Package ‘trafo’

November 27, 2018

Title Estimation, Comparison and Selection of Transformations
Version 1.0.1
Date 2018-11-26

Depends R (>= 3.2.4)
License GPL-2
Encoding UTF-8
LazyData true
RoxygenNote 6.1.1
Imports FNN, moments, pryr, graphics, grDevices, lmtest
Suggests testthat, R.rsp
VignetteBuilder R.rsp
NeedsCompilation no
Author Lily Medina [aut], Piedad Castro [aut], Ann-Kristin Kreutzmann [aut, cre], Natalia Rojas-Perilla [aut]
Maintainer Ann-Kristin Kreutzmann <ann-kristin.kreutzmann@fu-berlin.de>
Repository  CRAN
Date/Publication  2018-11-27 16:30:04 UTC

R topics documented:

as.data.frame.trafo ........................................... 3
assumptions ...................................................... 4
bickeldoksum .................................................... 5
boxcox .......................................................... 6
diagnostics ....................................................... 7
diagnostics.trafo_compare ...................................... 8
diagnostics.trafo_lm ........................................... 9
dual .............................................................. 10
glog ............................................................. 11
gpower .......................................................... 12
logshiftopt ..................................................... 13
logtrafo ......................................................... 14
manly ........................................................... 15
modulus .......................................................... 16
neglog ........................................................... 17
plot.trafo_compare ............................................. 18
plot.trafo_lm ................................................... 18
print.diagnostics.trafo_compare ............................... 19
print.diagnostics.trafo_lm .................................. 19
print.summary.trafo_compare ................................. 20
print.summary.trafo_lm ...................................... 20
print.trafo ...................................................... 21
print.trafo_compare .......................................... 21
print.trafo_lm .................................................. 22
reciprocal ....................................................... 22
sqrtshift ......................................................... 23
summary.trafo_compare ....................................... 24
summary.trafo_lm ............................................. 24
trafo ............................................................ 25
trafo_compare .................................................. 25
trafo_lm ........................................................ 26
yeojohnson ....................................................... 27

Index 29
as.data.frame.trafo  Data frame with transformed variables

Description

The data frame that is returned contains the variables that are used in the model and additionally a variable with the transformed dependent variable. To the variable name of the dependent variable a t is added for transformed.

Usage

## S3 method for class 'trafo'
as.data.frame(x, row.names = NULL, optional = FALSE, std = FALSE, ...)

Arguments

- **x**: an object of type trafo.
- **row.names**: NULL or a character vector giving the row names for the data frame. Missing values are not allowed.
- **optional**: logical. If TRUE, setting row names and converting column names (to syntactic names: see make.names) is optional. Note that all of R’s base package as.data.frame() methods use optional only for column names treatment, basically with the meaning of data.frame(*, check.names = !optional)
- **std**: logical. If TRUE, the data is transformed by the standardized/scaled transformation. Defaults to FALSE.
- **...**: other parameters that can be passed to the function.

Value

A data frame with the original variables and the transformed variable.

See Also

- bickeldoksum, boxcox, dual, glog, gpower, log, logshiftopt, manly, modulus, neglog, sqrtshift, yeojohnson

Examples

# Load data
data("cars", package = "datasets")

# Fit linear model
lm_cars <- lm(dist ~ speed, data = cars)

# Transform dependent variable using divergence minimization following # Kolmogorov-Smirnov
logshiftopt_trafo <- logshiftopt(object = lm_cars, method = "div.ks", plotit = FALSE)

# Get a data frame with the added transformed variable
as.data.frame(logshiftopt_trafo)

assumptions

First check of assumptions to find suitable transformations

Description

Gives a first overview if a transformation is useful and which transformation is promising to fulfill the model assumptions normality, homoscedasticity and linearity.

Usage

assumptions(object, method = "ml", std = FALSE, ...)

Arguments

object

an object of type lm.

method

a character string. Different estimation methods can be used for the estimation of the optimal transformation parameter: (i) Maximum likelihood approach ("ml"), (ii) Skewness minimization ("skew"), (iii) Kurtosis optimization ("kurt"), (iv) Divergence minimization by Kolmogorov-Smirnov ("div.ks"), by Cramer-von-Mises ("div.cvm") or by Kullback-Leibler ("div.kl"). Defaults to "ml".

std

logical. If TRUE, the transformed model is returned based on the standardized/scaled transformation. Defaults to FALSE.

