

# Package ‘t1m’

July 22, 2025

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

**Version** 0.2.0

**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. The methods are described in Barrera-Gomez  
and Basagana (2015) <[doi:10.1097/EDE.0000000000000247](https://doi.org/10.1097/EDE.0000000000000247)>.

**Depends** R (>= 4.1)

**Encoding** UTF-8

**LazyData** true

**Imports** graphics, stats, utils, boot

**Suggests** knitr, rmarkdown, xtable

**RoxygenNote** 7.3.2

**VignetteBuilder** knitr

**License** GPL (>= 3)

**NeedsCompilation** no

**Author** Jose Barrera-Gomez [aut, cre] (ORCID:  
<<https://orcid.org/0000-0002-2688-6036>>),  
Xavier Basagana [aut] (ORCID: <<https://orcid.org/0000-0002-8457-1489>>)

**Maintainer** Jose Barrera-Gomez <[jose.barrera@isglobal.org](mailto:jose.barrera@isglobal.org)>

**Repository** CRAN

**Date/Publication** 2025-01-07 11:10:02 UTC

## Contents

cotinine . . . . .	2
effect . . . . .	3
effectInfo . . . . .	5
feld1 . . . . .	6
glucose . . . . .	7

imt . . . . .	8
MY . . . . .	9
summary.tlm . . . . .	11
tlm . . . . .	12

<b>Index</b>	<b>15</b>
--------------	-----------

---

cotinine	<i>Birth Weight and Cord Serum Cotinine</i>
----------	---

---

## Description

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

## Usage

```
cotinine
```

## Format

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

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

**logcotinine** logarithm of cotinine.

**weight** 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	<i>Effects Estimate in Linear, Logistic and Poisson Regression Models with Transformed Variables</i>
--------	--

---

### Description

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 reported 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	object of class "t1m", a result of a call to <code>t1m</code> .
x1	numeric. The values of the explanatory variable where the effect should be computed. See Details below.
x2	numeric. The alternative values of the explanatory variable (changing from x1) for which the effect should be computed. See Details below.
c	numeric. The additive change in the explanatory variable. See Details below.
q	numeric. The multiplicative change in the explanatory variable. See Details below.
r	numeric. The percent change in the explanatory variable. See Details below.
npoints	numeric. The number of points where the effect should be computed. See Details below.
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 below.

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	for <code>print.effect</code> , an object of class "effect" (an output of the <code>effect</code> function).
...	additional arguments for <code>print.effect</code> .

## 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 arguments are passed, the interquartile range will be considered and a summary effect is 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

A list with class "effect" including the following items:

**effect** point estimate and confidence interval for the effect size.

**info** information on how to interpret the effect. Used by the function [effectInfo](#).

## References

Barrera-Gomez J, Basagana X. Models with transformed variables: interpretation and software. *Epidemiology*. 2015;26(2):e16-17.

## See Also

[t1m](#), [effectInfo](#), [MY](#).

## Examples

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

# model fitting:
modimt <- t1m(logimt ~ 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	<i>Interpretation of Effects in Linear, Logistic and Poisson Models with Transformed Variables</i>
------------	--

---

## Description

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 and interpret it.

## Usage

```
effectInfo(object)

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

## Arguments

object	object of class "t1m" obtained with the function <code>t1m</code> .
x	for <code>print.effectInfo</code> , an object of class "effectInfo" (an output of the <code>effectInfo</code> function).
...	additional arguments for <code>print.effectInfo</code> .

## Value

A list with class "effectInfo" including the following items:

**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.

## References

Barrera-Gomez J, Basagana X. Models with transformed variables: interpretation and software. *Epidemiology*. 2015;26(2):e16-17.

## See Also

[t1m](#), [effect](#), [MY](#).

## Examples

```
### Linear model with log transformation in the explanatory variable:
data(cotinine)
head(cotinine)
# model fitting:
modcot <- t1m(weight ~ 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 <- t1m(inv2glu ~ 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

feld1

## Format

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

**mattress** Feld d 1 concentration in the bed mattress ( $\mu\text{g/g}$ ).

**room** Feld d 1 concentration in the living room ( $\mu\text{g/g}$ ).

**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**

Basagana 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**

```
glucose
```

