

Package: pmxcv (via r-universe)

August 25, 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](https://doi.org/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>

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

Repository <https://jpryby.r-universe.dev>

RemoteUrl <https://github.com/cran/pmxcv>

RemoteRef HEAD

RemoteSha abaca41318fe7b0b7da19ad9d6b8bf91442ab3d3

Contents

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

Index[7](#)

dist.intcv	<i>Built-in integration-based %CV functions</i>
------------	---

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	<i>Built-in moment functions</i>
-------------	----------------------------------

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	<i>Integration-based CV%</i>
-------	------------------------------

Description

Integration-based CV%

Usage

intcv(...)

Arguments

... Arguments passed to moment()

Value

Percent CV

invcv	<i>Variance from CV%</i>
-------	--------------------------

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	<i>Moment function</i>
--------	------------------------

Description

Moment function

Usage

moment(...)

Arguments

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

Value

moment

moment_f	<i>Integratable moment function</i>
----------	-------------------------------------

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	<i>Box-Cox transform typically used in NONMEM</i>
--------------	---

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

<code>x</code>	numeric vector
<code>...</code>	other arguments for <code>sd()</code> and <code>mean()</code>

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