto\_p1k2\_cp 757

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

758 pareto\_p1k2\_cp

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),

pareto\_p1k2\_cp 759

• t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),

- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

### **Examples**

```
#
# example 1
x=fitdistcp::d056pareto_p1k2_example_data_v1_x
tt=fitdistcp::d056pareto_p1k2_example_data_v1_t
p=c(1:9)/10
n0=10
q=qpareto_p1k2_cp(x,tt,n0=n0,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qpareto_p1k2_cp)",
main="Pareto w/ p2: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

760 pareto\_p1k2\_f1fw

pareto\_p1k2\_f1fa

The first derivative of the density for DMGS

### **Description**

The first derivative of the density for DMGS

### Usage

```
pareto_p1k2_f1fa(x, t0, v1, v2, kscale)
```

### **Arguments**

x a vector of training data values

t0 a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameter v2 second parameter

kscale the known scale parameter

#### Value

Vector

pareto\_p1k2\_f1fw

The first derivative of the density for WAIC

### **Description**

The first derivative of the density for WAIC

### Usage

```
pareto_p1k2_f1fw(x, t, v1, v2, kscale)
```

### **Arguments**

x a vector of training data valuest a vector or matrix of predictors

v1 first parameter v2 second parameter

kscale the known scale parameter

#### Value

pareto\_p1k2\_f2fa 761

	410	COC
pareto	nikz	ナノナネ
pai cto_	DINZ_	_   _   0

The second derivative of the density for DMGS

### **Description**

The second derivative of the density for DMGS

### Usage

```
pareto_p1k2_f2fa(x, t0, v1, v2, kscale)
```

### **Arguments**

x a vector of training data values

t0 a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameter v2 second parameter

kscale the known scale parameter

#### Value

Matrix

pareto\_p1k2\_f2fw

The second derivative of the density for WAIC

### **Description**

The second derivative of the density for WAIC

### Usage

```
pareto_p1k2_f2fw(x, t, v1, v2, kscale)
```

### **Arguments**

x a vector of training data valuest a vector or matrix of predictors

v1 first parameter v2 second parameter

kscale the known scale parameter

#### Value

Matrix

762 pareto\_p1k2\_fdd

pareto_p1k2_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
pareto_p1k2_fd(x, t, v1, v2, v3)
```

## Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

### Value

Vector

pareto_p1k2_fdd	Second derivative of the density Created by Stephen Jewson using De-
	riv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
pareto_p1k2_fdd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

pareto\_p1k2\_ldda 763

### Value

Matrix

pareto\_p1k2\_ldda

The second derivative of the normalized log-likelihood

### **Description**

The second derivative of the normalized log-likelihood

### Usage

```
pareto_p1k2_ldda(x, t, v1, v2, kscale)
```

### **Arguments**

x a vector of training data valuest a vector or matrix of predictors

v1 first parameter v2 second parameter

kscale the known scale parameter

#### Value

Matrix

pareto\_p1k2\_lddda

The third derivative of the normalized log-likelihood

### **Description**

The third derivative of the normalized log-likelihood

#### Usage

```
pareto_p1k2_lddda(x, t, v1, v2, kscale)
```

### **Arguments**

x a vector of training data valuest a vector or matrix of predictors

v1 first parameter v2 second parameter

kscale the known scale parameter

764 pareto\_p1k2\_logfdd

### Value

3d array

pareto\_p1k2\_logf

Logf for RUST

### **Description**

Logf for RUST

### Usage

```
pareto_p1k2_logf(params, x, t, kscale)
```

### **Arguments**

params model parameters for calculating logf
x a vector of training data values
t a vector or matrix of predictors
kscale the known scale parameter

### Value

Scalar value.

pareto\_p1k2\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
pareto_p1k2_logfdd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

pareto\_p1k2\_logfddd 765

### Value

Matrix

## Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

### Usage

```
pareto_p1k2_logfddd(x, t, v1, v2, v3)
```

### **Arguments**

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter

v3 third parameter

## Value

3d array

### **Description**

observed log-likelihood function

## Usage

```
pareto_p1k2_loglik(vv, x, t, kscale)
```

## Arguments

VV	parameters

x a vector of training data valuest a vector or matrix of predictorskscale the known scale parameter

766 pareto\_p1k2\_means

### Value

Scalar

```
{\tt pareto\_p1k2\_logscores} \begin{tabular}{ll} Log\ scores\ for\ MLE\ and\ RHP\ predictions\ calculated\ using\ leave-one-out \\ \end{tabular}
```

### **Description**

Log scores for MLE and RHP predictions calculated using leave-one-out

### Usage

```
pareto_p1k2_logscores(logscores, x, t, kscale, debug)
```

## Arguments

logiscores logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data valuest a vector or matrix of predictorskscale the known scale parameter

debug debug flag

### Value

Two scalars

## Description

```
pareto_k1 distribution: RHP mean
```

# Usage

```
pareto_p1k2_means(
  means,
  t0,
  ml_params,
  lddi,
  lddd,
  lambdad_rhp,
  nx,
  dim = 2,
  kscale
)
```

pareto\_p1k2\_mu1fa 767

### Arguments

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

t0 a single value of the predictor (specify either t0 or n0 but not both)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data
dim number of parameters
kscale the known scale parameter

#### Value

Two scalars

pareto\_p1k2\_mu1fa

Minus the first derivative of the cdf, at alpha

### Description

Minus the first derivative of the cdf, at alpha

### Usage

```
pareto_p1k2_mu1fa(alpha, t0, v1, v2, kscale)
```

#### **Arguments**

alpha a vector of values of alpha (one minus probability)

to a single value of the predictor (specify either to or no but not both)

v1 first parameter v2 second parameter

kscale the known scale parameter

#### Value

768 pareto\_p1k2\_p1fa

pareto\_p1k2\_mu2fa

Minus the second derivative of the cdf, at alpha

### **Description**

Minus the second derivative of the cdf, at alpha

### Usage

```
pareto_p1k2_mu2fa(alpha, t0, v1, v2, kscale)
```

### **Arguments**

alpha a vector of values of alpha (one minus probability)

t0 a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameter v2 second parameter

kscale the known scale parameter

#### Value

Matrix

pareto\_p1k2\_p1fa

The first derivative of the cdf

### **Description**

The first derivative of the cdf

### Usage

```
pareto_p1k2_p1fa(x, t0, v1, v2, kscale)
```

### **Arguments**

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

v1 first parameter v2 second parameter

kscale the known scale parameter

#### Value

pareto\_p1k2\_p2fa 769

pareto	n1k2	n2fa

The second derivative of the cdf

## Description

The second derivative of the cdf

### Usage

```
pareto_p1k2_p2fa(x, t0, v1, v2, kscale)
```

## Arguments

X	a vector of training	data values

to a single value of the predictor (specify either to or no but not both)

v1 first parameter v2 second parameter

kscale the known scale parameter

#### Value

Matrix

pareto	n1k2	nd

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
pareto_p1k2_pd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

### Value

Vector

pareto_p1k2_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
pareto_p1k2_pdd(x, t, v1, v2, v3)
```

## Arguments

x	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

## Value

Matrix

```
pareto_p1k2_predictordata

Predicted Parameter and Generalized Residuals
```

## Description

Predicted Parameter and Generalized Residuals

## Usage

```
pareto_p1k2_predictordata(predictordata, x, t, t0, params, kscale)
```

pareto\_p1k2\_waic 771

### Arguments

predictordata logical that indicates whether to calculate and return predictordata

x a vector of training data valuest a vector or matrix of predictors

to a single value of the predictor (specify either to or no but not both)

params model parameters for calculating logf

kscale the known scale parameter

#### Value

Two vectors

pareto\_p1k2\_waic Waic

### **Description**

Waic

### Usage

```
pareto\_p1k2\_waic(waicscores,\ x,\ t,\ v1hat,\ v2hat,\ kscale,\ lddi,\ lambdad)
```

### **Arguments**

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data valuest a vector or matrix of predictors

v1hat first parameter v2hat second parameter

kscale the known scale parameter

1ddi inverse observed information matrix1ddd third derivative of log-likelihood

lambdad derivative of the log prior

## Value

Two numeric values.

772 pexp\_p1

	- 4
pcauchy_	nΙ
peaacity_	-P '

Cauchy-with-p1 distribution function

## Description

Cauchy-with-p1 distribution function

## Usage

```
pcauchy_p1(x, t0, ymn, slope, scale)
```

## Arguments

x a vector of training data values

t0 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor scale the scale parameter of the distribution

#### Value

Vector

pexp\_p1

Exponential-with-p1 distribution function

### **Description**

Exponential-with-p1 distribution function

### Usage

```
pexp_p1(x, t0, ymn, slope)
```

## Arguments

x a vector of training data values

t0 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor

#### Value

pfrechet\_p2k1 773

pfrechet_p2k1	Frechet_k1-with-p2 distribution function
pfrechet_p2k1	Frechet_k1-with-p2 distribution function

## Description

Frechet\_k1-with-p2 distribution function

### Usage

```
pfrechet_p2k1(x, t0, ymn, slope, lambda, kloc)
```

### **Arguments**

x a vector of training data values

t0 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor lambda the lambda parameter of the distribution

kloc the known location parameter

#### Value

Vector

pgev_p1	GEVD-with-p1: Distribution function

# Description

```
GEVD-with-p1: Distribution function
```

### Usage

```
pgev_p1(y, t0, ymn, slope, sigma, xi)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
t0	a single value of the predictor (specify either t0 or n0 but not both)
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma	the sigma parameter of the distribution
xi	the shape parameter of the distribution

774 pgev\_p123

## Value

Vector

pgev\_p12

GEVD-with-p1: Distribution function

# Description

GEVD-with-p1: Distribution function

# Usage

```
pgev_p12(y, t1, t2, ymn, slope, sigma1, sigma2, xi)
```

## **Arguments**

У		a vector of values at which to calculate the density and distribution functions
t	1	a vector of predictors for the mean
t	2	a vector of predictors for the sd
yı	mn	the location parameter of the function of the predictor
S	lope	the slope of the function of the predictor
S	igma1	first coefficient for the sigma parameter of the distribution
S	igma2	second coefficient for the sigma parameter of the distribution
Х	i	the shape parameter of the distribution

### Value

Vector

pgev\_p123

GEVD-with-p1: Distribution function

# Description

GEVD-with-p1: Distribution function

## Usage

```
pgev_p123(y, t1, t2, t3, ymn, slope, sigma1, sigma2, xi1, xi2)
```

pgev\_p1k3 775

# Arguments

У	a vector of values at which to calculate the density and distribution functions
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma1	first coefficient for the sigma parameter of the distribution
sigma2	second coefficient for the sigma parameter of the distribution
xi1	first coefficient for the shape parameter of the distribution
xi2	second coefficient for the shape parameter of the distribution

### Value

Vector

pgev_p1k3	GEV-with-known-shape-with-p1 distribution function

# Description

GEV-with-known-shape-with-p1 distribution function

## Usage

```
pgev_p1k3(x, t0, ymn, slope, sigma, kshape)
```

# Arguments

X	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
ciamo	the sigma parameter of the distribution

sigma the sigma parameter of the distribution

kshape the known shape parameter

## Value

776 pgumbel\_p1

pgev\_p1n

GEVD-with-p1: Distribution function

### **Description**

GEVD-with-p1: Distribution function

### Usage

```
pgev_p1n(y, t0, params)
```

### **Arguments**

y a vector of values at which to calculate the density and distribution functions

t0 a single value of the predictor (specify either t0 or n0 but not both)

params model parameters for calculating logf

#### Value

Vector

pgumbel\_p1

Gumbel-with-p1 distribution function

# Description

Gumbel-with-p1 distribution function

## Usage

```
pgumbel_p1(x, t0, ymn, slope, sigma)
```

#### **Arguments**

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution

### Value

plnorm\_p1 777

-		-
nl	.norm_	ni
בע	. 1 1 0 1 111_	_レ ו

Normal-with-p1 distribution function

### **Description**

Normal-with-p1 distribution function

### Usage

```
plnorm_p1(x, t0, ymn, slope, sigma)
```

### **Arguments**

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution

#### Value

Vector

-		_
nΙ	ogis	n1

Logistic-with-p1 distribution function

### **Description**

Logistic-with-p1 distribution function

#### Usage

```
plogis_p1(x, t0, ymn, slope, scale)
```

### **Arguments**

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor scale the scale parameter of the distribution

#### Value

778 pnorm\_p1

p]	lst.	_p1	k3
Ρ-		_~ .	

LST-with-p1 distribution function

## Description

LST-with-p1 distribution function

## Usage

```
plst_p1k3(x, t0, ymn, slope, sigma, kdf)
```

#### **Arguments**

Χ	a vector of training data values
---	----------------------------------

to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution kdf the known degrees of freedom parameter

#### Value

Vector

pnorm_	n	1
priorii_	ν	

Normal-with-p1 distribution function

### **Description**

Normal-with-p1 distribution function

### Usage

```
pnorm_p1(x, t0, ymn, slope, sigma)
```

### **Arguments**

Χ	a vector of training data values
^	a vector of training data values

t0 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution

### Value

pnorm\_p12 779

## Description

Normal-with-p12: Distribution function

# Usage

```
pnorm_p12(y, t01, t02, ymn, slope, sigma1, sigma2)
```

## Arguments

у	a vector of values at which to calculate the density and distribution functions
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma1	first coefficient for the sigma parameter of the distribution
sigma2	second coefficient for the sigma parameter of the distribution

#### Value

Vector

|--|--|--|

# Description

Linear regression formula, densities

## Usage

```
pnorm_p1_formula(y, tresid, tresid0, nx, muhat0, v3hat)
```

# Arguments

У	a vector of values at which to calculate the density and distribution functions
tresid	predictor residuals
tresid0	predictor residual at the point being predicted
nx	length of training data
muhat0	muhat at the point being predicted
v3hat	third parameter

780 punif\_formula

### Value

Vector

ppareto\_p1k2

pareto\_k1-with-p2 distribution function

### Description

```
pareto_k1-with-p2 distribution function
```

## Usage

```
ppareto_p1k2(x, t0, ymn, slope, kscale)
```

### **Arguments**

x a vector of training data values

t0 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor

kscale the known scale parameter

#### Value

Vector

punif\_formula

Predictive CDFs

## Description

Predictive CDFs

### Usage

```
punif_formula(x, y)
```

#### **Arguments**

x a vector of training data values

y a vector of values at which to calculate the density and distribution functions

#### Value

Two vectors

pweibull\_p2 781

		-	_
pwei	hul		ทว

Weibull-with-p1 distribution function

### **Description**

Weibull-with-p1 distribution function

### Usage

```
pweibull_p2(x, t0, shape, ymn, slope)
```

## **Arguments**

x a vector of training data values

to a single value of the predictor (specify either to or no but not both)

shape the shape parameter of the distribution

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor

#### Value

Vector

acauc	·h·	n 1
ucauc	.IIV	υı

Cauchy-with-p1 quantile function

### **Description**

Cauchy-with-p1 quantile function

### Usage

```
qcauchy_p1(p, t0, ymn, slope, scale)
```

### **Arguments**

p a vector of probabilities at which to generate predictive quantiles t0 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor scale the scale parameter of the distribution

#### Value

782 qfrechet\_p2k1

-with-p1 quantile function

### **Description**

-with-p1 quantile function

### Usage

```
qexp_p1(p, t0, ymn, slope)
```

### **Arguments**

p a vector of probabilities at which to generate predictive quantiles t0 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor

#### Value

Vector

qfrechet\_p2k1

Frechet\_k1-with-p2 quantile function

#### **Description**

Frechet\_k1-with-p2 quantile function

### Usage

```
qfrechet_p2k1(p, t0, ymn, slope, lambda, kloc)
```

#### **Arguments**

p a vector of probabilities at which to generate predictive quantiles t0 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor the lambda parameter of the distribution

kloc the known location parameter

#### Value

qgamma\_k1\_ppm 783

qgamma\_k1\_ppm

Temporary dummy for one of the cp models

#### **Description**

Temporary dummy for one of the cp models

#### Usage

```
qgamma_k1_ppm(x, p)
```

#### **Arguments**

- x a vector of training data values
- p a vector of probabilities at which to generate predictive quantiles

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.

