Package 'tlm'

September 8, 2014

Type Package

Title Effects under Linear, Logistic and Poisson Regression Models with Transformed Variables.

Version 0.1.2

Date 2014-09-08

Author Jose Barrera-Gómez and Xavier Basagaña

Maintainer Jose Barrera-Gómez<jbarrera@creal.cat>

Depends R (>= 3.0.1), stats, utils, boot

Description
  Computation of effects under linear, logistic and Poisson regression models with transformed variables. Logarithm and power transformations are allowed. Effects can be displayed both numerically and graphically in both the original and the transformed space of the variables.

License GPL (>=2)

R topics documented:

tlm-package	*Effects under Linear, Logistic and Poisson Regression Models with Transformed Variables*

Description

Computation of effects under linear, logistic and Poisson regression models with transformed variables. Logarithm and power transformations are allowed. Effects can be displayed both numerically and graphically in both the original and the transformed space of the variables.

Details

Package:	tlm
Type:	Package
Version:	0.1.2
Date:	2014-09-08
License:	GPL (>=2)
URL:	http://www.creal.cat/xbasagana/software.html
LazyLoad:	yes

Author(s)

Jose Barrera-Gómez and Xavier Basagaña

Maintainer: Jose Barrera-Gómez <jbarrera@creal.cat>

References

Barrera-Gómez J, Basagaña X. Interpretation of effects in linear, logistic and Poisson models with transformed variables: guidance for practice and software. (submitted).

Examples

```
# linear regression model with log-log tansformation:
data(feld1)
head(feld1)
modcat <- tlm(y = logroom, x = logmattress, z = cat, ypow = 0, xpow = 0, data = feld1)

# print and summary:
modcat
summary(modcat)

# plotting the fitted model in the original space:
plot(modcat, xname = "Mattress levels", yname = "room levels")

# plotting the fitted model in the transformed space (and adding the observations):
plot(modcat, xname = "Mattress levels", yname = "room levels", type = "transformed",
    observed = TRUE)
```

```
# diagnosis plot for the fitted model:
plot(modcat, type = "diagnosis")

# Expected geometric mean of the response, adjusting for variable 'cat':
MY(modcat)

# Expected mean of the transformed response, adjusting for variable 'cat':
MY(modcat, space = "transformed")

# Information on how to interpret effects:
effectInfo(modcat)

# Summary effect (default effect):
effect(modcat)
```

---

cotinine	*Birth Weight and Cord Serum Cotinine*

Description

Simulated data for birth weight and cord serum cotinine levels in 351 newborns.

Usage

```
data(cotinine)
```

Format

A data frame with 351 observations on the following 4 variables.

cotinine numeric cord serum cotinine level in the mother (ng/ml).

logcotinine logarithm of cotinine.

weight numeric birth weight (g).

underweight a factor with levels no and yes, indicating underweight (weight < 2500 g).

Details

Data were simulated to emulate true data pattern observed in a real study (see 'References').

Source

See 'References'.

References

Pichini S *et al.* Cord serum cotinine as a biomarker of fetal exposure to cigarette smoke at the end of pregnancy. *Environmental Health Perspectives*. 2000;108(11):1079-1083.

Examples

```
data(cotinine)
par(las = 1, mfrow = c(2, 2))
with(cotinine, plot(cotinine, weight))
with(cotinine, plot(logcotinine, weight))
with(cotinine, boxplot(cotinine ~ underweight))
with(cotinine, boxplot(logcotinine ~ underweight))
```

---

effect   *Effects Estimate in Linear, Logistic and Poisson Regression Models with Transformed Variables*

Description

This function estimates the effect of a explanatory variable of interest on a response variable, under a fitted linear, logistic or Poisson regression model with transformed variables. The effect is estimated in the original scale of the variables.