... other parameters that can be passed to the function, e.g. other lambdaranges. Self-defined lambdaranges are given to the function as an argument that is the combination of the name of the transformation and lr and the range needs to be a numeric vector of length 2. For instance, changing the lambdarange for the Manly transformation would mean to add an argument manly_lr = c(0.000005, 0.00005).

For the default values that are used for the lambdaranges see the documentation for the provided transformations.

Value

A table with tests for normality and homoscedasticity. Furthermore, scatterplots are returned to check the linearity assumption.

See Also

bickeldoksum, boxcox, dual, glog, gpower, log, logshiftopt, manly, modulus, neglog, sqrtshift, yeojohnson
Examples

# Load data
data("cars", package = "datasets")

# Fit linear model
lm_cars <- lm(dist ~ speed, data = cars)

assumptions(lm_cars)
assumptions(lm_cars, method = "skew", manly_lr = c(0.000005, 0.00005))

bickeldoksum

Bickel-Doksum transformation for linear models

Description

The function transforms the dependent variable of a linear model using the Bickel-Doksum transformation. The transformation parameter can either be estimated using different estimation methods or given.

Usage

bickeldoksum(object, lambda = "estim", method = "ml",
lambda.range = c(1e-11, 2), plotit = TRUE)

Arguments

object

an object of type lm.

lambda

either a character named "estim" if the optimal transformation parameter should be estimated or a numeric value determining a given value for the transformation parameter. Defaults to "estim".

method

a character string. Different estimation methods can be used for the estimation of the optimal transformation parameter: (i) Maximum likelihood approach ("ml"), (ii) Skewness minimization ("skew"), (iii) Kurtosis optimization ("kurt"), (iv) Divergence minimization by Kolmogorov-Smirnov ("div.ks"), by Cramer-von-Mises ("div.cvm") or by Kullback-Leibler ("div.kl"). Defaults to "ml".

lambda.range

a numeric vector with two elements defining an interval that is used for the estimation of the optimal transformation parameter. The Bickel-Doksum transformation is only defined for positive values of lambda. Defaults to c(1e-11, 2).

plotit

logical. If TRUE, a plot that illustrates the optimal transformation parameter or the given transformation parameter is returned. Defaults to TRUE.

Value

An object of class trafo. Methods such as as.data.frame.trafo and print.trafo can be used for this class.
References


Examples

```r
# Load data
data("cars", package = "datasets")

# Fit linear model
lm_cars <- lm(dist ~ speed, data = cars)

# Transform dependent variable using a maximum likelihood approach
bickeldoksum(object = lm_cars, plotit = FALSE)
```

---

**boxcox**

**Box-Cox transformation for linear models**

**Description**

The function transforms the dependent variable of a linear model using the Box-Cox transformation. The transformation parameter can either be estimated using different estimation methods or given. The Box-Cox transformation is only defined for positive response values. In case the response contains zero or negative values a shift is automatically added such that $y + \text{shift} > 0$.

**Usage**

```r
boxcox(object, lambda = "estim", method = "ml", lambdarange = c(-2, 2), plotit = TRUE)
```

**Arguments**

- `object` an object of type `lm`.
- `lambda` either a character named "estim" if the optimal transformation parameter should be estimated or a numeric value determining a given value for the transformation parameter. Defaults to "estim".
- `method` a character string. Different estimation methods can be used for the estimation of the optimal transformation parameter: (i) Maximum likelihood approach ("ml"), (ii) Skewness minimization ("skew"), (iii) Kurtosis optimization ("kurt"), (iv) Divergence minimization by Kolmogorov-Smirnov ("div.ks"), by Cramer-von-Mises ("div.cvm") or by Kullback-Leibler ("div.kl"). Defaults to "ml".
- `lambdarange` a numeric vector with two elements defining an interval that is used for the estimation of the optimal transformation parameter. Defaults to c(-2, 2).
- `plotit` logical. If TRUE, a plot that illustrates the optimal transformation parameter or the given transformation parameter is returned. Defaults to TRUE.
diagnostics

Value

An object of class trafo. Methods such as \texttt{as.data.frame.trafo} and \texttt{print.trafo} can be used for this class.