784 qgamma\_ppm

• cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

qgamma\_ppm

Temporary dummy for one of the ppm models

#### Description

Temporary dummy for one of the ppm models

#### Usage

```
qgamma_ppm(x, p)
```

### **Arguments**

- x a vector of training data values
- p a vector of probabilities at which to generate predictive quantiles

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

qgev\_k12\_ppm 785

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

qgev\_k12\_ppm

Temporary dummy for one of the ppm models

#### **Description**

Temporary dummy for one of the ppm models

## Usage

```
qgev_k12_ppm(x, p)
```

#### **Arguments**

- x a vector of training data values
- p a vector of probabilities at which to generate predictive quantiles

786 *qgev\_k12\_ppm* 

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

qgev\_mpd\_ppm 787

qgev\_mpd\_ppm

Temporary dummy for one of the ppm models

#### **Description**

Temporary dummy for one of the ppm models

#### Usage

```
qgev_mpd_ppm(x, p)
```

## **Arguments**

- x a vector of training data values
- p a vector of probabilities at which to generate predictive quantiles

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.

788 qgev\_p1

• cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

qgev\_p1

GEVD-with-p1: Quantile function

## Description

GEVD-with-p1: Quantile function

## Usage

```
qgev_p1(p, t0, ymn, slope, sigma, xi)
```

## Arguments

p	a vector of probabilities at which to generate predictive quantiles
t0	a single value of the predictor (specify either $t0$ or $n0$ but not both)
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma	the sigma parameter of the distribution
xi	the shape parameter of the distribution

. . . . . . .

#### Value

qgev\_p12 789

qgev\_p12

GEVD-with-p1: Quantile function

# Description

GEVD-with-p1: Quantile function

## Usage

```
qgev_p12(p, t1, t2, ymn, slope, sigma1, sigma2, xi)
```

# Arguments

р	a vector of probabilities at which to generate predictive quantiles
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma1	first coefficient for the sigma parameter of the distribution
sigma2	second coefficient for the sigma parameter of the distribution
xi	the shape parameter of the distribution

## Value

Vector

qgev\_p123

GEVD-with-p1: Quantile function

# Description

```
GEVD-with-p1: Quantile function
```

## Usage

```
qgev_p123(p, t1, t2, t3, ymn, slope, sigma1, sigma2, xi1, xi2)
```

790 *qgev\_p1k3* 

# Arguments

р	a vector of probabilities at which to generate predictive quantiles
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma1	first coefficient for the sigma parameter of the distribution
sigma2	second coefficient for the sigma parameter of the distribution
xi1	first coefficient for the shape parameter of the distribution
xi2	second coefficient for the shape parameter of the distribution

### Value

Vector

qgev_p1k3	GEV-with-known-shape-with-p1 quantile function

# Description

GEV-with-known-shape-with-p1 quantile function

## Usage

```
qgev_p1k3(p, t0, ymn, slope, sigma, kshape)
```

# Arguments

р	a vector of probabilities at which to generate predictive quantiles
t0	a single value of the predictor (specify either t0 or n0 but not both)
ymn	the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution

kshape the known shape parameter

## Value

*qgev\_p1n* 791

qgev\_p1n

GEVD-with-p1: Quantile function

## Description

GEVD-with-p1: Quantile function

## Usage

```
qgev_p1n(p, t0, params)
```

### **Arguments**

p a vector of probabilities at which to generate predictive quantiles t0 a single value of the predictor (specify either t0 or n0 but not both)

params model parameters for calculating logf

### Value

Vector

qgev\_p1\_ppm

Temporary dummy for one of the ppm models

### **Description**

Temporary dummy for one of the ppm models

## Usage

```
qgev_p1_pm(x, t, n0, p)
```

### **Arguments**

X	a vector	of training	data values

t a vector of predictors, such that length(t)=length(x)

no an index for the predictor (specify either to or no but not both)

p a vector of probabilities at which to generate predictive quantiles

792 qgev\_p1\_ppm

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

qgev\_ppm 793

qgev\_ppm

Temporary dummy for one of the ppm models

#### **Description**

Temporary dummy for one of the ppm models

### Usage

```
qgev_ppm(x, p)
```

### **Arguments**

- x a vector of training data values
- p a vector of probabilities at which to generate predictive quantiles

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.

794 *qgpd\_k1\_ppm* 

• cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

qgpd\_k1\_ppm

Temporary dummy for one of the ppm models

## **Description**

Temporary dummy for one of the ppm models

## Usage

```
qgpd_k1_ppm(x, p)
```

## **Arguments**

x a vector of training data values

p a vector of probabilities at which to generate predictive quantiles

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

qgumbel\_p1 795

For models with predictors, q\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

qgumbel\_p1

Gumbel-with-p1 quantile function

### **Description**

Gumbel-with-p1 quantile function

### Usage

```
qgumbel_p1(p, t0, ymn, slope, sigma)
```

796 qlnorm\_p1

### **Arguments**

p a vector of probabilities at which to generate predictive quantiles to a single value of the predictor (specify either to or no but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution

#### Value

Vector

qlnorm_p1 Normal-with-p1 quantile function
--------------------------------------------

## Description

Normal-with-p1 quantile function

## Usage

```
qlnorm_p1(p, t0, ymn, slope, sigma)
```

## **Arguments**

p a vector of probabilities at which to generate predictive quantiles t0 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution

### Value

qlogis\_p1 797

αI	0016	nΊ
4-	ogis_	י א-

Logistic-with-p1 quantile function

# Description

Logistic-with-p1 quantile function

## Usage

```
qlogis_p1(p, t0, ymn, slope, scale)
```

## Arguments

p	a vector of probabilities at which to generate predictive quantiles
t0	a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor scale the scale parameter of the distribution

### Value

Vector

	_	
alst	ומ	ĸЗ

LST-with-p1 quantile function

## Description

LST-with-p1 quantile function

## Usage

```
qlst_p1k3(p, t0, ymn, slope, sigma, kdf)
```

## Arguments

p	a vector of probabilities at which to generate predictive quantiles
t0	a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution kdf the known degrees of freedom parameter

### Value

798 qnorm\_p12

qnorm_p1	Normal-with-p1 quantile function

## Description

Normal-with-p1 quantile function

## Usage

```
qnorm_p1(p, t0, ymn, slope, sigma)
```

## **Arguments**

p a vector of probabilities at which to generate predictive quantiles
 t0 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor sigma the sigma parameter of the distribution

## Value

Vector

qnorm_p12	Normal-with-p12: Quantile function

## Description

Normal-with-p12: Quantile function

## Usage

```
qnorm_p12(p, t01, t02, ymn, slope, sigma1, sigma2)
```

## Arguments

р	a vector of probabilities at which to generate predictive quantiles
t01	a single value of the predictor (specify either t01 or n01 but not both)
t02	a single value of the predictor (specify either t02 or n02 but not both)
ymn	the location parameter of the function of the predictor
slope	the slope of the function of the predictor
sigma1	first coefficient for the sigma parameter of the distribution
sigma2	second coefficient for the sigma parameter of the distribution

qnorm\_p1\_formula 799

## Value

Vector

qnorm\_p1\_formula

Linear regression formula, quantiles

## Description

Linear regression formula, quantiles

## Usage

```
qnorm_p1_formula(alpha, tresid, tresid0, nx, muhat0, v3hat)
```

## **Arguments**

alpha a vector of values of alpha (one minus probability)

tresid predictor residuals

tresid0 predictor residual at the point being predicted

nx length of training data

muhat at the point being predicted

v3hat third parameter

## Value

Vector

qntt\_ppm

Temporary dummy for one of the ppm models

## Description

Temporary dummy for one of the ppm models

## Usage

```
qntt_ppm(x, p)
```

## **Arguments**

x a vector of training data values

p a vector of probabilities at which to generate predictive quantiles

800 qntt\_ppm

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

qpareto\_p1k2 801

qpareto\_p1k2

pareto\_k1-with-p2 quantile function

## Description

```
pareto_k1-with-p2 quantile function
```

## Usage

```
qpareto_p1k2(p, t0, ymn, slope, kscale)
```

## **Arguments**

a vector of probabilities at which to generate predictive quantiles
 a single value of the predictor (specify either t0 or n0 but not both)

ymn the location parameter of the function of the predictor

slope the slope of the function of the predictor

kscale the known scale parameter

## Value

Vector

qunif\_formula

Predictive Quantiles

## Description

**Predictive Quantiles** 

# Usage

```
qunif_formula(x, p)
```

# Arguments

x a vector of training data values

p a vector of probabilities at which to generate predictive quantiles

### Value

Two vectors

qweibull\_p2

Weibull-with-p1 quantile function

## **Description**

Weibull-with-p1 quantile function

### Usage

```
qweibull_p2(p, t0, shape, ymn, slope)
```

### **Arguments**

p a vector of probabilities at which to generate predictive quantiles
t0 a single value of the predictor (specify either t0 or n0 but not both)
shape the shape parameter of the distribution
ymn the location parameter of the function of the predictor
slope the slope of the function of the predictor

#### Value

Vector

reltest

Evaluation of Reliability for Models in the fitdistcp Package

## Description

Uses simulations to evaluate the reliability of the predictive quantiles produced by the q\*\*\*\*\_cp routines in the fitdistcp package.

## Usage

```
reltest(
  model = "exp",
  ntrials = 1000,
  nrepeats = 3,
  nx = 20,
  params = NA,
  alpha = seq(0.005, 0.995, 0.005),
  plotflag = TRUE,
  verbose = TRUE,
  dmgs = TRUE,
  debug = FALSE,
```

```
aderivs = TRUE,
unbiasedv = FALSE,
pwm = FALSE,
minxi = -10,
maxxi = 10
)
```

## **Arguments**

model	which distribution to test. Possibles values are "exp", "pareto_k1", "halfnorm", "unif", "norm", "norm_dmgs", "gnorm_k3", "lnorm", "lnorm_dmgs", "logis", "lst_k3", "cauchy", "gumbel", "frechet_k1", "weibull", "gev_k3", "exp_p1", "pareto_p1k3", "norm_p1", "lnorm_p1", "logis_p1", "lst_p1k4", "cauchy_p1", "gumbel_p1", "frechet_p2k1", "weibull_p2", "gev_p1k4", "norm_p12", "lst_p12k5", "gamma", "invgamma", "invgauss", "gev", "gpd_k1", "gev_p1". "gev_p12". "gev_p123".
ntrials	the number of trials to run. 5000 typically gives good results.
nrepeats	the number of entire repeats of the test to run, to check for convergence. 3 is a good choice.
nx	the length of the training data to use.
params	values for the parameters for the specified distribution
alpha	the exceedance probability values at which to test
plotflag	logical to turn the plotting on and off
verbose	logical to turn loop counting on and off
dmgs	logical to turn DMGS calculations on and off (to optimize speed for maxlik only calculations)
debug	logical for turning debug messages on and off
aderivs	logical for whether to use analytic derivatives (instead of numerical)
unbiasedv	logical for whether to use the unbiased variance instead of maxlik (for the normal)
pwm	logical for whether to use PWM instead of maxlik (for the GEV)
minxi	minimum value for EVT shape parameter
maxxi	maximum value for EVT shape parameter

### **Details**

The maximum likelihood quantiles (plotted in blue) do not give good reliability. They typically underestimate the tails (see panel (f)).

For "exp", "pareto\_k1", "unif", "norm", "lnorm", "norm\_p1" and "lnorm\_p1", the calibrating prior quantiles are calculated using the right Haar prior and an exact solution for the Bayesian prediction integral. They will converge towards exact reliability with a large enough number of trials, for any sample size.

```
For "halfnorm", "norm_dmgs", "lnorm_dmgs", "gnorm_k3", "logis", "lst_k3", "cauchy", "gumbel", "frechet_k1", "weibull", "gev_k3", "exp_p1", "pareto_p1k3", "gumbel_p1", "logis_p1" and
```

"lst\_p1k4" "cauchy\_p1", "gumbel\_p1", "frechet\_p2k1", "weibull\_p2", "gev\_p1k4", "norm\_p12", "lst\_p12k5" the calibrating prior quantiles are calculated using the right Haar prior, with the DMGS asymptotic solution for the Bayesian prediction integral. They will converge towards good reliability with a large enough number of trials, with the only deviation from exact reliability being due to the neglect of higher order terms in the asymptotic expansion. They will converge towards exact reliability with a large enough number of trials and a large enough sample size.

For "gamma", "invgamma", "invgauss", "gev", "gpd\_k1" and "gev\_p1", "gev\_p12", "gev\_p123", the calibrating prior quantiles are calculated using the "fitdistcp" recommended calibrating priors, with the DMGS asymptotic solution for the Bayesian prediction integral. The chosen priors give reasonably good reliability with a large enough number of trials, and for large sample sizes, but may give poor reliability for small sample sizes (e.g., n<20).

#### Value

A plot showing 9 different reliability checks, and a list containing various outputs, including the probabilities shown in the plot.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

• Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),

- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

# Examples

```
set.seed(1)
# example 1
# -runs the default settings, which test reliability for the exponential distribution
reltest()
```

reltest2	Evaluation of Reliability for Certain Additional Models in the
	fitdistcp Package

### **Description**

This routine is mainly for reproducing certain results in Jewson et al. (2025), and not of general interest.

