Package ‘pmxcv’

March 26, 2024

Title Integration-Based Coefficients of Variance

Version 0.0.1.0

Description Estimate coefficient of variance percent (CV%) for any arbitrary distribution, including some built-in estimates for commonly-used transformations in pharmacometrics. Methods are described in various sources, but applied here as summarized in: Prybylski, (2024) <doi:10.1007/s40262-023-01343-2>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author John Prybylski [aut, cre] (<`https://orcid.org/0000-0001-5802-0539`>)

Maintainer John Prybylski <`john.prybylski@pfizer.com`>

Repository CRAN

Date/Publication 2024-03-26 17:00:09 UTC

R topics documented:

dist.intcv 2
dist.moment 2
intcv 3
invcv 3
moment 4
moment_f 4
nonmemboxcox 5
numcv 6

Index **7**

dist.intcv *Built-in integration-based %CV functions*

Description

Built-in integration-based %CV functions

Usage

```
dist.intcv(
  dist = "log",
  ...,
  exact = ifelse(dist == "log", TRUE, FALSE),
  lambda = NULL,
  fun = FALSE
)
```

Arguments

`dist`	Selection of built-in distributions.
`...`	passed to moment()
`exact`	If there is an exact moment generating function, use that. Default TRUE only for log
`lambda`	shape parameter for nonmemboxcox()
`fun`	return function (for use in invcv())

Value

Percent CV

dist.moment *Built-in moment functions*

Description

Built-in moment functions

Usage

```
dist.moment(
  dist = "log",
  ...,
  exact = ifelse(dist == "log", TRUE, FALSE),
  lambda = NULL
)
```

Arguments

`dist`	Selection of built-in distributions.
`...`	passed to moment()
`exact`	If there is an exact moment generating function, use that. Default TRUE only for log
`lambda`	shape parameter for nonmemboxcox()

Value

moment

`intcv` *Integration-based CV%*

Description

Integration-based CV%

Usage

```
intcv(...)
```

Arguments

`...`	Arguments passed to moment()

Value

Percent CV

`invcv` *Variance from CV%*

Description

Variance from CV%

Usage

```
invcv(cvfun, cv, verbose = FALSE, ...)
```

Arguments

`cvfun`	intcv()-based function
`cv`	CV% generated from cvfun
`verbose`	extra output
`...`	Other parameters to pass to cvfun

Value

Best-fit variance

`moment` *Moment function*

Description

Moment function

Usage

```
moment(...)
```

Arguments

`...`	all arguments passed to moment_f()

Value

moment

`moment_f` *Integratable moment function*

Description

Integratable moment function

Usage

```
moment_f(x, u, v, n, pdist, qdist)
```

Arguments

`x`	numeric vector
`u`	mean
`v`	variance
`n`	moment number
`pdist`	un-transform function for transformed random variable (eg, exp())
`qdist`	transform function (eg, log())

Value

Point result of the moment function

`nonmemboxcox` *Box-Cox transform typically used in NONMEM*

Description

Parameters are typically treated as lognormally-distributed by NONMEM users. Box-Cox transforms are typically applied to the exponentiated individual ETA parameters; this means the parameter is neither Box-Cox distributed nor lognormally-distributed, but both. To get the "Box-Cox Transform" as it would be relevant for CV% calculation, these properties have to be considered.

Usage

```
nonmemboxcox(x, lambda, theta = 1, inv = FALSE)
```

Arguments

`x`	random vector. Must be positive.
`lambda`	shape parameter
`theta`	centrality parameter
`inv`	inverse transform

Value

Box-Cox transformed or untransformed vector

numcv *Numeric CV% of a sample*

Description

Numeric CV% of a sample

Usage

```
numcv(x, ...)
```

Arguments

`x`	numeric vector
`...`	other arguments for sd() and mean()

Value

Percent cv

Examples

```
test_x <- rnorm(1000, mean=50, sd=5)
cv <- numcv(test_x)
cv # expect ~ 10(%)
```

Index

dist.intcv, 2
dist.moment, 2

intcv, 3
invcv, 3

moment, 4
moment_f, 4

nonmemboxcox, 5
numcv, 6