References


Examples

\begin{verbatim}
# Load data
data("cars", package = "datasets")

# Fit linear model
lm_cars <- lm(dist ~ speed, data = cars)

# Transform dependent variable using skewness minimization
boxcox(object = lm_cars, method = "skew", plotit = FALSE)
\end{verbatim}

\begin{verbatim}
diagnostics                  Diagnostics for fitted models
\end{verbatim}

Description

Returns information about the transformation and selected diagnostics to check model assumptions.

Usage

diagnostics(object, ...)

Arguments

object an object that contains two models that should be compared.
...
other parameters that can be passed to the function.

Value

The return depends on the class of its argument. The documentation of particular methods gives detailed information about the return of that method.

See Also

diagnostics.trafo_lm, diagnostics.trafo_compare
Diagnostics for two differently transformed models

Description

Returns information about the applied transformations and selected diagnostics to check model assumptions. Two models are compared where the dependent variable is transformed by different transformations.

Usage

```r
# S3 method for class 'trafo_compare'
diagnostics(object, ...)
```

Arguments

- `object` an object of type `trafo_compare`
- `...` additional arguments that are not used in this method

Value

An object of class `diagnostics.trafo_compare`. The method `print.diagnostics.trafo_compare` can be used for this class.

Examples

```r
# Load data
data("cars", package = "datasets")

# Fit linear model
lm_cars <- lm(dist ~ speed, data = cars)

# Transform with Bickel-Doksum transformation
bd_trafo <- bickeldoksum(object = lm_cars, plotit = FALSE)

# Transform with Box-Cox transformation
bc_trafo <- boxcox(object = lm_cars, method = "skew", plotit = FALSE)

# Compare transformed models
compare <- trafo_compare(object = lm_cars, trafos = list(bd_trafo, bc_trafo))

# Get diagnostics
diagnostics(compare)
```
**Description**

Returns information about the applied transformation and selected diagnostics to check model assumptions. The return helps to compare the untransformed and the transformed model with regard to model assumptions.

**Usage**

```r
## S3 method for class 'trafo_lm'
diagnostics(object, ...)
```

**Arguments**

- `object`  
  an object of type `trafo_lm`
- `...`  
  additional arguments that are not used in this method

**Value**

An object of class `diagnostics.trafo_lm`. The method `print.diagnostics.trafo_lm` can be used for this class.

**Examples**

```r
# Load data
data("cars", package = "datasets")

# Fit linear model
lm_cars <- lm(dist ~ speed, data = cars)

# Compare transformed models
bd_lm <- trafo_lm(object = lm_cars, trafo = "bickeldoksum", 
  method = "skew", lambdarange = c(1e-11, 2))

# Get diagnostics
diagnostics(bd_lm)
```
**dual**

*Dual transformation for linear models*

**Description**

The function transforms the dependent variable of a linear model using the Dual transformation. The transformation parameter can either be estimated using different estimation methods or given.

**Usage**

```r
dual(object, lambda = "estim", method = "ml", lambdarange = c(0, 2), plotit = TRUE)
```

**Arguments**

- **object**: an object of type `lm`.
- **lambda**: either a character named "estim" if the optimal transformation parameter should be estimated or a numeric value determining a given value for the transformation parameter. Defaults to "estim".
- **method**: a character string. Different estimation methods can be used for the estimation of the optimal transformation parameter: (i) Maximum likelihood approach ("ml"), (ii) Skewness minimization ("skew"), (iii) Kurtosis optimization ("kurt"), (iv) Divergence minimization by Kolmogorov-Smirnov ("div.ks"), by Cramer-von-Mises ("div.cvm") or by Kullback-Leibler ("div.kl"). Defaults to "ml".
- **lambdarange**: a numeric vector with two elements defining an interval that is used for the estimation of the optimal transformation parameter. The Dual transformation is not defined for negative values of lambda. Defaults to `c(0, 2)`.
- **plotit**: logical. If `TRUE`, a plot that illustrates the optimal transformation parameter or the given transformation parameter is returned. Defaults to `TRUE`.