It uses simulations to evaluate the reliability of the predictive quantiles produced by the qgev\_cp, ggpd\_cp and qgev\_p1\_cp routines in the fitdistcp package. For each model, results for 5 models are calculated. This is to illustrate that the calibrating prior predictions dominate the ml, flat, crhp\_ml and jp predictions, in terms of reliability.

### Usage

```
reltest2(
  model = "gev",
  ntrials = 100,
  nrepeats = 3,
  nx = 50,
  params = c(0, 1, 0),
  alpha = seq(0.005, 0.995, 0.005),
  plotflag = TRUE,
  verbose = TRUE
)
```

#### **Arguments**

model	which distribution to test. Possibles values are "gev", "gpd_k1", "gev_p1".
ntrials	the number of trials to run. 5000 typically gives good results.
nrepeats	the number of entire repeats of the test to run, to check for convergence. $3$ is a good choice.
nx	the length of the training data.
params	values for the parameters for the specified distribution
alpha	the alpha values at which to test
plotflag	logical to turn the plotting on and off
verbose	logical to turn loop counting on and off

### **Details**

The maximum likelihood quantiles (plotted in blue) do not give good reliability. They typically underestimate the tails (see panel (f)).

The cp predictive quantiles generally give reasonably good reliability, especially for sample sizes of ~100. The other predictions generally give poor reliability.

#### Value

A plot showing 9 different reliability checks, and a list containing various outputs, including the probabilities shown in the plot.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),

808 reltest2\_cases

- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (lst\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

## **Examples**

```
set.seed(1)
# example 1
# -runs the default settings, which test reliability for the GEV distribution
reltest2(nrepeats=1)
```

reltest2\_cases

Cases

### **Description**

Cases

### Usage

```
reltest2_cases(model = "gev", nx = 50, params)
```

reltest2\_makeep 809

## **Arguments**

model which distribution to test. Possibles values are "gev", "gpd\_k1", "gev\_pred1".

nx length of training data params model parameters

## Value

Two integers

reltest2\_makeep Cases

# Description

Cases

## Usage

```
reltest2_makeep(model, pred1, tt0, params)
```

# Arguments

model which distribution to test. Possibles values are "gev", "gpd\_k1", "gev\_pred1".

pred1 quantile predictions

tt0 value of predictor vector

params model parameters

### Value

Vector

## Description

Plots 9 diagnostics related to predictive probability matching.

810 reltest2\_predict

### Usage

```
reltest2_plot(
  model,
  ntrials,
  nrepeats,
  nx,
  params,
  nmethods,
  alpha,
  freqexceeded,
  case
)
```

## **Arguments**

model which distribution to test. Possibles values are "gev", "gpd", "gev\_p1".

ntrials the number of trials o run. 5000 typically gives good results.

nrepeats the number of entire repeats of the test to run, to check for convergence

nx the length of the training data.

params values for the parameters for the specified distribution

nmethods the number of methods being tested alpha the values of alpha being tested

freqexceeded the exceedance counts

there are 3 cases (must be set to case=1 except for my testing)

### Value

Plots the results of reliability testing

## Description

Make prediction from one model

## Usage

```
reltest2_predict(model = "gev", xx, tt, n0, pp, params, case, nmethods)
```

reltest2\_simulate 811

## **Arguments**

model	<pre>which distribution to test. Possibles values are "exp", "pareto_k1", "halfnorm",   "norm", "lnorm", "gumbel", "frechet_k1", "weibull", "gev_k3", "logis",   "lst_k3", "cauchy", "norm_p1", "lnorm_p1", "logis_p1", "lst_k3p1", "gumbel_p1",   "norm_p12", "gev", "gpd", "gev_p1".</pre>
XX	training data
tt	predictor vector
n0	index for predictor vector
рр	probabilities to predict
params	model parameters
case	the case number: different models have different lists of methods

nmethods the number of methods: different models have different numbers of methods

## Value

Vector

reltest2\_simulate Random training data from one model

# Description

Random training data from one model

# Usage

```
reltest2_simulate(model = "gev", nx = 50, tt, params)
```

# Arguments

model which distribution to test. Possibles values are "gev", "gpd\_k1", "gev\_pred1".

nx the length of the training data.

tt the predictor

params values for the parameters for the specified distribution

## Value

812 reltest\_makemaxep

-			
re	test	mal	ceen

Calculate EP from one model

# Description

Calculate EP from one model

## Usage

```
reltest_makeep(model, pred1, tt0, tt10, tt20, tt30, params)
```

# Arguments

model	which distribution to test. Possibles values are "exp", "pareto_k2", "halfnorm", "unif", "norm", "norm_dmgs", "gnorm_k3", "lnorm", "lnorm_dmgs", "logis", "lst_k3", "cauchy", "gumbel", "frechet_k1", "weibull", "gev_k3", "exp_p1", "pareto_p1k2", "norm_p1", "lnorm_p1", "logis_p1", "lst_p1k3", "cauchy_p1", "gumbel_p1", "frechet_p2k1", "weibull_p2", "gev_p1k3", "norm_p12", "lst_p12k3", "gamma", "invgamma", "invgauss", "gev", "gpd_k1", "gev_p1". "gev_p12". "gev_p123".
pred1	quantile predictions
tt0	value of the predictor
tt10	value of predictor 1
tt20	value of predictor 2
tt30	value of predictor 3
params	the model parameters

## Value

Vector

 $reltest\_makemaxep$ 

Calculate MaxEP from one model

# Description

Calculate MaxEP from one model

## Usage

```
reltest_makemaxep(model, ml_max, tt0, tt10, tt20, tt30, params)
```

reltest\_params 813

# Arguments

model	which distribution to test. Possibles values are "gev", "gpd_k1", "gev_p "gev_p12". "gev_p123".	ე1".
ml_max	predicted max value	
tt0	value of the predictor	
tt10	value of predictor 1	
tt20	value of predictor 2	
tt30	value of predictor 3	
params	the model parameters	

## Value

Vector

 $reltest\_params$ 

Set default params for the chosen model

# Description

Set default params for the chosen model

# Usage

```
reltest_params(model = "exp", params)
```

# Arguments

model	which distribution to test. Possibles values are "exp", "pareto_k2", "halfnorm", "unif", "norm", "norm_dmgs", "gnorm_k3", "lnorm", "lnorm_dmgs", "logis", "lst_k3", "cauchy", "gumbel", "frechet_k1", "weibull", "gev_k3", "exp_p1", "pareto_p1k2", "norm_p1", "lnorm_p1", "logis_p1", "lst_p1k3", "cauchy_p1", "gumbel_p1", "frechet_p2k1", "weibull_p2", "gev_p1k3", "norm_p12", "lst_p12k3", "gamma", "invgamma", "invgauss", "gev", "gpd_k1", "gev_p1". "gev_p12". "gev_p123".
params	values for the parameters for the specified distribution

## Value

814 reltest\_predict

reltest\_predict

 ${\it Make prediction from one model}$ 

### **Description**

Make prediction from one model

## Usage

```
reltest_predict(
 model,
  ХХ,
  tt,
  tt1,
  tt2,
  tt3,
  n0,
  n10,
 n20,
  n30,
 pp,
 params,
  dmgs = TRUE,
  debug = FALSE,
  aderivs = TRUE,
 unbiasedv = FALSE,
 pwm = FALSE,
 minxi = -10,
 maxxi = 10
)
```

## **Arguments**

```
which distribution to test. Possibles values are "exp", "pareto_k2", "halfnorm",
model
                  "unif", "norm", "norm_dmgs", "gnorm_k3", "lnorm", "lnorm_dmgs", "logis",
                  "lst_k3", "cauchy", "gumbel", "frechet_k1", "weibull", "gev_k3", "exp_p1",
                  "pareto_p1k2", "norm_p1", "lnorm_p1", "logis_p1", "lst_p1k3", "cauchy_p1",
                  "gumbel_p1", "frechet_p2k1", "weibull_p2", "exp_p1k4", "norm_p12", "lst_p12k3",
                  "gamma", "invgamma", "invgauss", "gev", "gpd_k1", "gev_p1". "gev_p12".
                  "gev_p123".
                 training data
ХΧ
tt
                 predictor vector
tt1
                 predictor vector 1
tt2
                 predictor vector 2
tt3
                 predictor vector 3
```

reltest\_simulate 815

n0	index for predictor vector
n10	index for predictor vector 1
n20	index for predictor vector 2
n30	index for predictor vector 2
pp	probabilites at which to make quantile predictions
params	model parameters
dmgs	flag for whether to run dmgs calculations or not
debug	flag for turning debug messages on
aderivs	a logical for whether to use analytic derivatives (instead of numerical)
unbiasedv	a logical for whether to use the unbiased variance instead of maxlik (for the normal)
pwm	a logical for whether to use PWM instead of maxlik (for the GEV)
minxi	minimum value for EVT shape parameter

# Value

Two vectors

maxxi

 ${\tt reltest\_simulate}$ 

Random training data from one model

maximum value for EVT shape parameter

# Description

Random training data from one model

## Usage

```
reltest_simulate(
  model = "exp",
  nx = 20,
  tt,
  tt1,
  tt2,
  tt3,
  params,
  minxi = -10,
  maxxi = -10
)
```

816 rgev\_minmax

## Arguments

model	which distribution to test. Possibles values are "exp", "pareto_k2", "halfnorm", "unif", "norm", "norm_dmgs", "gnorm_k3", "lnorm", "lnorm_dmgs", "logis", "lst_k3", "cauchy", "gumbel", "frechet_k1", "weibull", "gev_k3", "exp_p1", "pareto_p1k2", "norm_p1", "lnorm_p1", "logis_p1", "lst_p1k3", "cauchy_p1", "gumbel_p1", "frechet_p2k1", "weibull_p2", "gev_p1k3", "norm_p12", "lst_p12k3", "gamma", "invgamma", "invgauss", "gev", "gpd_k1", "gev_p1". "gev_p12". "gev_p123".
nx	the length of the training data to use.
tt	predictor vector
tt1	predictor vector 1
tt2	predictor vector 2
tt3	predictor vector 2
params	values for the parameters for the specified distribution
minxi	minimum value for EVT shape parameter

## Value

Vector

maxxi

rgev_minmax	rgev but with maxlik xi guaranteed within bounds
-------------	--------------------------------------------------

## Description

rgev but with maxlik xi guaranteed within bounds

# Usage

```
rgev_minmax(nx, mu = 0, sigma = 1, xi = 0, minxi = -1, maxxi = 1)
```

maximum value for EVT shape parameter

## Arguments

nx	length of training data
mu	the location parameter of the distribution
sigma	the sigma parameter of the distribution
xi	the shape parameter of the distribution
minxi	minimum value of shape parameter xi
maxxi	maximum value of shape parameter xi

## Value

rgev\_p123\_minmax 817

rgev\_p123\_minmax

rgev for gev\_p123 but with maxlik xi within bounds

# Description

rgev for gev\_p123 but with maxlik xi within bounds

## Usage

```
rgev_p123_minmax(
    nx,
    mu = 0,
    sigma = 1,
    xi = 0,
    t1,
    t2,
    t3,
    minxi = -0.45,
    maxxi = 0.45,
    centering = TRUE
)
```

## Arguments

nx	length of training data
mu	the location parameter of the distribution
sigma	the sigma parameter of the distribution
xi	the shape parameter of the distribution
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
t3	a vector of predictors for the shape
minxi	minimum value of shape parameter xi
maxxi	maximum value of shape parameter xi
centering	indicates whether the routine should center the data or not

## Value

818 rgev\_p12\_minmax

rgev\_p12\_minmax

rgev for gev\_p12 but with maxlik xi within bounds

# Description

rgev for gev\_p12 but with maxlik xi within bounds

## Usage

```
rgev_p12_minmax(
    nx,
    mu = 0,
    sigma = 1,
    xi = 0,
    t1,
    t2,
    minxi = -0.45,
    maxxi = 0.45,
    centering = TRUE
)
```

## Arguments

nx	length of training data
mu	the location parameter of the distribution
sigma	the sigma parameter of the distribution
xi	the shape parameter of the distribution
t1	a vector of predictors for the mean
t2	a vector of predictors for the sd
minxi	minimum value of shape parameter xi
maxxi	maximum value of shape parameter xi
centering	indicates whether the routine should center the data or not

## Value

rgev\_p1n\_minmax 819

rgev\_p1n\_minmax

rgev for gev\_p1n but with maxlik xi within bounds

# Description

rgev for gev\_p1n but with maxlik xi within bounds

## Usage

```
rgev_p1n_minmax(
    nx,
    mu = 0,
    sigma = 1,
    xi = 0,
    tt,
    minxi = -0.45,
    maxxi = 0.45,
    centering = TRUE
)
```

# Arguments

nx	length of training data
mu	the location parameter of the distribution
sigma	the sigma parameter of the distribution
xi	the shape parameter of the distribution
tt	a vector of predictors
minxi	minimum value of shape parameter xi
maxxi	maximum value of shape parameter xi
centering	indicates whether the routine should center the data or not

## Value

820 rgev\_p1\_minmax

rgev\_p1\_minmax

rgev for gev\_p1 but with maxlik xi within bounds

# Description

rgev for gev\_p1 but with maxlik xi within bounds

## Usage

```
rgev_p1_minmax(
    nx,
    mu = 0,
    sigma = 1,
    xi = 0,
    tt,
    minxi = -0.45,
    maxxi = 0.45,
    centering = TRUE
)
```

# Arguments

nx	length of training data
mu	the location parameter of the distribution
sigma	the sigma parameter of the distribution
xi	the shape parameter of the distribution
tt	a vector of predictors
minxi	minimum value of shape parameter xi
maxxi	maximum value of shape parameter xi
centering	indicates whether the routine should center the data or not

## Value

rgpd\_k1\_minmax 821

rond	k1	minmax	

rgpd for gpd\_k1 but with maxlik xi within bounds

## Description

rgpd for gpd\_k1 but with maxlik xi within bounds

## Usage

```
rgpd_k1_minmax(nx, kloc, sigma, xi, minxi = -0.45, maxxi = 0.45)
```

## Arguments

nx	length of training data
kloc	the known location parameter
sigma	the sigma parameter of the distribution
xi	the shape parameter of the distribution
minxi	minimum value of shape parameter xi
maxxi	maximum value of shape parameter xi

## Value

Vector

rhn	dmgs	cpmet	hod
1 11P		_cpilic c	1100

Generates a comment about the method

## Description

Generates a comment about the method

## Usage

```
rhp_dmgs_cpmethod()
```

## Value

String

822 testppm\_plot

rust\_pumethod

Generates a comment about the method

## Description

Generates a comment about the method

## Usage

```
rust_pumethod()
```

#### Value

String

testppm\_plot

Plotting routine for testppm

### **Description**

Plots 9 diagnostics related to predictive probability matching.