Usage

```
effect(object, x1 = NULL, x2 = NULL, c = NULL, q = NULL, r = NULL, npoints = NULL,
     level = 0.95, nboot = 999, seed = 4321, verbose = TRUE)
## S3 method for class 'effect'
print(x, ...)
```

Arguments

object	an object of class "tlm", a result of a call to tlm.
x1	numeric. The values of the explanatory variable where the effect should be computed. See 'Details'.
x2	numeric. The alternative values of the explanatory variable (changing from x1) for which the effect should be computed. See 'Details'.
c	numeric. The additive change in the explanatory variable. See 'Details'.
q	numeric. The multiplicative change in the explanatory variable. See 'Details'.
r	numeric. The percent change in the explanatory variable. See 'Details'.
npoints	numeric. The number of points where the effect should be computed. See 'Details'.
level	numeric. Confidence level for the effect estimate. Default is 0.95.
nboot	numeric. The number of non parametric bootstrap samples to compute confidence intervals. Default is 999. See 'Details'.
seed	numeric. A single value, the seed for bootstrapping. Default is 4321.
verbose	logical. Whether to print detailed progress on R prompt. Default is TRUE.
x	an object of class "effect", a result of a call to effect.
...	further additional arguments passed to the method print.

Details

In order to compute the effect, both the initial and the final values of the explanatory should be provided. It can be done in several ways. For instance, providing, x1 and x2; x1 and one of c, q or r; x1, npoints and one of c, q or r. Only one of the arguments c, q or r is used, prevailing c and then q. If no enough argument are passed, the interquartile range will be considered and a summary effect should be computed, if it exists.

Confidence intervals are computed by transforming the endpoints of the intervals in the transformed scale when it is possible, while non-parametric bootstrap is used otherwise.

Value

effect	point estimate and confidence interval for the effect size.
info	information on how to interpret the effect. Used by the function effectInfo.

Author(s)

Barrera-Gómez J and Basagaña X.

References

Barrera-Gómez J, Basagaña X. Interpretation of effects in linear, logistic and Poisson models with transformed variables: guidance for practice and software. (submitted).

See Also

tlm, effectInfo, MY.

Examples

```
### Linear model with log transformation in the response variable:
data(imt)
head(imt)

# model fitting:
modimt <- tlm(y = logimt, x = age, data = imt, ypow = 0)
modimt

# information on interpreting the effect:
effectInfo(modimt)

# the function effect provides as default the expected change in IMT for an additive change in age
# equal to the interquartile range:
effect(modimt)

# other effects:
(minage <- min(imt$age))
(maxage <- max(imt$age))
effect(modimt, c = maxage - minage)

## Not run:
effect(modimt, x1 = minage, r = 50, npoints = 3)

## End(Not run)
```

---

effectInfo   *Interpretation of Effects in Linear, Logistic and Poisson Models with*
                                    Transformed Variables

Description

This function provides information on interpreting effects in linear, logistic and Poisson models with transformed variables. Specifically, if a summary measure for the effect exists, the function details how to obtain it.

Usage

```
effectInfo(object)
## S3 method for class 'effectInfo'
print(x, ...)
```

Arguments

object   an object of class "tlm", a result of a call to tlm.

x   an object of class "effectInfo", a result of a call to effectInfo.

...   further additional arguments for the print method.

Value

beta   regression coefficient estimate in the fitted model which is associated to the effect of the explanatory variable of interest on the response variable. NA corresponds to those models for which a summary effect does not exist.

Xincrease   type of change in the exploratory variable of interest (additive or realtive) for which a summary effect exists. NA corresponds to those models for which a summary effect does not exist.

effecttype   type of effect on the response variable for which a summary effect exists. NA corresponds to those models for which a summary effect is not available.

effectsize   formula for the summary effect size, if any. NA corresponds to those models for which a summary effect is not available.

furtherinfo   further information about how to interpret effects.

Author(s)

Barrera-Gómez J and Basagaña X.

References

Barrera-Gómez J, Basagaña X. Interpretation of effects in linear, logistic and Poisson models with transformed variables: guidance for practice and software. (submitted).