**Value**

An object of class `trafo`. Methods such as `as.data.frame.trafo` and `print.trafo` can be used for this class.

**References**


**Examples**

```r
# Load data
data("cars", package = "datasets")

# Fit linear model
lm_cars <- lm(dist ~ speed, data = cars)
```
# Transform dependent variable using divergence minimization following
# Cramer-von-Mises
dual(object = lm_carsL method = "div.cvm", plotit = TRUE)

---

### glog

**Glog transformation for linear models**

#### Description

The function transforms the dependent variable of a linear model using the Glog transformation.

#### Usage

```
glog(object)
```

#### Arguments

- `object` an object of type `lm`.

#### Value

An object of class `trafo`. Methods such as `as.data.frame.trafo` and `print.trafo` can be used for this class.

#### References


#### Examples

```
# Load data
data("cars", package = "datasets")

# Fit linear model
lm_cars <- lm(dist ~ speed, data = cars)

# Transform dependent variable
glog(object = lm_cars)
```
gpower

Gpower transformation for linear models

Description
The function transforms the dependent variable of a linear model using the Gpower transformation. The transformation parameter can either be estimated using different estimation methods or given.

Usage

gpower(object, lambda = "estim", method = "ml", lambdarange = c(-2, 2), plotit = TRUE)

Arguments

object    an object of type lm.
lambda    either a character named "estim" if the optimal transformation parameter should be estimated or a numeric value determining a given value for the transformation parameter. Defaults to "estim".
method    a character string. Different estimation methods can be used for the estimation of the optimal transformation parameter: (i) Maximum likelihood approach ("ml"), (ii) Skewness minimization ("skew"), (iii) Kurtosis optimization ("kurt"), (iv) Divergence minimization by Kolmogorov-Smirnov ("div.ks"), by Cramer-von-Mises ("div.cvm") or by Kullback-Leibler ("div.kl"). Defaults to "ml".
lambdarange a numeric vector with two elements defining an interval that is used for the estimation of the optimal transformation parameter. Defaults to c(-2, 2).
plotit    logical. If TRUE, a plot that illustrates the optimal transformation parameter or the given transformation parameter is returned. Defaults to TRUE.

Value
An object of class trafo. Methods such as as.data.frame.trafo and print.trafo can be used for this class.

References

Examples

# Load data
data("cars", package = "datasets")

# Fit linear model
lm_cars <- lm(dist ~ speed, data = cars)
logshiftopt

# Transform dependent variable using divergence minimization following # Kullback-Leibler
gpower(object = lm_carsL method = "div.kl", plotit = FALSE)

logshiftopt

*Log shift opt transformation for linear models*

**Description**

The function transforms the dependent variable of a linear model using the Log shift opt transformation. The transformation parameter can either be estimated using different estimation methods or given.

**Usage**

logshiftopt(object, lambda = "estim", method = "ml", lambdarange = NULL, plotit = TRUE)

**Arguments**

- **object**: an object of type lm.
- **lambda**: either a character named "estim" if the optimal transformation parameter should be estimated or a numeric value determining a given value for the transformation parameter. Defaults to "estim".
- **method**: a character string. Different estimation methods can be used for the estimation of the optimal transformation parameter: (i) Maximum likelihood approach ("ml"), (ii) Skewness minimization ("skew"), (iii) Kurtosis optimization ("kurt"), (iv) Divergence minimization by Kolmogorov-Smirnov ("div.ks"), by Cramer-von-Mises ("div.cvm") or by Kullback-Leibler ("div.kl"). Defaults to "ml".
- **lambdarange**: a numeric vector with two elements defining an interval that is used for the estimation of the optimal transformation parameter. Defaults to NULL. In this case the lambdarange is set to the range of the data. In case the lowest value is negative the absolute value of the lowest value plus 1 is the lower bound for the range.
- **plotit**: logical. If TRUE, a plot that illustrates the optimal transformation parameter or the given transformation parameter is returned. Defaults to TRUE.