## Usage

```
testppm_plot(
  model,
  ntrials,
  nrepeats,
  nx,
  params,
  nmethods,
  alpha,
  freqexceeded
)
```

### **Arguments**

model which distribution to test. Possibles values are

ntrials the number of trials to run. 5000 typically gives good results.

nrepeats the number of entire repeats of the test to run, to check for convergence

nx the length of the training data.

params values for the parameters for the specified distribution

nmethods the number of methods being tested alpha the values of alpha being tested

freqexceeded the exceedance counts

#### Value

Plots the results of reliability testing

unif\_cp

Uniform Distribution Predictions Based on a Calibrating Prior

### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

### Usage

```
qunif_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    means = FALSE,
    debug = FALSE,
    aderivs = TRUE
)

runif_cp(n, x, mlcp = TRUE, debug = FALSE, aderivs = TRUE)

dunif_cp(x, y = x, debug = FALSE, aderivs = TRUE)

punif_cp(x, y = x, debug = FALSE, aderivs = TRUE)
```

### **Arguments**

X	a vector of training data values
p	a vector of probabilities at which to generate predictive quantiles
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
debug	logical for turning on debug messages
aderivs	(for code testing only) logical for whether to use analytic derivatives (instead of numerical). By default almost all models now use analytical derivatives.
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
У	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).

• cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

### **Details of the Model**

The uniform distribution has probability density function

$$f(x; min, max) = \frac{1}{max - min}$$

and zero otherwise, where  $min \le x \le max$  is the random variable and min, max are the parameters.

The calibrating prior is given by the right Haar prior, which is

$$\pi(\lambda) \propto \frac{1}{max - min}$$

as given in Jewson et al. (2025).

### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\* optionally returns the following:

If rust=TRUE:

 ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

## **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

### **Details (analytic integration)**

For this model, the Bayesian prediction equation is integrated analytically.

### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),

- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (1st\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

### **Examples**

```
#
# example 1
x=fitdistcp::d025unif_example_data_v1
cat("length(x)=",length(x),"\n")
p=c(1:9)/10
q=qunif_cp(x,p)
xmin=min(q$m1_quantiles,q$cp_quantiles);
xmax=max(q$m1_quantiles,q$cp_quantiles);
plot(q$m1_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qunif_cp)",
main="unif: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
```

weibull\_cp

Weibull Distribution Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qweibull_cp(
    x,
    p = seq(0.1, 0.9, 0.1),
    means = FALSE,
    waicscores = FALSE,
    logscores = FALSE,
    dmgs = TRUE,
    rust = FALSE,
    nrust = 1e+05,
    debug = FALSE
)

rweibull_cp(n, x, rust = FALSE, mlcp = TRUE, debug = FALSE)

dweibull_cp(x, y = x, rust = FALSE, nrust = 1000, debug = FALSE)
```

```
pweibull_cp(x, y = x, rust = FALSE, nrust = 1000, debug = FALSE)
tweibull_cp(n, x, debug = FALSE)
```

#### **Arguments**

X	a vector of training data values
p	a vector of probabilities at which to generate predictive quantiles
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave-one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
debug	logical for turning on debug messages
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Weibull distribution has exceedance distribution function

$$S(x; k, \sigma) = \exp\left(-\left(\frac{x}{\sigma}\right)^k\right)$$

where  $x \ge 0$  is the random variable and  $k > 0, \sigma > 0$  are the parameters.

The calibrating prior is given by the right Haar prior, which is

$$\pi(k,\sigma) \propto \frac{1}{k\sigma}$$

as given in Jewson et al. (2025).

#### **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

#### If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

#### If means=TRUF:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

#### If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### **Details (RUST)**

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson < stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- · Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),

- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),
- t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),
- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

weibull\_f1fa 835

## **Examples**

```
#
# example 1
x=fitdistcp::d052weibull_example_data_v1
p=c(1:9)/10
q=qweibull_cp(x,p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),sub="(from qweibull_cp)",
main="Weibull: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

weibull\_f1fa

The first derivative of the density

## **Description**

The first derivative of the density

## Usage

```
weibull_f1fa(x, v1, v2)
```

# **Arguments**

x a vector of training data values

v1 first parameter

v2 second parameter

#### Value

Vector

weibull\_f2fa

The second derivative of the density

## **Description**

The second derivative of the density

```
weibull_f2fa(x, v1, v2)
```

836 weibull\_fdd

#### **Arguments**

x a vector of training data value	ies
-----------------------------------	-----

v1 first parameter v2 second parameter

#### Value

Matrix

weibull\_fd

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

#### Usage

```
weibull_fd(x, v1, v2)
```

## **Arguments**

x a vector of training data values

v1 first parameterv2 second parameter

# Value

Vector

weibull\_fdd Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## **Description**

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
weibull_fdd(x, v1, v2)
```

weibull\_ldda 837

# Arguments

x a vector of training data values

v1 first parameter

v2 second parameter

## Value

Matrix

weibull\_ldda

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

# Usage

```
weibull_ldda(x, v1, v2)
```

# **Arguments**

x a vector of training data values

v1 first parameter

v2 second parameter

## Value

Matrix

weibull\_lddda

The third derivative of the normalized log-likelihood

# Description

The third derivative of the normalized log-likelihood

```
weibull_lddda(x, v1, v2)
```

838 weibull\_logfdd

#### **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

# Value

3d array

weibull\_logf

Logf for RUST

# Description

Logf for RUST

#### Usage

```
weibull_logf(params, x)
```

## **Arguments**

params model parameters for calculating logf x a vector of training data values

## Value

Scalar value.

weibull\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

```
weibull_logfdd(x, v1, v2)
```

weibull\_logfddd 839

# **Arguments**

Χ	a vector of training data values
---	----------------------------------

v1 first parameterv2 second parameter

#### Value

Matrix

weibull\_logfddd

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
weibull_logfddd(x, v1, v2)
```

## **Arguments**

x a vector of training data values

v1 first parameterv2 second parameter

#### Value

3d array

weibull\_loglik

log-likelihood function

# Description

log-likelihood function

```
weibull_loglik(vv, x)
```

840 weibull\_means

#### **Arguments**

vv parameters

x a vector of training data values

#### Value

Scalar

weibull\_logscores

Log scores for MLE and RHP predictions calculated using leave-one-

# **Description**

Log scores for MLE and RHP predictions calculated using leave-one-out

## Usage

```
weibull_logscores(logscores, x)
```

## **Arguments**

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data values

#### Value

Two scalars

weibull\_means

MLE and RHP predictive means

# Description

MLE and RHP predictive means

```
weibull_means(means, ml_params, lddi, lddd, lambdad_rhp, nx, dim = 2)
```

weibull\_mu1fa 841

## **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

#### Value

Two scalars

weibull\_mu1fa

Minus the first derivative of the cdf, at alpha

# Description

Minus the first derivative of the cdf, at alpha

## Usage

```
weibull_mu1fa(alpha, v1, v2)
```

# Arguments

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

## Value

Vector

842 weibull\_p1fa

weibull\_mu2fa

Minus the second derivative of the cdf, at alpha

# Description

Minus the second derivative of the cdf, at alpha

# Usage

```
weibull_mu2fa(alpha, v1, v2)
```

# **Arguments**

alpha a vector of values of alpha (one minus probability)

v1 first parameter v2 second parameter

## Value

Matrix

weibull\_p1fa

The first derivative of the cdf

# Description

The first derivative of the cdf

# Usage

```
weibull_p1fa(x, v1, v2)
```

## **Arguments**

x a vector of training data values

v1 first parameter v2 second parameter

# Value

Vector

weibull\_p2fa 843

weibull_p2fa	wei	bu]	L1_	p2f	a
--------------	-----	-----	-----	-----	---

The second derivative of the cdf

# Description

The second derivative of the cdf

#### Usage

```
weibull_p2fa(x, v1, v2)
```

#### **Arguments**

x a vector of training data values

v1 first parameter

v2 second parameter

#### Value

Matrix

weibull\_p2\_cp

weibull Distribution with a Predictor on the Scale Parameter, Predictions Based on a Calibrating Prior

#### **Description**

The fitdistcp package contains functions that generate predictive distributions for various statistical models, with and without parameter uncertainty. Parameter uncertainty is included by using Bayesian prediction with a type of objective prior known as a calibrating prior. Calibrating priors are chosen to give predictions that give good reliability (i.e., are well calibrated), for any underlying true parameter values.

There are five functions for each model, each of which uses training data x. For model \*\*\*\* the five functions are as follows:

- q\*\*\*\*\_cp returns predictive quantiles at the specified probabilities p, and various other diagnostics.
- r\*\*\*\_cp returns n random deviates from the predictive distribution.
- d\*\*\*\*\_cp returns the predictive density function at the specified values y
- p\*\*\*\*\_cp returns the predictive distribution function at the specified values y
- t\*\*\*\*\_cp returns n random deviates from the posterior distribution of the model parameters.

The q, r, d, p routines return two sets of results, one based on maximum likelihood, and the other based on a calibrating prior. The prior used depends on the model, and is given under Details below.

Where possible, the Bayesian prediction integral is solved analytically. Otherwise, DMGS asymptotic expansions are used. Optionally, a third set of results is returned that integrates the prediction integral by sampling the parameter posterior distribution using the RUST rejection sampling algorithm.

```
qweibull_p2_cp(
 х,
  t,
  t0 = NA,
 n0 = NA
 p = seq(0.1, 0.9, 0.1),
 means = FALSE,
 waicscores = FALSE,
 logscores = FALSE,
 dmgs = TRUE,
 rust = FALSE,
 nrust = 1e+05,
 predictordata = TRUE,
 centering = TRUE,
  debug = FALSE
)
rweibull_p2_cp(
  n,
 Х,
  t,
  t0 = NA,
 n0 = NA,
 rust = FALSE,
 mlcp = TRUE,
 debug = FALSE
)
dweibull_p2_cp(
  Х,
  t,
  t0 = NA,
 n0 = NA,
 y = x,
 rust = FALSE,
 nrust = 1000,
 centering = TRUE,
  debug = FALSE
)
```

```
pweibull_p2_cp(
    x,
    t,
    t0 = NA,
    n0 = NA,
    y = x,
    rust = FALSE,
    nrust = 1000,
    centering = TRUE,
    debug = FALSE
)

tweibull_p2_cp(n, x, t, debug = FALSE)
```

# Arguments

x	a vector of training data values
t	a vector of predictors, such that length(t)=length(x)
t0	a single value of the predictor (specify either t0 or n0 but not both)
n0	an index for the predictor (specify either t0 or n0 but not both)
р	a vector of probabilities at which to generate predictive quantiles
means	logical that indicates whether to run additional calculations and return analytical estimates for the distribution means (longer runtime)
waicscores	logical that indicates whether to run additional calculations and return estimates for the WAIC1 and WAIC2 scores (longer runtime)
logscores	logical that indicates whether to run additional calculations and return leave- one-out estimates of the log-score (much longer runtime, non-EVT models only)
dmgs	logical that indicates whether DMGS calculations should be run or not (longer run time)
rust	logical that indicates whether RUST-based posterior sampling calculations should be run or not (longer run time)
nrust	the number of posterior samples used in the RUST calculations
predictordata	logical that indicates whether predictordata should be calculated
centering	logical that indicates whether the predictor should be centered
debug	logical for turning on debug messages
n	the number of random samples required
mlcp	logical that indicates whether maxlik and parameter uncertainty calculations should be performed (turn off to speed up RUST)
у	a vector of values at which to calculate the density and distribution functions

#### Value

q\*\*\*\* returns a list containing at least the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_value: the value of the log-likelihood at the maximum.
- standard\_errors: estimates of the standard errors on the parameters, from the inverse observed information matrix.
- ml\_quantiles: quantiles calculated using maximum likelihood.
- cp\_quantiles: predictive quantiles calculated using a calibrating prior.
- maic: the AIC score for the maximum likelihood model, times -1/2.
- cp\_method: a comment about the method used to generate the cp prediction.

For models with predictors, q\*\*\*\* additionally returns:

- predictedparameter: the estimated value for parameter, as a function of the predictor.
- adjustedx: the detrended values of x

r\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_deviates: random deviates calculated using maximum likelihood.
- cp\_deviates: predictive random deviates calculated using a calibrating prior.
- cp\_method: a comment about the method used to generate the cp prediction.

d\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_pdf: density function from maximum likelihood.
- cp\_pdf: predictive density function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

p\*\*\* returns a list containing the following:

- ml\_params: maximum likelihood estimates for the parameters.
- ml\_cdf: distribution function from maximum likelihood.
- cp\_cdf: predictive distribution function calculated using a calibrating prior (not available in EVT routines, for mathematical reasons, unless using RUST).
- cp\_method: a comment about the method used to generate the cp prediction.

t\*\*\* returns a list containing the following:

• theta\_samples: random samples from the parameter posterior.

#### **Details of the Model**

The Weibull distribution with predictor on the scale parameter has exceedance distribution function

$$S(x; k, a, b) = \exp\left(-\left(\frac{x}{\sigma(a, b)}\right)^k\right)$$

where  $x \ge 0$  is the random variable, k > 0 is the shape parameter and  $\sigma = e^{a+bt}$  is the scale parameter, modelled as a function of parameters a, b and predictor t.

The calibrating prior is given by the right Haar prior, which is

$$\pi(k,\sigma) \propto \frac{1}{k}$$

as given in Jewson et al. (2025).

# **Optional Return Values**

q\*\*\*\* optionally returns the following:

If rust=TRUE:

ru\_quantiles: predictive quantiles calculated using a calibrating prior, using posterior sampling with the RUST algorithm, based on inverting an empirical CDF based on nrust samples.

If waicscores=TRUE:

- waic1: the WAIC1 score for the calibrating prior model.
- waic2: the WAIC2 score for the calibrating prior model.

If logscores=TRUE:

- ml\_oos\_logscore: the leave-one-out logscore for the maximum likelihood prediction (not available in EVT routines, for mathematical reasons)
- cp\_oos\_logscore: the leave-one-out logscore for the parameter uncertainty model available in EVT routines, for mathematical reasons)

If means=TRUE:

- ml\_mean: analytic estimate of the mean of the MLE predictive distribution, where possible
- cp\_mean: analytic estimate of the mean of the calibrating prior predictive distribution, where mathematically possible. Can be compared with the mean estimated from random deviates.

r\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_deviates: nrust predictive random deviatives calculated using a calibrating prior, using posterior sampling with RUST.

d\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_pdf: predictive density calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust density functions.

p\*\*\*\* optionally returns the following:

If rust=TRUE:

• ru\_cdf: predictive probability calculated using a calibrating prior, using posterior sampling with RUST, averaging over nrust distribution functions.

Selecting these additional outputs increases runtime. They are optional so that runtime for the basic outputs is minimised. This facilitates repeated experiments that evaluate reliability over many thousands of repeats.

