

Package ‘rrscale’

March 13, 2019

Title Robust Re-Scaling to Better Recover Latent Effects in Data

Version 0.1.3

Description Non-linear transformations of data to better discover latent effects. Applies a sequence of three transformations (1) a Gaussianizing transformation, (2) a Z-score transformation, and (3) an outlier removal transformation.

Date 2019-3-11

Depends R (>= 3.4.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

Imports foreach, parallel, doParallel, DEoptim, abind

Suggests knitr, rmarkdown, testthat, ggplot2, reshape2

VignetteBuilder knitr

NeedsCompilation no

Author Gregory Hunt [aut, cre]

Maintainer Gregory Hunt <ghunt@wm.edu>

Repository CRAN

Date/Publication 2019-03-13 14:00:07 UTC

R topics documented:

asinh	2
box_cox	2
box_cox_exp	3
box_cox_negative	3
box_cox_plus1	4
box_cox_plusmin	4
box_cox_shift	5
center	5

gm_mean	6
list_transformations	6
log_box_cox	6
power	7
rrscale	7
svdc	9
winsor	10

Index	11
--------------	-----------

asinh	<i>Arc-hyperbolic-sine transformation</i>
-------	---

Description

- T the transformation with arguments Y, the data, lambda the parameter, and boolean inverse to calculate inverse transformation.
- T_deriv the transformation with arguments Y, the data, lambda the parameter.

Usage

asinh

Format

An object of class `list` of length 2.

box_cox	<i>Traditional box-cox power transformation. Accepts one real parameter</i>
---------	---

Description

- T the transformation with arguments Y, the data, lambda the parameter, and boolean inverse to calculate inverse transformation
- T_deriv the transformation with arguments Y, the data, lambda the parameter.

Usage

box_cox

Format

An object of class `list` of length 2.

box_cox_exp	<i>Exponential of the traditional box-cox transformation</i>
-------------	--

Description

- T the transformation with arguments Y, the data, lambda the parameter, and boolean inverse to calculate inverse transformation.
- T_deriv the transformation with arguments Y, the data, lambda the parameter.

Usage

```
box_cox_exp
```

Format

An object of class list of length 2.

box_cox_negative	<i>A generalized box-cox transformation that can handle negative data</i>
------------------	---

Description

- T the transformation with arguments Y, the data, lambda the parameter, and boolean inverse to calculate inverse transformation.
- T_deriv the transformation with arguments Y, the data, lambda the parameter.

Usage

```
box_cox_negative
```

Format

An object of class list of length 2.

box_cox_plus1	<i>Box-cox transformation with a shift of 1 added to the data</i>
---------------	---

Description

- T the transformation with arguments Y, the data, lambda the parameter, and boolean inverse to calculate inverse transformation.
- T_deriv the transformation with arguments Y, the data, lambda the parameter.

Usage

```
box_cox_plus1
```

Format

An object of class list of length 2.

box_cox_plusmin	<i>Box-cox transformation with the data shifted so that it is positive</i>
-----------------	--

Description

- T the transformation with arguments Y, the data, lambda the parameter, and boolean inverse to calculate inverse transformation.
- T_deriv the transformation with arguments Y, the data, lambda the parameter.

Usage

```
box_cox_plusmin
```

Format

An object of class list of length 2.

box_cox_shift	<i>Box-cox transformation of shifted variable</i>
---------------	---

Description

- T the transformation with arguments Y, the data, lambda the parameter, and boolean inverse to calculate inverse transformation. The parameter lambda has two real elements (1) the power and (2) the additive shift to the data.
- T_deriv the transformation with arguments Y, the data, lambda the parameter.

Usage

```
box_cox_shift
```

Format

An object of class list of length 2.

center	<i>Centers the data column-wise</i>
--------	-------------------------------------

Description

Centers the data column-wise

Usage

```
center(x)
```

Arguments

x the data.

gm_mean *Calculate the geometric mean*

Description

Calculate the geometric mean

Usage

```
gm_mean(x)
```

Arguments

x the data.

Examples

```
Y <- rlnorm(10)
gm <- gm_mean(Y)
```

list_transformations *List possible transformations*

Description

Returns list of transformations. Each transformation is a transformation function (“T”) accepting a parameter and the derivative of this transformation function (“T_deriv”).

Usage

```
list_transformations()
```

log_box_cox *Log of the traditional box-cox transformation*

Description

- T the transformation with arguments Y, the data, lambda the parameter, and boolean inverse to calculate inverse transformation.
- T_deriv the transformation with arguments Y, the data, lambda the parameter.

Usage

```
log_box_cox
```

Format

An object of class `list` of length 2.

power

Simple power transformation

Description

- `T` the transformation with arguments `Y`, the data, `lambda` the parameter, and boolean `inverse` to calculate inverse transformation.
- `T_deriv` the transformation with arguments `Y`, the data, `lambda` the parameter.