**Value**

An object of class trafo. Methods such as `as.data.frame.trafo` and `print.trafo` can be used for this class.
logtrafo

Log transformation for linear models

Description

The function transforms the dependent variable of a linear model using the Log transformation. The Log transformation is only defined for positive response values. In case the response contains zero or negative values a shift is automatically added such that \( y + \text{shift} > 0 \).

Usage

logtrafo(object)

Arguments

object an object of type lm.

Value

An object of class trafo. Methods such as \texttt{as.data.frame.trafo} and \texttt{print.trafo} can be used for this class.

References


Examples

# Load data
data("cars", package = "datasets")

# Fit linear model
lm_cars <- lm(dist ~ speed, data = cars)

# Transform dependent variable using divergence minimization following
# Kolmogorov-Smirnoff
logshiftopt(object = lm_cars, method = "div.ks", plotit = FALSE)
**Description**

The function transforms the dependent variable of a linear model using the Manly transformation. The transformation parameter can either be estimated using different estimation methods or given.

**Usage**

```r
manly(object, lambda = "estim", method = "ml", lambdarange = c(-2, 2), plotit = TRUE)
```

**Arguments**

- `object`: an object of type `lm`.
- `lambda`: either a character named "estim" if the optimal transformation parameter should be estimated or a numeric value determining a given value for the transformation parameter. Defaults to "estim".
- `method`: a character string. Different estimation methods can be used for the estimation of the optimal transformation parameter: (i) Maximum likelihood approach ("ml"), (ii) Skewness minimization ("skew"), (iii) Kurtosis optimization ("kurt"), (iv) Divergence minimization by Kolmogorov-Smirnov ("div.ks"), by Cramer-von-Mises ("div.cvm") or by Kullback-Leibler ("div.kl"). Defaults to "ml".
- `lambdarange`: a numeric vector with two elements defining an interval that is used for the estimation of the optimal transformation parameter. Defaults to `c(-2, 2)`.
- `plotit`: logical. If TRUE, a plot that illustrates the optimal transformation parameter or the given transformation parameter is returned. Defaults to TRUE.

**Value**

An object of class `trafo`. Methods such as `as.data.frame.trafo` and `print.trafo` can be used for this class.

**References**


**Examples**

```r
# Load data
data("cars", package = "datasets")

# Fit linear model
lm_cars <- lm(dist ~ speed, data = cars)
```
modulus

Modulus transformation for linear models

Description

The function transforms the dependent variable of a linear model using the Modulus transformation. The transformation parameter can either be estimated using different estimation methods or given.

Usage

modulus(object, lambda = "estim", method = "ml", lambdarange = c(-2, 2), plotit = TRUE)

Arguments

object an object of type lm.
lambda either a character named "estim" if the optimal transformation parameter should be estimated or a numeric value determining a given value for the transformation parameter. Defaults to "estim".
method a character string. Different estimation methods can be used for the estimation of the optimal transformation parameter: (i) Maximum likelihood approach ("ml"), (ii) Skewness minimization ("skew"), (iii) Kurtosis optimization ("kurt"), (iv) Divergence minimization by Kolmogorov-Smirnov ("div.ks"), by Cramer-von-Mises ("div.cvm") or by Kullback-Leibler ("div.kl"). Defaults to "ml".
lambdarange a numeric vector with two elements defining an interval that is used for the estimation of the optimal transformation parameter. Defaults to c(-2, 2).
plotit logical. If TRUE, a plot that illustrates the optimal transformation parameter or the given transformation parameter is returned. Defaults to TRUE.

Value

An object of class trafo. Methods such as as.data.frame.trafo and print.trafo can be used for this class.

References

**neglog**  

**Examples**

```r
# Load data
data("cars", package = "datasets")

# Fit linear model
lm_cars <- lm(dist ~ speed, data = cars)

# Transform dependent variable with fixed lambda
modulus(object = lm_cars, lambda = 0.8, plotit = FALSE)
```

---

**neglog**  

**Neg log transformation for linear models**

**Description**

The function transforms the dependent variable of a linear model using the Neg log transformation.

**Usage**

```r
neglog(object)
```

**Arguments**

- `object`: an object of type `lm`.

**Value**

An object of class `trafo`. Methods such as `as.data.frame.trafo` and `print.trafo` can be used for this class.