See Also

tlm, effect, MY.

Examples

```
### Linear model with log transformation in the explanatory variable:
data(cotinine)
head(cotinine)
# model fitting:
modcot <- tlm(y = weight, x = logcotinine, data = cotinine, xpow = 0)
modcot
# information on interpreting the effect:
effectInfo(modcot)

### Linear model with no summary measure of effect:
data(glucose)
head(glucose)
# transformations Y^(-2) and X^(-1/2):
modgluco <- tlm(y = inv2glu, x = inv12tri, data = glucose, ypow = -2, xpow = -1/2)
modgluco
effectInfo(modgluco)
```

---

feld1   *Cat Allergen Concentrations*

Description

Simulated data for cat allergen concentrations (Fel d 1) in 471 homes, measured in both the living room and the bed mattress.

Usage

```
data(feld1)
```

Format

A data frame with 471 observations on the following 5 variables.

mattress numeric Feld d 1 concentration in the bed mattress (μg/g).

room numeric Feld d 1 concentration in the living room (μg/g).

logmattress logarithm of mattress.

logroom logarithm of room.

cat a factor with levels no and yes, indicating cat ownership.

Details

Data were simulated to emulate true data pattern observed in a real study (see 'References').

Source

See 'References'.

References

Basagaña X *et al*. Domestic aeroallergen levels in Barcelona and Menorca (Spain). *Pediatric Allergy and Immunology*. 2002;13(6):412-417.

Examples

```
data(feld1)
par(las = 1, mfrow = c(1, 2))
with(feld1, plot(mattress, room, col = as.numeric(cat)))
with(feld1, plot(logmattress, logroom, col = as.numeric(cat)))
```

---

glucose   *Glucose and Triglycerides Levels in Blood*

Description

Simulated data for glucose and triglycerides levels in blood in 400 adults.

Usage

```
data(glucose)
```

Format

A data frame with 400 observations on the following 4 variables.

trigly numeric triglycerides levels in blood (mg/dl).

gluco numeric glucose levels in blood (mg/dl).

inv12tri numeric reciprocal of the square root of trigly (i.e., -1/2 power transformation).

inv2glu numeric reciprocal of the gluco square (i.e., -2 power transformation).

Details

Data were simulated to emulate true data pattern observed in a real study (see 'References').

Source

See 'References'.

References

Rivera M *et al*. Association between long-term exposure to traffic-related air pollution and subclinical atherosclerosis: the REGICOR Study. *Environmental Health Perspectives*. 2013;121(2):223-230.

Examples

```
data(glucose)
par(las = 1, mfrow = c(1, 2))
with(glucose, plot(trigly, gluco))
with(glucose, plot(inv12tri, inv2glu))
```

---

imt   *Intima Media Thickness of the Carotid Artery*

Description

Simulated data for intima media thickness of the carotid artery and age in 2784 adults.

Usage

data(imt)

Format

A data frame with 2784 observations on the following 3 variables.

age numeric age of the individual (years).

imt numeric intima media thickness of the carotid artery (mm).

logimt logarithm of imt.

Details

Data were simulated to emulate true data pattern observed in a real study (see 'References').

Source

See 'References'.

References

Rivera M *et al*. Association between long-term exposure to traffic-related air pollution and subclinical atherosclerosis: the REGICOR Study. *Environmental Health Perspectives*. 2013;121(2):223-230.

Examples

```
data(imt)
par(las = 1, mfrow = c(1, 2))
with(imt, plot(age, imt))
with(imt, plot(age, logimt))
```

MY   *Expected Adjusted Median or Generalized Mean*

Description

This function computes expected measures of the response variable under a linear, logistic or Poisson regression fitted model with transformed variables. Measures can be calculated in both the original and the transformed space. The function automatically provides the name of the measure depending on the fitted model.