#### **Details (homogeneous models)**

This model is a homogeneous model, and the cp results are based on the right Haar prior. For homogeneous models (models with sharply transitive transformation groups), a Bayesian prediction based on the right Haar prior gives exact reliability, as shown by Severini et al. (2002), even when the true parameters are unknown. This means that probabilities in the prediction will correspond to frequencies of future outcomes in repeated trials (if the model is correct).

Maximum likelihood prediction does not give reliable predictions, even when the model is correct, because it does not account for parameter uncertainty. In particular, maximum likelihood predictions typically underestimate the tail in repeated trials.

The reliability of the maximum likelihood and the calibrating prior predictive quantiles produced by the q\*\*\*\*\_cp routines in fitdistcp can be quantified using repeated simulations with the routine reltest.

#### **Details (DMGS integration)**

For this model, the Bayesian prediction equation cannot be solved analytically, and is approximated using the DMGS asymptotic expansions given by Datta et al. (2000). This approximation seems to work well for medium and large sample sizes, but may not work well for small sample sizes (e.g., <20 data points). For small sample sizes, it may be preferable to use posterior sampling by setting rust=TRUE and looking at the ru outputs. The performance for any sample size, in terms of reliability, can be tested using reltest.

#### Details (RUST)

The Bayesian prediction equation can also be integrated using ratio-of-uniforms-sampling-with-transformation (RUST), using the option rust=TRUE. fitdistcp then calls Paul Northrop's rust package (Northrop, 2023). The RUST calculations are slower than the DMGS calculations.

For small sample sizes (e.g., n<20), and the very extreme tail, the DMGS approximation is somewhat poor (although always better than maximum likelihood) and it may be better to use RUST. For medium sample sizes (30+), DMGS is reasonably accurate, except for the very far tail.

It is advisable to check the RUST results for convergence versus the number of RUST samples.

It may be interesting to compare the DMGS and RUST results.

#### Author(s)

Stephen Jewson <stephen.jewson@gmail.com>

#### References

If you use this package, we would be grateful if you would cite the following reference, which gives the various calibrating priors, and tests them for reliability:

 Jewson S., Sweeting T. and Jewson L. (2025): Reducing Reliability Bias in Assessments of Extreme Weather Risk using Calibrating Priors; ASCMO Advances in Statistical Climatology, Meteorology and Oceanography), https://ascmo.copernicus.org/articles/11/1/ 2025/.

#### See Also

An introduction to fitdistcp, with more examples, is given on this webpage.

The fitdistcp package currently includes the following models (in alphabetical order):

- Cauchy (cauchy),
- Cauchy with linear predictor on the mean (cauchy\_p1),
- Exponential (exp),
- Exponential with log-linear predictor on the scale (exp\_p1),
- Frechet with known location parameter (frechet\_k1),
- Frechet with log-linear predictor on the scale and known location parameter (frechet\_p2k1),
- Gamma (gamma),
- Generalized normal (gnorm),
- GEV (gev),
- GEV with linear predictor on the location (gev\_p1),
- GEV with 1-3 linear predictors on the location (gev\_p1n),
- GEV with linear predictor on the location and log-linear prediction on the scale (gev\_p12),
- GEV with linear predictor on the location, log-linear prediction on the scale, and linear predictor on the shape (gev\_p123),
- GEV with linear predictor on the location and known shape (gev\_p1k3),
- GEV with known shape (gev\_k3),
- GPD with known location (gpd\_k1),
- Gumbel (gumbel),
- Gumbel with linear predictor on the mean(gumbel\_p1),
- Half-normal (halfnorm),
- Inverse gamma (invgamma),
- Inverse Gaussian (invgauss),
- t distribution with unknown location and scale and known DoF (1st\_k3),

• t distribution with unknown location and scale, linear predictor on the location, and known DoF (lst\_p1k3),

- Logistic (logis),
- Logistic with linear predictor on the location (logis\_p1),
- Log-normal (lnorm),
- Log-normal with linear predictor on the location (lnorm\_p1),
- Normal (norm),
- Normal with predictor on the mean (norm\_p1),
- Normal with predictors on the mean and sd (norm\_p12),
- Pareto with known scale (pareto\_k2),
- Pareto with log-linear predictor on the shape and known scale (pareto\_p1k2),
- Uniform (unif),
- Weibull (weibull),
- Weibull with linear predictor on the scale (weibull\_p2),

The level of predictive probability matching achieved by the maximum likelihood and calibrating prior quantiles, for any model, sample size and true parameter values, can be demonstrated using the routine reltest.

Model selection among models can be demonstrated using the routines ms\_flat\_1tail, ms\_flat\_2tail, ms\_predictors\_1tail, and ms\_predictors\_2tail,

## **Examples**

```
#
# example 1
x=fitdistcp::d073weibull_p2_example_data_v1_x
tt=fitdistcp::d073weibull_p2_example_data_v1_t
p=c(1:9)/10
n0=10
q=qweibull_p2_cp(x,tt,n0=n0,p=p,rust=TRUE,nrust=1000)
xmin=min(q$ml_quantiles,q$cp_quantiles);
xmax=max(q$ml_quantiles,q$cp_quantiles);
plot(q$ml_quantiles,p,xlab="quantile estimates",xlim=c(xmin,xmax),
sub="(from qweibull_p2_cp)",
main="Weibull w/ p2: quantile estimates");
points(q$cp_quantiles,p,col="red",lwd=2)
points(q$ru_quantiles,p,col="blue")
```

weibull\_p2\_f1fa 851

	1 7	٦.	- ^	C1	C -
wei	pul	. L_I	02_	ŤΙ	та

The first derivative of the density for DMGS

## **Description**

The first derivative of the density for DMGS

# Usage

```
weibull_p2_f1fa(x, t0, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t0	a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameterv2 second parameterv3 third parameter

## Value

Vector

weibull\_p2\_f1fw

The first derivative of the density for WAIC

# Description

The first derivative of the density for WAIC

## Usage

```
weibull_p2_f1fw(x, t, v1, v2, v3)
```

# Arguments

Χ	a vector of training data values
t	a vector or matrix of predictors

v1 first parameter v2 second parameter v3 third parameter

## Value

Vector

weibull\_p2\_f2fw

weibull\_p2\_f2fa

The second derivative of the density for DMGS

## **Description**

The second derivative of the density for DMGS

## Usage

```
weibull_p2_f2fa(x, t0, v1, v2, v3)
```

## **Arguments**

x a vector of training data values

t0 a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

weibull\_p2\_f2fw

The second derivative of the density for WAIC

# Description

The second derivative of the density for WAIC

## Usage

```
weibull_p2_f2fw(x, t, v1, v2, v3)
```

## **Arguments**

x a vector of training data valuest a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

weibull\_p2\_fd 853

weibull_p2_fd	First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
weibull_p2_fd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

#### Value

Vector

weibull_p2_fdd	Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
weibull_p2_fdd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter
v3	third parameter

854 weibull\_p2\_lddda

## Value

Matrix

weibull\_p2\_ldda

The second derivative of the normalized log-likelihood

# Description

The second derivative of the normalized log-likelihood

## Usage

```
weibull_p2_ldda(x, t, v1, v2, v3)
```

# Arguments

x a vector of training data values
t a vector or matrix of predictors
v1 first parameter

v2 second parameter v3 third parameter

## Value

Matrix

weibull\_p2\_lddda

The third derivative of the normalized log-likelihood

# **Description**

The third derivative of the normalized log-likelihood

## Usage

```
weibull_p2_lddda(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

weibull\_p2\_logf 855

## Value

3d array

 $weibull\_p2\_logf$ 

Logf for RUST

# Description

Logf for RUST

## Usage

```
weibull_p2_logf(params, x, t)
```

## **Arguments**

params model parameters for calculating logf
x a vector of training data values
t a vector or matrix of predictors

## Value

Scalar value.

weibull\_p2\_logfdd

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
weibull_p2_logfdd(x, t, v1, v2, v3)
```

# Arguments

Х	a vector of training data values
t	a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

weibull\_p2\_loglik

## Value

Matrix

# Description

Third derivative of the log density Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
weibull_p2_logfddd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

#### Value

3d array

# Description

observed log-likelihood function

## Usage

```
weibull_p2_loglik(vv, x, t)
```

# Arguments

parameters

x a vector of training data valuest a vector or matrix of predictors

weibull\_p2\_logscores 857

## Value

Scalar

# Description

Log scores for MLE and RHP predictions calculated using leave-one-out

## Usage

```
weibull_p2_logscores(logscores, x, t)
```

# **Arguments**

logical that indicates whether to return leave-one-out estimates estimates of the

log-score (much longer runtime)

x a vector of training data values

t a vector or matrix of predictors

# Value

Two scalars

# Description

weibull distribution: RHP mean

```
weibull_p2_means(means, t0, ml_params, lddi, lddd, lambdad_rhp, nx, dim)
```

weibull\_p2\_mu1fa

# **Arguments**

means logical that indicates whether to return analytical estimates for the distribution

means (longer runtime)

to a single value of the predictor (specify either to or no but not both)

ml\_params parameters

lddi inverse observed information matrixlddd third derivative of log-likelihoodlambdad\_rhp derivative of the log RHP prior

nx length of training data dim number of parameters

#### Value

Two scalars

weibull\_p2\_mu1fa

Minus the first derivative of the cdf, at alpha

#### **Description**

Minus the first derivative of the cdf, at alpha

#### Usage

```
weibull_p2_mu1fa(alpha, t0, v1, v2, v3)
```

# Arguments

alpha a vector of values of alpha (one minus probability)

t0 a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameterv2 second parameterv3 third parameter

# Value

Vector

weibull\_p2\_mu2fa 859

weibu]	11	n2	mu2fa
werbu.	$LL_{-}$	$DZ_{-}$	_IIIUZT d

Minus the second derivative of the cdf, at alpha

## **Description**

Minus the second derivative of the cdf, at alpha

## Usage

```
weibull_p2_mu2fa(alpha, t0, v1, v2, v3)
```

## **Arguments**

alpha a vector of values of alpha (one minus probability)
to a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameterv2 second parameterv3 third parameter

## Value

Matrix

weibull\_p2\_p1fa

The first derivative of the cdf

# **Description**

The first derivative of the cdf

## Usage

```
weibull_p2_p1fa(x, t0, v1, v2, v3)
```

# Arguments

Χ	a vector of training data values
---	----------------------------------

to a single value of the predictor (specify either to or no but not both)

v1 first parameter v2 second parameter v3 third parameter

#### Value

Vector

			_	
WE.	i bu	ш	n2	n2fa

The second derivative of the cdf

# Description

The second derivative of the cdf

## Usage

```
weibull_p2_p2fa(x, t0, v1, v2, v3)
```

# Arguments

x a vector of training data value
-----------------------------------

t0 a single value of the predictor (specify either t0 or n0 but not both)

v1 first parameterv2 second parameterv3 third parameter

#### Value

Matrix

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
weibull_p2_pd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors

v1 first parameterv2 second parameterv3 third parameter

weibull\_p2\_pdd 861

# Value

Vector

weibull_p2_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()
	by Andrew Clausen and Serguei Sokol

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
weibull_p2_pdd(x, t, v1, v2, v3)
```

# Arguments

X	a vector of training data values
t	a vector or matrix of predictors
v1	first parameter
v2	second parameter

third parameter

# Value

Matrix

v3

```
weibull\_p2\_predictordata
```

Predicted Parameter and Generalized Residuals

# Description

Predicted Parameter and Generalized Residuals

```
weibull_p2_predictordata(predictordata, x, t, t0, params)
```

862 weibull\_p2\_waic

#### **Arguments**

predictordata logical that indicates whether to calculate and return predictordata

x a vector of training data valuest a vector or matrix of predictors

t0 a single value of the predictor (specify either t0 or n0 but not both)

params model parameters for calculating logf

#### Value

Two vectors

weibull\_p2\_waic
Waic

## **Description**

Waic

## Usage

```
weibull_p2_waic(waicscores, x, t, v1hat, v2hat, v3hat, lddi, lddd, lambdad)
```

## **Arguments**

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data valuest a vector or matrix of predictors

v1hat first parameter
v2hat second parameter
v3hat third parameter

1ddi inverse observed information matrix1ddd third derivative of log-likelihood

lambdad derivative of the log prior

#### Value

Two numeric values.

weibull\_pd 863

weibull_pd	First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol
	Thater Clauser and serguet solor

# Description

First derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

# Usage

```
weibull_pd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

## Value

Vector

weibull_pdd	Second derivative of the cdf Created by Stephen Jewson using Deriv()	
	by Andrew Clausen and Serguei Sokol	

# Description

Second derivative of the cdf Created by Stephen Jewson using Deriv() by Andrew Clausen and Serguei Sokol

## Usage

```
weibull_pdd(x, v1, v2)
```

# Arguments

x a vector of training data values

v1 first parameter v2 second parameter

#### Value

Matrix

864 weibull\_waic

weibull_waic	Waic for RUST		
--------------	---------------	--	--

# Description

Waic for RUST

# Usage

