

# Package ‘indirect’

April 12, 2018

**Type** Package

**Title** Elicitation of Independent Conditional Means Priors for Generalised Linear Models

**Version** 0.2.0

**Date** 2018-04-10

**Author** Geoffrey R. Hosack

**Maintainer** Geoff Hosack <geoff.hosack@csiro.au>

**Description** Functions are provided to facilitate prior elicitation for Bayesian generalised linear models using independent conditional means priors. The package supports the elicitation of multivariate normal priors for generalised linear models. The approach can be applied to indirect elicitation for a generalised linear model that is linear in the parameters. The package is designed such that the facilitator executes functions within the R console during the elicitation session to provide graphical and numerical feedback at each design point. Various methodologies for eliciting fractiles (equivalently, percentiles or quantiles) are supported, including versions of the approach of Hosack et al. (2017) <doi:10.1016/j.res.2017.06.011>. For example, experts may be asked to provide central credible intervals that correspond to a certain probability. Or experts may be allowed to vary the probability allocated to the central credible interval for each design point. Additionally, a median may or may not be elicited.

**License** GPL-3

**Depends** R (>= 3.1.0)

**Imports** MASS, gplots

**Suggests** tools, utils

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-04-12 10:34:10 UTC

## R topics documented:

checkX	2
CNdiag	2

designLink . . . . .	3
dGompertzNorm . . . . .	4
dLogitNorm . . . . .	5
elicitPt . . . . .	6
indirect . . . . .	7
makeSweave . . . . .	7
muSigma . . . . .	8
mV . . . . .	9
pdist . . . . .	10
plotDesignPoint . . . . .	11
saveRecord . . . . .	13

<b>Index</b>	<b>15</b>
--------------	-----------

---

checkX	<i>Helper function that checks for sensible covariate matrix.</i>
--------	---

---

### Description

Helper function that checks for sensible covariate matrix.

### Usage

```
checkX(X)
```

### Arguments

X	numeric matrix of covariates, $n$ design points by $p$ covariates, for a given model and design points.
---	---

### Value

throws an error if not full rank.

---

CNdiag	<i>Function to check condition number diagnostic.</i>
--------	---

---

### Description

This function calculates the condition number of the rescaled  $n \times p$  design matrix  $X$  such that each column has unit length.

### Usage

```
CNdiag(X)
```

**Arguments**

X                      Design matrix

**Value**

a scalar giving the condition number of the rescaled design matrix

**Examples**

```
X <- matrix(rnorm(16), nrow = 4)
CNdiag(X)
```

---

designLink                      *Create list with information for the elicitation session*

---

**Description**

This builds the structure that will store elicited data. The linear predictor has a normal prior  $g(\theta) N(m, V)$ ,  $\theta$  is the elicitation target. Link functions  $g(\cdot)$ : logit, log, cloglog, identity.

**Usage**

```
designLink(design, link = "identity", target = "Target", CI.prob = 1/2,
  expertID = "Expert", facilitator = "Facilitator", rapporteur = "none",
  intro.comments = "This is a record of the elicitation session.",
  fit.method = "KL")
```

**Arguments**

design	a dataframe with covariate values that will be displayed to the expert(s) during the elicitation session.
link	character logit, log, cloglog, identity
target	character, name of target parameter of elicitation exercise
CI.prob	numeric, a fraction between 0 and 1 that defines probability attributed to central credible interval. For example, 1/2 for a central credible interval of probability 0.5, or 1/3 for a central credible interval of probability 0.333... The default is probability 1/2.
expertID	character, identifier for expert or group of experts
facilitator	character, facilitator identifier
rapporteur	character, rapporteur identifier. Default "none".
intro.comments	character, text with any prefacing comments. This may include, for example, the definition of the target parameter for the elicitation session. Beware of non-ASCII text and special characters, which may affect the ability to save the elicitation record with function <a href="#">saveRecord</a> or create a summary report with function <a href="#">makeSweave</a> if called by the function <a href="#">makeSweave</a> may affect ability to render by means of <a href="#">Sweave</a> or <a href="#">knitr</a> etc.

`fit.method` character, method used to fit conditional means prior: KL (default), moment, SS (see vignette and [mV](#) for more information on these options)

### Details

Assumption: at least two fractiles selected from the median, upper and lower bounds of the central credible interval of probability `CI.prob` will be elicited at each design point. The probabilities assigned to the central credible intervals can vary across design points. The argument `CI.prob` can later be adjusted by design point during the elicitation exercise, see function [elicitPt](#). In the first instance, it is set to a global value specified by `CI.prob` in function [designLink](#) with default value 0.5.