**References**


**Examples**

```r
# Load data
data("cars", package = "datasets")

# Fit linear model
lm_cars <- lm(dist ~ speed, data = cars)

# Transform dependent variable
neglog(object = lm_cars)
```
plot.trafo_compare  Plots for linear models with transformed dependent variable

Description
For the two transformed models a range of plots is returned in order to check model assumptions graphically.

Usage
```r
## S3 method for class 'trafo_compare'
plot(x, ...)
```

Arguments
- `x`  an object of type `trafo_compare`
- `...` additional arguments that are not used in this method

plot.trafo_lm  Plot for regression models with untransformed and transformed dependent variable

Description
For the untransformed and transformed model a range of plots is returned in order to check model assumptions graphically.

Usage
```r
## S3 method for class 'trafo_lm'
plot(x, ...)
```

Arguments
- `x`  an object of type `trafo_lm`
- `...` additional arguments that are not used in this method
Description

Prints diagnostics of two trafo objects.

Usage

```r
## S3 method for class 'diagnostics.trafo_compare'
print(x, ...)
```

Arguments

- `x`: an object of type `diagnostics.trafo_compare`
- `...`: additional arguments that are not used in this method

Description

Prints diagnostics of an untransformed and a transformed model.

Usage

```r
## S3 method for class 'diagnostics.trafo_lm'
print(x, ...)
```

Arguments

- `x`: an object of type `diagnostics.trafo_lm`
- `...`: additional arguments that are not used in this method
print.summary.trafo_compare

Prints summary of trafo_compare objects

Description

Prints objects to be shown in the summary function for objects of type trafo_compare.

Usage

### S3 method for class 'summary.trafo_compare'
print(x, ...)

Arguments

- x: an object of type summary.trafo_compare
- ...: additional arguments that are not used in this method

print.summary.trafo_lm

Print summary trafo

Description

Prints objects to be shown in the summary function for objects of type trafo_lm

Usage

### S3 method for class 'summary.trafo_lm'
print(x, ...)

Arguments

- x: an object of type summary.trafo_lm
- ...: additional arguments that are not used in this method
Description
Prints object of type trafo

Usage
```r
## S3 method for class 'trafo'
print(x, ...)
```

Arguments
- `x`: an object of type trafo.
- `...`: other parameters that can be passed to the function.

Description
Prints object of type trafo_compare

Usage
```r
## S3 method for class 'trafo_compare'
print(x, ...)
```

Arguments
- `x`: an object of type trafo_compare.
- `...`: other parameters that can be passed to the function.
**print.trafo_lm** *Prints object of type trafo_lm*

**Description**

Prints object of type trafo_lm

**Usage**

```r
## S3 method for class 'trafo_lm'
print(x, ...)
```

**Arguments**

- `x` an object of type trafo_lm.
- `...` other parameters that can be passed to the function.

**reciprocal** *Reciprocal transformation for linear models*

**Description**

The function transforms the dependent variable of a linear model using the Reciprocal transformation.

**Usage**

```r
reciprocal(object)
```

**Arguments**

- `object` an object of type lm.

**Value**

An object of class trafo. Methods such as `as.data.frame.trafo` and `print.trafo` can be used for this class.

**Examples**

```r
# Load data
data("cars", package = "datasets")

# Fit linear model
lm_cars <- lm(dist ~ speed, data = cars)

# Transform dependent variable
reciprocal(object = lm_cars)
```
The function transforms the dependent variable of a linear model using the Square-root shift transformation. The transformation parameter can either be estimated using different estimation methods or given.

Usage

```
sqrtshift(object, lambda = "estim", method = "ml",
            lambdarange = NULL, plotit = TRUE)
```

Arguments

- `object`: an object of type lm.
- `lambda`: either a character named "estim" if the optimal transformation parameter should be estimated or a numeric value determining a given value for the transformation parameter. Defaults to "estim".
- `method`: a character string. Different estimation methods can be used for the estimation of the optimal transformation parameter: (i) Maximum likelihood approach ("ml"), (ii) Skewness minimization ("skew"), (iii) Kurtosis optimization ("kurt"), (iv) Divergence minimization by Kolmogorov-Smirnov ("div.ks"), by Cramer-von-Mises ("div.cvm") or by Kullback-Leibler ("div.kl"). Defaults to "ml".
- `lambdarange`: a numeric vector with two elements defining an interval that is used for the estimation of the optimal transformation parameter. Defaults to NULL. In this case the lambdarange is set to the range of the data. In case the lowest value is negative the absolute value of the lowest value plus 1 is the lower bound for the range.
- `plotit`: logical. If TRUE, a plot that illustrates the optimal transformation parameter or the given transformation parameter is returned. Defaults to TRUE.

