Package: normalr (via r-universe)

August 23, 2024

Title Normalisation of Multiple Variables in Large-Scale Datasets

Version 1.0.1

Description The robustness of many of the statistical techniques, such as factor analysis, applied in the social sciences rests upon the assumption of item-level normality. However, when dealing with real data, these assumptions are often not met. The Box-Cox transformation (Box & Cox, 1964) <http://www.jstor.org/stable/2984418> provides an optimal transformation for non-normal variables. Yet, for large datasets of continuous variables, its application in current software programs is cumbersome with analysts having to take several steps to normalise each variable. We present an R package 'normalr' that enables researchers to make convenient optimal transformations of multiple variables in datasets. This R package enables users to quickly and accurately: (1) anchor all of their variables at 1.00, (2) select the desired precision with which the optimal lambda is estimated, (3) apply each unique exponent to its variable, (4) rescale resultant values to within their original X1 and X(n) ranges, and (5) provide original and transformed estimates of skewness, kurtosis, and other inferential assessments of normality.

Depends R (>= 3.3.0)

License GPL

Encoding UTF-8

LazyData true

Imports MASS, parallel, purrr, magrittr, rlang, shiny

Suggests testthat, covr

RoxygenNote 7.1.1

URL https://github.com/kcha193/normalr

BugReports https://github.com/kcha193/normalr/issues

Config/testthat/edition 3

Repository https://kcha193.r-universe.dev

RemoteUrl https://github.com/kcha193/normalr

RemoteRef HEAD

RemoteSha 0f97851949f572bd69ce04fe9a74eb00299e0276

Contents

getLambda 2
normalise 3
normaliseData 3
normaliseDataInfo 4
normalrShiny 5
testData 5

Index **6**

getLambda *Get Optimal Lambda value*

Description

Computes optimal lambda value using boxcox function from the provided data.

Usage

```
getLambda(dat, lambda = seq(-10, 10, 1/100), parallel = TRUE)
```

Arguments

`dat`	a data frame containing the variables of numeric or integer vectors.
`lambda`	a vector of values of lambda – default (-10, 10) in steps of 0.01.
`parallel`	perform the computation in parallel, default setting is TRUE.

Value

a numeric vector

References

Courtney, M. G. R., and Chang, K. C. (2018) Dealing with non-normality: an introduction and step-by-step guide using R. TEST, doi: 10.1111/test.12154.

Examples

```
## Not run:  getLambda(mtcars)
getLambda(mtcars, parallel = FALSE)
```

normalise — *Apply normalisation on a numeric vector using a specific Lambda value*

Description

Apply normalisation on a numeric vector using a specific Lambda value

Usage

```
normalise(x, lambda = 3)
```

Arguments

`x` a numeric vector to be normalised.

`lambda` a numeric vector from the boxcox function

Value

a numeric vector with lambda, scale and the transformed variable

References

Courtney, M. G. R., and Chang, K. C. (2018) Dealing with non-normality: an introduction and step-by-step guide using R. TEST, doi: 10.1111/test.12154.

Examples

```
x <- c(1, 5, 9, 9, 9, 9, 10, 10, 10, 11, 11, 12)
normalise(x, lambda = 3)
```

normaliseData — *Apply normalisation on a data frame using specific Lambda value*

Description

Apply normalisation on a data frame using specific Lambda value

Usage

```
normaliseData(dat, lambdas)
```

Arguments

`dat` a data frame containing the variables.

`lambdas` a numeric vector from the boxcox function

Value

a data frame

References

Courtney, M. G. R., and Chang, K. C. (2018) Dealing with non-normality: an introduction and step-by-step guide using R. TEST, doi: 10.1111/test.12154.

Examples

```
## Not run: normaliseData(mtcars, getLambda(mtcars, parallel = FALSE))
normaliseData(mtcars, getLambda(mtcars, parallel = FALSE))
```

normaliseDataInfo *Apply normalisation on a data frame using specific Lambda value*

Description

Apply normalisation on a data frame using specific Lambda value

Usage

```
normaliseDataInfo(dat, lambdas)
```

Arguments

dat	a data frame containing the variables.
lambdas	a numeric vector from the boxcox function

Value

a list of three elements, the first element, lambdas, is a numeric vector containing lambdas, the second element, scaled, is a numeric vector containing scale adjustment, the last element, transDat, is a data frame comprising the transformed variables.

References

Courtney, M. G. R., and Chang, K. C. (2018) Dealing with non-normality: an introduction and step-by-step guide using R. TEST, doi: 10.1111/test.12154.

Examples

```
## Not run: normaliseDataInfo(mtcars, getLambda(mtcars, parallel = FALSE))
normaliseDataInfo(mtcars, getLambda(mtcars, parallel = FALSE))
```

`normalrShiny` *Shiny application of the normalr*

Description

Shiny application of the normalr

Usage

```
normalrShiny(example = "normalr")
```

Arguments

`example`	name of the shiny apps

Examples

```
## Not run: normalrShiny()
```

`testData` *Test dataset for the paper*

Description

Test dataset for the paper

Usage

```
testData
```

Format

An object of class `data.frame` with 957 rows and 9 columns.

Index

∗ **datasets**
    testData, 5

boxcox, *2–4*

getLambda, 2

normalise, 3
normaliseData, 3
normaliseDataInfo, 4
normalrShiny, 5

testData, 5