### Value

list of design with entries: `theta`, a  $n \times 4$  matrix with columns that give lower, median and upper quantiles followed by `CI.prob` and  $n$  equal to the number of design points (scenarios); `link`, the link function used; `target`; `expert facilitator`; `rapporteur`; `date`; `intro.comments`; `fit.method`.

### Examples

```
X <- matrix(c(1, 1, 0, 1), nrow = 2) # design
Z <- designLink(design = X, link = "logit", target = "target",
  CI.prob = 1/2, expertID = "Expert", facilitator = "facilitator")
```

---

dGompertzNorm                      *density for Gompertz transformed univariate Gaussian*

---

### Description

density for Gompertz transformed univariate Gaussian

### Usage

```
dGompertzNorm(x, mu, sigma)
```

### Arguments

`x`,                      numeric real  
`mu`,                      numeric real  
`sigma`,                    numeric real positive

### Value

transformed density on support (0, 1)

**Examples**

```
mu <- -1
sigma <- 1
z <- rnorm(10000, mu, sigma)
hist(1 - exp(-exp(z)), freq = FALSE)
curve(dGompertzNorm(x, mu = mu, sigma = sigma), col = 'red', add = TRUE, from = 0.01, to = 0.99)
integrate(function(x) dGompertzNorm(x, mu = mu, sigma = sigma), lower = 0, upper = 1) # equals 1
```

---

dLogitNorm

*density for logit transformed univariate Gaussian*

---

**Description**

density for logit transformed univariate Gaussian

**Usage**

```
dLogitNorm(x, mu, sigma)
```

**Arguments**

x,	numeric real
mu,	numeric real
sigma,	numeric real positive

**Value**

transformed density on support (0, 1)

**Examples**

```
mu <- -1
sigma <- 1
z <- rnorm(10000, mu, sigma)
hist(exp(z)/(1 + exp(z)), freq = FALSE)
curve(dLogitNorm(x, mu = mu, sigma = sigma), col = 'red', add = TRUE, from = 0.01, to = 0.99)
integrate(function(x) dLogitNorm(x, mu = mu, sigma = sigma), lower = 0, upper = 1) # equals 1
```

---

 elicitPt

*Function to create or update elicitation at a given design point.*


---

## Description

Function to create or update elicitation at a given design point.

## Usage

```
elicitPt(Z, design.pt = NULL, lower.CI.bound = NA, median = NA,
         upper.CI.bound = NA, CI.prob = NULL, comment = " ")
```

## Arguments

Z	list of design with entries: theta, a $n \times 4$ matrix with columns that give lower, median and upper quantiles of the central credible interval followed by the probability CI.prob allocated to the interval; link, the link function used; and target. This list object is created by <a href="#">designLink</a>
design.pt	single integer that denotes design point of interest
lower.CI.bound	scalar that gives the lower bound of the central credible interval, default NA.
median	scalar value, default NA
upper.CI.bound	scalar that gives the upper bound of the central credible interval, default NA.
CI.prob	numeric, a fraction between 0 and 1 that defines probability attributed to central credible interval. For example, 1/2 for quartiles or 1/3 for tertiles. Default NULL uses the initial CI.prob as defined by <a href="#">designLink</a> .
comment	character, ASCII text providing contributed commentary associated with elicitation design point. It is recommended to avoid special characters such as quotation marks etc.

## Value

Z, a list of design with entries: theta, a  $n \times 4$  matrix with columns that give lower, median and upper quantiles followed by CI.prob with updated entries for row specified by argument design.pt; link, the link function used; and target.

## Examples

```
X <- matrix(c(1, 1, 0, 1), nrow = 2) # design
Z <- designLink(design = X)
Z <- elicitPt(Z, design.pt = 1,
             lower.CI.bound = -1,
             median = 0,
             upper.CI.bound = 1,
             comment = "A completed elicitation scenario.")
```

---

indirect	<i>indirect: A package for assisting indirect elicitation of priors for generalised linear models.</i>
----------	--

---

## Description

The indirect package provides three categories of functions: elicitation functions, fitting functions and visualisation functions.

## Elicitation functions

These are the functions that are used to record expert opinion. This is where edits will be made and so on. The key function is [designLink](#), which defines a list object that contains information about the design and elicitation. The elicitations are recorded and updated via function [elicitPt](#).

## Fitting functions

These are generally helper functions except for the function [muSigma](#) that is used for estimating the mean vector and covariance matrix of the unknown coefficients for the multivariate normal prior. Helper functions include [mV](#) for the elicited moments of conditional means priors.