```
weibull_waic(waicscores, x, v1hat, v2hat, lddi, lddd, lambdad)
```

# Arguments

waicscores logical that indicates whether to return estimates for the waic1 and waic2 scores

(longer runtime)

x a vector of training data values

v1hat first parameter v2hat second parameter

lddi inverse observed information matrixlddd third derivative of log-likelihood

lambdad derivative of the log prior

#### Value

Two numeric values.

## **Index**

adhoc_dmgs_cpmethod, 24	cauchy_p1_1mnp, 59
analytic_cpmethod, 24	cauchy_p1_logf, 59
	cauchy_p1_logfdd, 60
<pre>bayesian_dq_4terms_v1, 24</pre>	cauchy_p1_logfddd, 61
	cauchy_p1_loglik, 61
calc_revert2ml, 25	cauchy_p1_logscores, 62
cauchy_cp, 26	cauchy_p1_means, 62
cauchy_f1f, 33	cauchy_p1_mu1f, 63
cauchy_f1fa, 33	cauchy_p1_mu2f, 64
cauchy_f2f, 34	cauchy_p1_p1f, 64
cauchy_f2fa, 34	cauchy_p1_p2f, 65
cauchy_fd, 35	cauchy_p1_predictordata,66
cauchy_fdd, 35	cauchy_p1_waic,66
cauchy_ldd, 36	cauchy_p1f, 43
cauchy_ldda, 36	cauchy_p2f, 67
cauchy_lddd, 37	cauchy_waic,68
cauchy_lddda, 37	crhpflat_dmgs_cpmethod,69
cauchy_lmn, 38	
cauchy_lmnp, 38	d010exp_example_data_v1, 69
cauchy_logf, 39	d011pareto_k2_example_data_v1,69
cauchy_logfdd, 40	d020halfnorm_example_data_v1, 69
cauchy_logfddd, 40	d025unif_example_data_v1,70
cauchy_loglik, 41	d030norm_example_data_v1,70
cauchy_logscores, 41	d031norm_dmgs_example_data_v1, 70
cauchy_mu1f, 42	d032gnorm_k3_example_data_v1,70
cauchy_mu2f, 42	d035lnorm_example_data_v1,70
cauchy_p1_cp, 43	d036lnorm_dmgs_example_data_v1,71
cauchy_p1_f1f, 51	d040logis_example_data_v1,71
cauchy_p1_f1fa, 52	d0411st_k3_example_data_v1, 71
cauchy_p1_f1fw, 52	d042cauchy_example_data_v1, 71
cauchy_p1_f2f, 53	d050gumbel_example_data_v1, 71
cauchy_p1_f2fa, 53	d051frechet_k1_example_data_v1,72
cauchy_p1_f2fw, 54	d052weibull_example_data_v1,72
cauchy_p1_fd, 54	d053gev_k3_example_data_v1,72
cauchy_p1_fdd, 55	d055exp_p1_example_data_v1_t, 72
cauchy_p1_ldd, 55	d055exp_p1_example_data_v1_x, 72
cauchy_p1_ldda, 56	d056pareto_p1k2_example_data_v1_t, 73
cauchy_p1_lddd, 57	d056pareto_p1k2_example_data_v1_x, 73
cauchy_p1_lddda, 57	d060norm_p1_example_data_v1_t, 73
cauchy_p1_1mn, 58	d060norm_p1_example_data_v1_x, 73

d061lnorm_p1_example_data_v1_t, 73	dexp_cp (exp_cp), 114
d061lnorm_p1_example_data_v1_x,74	dexp_p1, 86
d062logis_p1_example_data_v1_t, 74	dexp_p1_cp (exp_p1_cp), 126
d062logis_p1_example_data_v1_x, 74	dexp_p1sub, 86
d063lst_p1k3_example_data_v1_t, 74	dexpsub, 85
d063lst_p1k3_example_data_v1_x, 74	dfrechet_k1_cp (frechet_k1_cp), 148
d064cauchy_p1_example_data_v1_t, 75	dfrechet_p2k1, 87
d064cauchy_p1_example_data_v1_x, 75	dfrechet_p2k1_cp (frechet_p2k1_cp), 164
d070gumbel_p1_example_data_v1_t, 75	dfrechet_p2k1sub, 88
d070gumbel_p1_example_data_v1_x, 75	dfrechetsub, 87
d071frechet_p2k1_example_data_v1_t, 75	dgamma_cp (gamma_cp), 185
d071frechet_p2k1_example_data_v1_x, 76	dgammasub, 88
d072weibull_p1_example_data_v1_t, 76	dgev_cp (gev_cp), 207
d072weibull_p1_example_data_v1_x, 76	dgev_k3_cp (gev_k3_cp), 220
d073weibull_p2_example_data_v1_t, 76	dgev_k3sub, 89
d073weibull_p2_example_data_v1_x, 76	$dgev_p1, 90$
d074gev_p1k3_example_data_v1_t, 77	dgev_p12, 90
d074gev_p1k3_example_data_v1_x, 77	dgev_p123, 91
d080norm_p12_example_data_v1_t1,77	dgev_p123_cp (gev_p123_cp), 242
d080norm_p12_example_data_v1_t2,77	dgev_p123sub, 92
d080norm_p12_example_data_v1_x, 77	dgev_p12_cp (gev_p12_cp), 275
d0811st_p12k3_example_data_v1_t1,78	dgev_p12sub, 92
d0811st_p12k3_example_data_v1_t2, 78	dgev_p1_cp (gev_p1_cp), 358
d0811st_p12k3_example_data_v1_x, 78	dgev_p1k3, 93
d082weibull_p12_example_data_v1_t1,78	dgev_p1k3_cp (gev_p1k3_cp), 324
d082weibull_p12_example_data_v1_t2, 78	dgev_p1k3sub, 94
d082weibull_p12_example_data_v1_x, 79	dgev_p1n, 95
d100gamma_example_data_v1,79	dgev_p1n_cp (gev_p1n_cp), 344
d101invgamma_example_data_v1, 79	dgev_p1nsub, 95
d102invgauss_example_data_v1, 79	dgev_p1sub, 96
d105burr_example_data_v1,79	dgevsub, 89
d110gev_example_data_v1, 80	dgnorm_k3_cp (gnorm_k3_cp), 373
d120gpd_k1_example_data_v1, 80	dgnorm_k3sub, 96
d150gev_p1_example_data_v1_t, 80	dgpd_k1_cp (gpd_k1_cp), 400
d150gev_p1_example_data_v1_x, 80	dgpdsub, 97
d151gev_p12_example_data_v1_t, 80	dgumbel_cp (gumbel_cp), 418
d151gev_p12_example_data_v1_x, 81	dgumbel_p1, 98
d152gev_p123_example_data_v1_t, 81	dgumbel_p1_cp (gumbel_p1_cp), 432
d152gev_p123_example_data_v1_x, 81	dgumbel_p1sub, 99
dcauchy_cp (cauchy_cp), 26	dgumbelsub, 98
dcauchy_p1, 82	dhalfnorm_cp (halfnorm_cp), 453
dcauchy_p1_cp (cauchy_p1_cp), 43	dhalfnormsub, 99
dcauchy_p1sub, 82	dinvgamma_cp (invgamma_cp), 473
dcauchysub, 81	dinvgammasub, 100
deriv_copyfdd, 83	dinvgauss_cp (invgauss_cp), 492
deriv_copylda, 85	dinvgausssub, 100
deriv_copyldd, 84	dlnorm_cp (lnorm_cp), 513
deriv_copyldd, 85	dlnorm_dmgs_cp (lnorm_dmgs_cp), 519
uci iv_copyiuuu, oo	arnor iii_uiiiga_cp (rnor iii_uiiiga_cp), 319

dlnorm_dmgssub, 101	exp_logfddd, 124
dlnorm_p1, 102	exp_logscores, 125
dlnorm_p1_cp (lnorm_p1_cp), 535	exp_p1_cp, 126
dlnorm_p1sub, 102	exp_p1_f1fa, 133
dlnormsub, 101	exp_p1_f1fw, 134
dlogis2sub, 103	exp_p1_f2fa, 134
dlogis_cp (logis_cp), 555	exp_p1_f2fw, 135
dlogis_p1, 103	exp_p1_fd, 135
dlogis_p1_cp (logis_p1_cp), 568	exp_p1_fdd, 136
dlogis_p1sub, 104	exp_p1_ldda, 136
dlst_k3_cp (lst_k3_cp), 589	exp_p1_lddda, 137
dlst_k3sub, 104	exp_p1_logf, 137
dlst_p1k3, 105	exp_p1_logfdd, 138
dlst_p1k3_cp (lst_p1k3_cp), 609	exp_p1_logfddd, 138
dlst_p1k3sub, 105	exp_p1_loglik, 139
dmgs, 106	exp_p1_logscores, 139
dnorm_cp (norm_cp), 668	$exp_p1_means, 140$
dnorm_dmgs_cp (norm_dmgs_cp), 674	exp_p1_mu1fa, 140
dnorm_dmgssub, 107	exp_p1_mu2fa, 141
dnorm_p1, 108	exp_p1_p1fa, 141
dnorm_p12, 108	exp_p1_p2fa, 142
dnorm_p12_cp (norm_p12_cp), 690	exp_p1_pd, 142
dnorm_p12dmgs, 109	exp_p1_pdd, 143
dnorm_p1_cp (norm_p1_cp), 712	exp_p1_predictordata, 143
dnorm_p1_formula, 110	exp_p1_waic, 144
dnorm_p1sub, 109	exp_p1fa, <u>125</u>
dnormsub, 107	exp_p2fa, 144
dpareto_k2_cp (pareto_k2_cp), 735	exp_pd, 145
dpareto_k2_sub, 110	exp_pdd, 146
dpareto_p1k2, 111	exp_waic, 146
dpareto_p1k2_cp (pareto_p1k2_cp), 752	
dpareto_p1k2sub, 111	findnt, 147
dunif_cp (unif_cp), 823	fixgevrange, 147
dunif_formula, 112	fixgpdrange, 148
dweibull_cp (weibull_cp), 829	frechet_k1_cp, 148
dweibull_p2, 113	frechet_k1_f1fa, 154
dweibull_p2_cp (weibull_p2_cp), 843	frechet_k1_f2fa, 155
dweibull_p2sub, 113	frechet_k1_fd, 156
dweibullsub, 112	frechet_k1_fdd, 156
	frechet_k1_ldda, 157
exp_cp, 114	frechet_k1_lddda, 157
exp_f1fa, 120	frechet_k1_logf, 158
exp_f2fa, 120	frechet_k1_logfdd, 158
exp_fd, 121	frechet_k1_logfddd, 159
exp_fdd, 121	frechet_k1_mu1fa, 159
exp_ldda, 122	frechet_k1_mu2fa, 160
exp_lddda, 123	frechet_k1_p1fa, 160
exp_logf, 123	frechet_k1_p2fa, 161
exp_logfdd, 124	frechet_k1_pd, 161

frechet_k1_pdd, 162	gamma_means, 203
frechet_k1_waic, 162	gamma_mu1f, 203
frechet_loglik, 163	gamma_mu2f, 204
frechet_logscores, 163	gamma_p1f, 204
frechet_means, 164	
frechet_p2k1_cp, 164	gamma_p2f, 205
frechet_p2k1_cp, 104 frechet_p2k1_f1fa, 172	gamma_waic, 206
frechet_p2k1_f1fw, 172	gev_boot, 206
frechet_p2k1_f2fa, 173	gev_checkmle, 207
·	gev_cp, 207
frechet_p2k1_f2fw, 173	gev_f1fa, 216
frechet_p2k1_fd, 174	gev_f2fa, 217
frechet_p2k1_fdd, 174	gev_fd, 217
frechet_p2k1_ldda, 175	gev_fdd, 218
frechet_p2k1_lddda, 176	<pre>gev_k12_ppm_minusloglik, 218</pre>
frechet_p2k1_logf, 176	gev_k3_cp, 220
frechet_p2k1_logfdd, 177	gev_k3_f1fa, 226
frechet_p2k1_logfddd, 177	gev_k3_f2fa, 227
frechet_p2k1_loglik, 178	gev_k3_fd, 228
frechet_p2k1_logscores, 179	gev_k3_fdd, 228
frechet_p2k1_means, 179	gev_k3_1dda, 229
frechet_p2k1_mu1fa, 180	gev_k3_1ddda, 229
frechet_p2k1_mu2fa, 181	gev_k3_logf, 230
frechet_p2k1_p1fa, 181	gev_k3_logfdd, 230
frechet_p2k1_p2fa, 182	gev_k3_logfddd, 231
frechet_p2k1_pd, 182	gev_k3_loglik, 231
frechet_p2k1_pdd, 183	gev_k3_means, 232
frechet_p2k1_predictordata, 184	gev_k3_mu1fa, 232
frechet_p2k1_waic, 184	gev_k3_mu2fa, 233
commo on 105	gev_k3_pd, 233
gamma_cp, 185	gev_k3_pdd, 234
gamma_f1f, 192	gev_k3_waic, 234
gamma_f1fa, 193	gev_ld12a, 235
gamma_f2f, 193	gev_lda, 236
gamma_f2fa, 194	gev_1dda, 236
gamma_fd, 194	gev_lddda, 237
gamma_fdd, 195	gev_logf, 237
gamma_gg, 195	gev_logfd, 238
gamma_gmn, 196	gev_logfdd, 238
gamma_ldd, 196	gev_logfddd, 239
gamma_ldda, 197	gev_loglik, 239
gamma_lddd, 198	
gamma_lddda, 198	gev_means, 240
gamma_lmn, 199	gev_mu1fa, 241
gamma_lmnp, 199	gev_mu2fa, 241
gamma_logf, 200	gev_p123_checkmle, 242
gamma_logfdd, 201	gev_p123_cp, 242
gamma_logfddd, 201	gev_p123_f1fa, 251
gamma_loglik, 202	gev_p123_f1fw, 252
gamma_logscores, 202	gev_p123_f2fa, 253

gev_p123_f2fw, 253	gev_p12k3_logfddd, 271
gev_p123_fd, 254	gev_p12k3_mu1fa, <mark>27</mark> 1
gev_p123_fdd, 255	gev_p12k3_mu2fa, 272
gev_p123_ldda, 255	gev_p12k3_pd, 273
gev_p123_1ddda, 256	gev_p12k3_pdd, 273
gev_p123_logf, 257	gev_p1_checkmle, 358
gev_p123_logfdd, 257	gev_p1_cp, 358
gev_p123_logfddd, 258	gev_p1_logf, 367
gev_p123_loglik, 259	gev_p1_loglik, 368
gev_p123_means, 259	$gev_p1_means, 368$
gev_p123_mu1fa, 260	gev_p1_predictordata,369
gev_p123_mu2fa, 261	<pre>gev_p1_setics, 369</pre>