Usage

power

Format

An object of class `list` of length 2.

rrscale

Re-scale a data matrix

Description

This transformation is three steps (1) Gaussianize the data, (2) z-score Transform the data, and (3) remove extreme outliers from the data. The sequence of these transformations helps focus further analyses on consequential variance in the data rather than having it be focused on variation resulting from the feature's measurement scale or outliers.

Usage

```
rrscale(Y, trans_list = list(box_cox_negative = box_cox_negative, asinh =
  asinh), lims_list = list(box_cox_negative = c(-100, 100), asinh =
  list(0, 100)), opt_control = NULL, ncores = NULL, z = 4,
  q = 0.001, run_parallel = TRUE, verbose = FALSE, zeros = FALSE,
  opts = FALSE, seed = NULL)
```

Arguments

Y	Data matrix to be transformed.
trans_list	List of transformations to be considered. See function <code>list_transformations</code> . Each element of the list should be a list containing the transformation function as the first element and the derivative of the transformation function as the second argument. The first argument of each function should be the data, the second the transformation parameter.
lims_list	List of optimization limits for each transformation from <code>trans_list</code> . This should be a list with one element per transformation parameter. The element of the list for each transformation family should be a list of two-element vectors control the limits for each parameter of the transformation family.
opt_control	Optional optimization controlling parameters for DEoptim control argument. See the DEoptim package for details.
ncores	Number of cores to use if running parallel.
z	The O-step cutoff value. Points are removed if their robust z-score is above z in magnitude.
q	The Z-step winsorizing quantile cutoff. The quantile at which to winsorize the data when calculating the robust z-scores.
run_parallel	a boolean, if TRUE the method will be run run column-wise in parallel.
verbose	a boolean, if TRUE then save optimization output in local directory <code>'.rrscale'</code>
zeros	How to deal with zeros in the data set. If set to FALSE the algorithm will fail if it encounters a zero. If set to a number or 'NA' then the zeros are replaced by this number or 'NA'.
opts	Boolean determining if optimization output is returned. Defaults to FALSE.
seed	Sets the seed before running any other analyses.

Value

A list of output:

- `opts`: the optimization output for all transformation families and all columns
- `pars`: the optimal parameters for each column for the optimal family
- `par_hat`: the estimated optimal parameter
- `NT`: the original data
- `RR`: the robust-rescaled data
- `G`: gaussianized data
- `Z`: robust z-transformed data
- `O`: data with outliers removed
- `rr_fn`: a function to apply the estimated RR transformation to new data. Takes arguments
 - Y: the data,
 - z: the z-score cutoff (defaults to 4),
 - q: the winsorizing quantile cutoff (defaults to 0.001),

- lambda: the transformation parameter to use (defaults to the estimated one),
 - T: the transformation function family (defaults to the optimal estimated family),
 - mu: the mean to be used in the robust z-score step (re-estimates if NULL)
 - sigma: the s.d. to be used in the robust z-score step (re-estimates if NULL)
- T: the optimal family
 - T_deriv: the derivative of the optimal family
 - T_name: name of the optimal family
 - alg_control: the parameters passed to the algorithm

Examples

```
Y <- rlnorm(10)**t(rlnorm(10))
rr.out <- rrscale(Y,run_parallel=FALSE)
Yt <- rr.out$RR
```

svdc

The completed SVD

Description

This calculates right and left singular vectors of a data matrix possibly containing missing values.

Usage

```
svdc(X, nu = NULL, nv = NULL)
```

Arguments

X	the data matrix of which to calculate the completed SVD.
nu	the number of left singular vectors to calculate
nv	the number of right singular vectors to calculate

Examples

```
Y <- rnorm(10)**t(rnorm(10))
Y[1,1] <- NA
svdc.out <- svdc(Y)
```

winsor	<i>Winsorizes the data</i>
--------	----------------------------

Description

Winsorizes the data

Usage

```
winsor(x, fraction = 0.01)
```

Arguments

x	the data.
fraction	the top and bottom quantiles to cap.

Examples

```
Y <- rlnorm(10)%*%t(rlnorm(10))  
Yw <- winsor(Y,1E-2)
```

Index

*Topic **datasets**

- asinh, [2](#)
- box_cox, [2](#)
- box_cox_exp, [3](#)
- box_cox_negative, [3](#)
- box_cox_plus1, [4](#)
- box_cox_plusmin, [4](#)
- box_cox_shift, [5](#)
- log_box_cox, [6](#)
- power, [7](#)

asinh, [2](#)

box_cox, [2](#)

box_cox_exp, [3](#)

box_cox_negative, [3](#)

box_cox_plus1, [4](#)

box_cox_plusmin, [4](#)

box_cox_shift, [5](#)

center, [5](#)

gm_mean, [6](#)

list_transformations, [6](#)

log_box_cox, [6](#)

power, [7](#)

rrscale, [7](#)

svdc, [9](#)

winsor, [10](#)