Value

An object of class `trafo`. Methods such as `as.data.frame.trafo` and `print.trafo` can be used for this class.

Examples

```
# Load data
data("cars", package = "datasets")

# Fit linear model
lm_cars <- lm(dist ~ speed, data = cars)

# Transform dependent variable using a maximum likelihood approach
sqrtshift(object = lm_cars, plotit = TRUE)
```
**summary.trafo_compare**  Summary for two differently transformed models

**Description**

The summary contains the summary for two transformed models. The summary is based on the summary for objects of type lm.

**Usage**

```r
## S3 method for class 'trafo_compare'
summary(object, ...)
```

**Arguments**

- `object`: an object of type `trafo_compare`
- `...`: additional arguments that are not used in this method

**Value**

An object of class `summary.trafo_compare`. The method `print.summary.trafo_compare` can be used for this class.

---

**summary.trafo_lm**  Summary for linear models with untransformed and transformed dependent variable

**Description**

The summary method for class `trafo_lm` contains a summary for an untransformed and a transformed model. The resulting summary is based on the summary for objects of type lm.

**Usage**

```r
## S3 method for class 'trafo_lm'
summary(object, ...)
```

**Arguments**

- `object`: an object of type `trafo_lm`
- `...`: additional arguments that are not used in this method

**Value**

An object of class `summary.trafo_lm`. The method `print.summary.trafo_lm` can be used for this class.
An R package supporting the selection of a suitable transformation

Description

Estimation, selection and comparison of several families of transformations. The families of transformations included in the package are the following: Bickel-Doksum, Box-Cox, Dual, Glog, Gpower, Log, Log-shift opt, Manly, Modulus, Neglog, Reciprocal and Yeo-Johnson. The package simplifies to compare linear models with untransformed and transformed dependent variable as well as linear models where the dependent variable is transformed with different transformations. Furthermore, the package employs maximum likelihood approaches, skewness and divergence minimization to estimate the optimal transformation parameter.

Details

An overview of all currently provided functions can be requested by library(help=trafo).

trafo_compare

Compares linear models with transformed dependent variable

Description

Function trafo_compare compares linear models where the dependent variable is transformed by different transformations.

Usage

trafo_compare(object, trafos, std = FALSE)

Arguments

object an object of type lm
trafos a list of two trafo objects based on the same model given in object.
std logical. If TRUE, the transformed models are returned based on the standardized/scaled transformation. Defaults to FALSE.

Value

An object of class trafo_compare. Methods such as diagnostics.trafo_compare, print.trafo_compare, plot.trafo_compare and summary.trafo_compare can be used for this class.

See Also

bickeldoksum, boxcox, dual, glog, gpower, log, logshiftopt, manly, modulus, neglog, sqrtshift, yeojohnson
Examples

# Load data
data("cars", package = "datasets")

# Fit linear model
lm_cars <- lm(dist ~ speed, data = cars)

# Transform with Bickel-Doksum transformation
bd_trafo <- bickeldoksum(object = lm_cars, plotit = FALSE)

# Transform with Box-Cox transformation
bc_trafo <- boxcox(object = lm_cars, method = "skew", plotit = FALSE)

# Compare transformed models
trafo_compare(object = lm_cars, trafos = list(bd_trafo, bc_trafo))

trafo_lm

Fits transformed linear models

Description

Function trafo_lm fits linear models with transformed dependent variable. The main return are two lm objects where one is the untransformed linear model and the other one the transformed linear model.