gev_p123_pd, 261	gev_p1_waic, 370
gev_p123_pdd, 262	gev_p1a_f1fa, <mark>297</mark>
gev_p123_predictordata, 263	gev_p1a_f1fw, 297
gev_p123_setics, 263	gev_p1a_f2fa, <mark>298</mark>
gev_p123_waic, 264	gev_p1a_f2fw, 298
gev_p12_checkmle, 274	gev_p1a_fd, 299
gev_p12_cp, 275	gev_p1a_fdd, 299
gev_p12_f1fa, 283	gev_p1a_ldda, 300
gev_p12_f1fw, 284	gev_p1a_lddda, 301
gev_p12_f2fa, 285	gev_p1a_logfdd, 301
gev_p12_f2fw, 285	gev_p1a_logfddd,302
gev_p12_fd, 286	gev_p1a_mu1fa,302
gev_p12_fdd, 287	gev_p1a_mu2fa, 303
gev_p12_ldda, 287	gev_p1a_pd, 304
gev_p12_lddda, 288	gev_p1a_pdd, 304
gev_p12_logf, 289	gev_p1b_f1fa, 305
gev_p12_logfdd, 289	gev_p1b_f1fw, 306
gev_p12_logfddd, 290	gev_p1b_f2fa, 306
gev_p12_loglik, 290	gev_p1b_f2fw, 307
gev_p12_means, 291	gev_p1b_fd, 308
gev_p12_mu1fa, 292	gev_p1b_fdd, 308
gev_p12_mu2fa, 292	gev_p1b_ldda, 309
gev_p12_pd, 293	gev_p1b_lddda, 310
gev_p12_pdd, 294	gev_p1b_logfdd, 310
gev_p12_predictordata, 294	<pre>gev_p1b_logfddd, 311</pre>
gev_p12_setics, 295	gev_p1b_mu1fa,312
gev_p12_waic, 296	gev_p1b_mu2fa, 312
gev_p12k3_f1fa, 265	gev_p1b_pd, 313
gev_p12k3_f1fw, 265	gev_p1b_pdd, 314
gev_p12k3_f2fa, 266	gev_p1c_f1fa, 314
gev_p12k3_f2fw, 267	gev_p1c_f1fw, 315
gev_p12k3_fd, 267	gev_p1c_f2fa, 316
gev_p12k3_fdd, 268	gev_p1c_f2fw, 317
gev_p12k3_1dda, 269	gev_p1c_fd, 317
gev_p12k3_1ddda, 269	gev_p1c_fdd, 318
gev_p12k3_logfdd, 270	gev_p1c_ldda, 319

gev_p1c_1ddda, 319	gnorm_k3_fdd, 383
gev_p1c_logfdd, 320	gnorm_k3_1dd, 384
gev_p1c_logfddd, 321	gnorm_k3_ldda, 385
gev_p1c_mu1fa, 321	gnorm_k3_1ddd, 385
gev_p1c_mu2fa, 322	gnorm_k3_lddda,386
gev_p1c_pd, 323	gnorm_k3_1mn, 386
gev_p1c_pdd, 324	gnorm_k3_logf, 387
gev_p1k3_cp, 324	gnorm_k3_logfdd, 387
gev_p1k3_f1fa, 332	gnorm_k3_logfddd, 388
gev_p1k3_f1fw, 333	gnorm_k3_loglik, 388
gev_p1k3_f2fa, 333	<pre>gnorm_k3_logscores, 389</pre>
gev_p1k3_f2fw, 334	gnorm_k3_mu1f, 389
gev_p1k3_fd, 334	gnorm_k3_mu2f, 390
gev_p1k3_fdd, 335	gnorm_k3_p1f, 391
gev_p1k3_1dda, 336	gnorm_k3_p2f, 391
gev_p1k3_lddda, 336	gnorm_lmnp, 392
gev_p1k3_logf, 337	gnorm_waic, 393
gev_p1k3_logfdd, 337	gpd_k13_f1fa, 394
gev_p1k3_logfddd, 338	gpd_k13_f2fa, 394
gev_p1k3_loglik, 338	gpd_k13_fd, 395
gev_p1k3_means, 339	gpd_k13_fdd, 395
$gev_p1k3_mu1fa, 340$	gpd_k13_1dda, 396
gev_p1k3_mu2fa, 340	gpd_k13_1ddda, 396
gev_p1k3_pd, 341	gpd_k13_logfdd, 397
gev_p1k3_pdd, 341	gpd_k13_logfddd, 397
gev_p1k3_predictordata, 342	gpd_k13_mu1fa, 398
gev_p1k3_waic, 343	gpd_k13_mu2fa, 398
gev_p1n_checkmle, 344	gpd_k13_pd, 399
gev_p1n_cp, 344	gpd_k13_pdd, 399
gev_p1n_logf, 353	gpd_k1_checkmle, 400
gev_p1n_loglik, 354	gpd_k1_cp, 400
gev_p1n_means, 354	gpd_k1_f1fa, 408
gev_p1n_n1_exampledata, 355	gpd_k1_f2fa, 409
gev_p1n_n2_exampledata, 356	gpd_k1_fd, 410
gev_p1n_predictordata, 356	gpd_k1_fdd, 410
gev_p1n_setics, 357	gpd_k1_ldda, 411
gev_p1n_waic, 357	gpd_k1_lddda, 411
gev_pd, 370	gpd_k1_logf, 412
gev_pdd, 371	gpd_k1_logfdd, 412
gev_pwm_params, 372	gpd_k1_logfddd, 413
gev_setics, 372	gpd_k1_loglik, 413
gev_waic, 373	gpd_k1_means, 414
gnorm_k3_cp, 373	gpd_k1_mu1fa, 415
gnorm_k3_f1f, 380	gpd_k1_mu2fa, 415
gnorm_k3_f1fa, 381	gpd_k1_pd, 416
gnorm_k3_f2f, 382	gpd_k1_pdd, 416
gnorm_k3_f2fa, 382	<pre>gpd_k1_setics, 417</pre>
gnorm_k3_fd, 383	gpd_k1_waic, 417

<pre>gumbel_cp, 418</pre>	halfnorm_fd, 462
<pre>gumbel_f1fa, 424</pre>	halfnorm_fdd,462
gumbel_f2fa, 424	halfnorm_gg, 463
<pre>gumbel_fd, 425</pre>	halfnorm_gg11,463
<pre>gumbel_fdd, 425</pre>	halfnorm_1111,464
gumbel_ldda, 426	halfnorm_ldd,464
<pre>gumbel_lddda, 426</pre>	halfnorm_ldda,465
<pre>gumbel_logf, 427</pre>	halfnorm_lddd,466
<pre>gumbel_logfdd, 427</pre>	halfnorm_lddda,466
<pre>gumbel_logfddd, 428</pre>	halfnorm_logf,467
<pre>gumbel_loglik, 428</pre>	halfnorm_logfdd,467
<pre>gumbel_logscores, 429</pre>	halfnorm_logfddd,468
<pre>gumbel_means, 429</pre>	halfnorm_loglik,468
<pre>gumbel_mu1fa, 430</pre>	halfnorm_logscores, 469
<pre>gumbel_mu2fa, 431</pre>	halfnorm_means, 469
<pre>gumbel_p1_cp, 432</pre>	halfnorm_mu1f,470
<pre>gumbel_p1_f1fa, 439</pre>	halfnorm_mu2f,470
<pre>gumbel_p1_f1fw, 440</pre>	halfnorm_p1f,471
<pre>gumbel_p1_f2fa, 440</pre>	halfnorm_p2f,471
<pre>gumbel_p1_f2fw, 441</pre>	halfnorm_waic,472
<pre>gumbel_p1_fd, 441</pre>	
<pre>gumbel_p1_fdd, 442</pre>	ifvectorthenmatrix, 472
<pre>gumbel_p1_ldda, 442</pre>	invgamma_cp, 473
<pre>gumbel_p1_lddda, 443</pre>	invgamma_f1f,480
<pre>gumbel_p1_logf, 443</pre>	invgamma_f1fa,480
<pre>gumbel_p1_logfdd, 444</pre>	invgamma_f2f,481
<pre>gumbel_p1_logfddd, 444</pre>	invgamma_f2fa,481
<pre>gumbel_p1_loglik, 445</pre>	invgamma_fd, 482
<pre>gumbel_p1_logscores, 445</pre>	invgamma_fdd,482
<pre>gumbel_p1_means, 446</pre>	invgamma_ldd, 483
<pre>gumbel_p1_mu1fa, 446</pre>	invgamma_ldda,483
<pre>gumbel_p1_mu2fa, 447</pre>	invgamma_lddd,484
<pre>gumbel_p1_p1fa, 447</pre>	invgamma_lddda,484
<pre>gumbel_p1_p2fa, 448</pre>	invgamma_lmn, 485
<pre>gumbel_p1_pd, 448</pre>	invgamma_lmnp, 485
<pre>gumbel_p1_pdd, 449</pre>	invgamma_logf, 486
<pre>gumbel_p1_predictordata, 449</pre>	invgamma_logfdd, 487
<pre>gumbel_p1_waic, 450</pre>	invgamma_logfddd,487
<pre>gumbel_p1fa, 431</pre>	invgamma_loglik, 488
<pre>gumbel_p2fa, 451</pre>	invgamma_logscores, 488
<pre>gumbel_pd, 451</pre>	invgamma_mu1f,489
<pre>gumbel_pdd, 452</pre>	invgamma_mu2f,489
<pre>gumbel_waic, 452</pre>	invgamma_p1f,490
	invgamma_p2f, 490
halfnorm_cp, 453	invgamma_waic, 491
halfnorm_f1f, 460	invgauss_cp, 492
halfnorm_f1fa, 460	invgauss_f1f, 499
halfnorm_f2f, 461	invgauss_f1fa, 500
halfnorm_f2fa, 461	invgauss_f2f, 500

invgauss_f2fa, 501	1norm_p1_fdd, 544
invgauss_fd, 501	lnorm_p1_ldda, 545
invgauss_fdd, 502	lnorm_p1_lddda,545
invgauss_ldd, 502	<pre>lnorm_p1_logf, 546</pre>
invgauss_ldda, 503	<pre>lnorm_p1_logfdd, 546</pre>
invgauss_lddd, 503	<pre>lnorm_p1_logfddd, 547</pre>
invgauss_lddda, 504	<pre>lnorm_p1_loglik, 547</pre>
invgauss_lmn, 504	<pre>lnorm_p1_logscores, 548</pre>
invgauss_lmnp, 505	lnorm_p1_mu1fa, <u>548</u>
invgauss_logf, 505	lnorm_p1_mu2fa, 549
invgauss_logfdd, 506	lnorm_p1_p1fa, 549
invgauss_logfddd, 506	lnorm_p1_p2fa, 550
invgauss_loglik, 507	lnorm_p1_pd, 550
invgauss_logscores, 507	lnorm_p1_pdd, <u>551</u>
invgauss_means, 508	lnorm_p1_predictordata, 551
invgauss_mu1f, 509	<pre>lnorm_p1_waic, 552</pre>
invgauss_mu2f, 509	lnorm_p1fa, 534
invgauss_p1f, 510	lnorm_p2fa, 553
invgauss_p2f, 510	lnorm_pd, 553
invgauss_waic, 511	lnorm_pdd, 554
	lnorm_waic, 554
jpf2p, 512	logis_cp, 555
jpf3p, 512	logis_f1fa, 561
jpf4p, 513	logis_f2fa, 561
lnorm on 512	logis_fd, 562
lnorm_cp, 513	logis_fdd, <u>562</u>
<pre>lnorm_dmgs_cp, 519 lnorm_dmgs_loglik, 526</pre>	logis_ldda, 563
	logis_lddda, 563
lnorm_dmgs_logscores, 526	logis_logf, 564
<pre>lnorm_dmgs_means, 527 lnorm_dmgs_waic, 527</pre>	logis_logfdd, 564
lnorm_f1fa, 528	logis_logfddd, 565
Inorm_f2fa, 529	logis_loglik, 565
Inorm_fd, 529	logis_logscores, 566
lnorm_fdd, 530	logis_mu1fa, 566
lnorm_ldda, 530	logis_mu2fa, 567
lnorm_lddda, 531	logis_p1_cp, 568
lnorm_logf, 531	logis_p1_f1fa, 575
lnorm_logfdd, 532	logis_p1_f1fw, 576
lnorm_logfddd, 532	logis_p1_f2fa, 576
lnorm_logscores, 533	logis_p1_f2fw, 577
lnorm_mu1fa, 533	logis_p1_fd, 577
Inorm_mu2fa, 534	logis_p1_fdd, 578
lnorm_p1_cp, 535	logis_p1_ldda, 578
lnorm_p1_f1fa, 542	logis_p1_lddda, 579
Inorm_p1_f1fw, 542	logis_p1_logf, 579
Inorm_p1_f2fa, 543	logis_p1_logfdd, 580
Inorm_p1_f2fw, 543	logis_p1_logfddd, 580
Inorm_p1_fd, 544	logis_p1_loglik, 581
11101 m_p1_1 u, 277	10813_PI_10811K, 301

logis_p1_logscores, 581	lst_p1k3_ldda, 623
logis_p1_means, 582	lst_p1k3_lddd, 623
logis_p1_mu1fa,582	lst_p1k3_lddda, 624
logis_p1_mu2fa, 583	lst_p1k3_lmn, 625
logis_p1_p1fa, 583	lst_p1k3_lmnp, 625
logis_p1_p2fa, 584	lst_p1k3_logf, 626
logis_p1_pd, 584	lst_p1k3_logfdd,627
logis_p1_pdd, 585	lst_p1k3_logfddd,627
logis_p1_predictordata, 585	lst_p1k3_loglik,628
logis_p1_waic, 586	lst_p1k3_logscores,629
logis_p1fa, 567	lst_p1k3_mu1f, 629
logis_p2fa, 587	lst_p1k3_mu2f, 630
logis_pd, 587	lst_p1k3_p1f, 631
logis_pdd, 588	lst_p1k3_p2f, 631
logis_waic, 588	lst_p1k3_predictordata,632
lst_k3_cp, 589	lst_p1k3_setics, 633
lst_k3_f1f, 596	lst_p1k3_waic, 633
lst_k3_f1fa, 597	
lst_k3_f2f, 597	make_cwaic, 637
lst_k3_f2fa, 598	make_maic, 638
lst_k3_fd, 598	make_se, 638
lst_k3_fdd, 599	make_waic, 639
lst_k3_ldd, 599	makebetat0, 634
lst_k3_ldda, 600	makebetatm, 635
lst_k3_lddd, 600	makemuhat0, 635
lst_k3_lddda, 601	makeq, 636
lst_k3_lmn, 601	maket0, 636
lst_k3_lmnp, 602	maketresid0,637
lst_k3_logf, 603	man, 639
lst_k3_logfdd, 603	man1f, 649
lst_k3_logfddd, 604	man2f, 650
lst_k3_loglik, 604	manboot, 650
lst_k3_logscores, 605	mancheckmle, 650
lst_k3_mu1f, 605	mandsub, 651
lst_k3_mu2f, 606	manf, 651
lst_k3_p1f, 607	manldd, 658
lst_k3_p2f, 607	manlddd, 658
lst_k3_waic, 608	manlnn, 659
lst_p1k3_cp, 609	manlnnn, 659
lst_p1k3_f1f, 617	manlogf, 659
lst_p1k3_f1fa, 617	manloglik, 660
lst_p1k3_f1fw, 618	manlogscores, 660
lst_p1k3_f2f, 619	manmeans, 660
	manpredictor, 661
1st_p1k3_f2fa, 619	manvector, 661
lst_p1k3_f2fw, 620	manwaic, 661
1st_p1k3_fd, 621	movexiawayfromzero, 662
lst_p1k3_fdd, 621	ms_flat_1tail,662
lst_p1k3_ldd, 622	ms_flat_2tail,664