Usage

MY(object, x = NULL, npoints = 10, space = c("original", "transformed"), level = 0.95)
S3 method for class 'MY'
print(x, ...)

Arguments

object	an object of class "tlm", a result of a call to tlm.
x	for MY: numeric. a number or a numeric vector containing the values of the explanatory variable of interest for which the expected measure of the response variable are required. Default is NULL. For print.MY: an object of class "MY", a result of a call to MY.
npoints	numeric. If x is NULL, the number of points where the measure should be measured. Default is 10. See 'Details'.
space	character. If "original" (default), the measure is computed in the original space of the variables. If "transformed", the measure is computed in the transformed space of the variables, where the model is fitted.
level	numeric. The confidence level for measures. Default is 0.95.
...	further additional arguments for the print.MY method.

Details

In order to compute adjusted measures, all explanatory variables in the model different than the explanatory variable of interest are set at their means.

If space is "original", then the mean (for Poisson response) or the probability (for binary response) is computed. For gaussian response, the mean is computed if the response variable is not transformed; otherwise, the geometric mean (for log transformation in the response) or the median (for power transformation) is computed.

If space is "transformed", then the mean (for Poisson response or transformed gaussian response), or the logodds (for binary response) is computed.

If x is NULL, the measure is computed in npoints values of the explanatory variable of interest. Those values are choosen to be in arithmetic progression in the given space, inside the observed range of the explanatory variable.

Value

M	adjusted measure of the response variable. See 'Details'.
ymeasure	the type of measure for M.
space	space where measures has been computed ("original" or "transformed").
ypow	numeric power transformation assumed in the response variable. See tlm.
xpow	numeric power transformation assumed in the explanatory variable of interest. See tlm.

Author(s)

Barrera-Gómez J and Basagaña X.

References

Barrera-Gómez J, Basagaña X. Interpretation of effects in linear, logistic and Poisson models with transformed variables: guidance for practice and software. (submitted).

See Also

tlm, effectInfo, effect.

Examples

```
data(feld1)
head(feld1)

# Linear model with log-log transformation, adjusting for variable 'cat':
modcat <- tlm(y = logroom, x = logmattress, z = cat, data = feld1, ypow = 0, xpow = 0)
summary(modcat)

# Geometric mean of the response as a function of the explanatory variable, adjusted for
# variable 'cat':
MY(modcat)
MY(modcat, npoints = 3)
# computed at 1st and 3rd quartiles of the explanatory variable:
MY(modcat, x = quantile(feld1$mattress, probs = c(1, 3)/4))

# Mean of the log(response) as a function of the log explanatory variable, adjusted for
# variable 'cat':
MY(modcat, space = "transformed")
```

---

summary.tlm	*Summarizing Linear, Logistic and Poisson Models Fits with Transformed Variables.*

Description

summary method for an object created by the the function tlm.

Usage

```
## S3 method for class 'tlm'
summary(object, ...)
## S3 method for class 'summary.tlm'
print(x, ...)
```

Arguments

object	an object of class "tlm", a result of a call to tlm.
x	an object of class "summary.tlm", a result of a call to summary.tlm.
...	further additional arguments for summary and print methods.

Details

Essentially, the output of summary.lm or summary.glm is displayed. In addition, further information on the fitted model is also displayed.

Value

model	the fitted model in the transformed space.
ypow	the value of ypow.
xpow	the value of xpow.
summary	the summary of the fitted model provide by summary.lm (for gaussian response) or summary.glm (otherwise).

Author(s)

Barrera-Gómez J and Basagaña X.

References

Barrera-Gómez J, Basagaña X. Interpretation of effects in linear, logistic and Poisson models with transformed variables: guidance for practice and software. (submitted).

See Also

tlm, summary.lm, summary.glm.