Usage

trafo_lm(object, trafo = "boxcox", lambda = "estim", method = "ml",
         lambdarange = NULL, std = FALSE, custom_trafo = NULL)

Arguments

object

trafo

lambda

method

an object of type lm.
a character string. Different transformations can be used for transforming the dependent variable in a linear model: (i) "bickeldoksum", (ii) "boxcox", (iii) "dual", (iv) "glog", (v) "gpower", (vi) "log", (vii) "logshiftopt", (viii) "manly", (ix) "modulus", (x) "neglog", (xi) "reciprocal", (xii) "yeojohnson". Defaults to "boxcox".
either a character named "estim" if the optimal transformation parameter should be estimated or a numeric value determining a given value for the transformation parameter. Defaults to "estim".
a character string. Different estimation methods can be used for the estimation of the optimal transformation parameter: (i) Maximum likelihood approach ("ml"), (ii) Skewness minimization ("skew"), (iii) Kurtosis optimization ("kurt"), (iv) Divergence minimization by Kolmogorov-Smirnov ("div.ks"), by Cramer-von-Mises ("div.cvm") or by Kullback-Leibler ("div.kl"). Defaults to "ml".
Yeo-Johnson transformation for linear models

Description

The function transforms the dependent variable of a linear model using the Yeo-Johnson transformation. The transformation parameter can either be estimated using different estimation methods or given.

Usage

yeojohnson(object, lambda = "estim", method = "ml",
            lambdarange = c(-2, 2), plotit = TRUE)
Arguments

- **object**: an object of type `lm`.
- **lambda**: either a character named "estim" if the optimal transformation parameter should be estimated or a numeric value determining a given value for the transformation parameter. Defaults to "estim".
- **method**: a character string. Different estimation methods can be used for the estimation of the optimal transformation parameter: (i) Maximum likelihood approach ("ml"), (ii) Skewness minimization ("skew"), (iii) Kurtosis optimization ("kurt"), (iv) Divergence minimization by Kolmogorov-Smirnov ("div.ks"), by Cramer-von-Mises ("div.cvm") or by Kullback-Leibler ("div.kl"). Defaults to "ml".
- **lambdarange**: a numeric vector with two elements defining an interval that is used for the estimation of the optimal transformation parameter. Defaults to c(-2, 2).
- **plotit**: logical. If TRUE, a plot that illustrates the optimal transformation parameter or the given transformation parameter is returned. Defaults to TRUE.

Value

An object of class `trafo`. Methods such as `as.data.frame.trafo` and `print.trafo` can be used for this class.

References


Examples

```r
# Load data
data("cars", package = "datasets")

# Fit linear model
lm_cars <- lm(dist ~ speed, data = cars)

# Transform dependent variable using a maximum likelihood approach
eyeojohnson(object = lm_cars, plotit = FALSE)
```
Index

*Topic diagnostics
diagnostics, 7

as.data.frame.trafo, 3, 5, 7, 10–17, 22, 23, 28
assumptions, 4

bickeldoksum, 3, 4, 5, 25, 27
boxcox, 3, 4, 6, 25, 27
diagnostics, 7
diagnostics.trafo_compare, 7, 8, 25
diagnostics.trafo_lm, 7, 9, 27
dual, 3, 4, 10, 25, 27
glog, 3, 4, 11, 25, 27
gpower, 3, 4, 12, 25, 27

log, 3, 4, 25, 27
logshiftopt, 3, 4, 13, 25, 27
logtrafo, 14

manly, 3, 4, 15, 25, 27
modulus, 3, 4, 16, 25, 27

neglog, 3, 4, 17, 25, 27

plot.trafo_compare, 18, 25
plot.trafo_lm, 18, 27
print.diagnostics.trafo_compare, 8, 19
print.diagnostics.trafo_lm, 9, 19
print.summary.trafo_compare, 20, 24
print.summary.trafo_lm, 20, 24
print.trafo, 5, 7, 10–17, 21, 22, 23, 28
print.trafo_compare, 21, 25
print.trafo_lm, 22, 27

reciprocal, 22

sqrtshift, 3, 4, 23, 25, 27
summary.trafo_compare, 24, 25

summary.trafo_lm, 24, 27
trafo, 25
trafo-package (trafo), 25
trafo_compare, 25
trafo_lm, 26
yeojohnson, 3, 4, 25, 27, 27