## Visualisation functions

These are functions for visualisation. The core function is [plotDesignPoint](#).

## References

Hosack, G. R., Hayes, K. R., & Barry, S. C. (2017). Prior elicitation for Bayesian generalised linear models with application to risk control option assessment. *Reliability Engineering and System Safety*, 167:351-361. doi:10.1016/j.res.2017.06.011

---

makeSweave	<i>Function to create summary document from a saved elicitation record.</i>
------------	---

---

## Description

Creates a Sweave file that can be used to generate a pdf document of the summary report.

## Usage

```
makeSweave(filename.rds = "", reportname = "",  
           title = "Elicitation record", contact.details = "none",  
           fitted.fractiles = TRUE, cumul.prob.bounds = c(0.05, 0.95))
```

**Arguments**

filename.rds	character, filename of the record saved as an RDS object, see ?saveRDS.
reportname	character, filename without extension to be used for the generated Sweave (.Rnw) file. The Sweave file supports the creation of report (.pdf) documentation and accompanying files such as the .tex file generated by using Sweave followed by tools::texi2pdf().
title	character, a title for the report
contact.details	character, an email address or other mechanism by which the expert may contact the facilitator or rapporteur
fitted.fractiles	logical or numeric vector. A logical value of FALSE will not plot any fitted fractiles from the fitted subjective probability distribution. A logical value of TRUE will plot the fitted fractiles that correspond to the final iteration of the raw elicited fractiles. Alternatively, a numeric vector can specify arbitrary fractiles for plotting from the fitted distribution, e.g., c(1/10, 1/4, 1/2, 3/4, 9/10)
cumul.prob.bounds	numeric vector that specifies the upper and lower plot bounds determined by this credible interval. The default is the 0.90 central credible interval, c(0.05, 0.95)

**Examples**

```
## Not run:
X <- matrix(c(1, 1, 0, 1), nrow = 2) # design
Z <- designLink(design = X)
Z <- elicitPt(Z, design.pt = 1,
  lower.CI.bound = -1,
  median = 0,
  upper.CI.bound = 1,
  comment = "A completed elicitation scenario.")
tmp.rds <- tempfile(pattern = "record", fileext = ".rds")
saveRecord(Z, file = tmp.rds)
tmpReport <- tempfile(pattern = "report")
makeSweave(filename.rds = tmp.rds, reportname = tmpReport)
setwd(tempdir())
utils::Sweave(paste0(tmpReport, ".Rnw"))
tools::texi2pdf(paste0(tmpReport, ".tex"))

## End(Not run)
```

---

muSigma

*Function to estimate mean and covariance for unknown parameters  $\beta$ .*


---

**Description**

Function to estimate mean and covariance for unknown parameters  $\beta$ .



**Usage**

```
muSigma(Z, X = NULL, fit.method = "KL", wls.method = "default")
```

**Arguments**

Z	list of design points and link function that is an output of function <code>designLink</code>
X	model matrix for model formula and design points. The covariates must correspond to the description of design points in Z, but can be transformed etc. If NULL then X will be coerced by applying <code>as.matrix()</code> to <code>Z\$design</code> . The matrix X should be full rank when subsetted to the elicited design points. If a column of X has the name <code>offset</code> then this column is treated as an offset during estimation
<code>fit.method</code>	character, moment, KL. See <a href="#">mV</a> . Default is KL.
<code>wls.method</code>	character giving the numerical solution method: QR, using the QR decomposition, SVD, using the singular value decomposition, or option <code>default</code> that uses <code>solve()</code>

**Value**

list of `mu`, numeric vector of location parameters for the normal prior; `Sigma`, the covariance matrix; and `log.like`, a scalar

**Examples**

```
X <- matrix(c(1, 1, 0, 1), nrow = 2) # design
Z <- designLink(design = X)
Z <- elicitPt(Z, design.pt = 1,
  lower.CI.bound = -1,
  median = 0,
  upper.CI.bound = 1,
  comment = "The first completed elicitation scenario.")
Z <- elicitPt(Z, design.pt = 2,
  lower.CI.bound = -2,
  median = 1,
  upper.CI.bound = 2,
  comment = "The second completed elicitation scenario.")
prior <- muSigma(Z, X, fit.method = "KL")
prior$mu
prior$Sigma
```

**Description**

The default for `fit.method` is option KL. This option uses an objective function that minimises a discretised directed divergence from a cumulative distribution implied by raw elicited fractiles to a normal conditional mean prior for the linear predictor. An alternative method `moment` assigns the location parameter of the normal conditional mean prior to the elicited median on the linear predictor scale. The variance parameter is estimated as  $V = ((g(f_u) - g(f_l))/(qnorm(u) - qnorm(l)))^2$ , where  $l$  is the probability associated with the fractile  $f_l$  that defines the lower bound for the central credible interval and  $u$  is the probability associated with the fractile  $f_u$  that defines the upper bound for the central credible interval. This is also used to initialise the optimisation for the KL method. Another optimisation method that minimises the sum of squares is also available as method `SS`. See the vignette for more details on the choice of objective function for KL and SS.