Examples

```
### linear model with log-log transformation:
data(feld1)
modcat <- tlm (y = logroom, x = logmattress, z = cat, data = feld1, ypow = 0, xpow = 0)
modcat
summary(modcat)
```

tlm	*Fitting Linear, Logistic and Poisson Regression Models with Transformed Variables*

Description

This function fits a linear, logistic or Poisson regression model with transformed variables and creates an object which can be used to extract effects of the explanatory variable of interest on the response variable, in the original scale of the variables.

Usage

```
tlm(y, x, z = "none", family = gaussian, data, ypow = 1, xpow = 1, ...)
## S3 method for class 'tlm'
print(x, ...)
## S3 method for class 'tlm'
plot(x, type = c("original", "transformed", "diagnosis"), observed = FALSE, xname = "x",
    yname = "y", level = 0.95, ...)
```

Arguments

y	name of the response variable (unquoted).
x	name of the explanatory variable of interest (unquoted).
z	additional explanatory variables in the model. Default is "none". For a single variable, the name of the variable (unquoted); for more than one variable, the names should be passed unquoted and separated by the plus sign; for all the remaining variables in data, use just a period (.). No interaction terms are allowed.
family	the link function. For linear model, gaussian (default); for logistic regression, binomial; for Poisson regression, poisson. Quoted values are optional. In the case of binomial, the logit link is assumed.
data	a data.frame containing the variables in the model.
ypow	numeric power transformation already done in y. See 'Details'.
xpow	numeric power transformation already done in x. See 'Details'.
type	character indicating the type of plot for the fitted model. If "original" (default), the fitted model is plotted in the original space of the variables; if "transformed", the fitted model is plotted in the transformed space of the variables (where the model has been fitted); if "diagnosis", a model diagnostics plot is shown.
observed	logical indicating whether the observations should be added to the plot. Assumed to be FALSE if x is categorical. Default is FALSE.
xname	character. The name of the x variable for labeling the horizontal plot axis. Default is "x".
yname	character. The name of the y variable for labeling the vertical plot axis. Default is "y". The name of the measure for the response variable is automatically added before yname.
level	numeric. Confidence level for the prediction of the measure of y plotted. Default is 0.95.
...	further additional arguments passed to tlm or methods print and plot.

Details

The transformations already done in y and x are passed by ypow and xpow, respectively. They should be a number. Default is 1 (no transformation). The value 0 corresponds to the logarithmic transformation. If family is not gaussian, y is assumed non transformed. If x is categorical or presents only two different values, the value of xpow is assumed to be 1.

Value

model	the fitted model in the transformed space.
ypow	the value of ypow.
xpow	the value of xpow.

Author(s)

Barrera-Gómez J and Basagaña X.

References

Barrera-Gómez J, Basagaña X. Interpretation of effects in linear, logistic and Poisson models with transformed variables: guidance for practice and software. (submitted).

See Also

MY, effect.

Examples

```
### Linear model with log-log transformation:
data(feld1)
head(feld1)
# model fitting in the transformed space:
modcat <- tlm (y = logroom, x = logmattress, z = cat, data = feld1, ypow = 0, xpow = 0)
modcat
summary(modcat)
# plot of the geometric mean of the response (original space), adjusting for variable 'cat':
plot(modcat, xname = "Mattress levels", yname = "room levels")
# plot of the mean of the log of response (transformed space), adjusting for variable 'cat' and
# adding the observations:
plot(modcat, type = "transformed", xname = "mattress levels", yname = "room levels",
    observed = TRUE)
# diagnosis plot:
plot(modcat, type = "diagnosis")

### The same model but now considering 'cat' as the explanatory variable of interest:
modcat2 <- tlm (y = logroom, x = cat, z = logmattress, data = feld1, ypow = 0)
summary(modcat2)
# plot of the geometric mean of the response (original space), adjusting for mattress levels:
plot(modcat2, xname = "Cat", yname = "room levels")
# plot of the mean of the log of response (transformed space), adjusting for mattress levels:
plot(modcat2, type = "transformed", xname = "Cat", yname = "room levels")
```

Index