**Format**

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

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

**gluco** 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

imt

## Format

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

**age** age of the individual (years).

**imt** 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**

Computes expected measures of the response variable under a linear, logistic or Poisson regression fitted model with transformed variables. Measures can be reported 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	object of class <code>tlm</code> , a result of a call to <code>tlm</code> .
x	For <code>MY</code> , 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 <code>NULL</code> . For <code>print.MY</code> , an object of class <code>"MY"</code> (an output of the <code>MY</code> function).
npoints	numeric. If <code>x</code> is <code>NULL</code> , the number of points where the measure should be measured. Default is 10. See Details below.
space	character. If <code>"original"</code> (default), the measure is reported in the original space of the variables. If <code>"transformed"</code> , the measure is reported in the transformed space of the variables, where the model is fitted.
level	numeric. The confidence level for measures. Default is 0.95.
...	additional arguments for <code>print.MY</code> .

## 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` argument in `MY` is `NULL`, the measure is computed in `npoints` values of the explanatory variable of interest. Those values are chosen to be in arithmetic progression in the given space, inside the observed range of the explanatory variable.

## Value

A list with class "MY" including the following items:

**M** adjusted measure of the response variable. See Details below.

**ymeasure** the type of measure for M.

**ypow** numeric power transformation assumed in the response variable. See [t1m](#).

**xpow** numeric power transformation assumed in the explanatory variable of interest. See [t1m](#).

## References

Barrera-Gomez J, Basagana X. Models with transformed variables: interpretation and software. *Epidemiology*. 2015;26(2):e16-17.

## See Also

[t1m](#), [effectInfo](#), [effect](#).

## Examples

```
data(feld1)
head(feld1)

# Linear model with log-log transformation, adjusting for variable 'cat':
modcat <- t1m(logroom ~ logmattress + cat, data = feld1, ypow = 0, xpow = 0)
summary(modcat)

# Geometric mean of the response as a function of the explanatory variable,
# adjusted for '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 'cat':
```

```
MY(modcat, space = "transformed")
```

---

summary.tlm

*Summarizing the Model Fitting*

---

## Description

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

## Usage

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

## Arguments

<code>object</code>	an object of class "tlm", an output of the <code>tlm</code> function.
<code>...</code>	additional arguments.
<code>x</code>	an object of class "summary.tlm" (an output of <code>summary.tlm</code> ).

## Details

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

## Value

A list with class "summary.tlm" including the following items:

**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).

## References

Barrera-Gomez J, Basagana X. Models with transformed variables: interpretation and software. *Epidemiology*. 2015;26(2):e16-17.

## See Also

[tlm](#), [effectInfo](#), [MY](#).

## Examples

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

---

tlm	<i>Fitting, Reporting and Visualizing Linear, Logistic and Poisson Regression Models with Transformed Variables</i>
-----	---

---

## Description

tlm is the main function of the package. It fits a linear, logistic or Poisson regression model with transformed variables and creates an object which can be subsequently used to compute adjusted measures of the response variable (with [MY](#)) and compute and interpret adjusted effects of the explanatory variable of interest on the response variable (with [effect](#) and [effectInfo](#), respectively), in the natural scale of the variables. In addition, the fitted model can be visualized with the `plot.tlm` method.