**Usage**

```
mV(Z, fit.method = "KL")
```

**Arguments**

`Z` list object that contains matrix `theta` of elicitation and character `link`, see [plotDesignPoint](#)

`fit.method` character, `moment`, `KL`, `SS`. Default is `KL`.

**Value**

A list with vector of means `m` and diagonal covariance matrix `V`.

---

pdist	<i>Helper function that gives the probability distribution function for design point.</i>
-------	---

---

**Description**

Helper function that gives the probability distribution function for design point.

**Usage**

```
pdist(x, Z, design.pt = NULL, fit.method = "KL")
```

**Arguments**

`x` numeric: coordinate

`Z` list of design points and link function, see [designLink](#)

`design.pt` integer: design point

`fit.method` character: method for fit in `mV`, default is `KL`

**Examples**

```

# design matrix: two scenarios
X <- matrix(c(1, 1, 0, 1), nrow = 2)
rownames(X) <- c("scenario1", "scenario2")
colnames(X) <- c("covariate1", "covariate2")
#' # logit link
# central credible intervals with probability = 1/2
Z <- designLink(design = X, link = "logit", CI.prob = 0.5)
#' # lower and upper quartiles and median
Z <- indirect::elicitPt(Z, design.pt = 1,
  lower.CI.bound = 0.2,
  median = 0.4,
  upper.CI.bound = 0.6,
  comment = "Completed.")
indirect::plotDesignPoint(Z, design.pt = 1,
  elicited.fractiles = TRUE, theta.bounds = c(0, 1),
  fitted.fractiles = TRUE, fitted.curve = TRUE)

# probability that target is below 0.1 and
# probability that target is below 0.9
indirect::pdist(c(0.1, 0.9), Z, design.pt = 1)

```

---

plotDesignPoint

*Plot elicited data, fitted marginals or model output*


---

**Description**

Plot elicited data, fitted marginals or model output

**Usage**

```

plotDesignPoint(Z, X = NULL, design.pt = NULL, elicited.fractiles = TRUE,
  fitted.fractiles = FALSE, fitted.curve = FALSE, CI.prob = NULL,
  estimated.probs = NULL, modelled.fractiles = FALSE,
  modelled.curve = FALSE, cumul.prob.bounds = c(0.05, 0.95),
  theta.bounds = NULL, ylim.max = NULL, xlog = FALSE,
  design.table = TRUE, n.pts = 101)

```

**Arguments**

Z	list object that contains matrix theta of elicitations, character link and character target as initialised by <a href="#">designLink</a> and updated by <a href="#">elicitPt</a>
X	design matrix (can be NULL, unless modelled output is requested)
design.pt	single integer that denotes design point of interest
elicited.fractiles	logical, plot vertical lines for elicited fractiles?

fitted.fractiles	logical, plot vertical lines for fitted conditional mean prior fractiles for this design point? Alternatively, a numeric vector of arbitrary fractiles to be plotted from the fitted elicitation distribution. If TRUE then the fractiles corresponding to the median, upper and lower level central CI are plotted
fitted.curve,	logical plot fitted conditional mean prior density for this design point?
CI.prob	numeric scalar, locally specified probability assigned to the elicited central credible interval of the current design point. Defaults to NULL in which case the global value initially assigned by <code>designLink</code> or as updated by <code>elicitPt</code> is used
estimated.probs	numeric vector of values for which estimated probabilities are to be estimated from the fitted elicitation distribution for the target theta. Default is NULL. The result is output to the console.
modelled.fractiles	logical, plot vertical lines for modelled fractiles from the conditional mean prior distribution fit to all design points? This option requires a design matrix X of full column rank.
modelled.curve	logical, plot modelled conditional mean prior density for the entire model? This option requires a design matrix X of full column rank.
cumul.prob.bounds	numeric vector of length two, giving plot bounds by cumulative probability. This argument is ignored if there is not enough data to fit a parametric distribution or if <code>theta.bounds</code> is not NULL
theta.bounds	numeric vector giving support of response for plotting purposes (can be NULL). This will overwrite <code>cumul.prob.bounds</code> , if applicable
ylim.max	numeric maximum value of y-axis (can be NULL)
xlog	logical log x-axis
design.table	logical include design dataframe, elicited fractiles and modelled or fitted fractiles
n.pts	numeric giving number of point to evaluate density curve (if plotted)

**Value**

a plot to the current device. See `dev.cur()` to check.

