Package ‘jtrans’

August 29, 2016

Title Johnson Transformation for Normality

Version 0.2.1

Description Transforming univariate non-normal data to normality using Johnson families of distributions. Johnson family is a comprehensive distribution family that accommodates many kinds of non-normal distributions. A bunch of distributions with various parameters will be fit and the corresponding p-values under a user-specified normality test will be given. The final transformation will be the one with the largest p-value under the given normality test.

Depends R (>= 3.1.0)

Suggests nortest, knitr

License GPL-2

Author Yuchen Wang [aut, cre]

Maintainer Yuchen Wang <ycwang0712@gmail.com>

NeedsCompilation no

Repository CRAN

Date/Publication 2015-03-18 09:52:46

R topics documented:

fit_sb *Fit Functions for Johnson Curves*

Description

Fit functions for three Johnson Curves.

Usage

```
fit_sb(x, q)

fit_su(x, q)

fit_sl(x, q)

qtls(x, z)
```

Arguments

x	the non-normal numerical data.
q	the quantiles and some statistics generated by quantiles, it must be the return value of `qtls` function.
z	a single z value for model fitting. It's returned by `jtrans`

Details

Three types of transformations are SB, SL and SU. Their forms are described below:

$$S_B : Z = \gamma + \eta * ln((X - \epsilon)/(\lambda + \epsilon - X))$$

$$S_L : Z = \gamma + \eta * ln(X - \epsilon)$$

$$S_U : Z = \gamma + \eta * asinh((X - \epsilon)/\lambda)$$

in whihc Z is the standard normal varible, and X is the non-normal original data, all the necessary parameters will be returned. Before fitting these curves, sample quantiles should be calculated according to z values. the `qtls` function here is to provide every useful parameters for Johnson curve fitting.

These functions could also be used for predicting new values when you have already fitted a model and obtained a `jtrans` object. This could be done by set the `newx` parameter. See examples for details.

Note that when predicting new data, the new data should be from the same distribution as the original data used for fitting the model. All fits have certain restrictions on data range, if the new data is outside the range of the model, the prediction will return NA for all the values. Try to exclude some out-of-range values and predict again may fix this problem.

Value

return NA when the prediction failed, return a list with 2 component when fit succeeded. The first component `trans` is the transformed value and the second component `params` is the parameters used in the transformation.

`jtrans` *Johnson Transformation for Normality*

Description

`jtrans` transforms a continuous univariate vector to a random vector from standard normal distribution.

Usage

```
jtrans(x, test = "shapiro.test", exclude_original = TRUE, z_lim = c(0.25,
  1.25), z_length = 101)
```

Arguments

`x`	the non-normal numerical data.
`test`	the normality test used to select fits, defaults to `shapiro.test`
`exclude_original`	whether the original data should be excluded when comparing fits.
`z_lim`	two values vector defining the range of the z values, defaults to 0.25 to 1.25, which is recommended by Mandraccia, Halverson and Chou (1996).
`z_length`	the length of the z vector, default to 101. The number of different fits estiamted in the algorithm. Set larger z.length value if you want extra precision.

Details

`jtrans` fits data to a set of distributions from Johnson family. A normality test is used to find the best fit by choosing the fit with maximum p.value under that given test. It returns the transformed data, the corresponding type of Johnson curve and parameter estimations.

Since the default Shapiro-Wilk test can only accept sample size between 3 and 5000, one should specify another normality test in the test parameter, generally the `ad.test` in the **nortest** package is recommended.

Sometimes, this algorithm may return poor fits. The most extreme case is that all the transformed data have smaller p.values than the original data's. In such cases, the `exclude_original` flag should be set to FALSE, so `jtrans` will return the original data as the transformed data.

Value

A list with two classes: the first one is the type of transformation used, the same as the `type` component, could be "sb", "su" or "sl"; The second one is "jtrans". The list containsthe following components:

`original` original data.

`transformed` transformed data.

`type` type of transformation selected.

`test` normality test used to select transformations.

`z` selected z value among 101 values from 0.25 to 1.25.

`eta, gamma, lambda, epsilon`
transformation parameters.

`p.value` the maximum p.value returned by test

References

Chou, Y. M., Polansky, A. M., & Mason, R. L. (1998). Transforming non-normal data to normality in statistical process control. Journal of Quality Technology, 30(2), 133-141.

Examples

```
# generate 100 non-normal data and transform it.
x <- rexp(50, .2)
jt <- jtrans(x)
jt

plot(density(x))
plot(density(jt$transformed))
qqnorm(jt$transformed)
qqline(jt$transformed)

## Not run:
# Using another normality test
require(nortest)
jtrans(x, test = "ad.test")

## End(Not run)
```

`predict.sb` *Predict functions for Johnson Transformation*

Description

Generic functions to apply the fitted Johnson transformation to transform new data.

Usage

```
## S3 method for class 'sb'
predict(object, newdata, ...)

## S3 method for class 'su'
predict(object, newdata, ...)

## S3 method for class 'sl'
predict(object, newdata, ...)
```

Arguments

`object` a `jtrans` object with a specific type

`newdata` new data to be fitted

`...` further arguments to match `predict`

Details

After the johnson transformation is used, if you want to use the same transformation on different data, you can use these functions. This is designed to be the same functionality as the `predict` functions.

Value

Numeric vector of the transformed values

#'

Examples

```
#' # if you want to predict based on a fitted distribution, you must set the
# parameters in the qtls() function using the fitted model object jt.

jt <- jtrans(rexp(300, .4))

# good prediction
predict(jt, rexp(10, .4))

# will generate NaN because newx is from different distribution
predict(jt, rexp(10, .1))
```

`transeq` *Export the transformation equation into LaTeX*

Description

`transeq` turns a `jtrans` object into a LaTeX equation for display.

Usage

```
transeq(obj, digits = 4)

## S3 method for class 'sb'
transeq(obj, digits)

## S3 method for class 'su'
transeq(obj, digits)
```

Arguments

`obj`	a `jtrans` object with a specific transformation type
`digits`	digits displayed in the equation

Details

A LaTeX equation in the display mode, e.g. between `\[` and `\]` is returned with the formula used in the transformation. Note that when it's displayed in the R console, the backslashes are escaped. So it's always double backslash when in print it in the terminal.

This is designed to work with **knitr** and **rmarkdown**. In this case you can set the chunk option `results='asis'` and output it to a PDF document. Then the LaTeX equation will be properly formatted and can be easily included in your report.

Examples

~~~
## Not run:
# designed to be used with R Markdown and chunk options
```{r, results='asis'}
library(jtrans)
jt <- jtrans(rexp(30, .3))
transeq(jt)
```

## End(Not run)
~~~

Index