ms_predictors_1tail, 665	norm_p12_waic, <b>711</b>
ms_predictors_2tail,666	norm_p1_cp, 712
	norm_p1_f1fa,719
nopdfcdfmsg, 667	norm_p1_f1fw, 720
norm_boot, 668	norm_p1_f2fa, 720
norm_cp, 668	norm_p1_f2fw, 721
norm_dmgs_cp, 674	norm_p1_fd, 721
norm_dmgs_loglik, 680	norm_p1_fdd, 722
norm_dmgs_logscores, 680	norm_p1_1dda, 723
norm_dmgs_means, 681	norm_p1_lddda, 723
norm_dmgs_waic, 682	norm_p1_logf, 724
norm_f1fa, 682	norm_p1_logfdd, 725
norm_f2fa, 683	norm_p1_logfddd, 725
norm_fd, 683	norm_p1_loglik, 726
norm_fdd, 684	norm_p1_logscores, 727
norm_ldda, 684	norm_p1_mlparams, 727
norm_lddda, 685	norm_p1_mu1fa, 728
norm_logf, 685	norm_p1_mu2fa, 728
norm_logfdd, 686	norm_p1_p1fa, 729
norm_logfddd, 686	norm_p1_p2fa, 730
norm_logscores, 687	norm_p1_pd, 730
norm_ml_params, 687	norm_p1_pdd, 731
norm_mu1fa, 688	norm_p1_predictordata, 732
norm_mu2fa, 688	norm_p1_waic, 732
norm_p12_boot, 689	norm_p1fa, 712
norm_p12_checkmle, 689	norm_p2fa, 733
norm_p12_cp, 690	norm_pd, 733
norm_p12_exampledata, 698	norm_pdd, 734
norm_p12_f1fa, 698	norm_unbiasedv_params, 734
norm_p12_f1fw, 699	norm_waic, 735
norm_p12_f2fa, 699	norm_wate, 755
norm_p12_f2fw, 700	pareto_k2_cp, 735
norm_p12_fd, 700	pareto_k2_f1fa, 741
norm_p12_fdd, 701	pareto_k2_f2fa, 742
norm_p12_1dda, 702	pareto_k2_fd, 743
norm_p12_lddda, 702	pareto_k2_fdd, 743
norm_p12_logf, 703	pareto_k2_ldda, 744
norm_p12_logfdd, 703	pareto_k2_lddda, 745
norm_p12_logfddd, 704	pareto_k2_logf, 745
norm_p12_loglik, 705	pareto_k2_logfdd, 746
norm_p12_logscores, 705	pareto_k2_logfddd, 746
norm_p12_mu1fa, 706	pareto_k2_logscores, 747
norm_p12_mu2fa, 706	pareto_k2_ml_params, 748
norm_p12_mu21a, 700	pareto_k2_m1_params, 748
norm_p12_p2fa, 707	pareto_k2_mu2fa, 749
norm_p12_pd, 708	pareto_k2_mu21a, 749 pareto_k2_p1fa, 749
norm_p12_pdd, 709	pareto_k2_p11a, 749 pareto_k2_p2fa, 750
norm_p12_predictordata, 709	pareto_k2_p21a, 750 pareto_k2_pd, 750
norm_p12_setics, 710	pareto_k2_pdd, 751

pareto_k2_waic, 752	pgumbel_cp(gumbel_cp), 418
pareto_p1k2_cp, 752	pgumbel_p1, 776
pareto_p1k2_f1fa, 760	pgumbel_p1_cp (gumbel_p1_cp), 432
pareto_p1k2_f1fw, 760	phalfnorm_cp (halfnorm_cp), 453
pareto_p1k2_f2fa, 761	pinvgamma_cp(invgamma_cp), 473
pareto_p1k2_f2fw, 761	pinvgauss_cp (invgauss_cp), 492
pareto_p1k2_fd, 762	plnorm_cp(lnorm_cp), 513
pareto_p1k2_fdd, 762	plnorm_dmgs_cp(lnorm_dmgs_cp), 519
pareto_p1k2_1dda, 763	plnorm_p1,777
pareto_p1k2_lddda, 763	plnorm_p1_cp (lnorm_p1_cp), 535
pareto_p1k2_logf, 764	plogis_cp(logis_cp), 555
pareto_p1k2_logfdd, 764	plogis_p1,777
pareto_p1k2_logfddd, 765	plogis_p1_cp (logis_p1_cp), 568
pareto_p1k2_loglik, 765	plst_k3_cp (lst_k3_cp), 589
pareto_p1k2_logscores, 766	plst_p1k3,778
pareto_p1k2_means, 766	plst_p1k3_cp (lst_p1k3_cp), 609
pareto_p1k2_mu1fa, 767	pnorm_cp (norm_cp), 668
pareto_p1k2_mu2fa, 768	pnorm_dmgs_cp (norm_dmgs_cp), 674
pareto_p1k2_p1fa, 768	pnorm_p1, 778
pareto_p1k2_p2fa, 769	pnorm_p12, 779
pareto_p1k2_pd, 769	pnorm_p12_cp (norm_p12_cp), 690
pareto_p1k2_pdd, 770	pnorm_p1_cp (norm_p1_cp), 712
pareto_p1k2_predictordata, 770	pnorm_p1_formula, 779
pareto_p1k2_waic, 771	ppareto_k2_cp (pareto_k2_cp), 735
pcauchy_cp (cauchy_cp), 26	ppareto_p1k2, 780
pcauchy_p1, 772	ppareto_p1k2_cp (pareto_p1k2_cp), 752
	<pre>punif_cp (unif_cp), 823</pre>
pcauchy_p1_cp (cauchy_p1_cp), 43	punif_formula, 780
pexp_cp (exp_cp), 114	<pre>pweibull_cp (weibull_cp), 829</pre>
pexp_p1, 772	pweibull_p2,781
pexp_p1_cp (exp_p1_cp), 126	<pre>pweibull_p2_cp (weibull_p2_cp), 843</pre>
pfrechet_k1_cp (frechet_k1_cp), 148	
pfrechet_p2k1, 773	qcauchy_cp (cauchy_cp), 26
pfrechet_p2k1_cp (frechet_p2k1_cp), 164	qcauchy_p1, 781
pgamma_cp (gamma_cp), 185	qcauchy_p1_cp (cauchy_p1_cp), 43
pgev_cp (gev_cp), 207	qexp_cp (exp_cp), 114
pgev_k3_cp (gev_k3_cp), 220	qexp_p1, 782
pgev_p1, 773	qexp_p1_cp (exp_p1_cp), 126
pgev_p12, 774	qfrechet_k1_cp (frechet_k1_cp), 148
pgev_p123, 774	qfrechet_p2k1, 782
pgev_p123_cp (gev_p123_cp), 242	qfrechet_p2k1_cp (frechet_p2k1_cp), 164
pgev_p12_cp (gev_p12_cp), 275	qgamma_cp (gamma_cp), 185
pgev_p1_cp (gev_p1_cp), 358	qgamma_k1_ppm, 783
pgev_p1k3, 775	qgamma_ppm, 784
pgev_p1k3_cp (gev_p1k3_cp), 324	qgev_cp (gev_cp), 207
pgev_p1n, 776	qgev_k12_ppm, 785
pgev_p1n_cp (gev_p1n_cp), 344	qgev_k3_cp (gev_k3_cp), 220
pgnorm_k3_cp (gnorm_k3_cp), 373	qgev_mpd_ppm, 787
pgpd_k1_cp (gpd_k1_cp), 400	qgev_p1, 788

qgev_p12, 789	reltest, 802
qgev_p123, 789	reltest2, 806
qgev_p123_cp (gev_p123_cp), 242	reltest2_cases, 808
qgev_p12_cp (gev_p12_cp), 275	reltest2_makeep, $809$
qgev_p1_cp (gev_p1_cp), 358	reltest2_plot, 809
qgev_p1_ppm, 791	reltest2_predict, 810
qgev_p1k3, 790	reltest2_simulate, 811
qgev_p1k3_cp (gev_p1k3_cp), 324	reltest_makeep, 812
qgev_p1n, 791	reltest_makemaxep, 812
qgev_p1n_cp (gev_p1n_cp), 344	reltest_params, 813
qgev_ppm, 793	reltest_predict, 814
qgnorm_k3_cp (gnorm_k3_cp), 373	reltest_simulate, 815
qgpd_k1_cp (gpd_k1_cp), 400	rexp_cp (exp_cp), 114
qgpd_k1_ppm, 794	rexp_p1_cp (exp_p1_cp), 126
qgumbel_cp (gumbel_cp), 418	rfrechet_k1_cp (frechet_k1_cp), 148
qgumbel_p1, 795	<pre>rfrechet_p2k1_cp (frechet_p2k1_cp), 164</pre>
qgumbel_p1_cp (gumbel_p1_cp), 432	rgamma_cp (gamma_cp), 185
qhalfnorm_cp (halfnorm_cp), 453	rgev_cp (gev_cp), 207
qinvgamma_cp (invgamma_cp), 473	rgev_k3_cp (gev_k3_cp), 220
qinvgauss_cp (invgauss_cp), 492	rgev_minmax, 816
qlnorm_cp (lnorm_cp), 513	rgev_p123_cp (gev_p123_cp), 242
qlnorm_dmgs_cp (lnorm_dmgs_cp), 519	rgev_p123_minmax, 817
qlnorm_p1, 796	rgev_p12_cp (gev_p12_cp), 275
qlnorm_p1_cp (lnorm_p1_cp), 535	rgev_p12_minmax, 818
qlogis_cp (logis_cp), 555	rgev_p1_cp (gev_p1_cp), 358
qlogis_p1,797	rgev_p1_minmax,820
qlogis_p1_cp (logis_p1_cp), 568 qlst_k3_cp (lst_k3_cp), 589	rgev_p1k3_cp (gev_p1k3_cp), 324
q1st_k3_cp (1st_k3_cp), 389 q1st_p1k3, 797	rgev_p1n_cp (gev_p1n_cp), 344
q1st_p1k3, 797 q1st_p1k3_cp (1st_p1k3_cp), 609	rgev_p1n_minmax, 819
qnorm_cp (norm_cp), 668	rgnorm_k3_cp (gnorm_k3_cp), 373
qnorm_dmgs_cp (norm_dmgs_cp), 674	rgpd_k1_cp (gpd_k1_cp), 400
qnorm_p1, 798	rgpd_k1_minmax, 821
qnorm_p12, 798	rgumbel_cp(gumbel_cp),418
qnorm_p12_cp (norm_p12_cp), 690	rgumbel_p1_cp (gumbel_p1_cp), 432
qnorm_p1_cp (norm_p1_cp), 712	rhalfnorm_cp (halfnorm_cp), 453
qnorm_p1_formula, 799	rhp_dmgs_cpmethod, 821
qntt_ppm, 799	rinvgamma_cp (invgamma_cp), 473
qpareto_k2_cp (pareto_k2_cp), 735	rinvgauss_cp (invgauss_cp), 492
qpareto_p1k2, 801	rlnorm_cp (lnorm_cp), 513
qpareto_p1k2_cp (pareto_p1k2_cp), 752	rlnorm_dmgs_cp (lnorm_dmgs_cp), 519
qunif_cp (unif_cp), 823	rlnorm_p1_cp (lnorm_p1_cp), 535
qunif_formula, 801	rlogis_cp(logis_cp), 555
qweibull_cp (weibull_cp), 829	rlogis_p1_cp(logis_p1_cp), 568
qweibull_p2, 802	rlst_k3_cp (lst_k3_cp), 589
qweibull_p2_cp (weibull_p2_cp), 843	rlst_p1k3_cp (lst_p1k3_cp), 609
	rnorm_cp (norm_cp), 668
rcauchy_cp (cauchy_cp), 26	<pre>rnorm_dmgs_cp (norm_dmgs_cp), 674</pre>
rcauchy_p1_cp (cauchy_p1_cp), 43	rnorm_p12_cp (norm_p12_cp), 690

rnorm_p1_cp (norm_p1_cp), 712	weibull_fd, 836
rpareto_k2_cp (pareto_k2_cp), 735	weibull_fdd, 836
rpareto_p1k2_cp (pareto_p1k2_cp), 752	weibull_ldda, 837
runif_cp (unif_cp), 823	weibull_lddda, 837
rust_pumethod, 822	weibull_logf, 838
rweibull_cp (weibull_cp), 829	weibull_logfdd, 838
rweibull_p2_cp (weibull_p2_cp), 843	weibull_logfddd, 839
	weibull_loglik, 839
tcauchy_cp (cauchy_cp), 26	weibull_logscores, 840
tcauchy_p1_cp (cauchy_p1_cp), 43	weibull_means, 840
testppm_plot, 822	weibull_mu1fa, 841
texp_cp (exp_cp), 114	weibull_mu2fa, 842
texp_p1_cp (exp_p1_cp), 126	weibull_p1fa, 842
tfrechet_k1_cp (frechet_k1_cp), 148	weibull_p2_cp, 843
tfrechet_p2k1_cp (frechet_p2k1_cp), 164	weibull_p2_f1fa, 851
tgamma_cp (gamma_cp), 185	weibull_p2_f1fw, 851
tgev_cp (gev_cp), 207	weibull_p2_f2fa, 852
tgev_k3_cp (gev_k3_cp), 220	weibull_p2_f2fw, 852
tgev_p123_cp (gev_p123_cp), 242	weibull_p2_fd, 853
tgev_p12_cp (gev_p12_cp), 275	weibull_p2_fdd, 853
tgev_p1_cp (gev_p1_cp), 358	weibull_p2_ldda, 854
tgev_p1k3_cp (gev_p1k3_cp), 324	weibull_p2_lddda, 854
tgev_p1n_cp (gev_p1n_cp), 344	weibull_p2_logf, 855
tgnorm_k3_cp (gnorm_k3_cp), 373	weibull_p2_logfdd, 855
tgpd_k1_cp (gpd_k1_cp), 400	weibull_p2_logfddd, 856
tgumbel_cp (gumbel_cp), 418	weibull_p2_loglik, 856
tgumbel_p1_cp (gumbel_p1_cp), 432	weibull_p2_logscores, 857
thalfnorm_cp (halfnorm_cp), 453	weibull_p2_means, 857
tinvgamma_cp (invgamma_cp), 473	weibull_p2_mu1fa, 858
tinvgauss_cp (invgauss_cp), 473	weibull_p2_mu2fa, 859
tlnorm_cp (lnorm_cp), 513	weibull_p2_p1fa, 859
tlnorm_p1_cp (lnorm_p1_cp), 535	weibull_p2_p1fa, 859 weibull_p2_p2fa, 860
tlogis_cp (logis_cp), 555	weibull_p2_pd, 860
tlogis_cp (logis_cp), 555 tlogis_p1_cp (logis_p1_cp), 568	weibull_p2_pdd, 861
tlst_k3_cp (lst_k3_cp), 589	weibull_p2_predictordata, 861
	weibull_p2_waic, 862
tlst_p1k3_cp (lst_p1k3_cp), 609	weibull_p2fa, 843
tnorm_cp (norm_cp), 668	
tnorm_p12_cp (norm_p12_cp), 690	weibull_pd, 863 weibull_pdd, 863
tnorm_p1_cp (norm_p1_cp), 712 tpareto_k2_cp (pareto_k2_cp), 735	weibull_waic, 864
	welbull_walc, 804
tpareto_p1k2_cp (pareto_p1k2_cp), 752	
tweibull_cp (weibull_cp), 829	
tweibull_p2_cp (weibull_p2_cp), 843	
unif_cp, 823	
weibull_cp, 829	
weibull_f1fa, 835	
weibull f2fa, 835	