**Examples**

```
# design matrix: two scenarios
X <- matrix(c(1, 1, 0, 1), nrow = 2)
rownames(X) <- c("scenario1", "scenario2")
colnames(X) <- c("covariate1", "covariate2")

# logit link
# central credible intervals with probability = 1/2
Z <- designLink(design = X, link = "logit", CI.prob = 0.5)
```

```

# 1st design point
# no elicited fractiles
indirect::plotDesignPoint(Z, design.pt = 1)
# elicited median
Z <- indirect::elicitPt(Z, design.pt = 1,
  lower.CI.bound = NA,
  median = 0.4,
  upper.CI.bound = NA,
  CI.prob = NULL)
indirect::plotDesignPoint(Z, design.pt = 1,
  elicited.fractiles = TRUE, theta.bounds = c(0, 1))
# lower and upper quartiles and median
Z <- indirect::elicitPt(Z, design.pt = 1,
  lower.CI.bound = 0.2,
  median = 0.4,
  upper.CI.bound = 0.6,
  comment = "Completed.")
indirect::plotDesignPoint(Z, design.pt = 1,
  elicited.fractiles = TRUE, theta.bounds = c(0, 1),
  fitted.fractiles = TRUE, fitted.curve = TRUE)
indirect::plotDesignPoint(Z, design.pt = 1,
  elicited.fractiles = TRUE, theta.bounds = c(0, 1),
  fitted.fractiles = c(1/10, 1/4, 1/2, 3/4, 9/10),
  fitted.curve = TRUE)

# second design point
# central credible intervals with probability = 1/3
# elicit upper and lower tertiles
Z <- elicitPt(Z, design.pt = 2,
  lower.CI.bound = 0.1,
  upper.CI.bound = 0.3,
  CI.prob = 1/3,
  comment = "Switched to tertiles.")
indirect::plotDesignPoint(Z, design.pt = 2,
  elicited.fractiles = TRUE, theta.bounds = c(0, 1))
indirect::plotDesignPoint(Z, design.pt = 2,
  elicited.fractiles = TRUE, theta.bounds = c(0, 1),
  fitted.fractiles = TRUE, fitted.curve = TRUE)
indirect::plotDesignPoint(Z, design.pt = 2,
  elicited.fractiles = TRUE, theta.bounds = c(0, 1),
  fitted.fractiles = c(1/10, 1/3, 1/2, 2/3, 9/10),
  fitted.curve = TRUE)

```

---

saveRecord

*Function to save elicitation record.*


---

## Description

Function to save elicitation record.

**Usage**

```
saveRecord(designLink.obj,  
           conclusion.comments = "This concludes the elicitation record.", file = "")
```

**Arguments**

`designLink.obj` list object initially created by function [designLink](#) and subsequently updated by function [elicitPt](#)

`conclusion.comments` character, comments to conclude session. Beware of non-ASCII text and special characters, which may affect ability to save or generate a Sweave document by using [makeSweave](#)

`file` character providing filename.

**Value**

an RDS file is created with filename `file`. A timestamp is added to `designLink.obj` using `Sys.time()`.

**Examples**

```
## Not run:  
X <- matrix(c(1, 1, 0, 1), nrow = 2) # design  
Z <- designLink(design = X)  
tmp <- tempfile(pattern = "report", fileext = ".rds")  
saveRecord(Z, file = tmp)  
  
## End(Not run)
```

# Index

checkX, [2](#)

CNdiag, [2](#)

designLink, [3](#), [4](#), [6](#), [7](#), [10–12](#), [14](#)

dGompertzNorm, [4](#)

dLogitNorm, [5](#)

elicitPt, [4](#), [6](#), [7](#), [11](#), [12](#), [14](#)

indirect, [7](#)

indirect-package (indirect), [7](#)

makeSweave, [3](#), [7](#), [14](#)

muSigma, [7](#), [8](#)

mV, [4](#), [7](#), [9](#), [9](#), [10](#)

pdist, [10](#)

plotDesignPoint, [7](#), [10](#), [11](#)

saveRecord, [3](#), [13](#)

Sweave, [3](#), [8](#)