## Usage

```
tlm(
  formula,
  family = gaussian,
  data,
  ypow = 1,
  xpow = 1,
  ...,
  y,
  x,
  z = "none"
)

## 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

`formula` model [formula](#). Left-hand-side indicates the response variable (whose values are assumed to be already transformed). First term in right-hand-side indicates the explanatory variable of interest (whose values are assumed to be already transformed). Right-hand-side can include additional terms (e.g. adjusting variables) but the explanatory variable of interest cannot be involved in any of them.

family	the response variable <code>family</code> . Allowed values are: 1) gaussian (default), for linear regression; 2) binomial, for logistic regression, and 3) poisson, for Poisson regression with logarithmic link. Quoted values are allowed.
data	a <code>data.frame</code> containing the variables in the model.
ypow	numeric. Power transformation already done in the response variable. See Details below.
xpow	numeric. Power transformation already done in the explanatory variable of interest. See Details below.
...	for <code>t1m</code> , additional arguments for the underlying <code>lm</code> or <code>glm</code> fitting. For <code>print.t1m</code> and <code>plot.t1m</code> , additional arguments for generic methods <code>print</code> and <code>plot</code> , respectively.
y, z	old arguments for back compatibility only. To be removed, see Details below.
x	for <code>t1m</code> , old argument for back compatibility only. To be removed, see Details below. For <code>print.t1m</code> and <code>plot.t1m</code> , an object of class "t1m" (an output of the <code>t1m</code> function).
type	For <code>plot.t1m</code> , 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 "diagnosis", a model diagnostics plot is shown. If "transformed", the fitted model is plotted in the transformed space of the variables (where the model has been fitted). Resulting plots under "transformed" should not be reported (see Details below).
observed	For <code>plot.t1m</code> , logical indicating whether the observations are shown in the plot. Assumed to be FALSE if the explanatory variable of interest is categorical. Default is FALSE.
xname, yname	For <code>plot.t1m</code> , character indicating the name of the explanatory and the response variable of interest for labeling the plot axes. Default are "x" and "y", respectively.
level	For <code>plot.t1m</code> , numeric indicating the confidence level for the confidence of the expectation of the response variable according to the fitted model. Default is 0.95.

## Details

The transformations already done in the response variable and in the explanatory variable of interest are passed by `ypow` and `xpow`, respectively, and must be numbers. Default is 1, meaning no transformation. The value 0 corresponds to the logarithmic transformation. If `family` is not gaussian, the response variable is assumed non transformed. If the explanatory variable of interest is categorical or takes only two different values, the value of `xpow` is assumed to be 1. If the explanatory variable of interest takes only two different values, it is handled as a binary variable.

Plots obtained for `type = "transformed"` are intended to visually explore the model goodness of fit and should not be reported because values of the transformed variables are meaningless (e.g. `log(cotinine)` has no sense).

Old arguments `y`, `x` and `z`, are deprecated and bugs are no longer fixed. They will be removed in the first version posterior to 0.2.0. Use argument `formula` instead.

**Value**

A list with class "t1m" including the following items:

**model** the fitted model in the transformed space.

**ypow** the value of ypow.

**xpow** the value of xpov.

**References**

Barrera-Gomez J, Basagana X. Models with transformed variables: interpretation and software. *Epidemiology*. 2015;26(2):e16-17.

**See Also**

[MY](#), [effect](#), [formula](#).

**Examples**

```
### Linear model with log-log transformation:
### effect of mattress levels on room levels, adjusting for cat:
# model fitting in the transformed space:
data(feld1)
head(feld1)
modcat <- t1m(logroom ~ logmattress + cat, data = feld1, ypow = 0, xpov = 0)
modcat
summary(modcat)
# plot of the geometric mean of the response (original space), adjusting for 'cat':
plot(modcat, xname = "Mattress levels", yname = "room levels")
# plot of the mean of the log of response (transformed space), adjusting for 'cat' and
# adding the observations:
plot(modcat, type = "transformed", xname = "mattress levels", yname = "room levels",
      observed = TRUE)
# diagnosis plot:
plot(modcat, type = "diagnosis")

### effect of cat in house on room levels, adjusting for mattress levels:
modcat2 <- t1m(logroom ~ cat + 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

## \* datasets

- cotinine, 2
- feld1, 6
- glucose, 7
- imt, 8

cotinine, 2

effect, 3, 6, 10, 12, 14  
effectInfo, 4, 5, 10–12

family, 13  
feld1, 6  
formula, 12, 14

glucose, 7

imt, 8

MY, 4, 6, 9, 11, 12, 14

plot.tlm (tlm), 12  
print.effect (effect), 3  
print.effectInfo (effectInfo), 5  
print.MY (MY), 9  
print.summary.tlm (summary.tlm), 11  
print.tlm (tlm), 12

summary.glm, 11  
summary.lm, 11  
summary.tlm, 11

tlm, 3–6, 9–11, 12