Package: twopartm (via r-universe)

October 11, 2026

Type Package

Title Two-Part Model with Marginal Effects

Version 0.1.0

Maintainer Yajie Duan <yajieritaduan@gmail.com>

Description Fit two-part regression models for zero-inflated data. The models and their components are represented using S4 classes and methods. Average Marginal effects and predictive margins with standard errors and confidence intervals can be calculated from two-part model objects. Belotti, F., Deb, P., Manning, W. G., & Norton, E. C. (2015) <doi:10.1177/1536867X1501500102>.

License GPL (>= 2)

Depends R (>= 2.10)

Imports methods, graphics, stats, data.table, MASS

Encoding UTF-8

LazyData true

NeedsCompilation no

RoxygenNote 7.2.1

Author Yajie Duan [aut, cre], Birol Emir [aut], Griffith Bell [aut], Javier Cabrera [aut], Pfizer Inc. [cph, fnd]

Repository https://yajie1020.r-universe.dev

Date/Publication 2022-10-14 07:50:02 UTC

RemoteUrl https://github.com/cran/twopartm

RemoteRef HEAD

RemoteSha f94c566708c58cea095d7b9680d11e8df6325936

Contents

AME 2
bioChemists 5
coef-methods 6

FiellerRatio 7
logLik-methods 9
margin 10
meps 14
plot-methods 15
predict-methods 17
residuals-methods 19
tpm 20
twopartm-class 23

Index **26**

AME *Average Marginal Effect (AME) with CIs for Two-part Model Objects*

Description

Calculate average marginal effects (AMEs) with CIs for variables from a fitted two-part regression model object of class `twopartm`.

Usage

```
## S4 method for signature 'twopartm'
AME(object, newdata = NULL, term = NULL, at = NULL, se = TRUE,
se.method = c("delta","bootstrap"), CI = TRUE,CI.boots = FALSE,
level = 0.95,eps = 1e-7,na.action = na.pass, iter = 50)
```

Arguments

`object`	a fitted two-part model object of class `twopartm` as returned by `tpm`.
`newdata`	optionally, a data frame in which to look for variables with which to calculate average marginal effects. If omitted, the original observations are used.
`term`	A character vector with the names of variables for which to compute the average marginal effects. The default (NULL) returns average marginal effects for all variables.
`at`	A list of one or more named vectors, specifically values at which to calculate the average marginal effects. The specified values are fully combined (i.e., a cartesian product) to find AMEs for all combinations of specified variable values. These are used to modify the value of data when calculating AMEs across specified values. Note: This does not calculate AMEs for subgroups but rather for counterfactual datasets where all observations take the specified values; to obtain subgroup effects, subset data directly.
`se`	logical switch indicating if standard errors are required.
`se.method`	A character string indicating the type of estimation procedure to use for estimating variances of AMEs. The default ("delta") uses the delta method. The alternative is "bootstrap", which uses bootstrap estimation.

CI	logical switch indicating if confidence intervals are required.
CI.boots	if se.method == "bootstrap", logical switch indicating if confidence intervals are obtained by normal-based or by bootstrap quantiles.
level	A numeric value specifying the confidence level for calculating p-values and confidence intervals.
eps	A numeric value specifying the "step" to use when calculating numerical derivatives.
na.action	function determining what should be done with missing values in newdata. The default is to predict NA.
iter	if se.method == "bootstrap", the number of bootstrap iterations.

Details

For factor variables, the average value of discrete first-differences in predicted outcomes are calculated as AME (i.e., change in predicted outcome when factor is set to a given level minus the predicted outcome when the factor is set to its baseline level). If you want to use numerical differentiation for factor variables (which you probably do not want to do), enter them into the original modeling function as numeric values rather than factors. For logical variables, the same approach as factors is used, but always moving from FALSE to TRUE.

For numeric (and integer) variables, the method calculates an instantaneous marginal effect using a simple "central difference" numerical differentiation:

$$\frac{f(x + \frac{1}{2}h) - f(x - \frac{1}{2}h)}{dh}$$

, where ($h = \max(|x|, 1)\sqrt{\epsilon}$ and the value of ϵ is given by argument eps. Then AMEs are calculated by taking average values of marginal effects from all the observations.

If at = NULL (the default), AMEs are calculated based on the original observations used in the fitted two-part model, or the new data set that newdata inputs. Otherwise, AMEs are calculated based upon modified data by the number of combinations of values specified in at.

The standard errors of AMEs could be calculated using delta method or bootstrap method. The delta method for two-part model considers the difference between average Jacobian vectors for factor or logical variables, or the second-order partial derivatives of prediction with respect to both models' parameters, assuming independence between models from two parts. The Jacobian vectors and derivatives are approximated by numerical differentiations. The bootstrap method generates bootstrap samples to fit two-part models, and get variances and inverted bootstrap quantile CIs or normal-based CIs of AMEs. If se == T, the returned data frames also have columns indicating z-statistics and p-values that are calculated by normal assumption and input level, and with CIs if CI == T.

Value

A data frame of estimated average marginal effects for all independent variables in the fitted two-part model or the variables that term specifies, if se == T, with standard errors of AMEs, z-statistics and p-values that are calculated by normal assumption and input level, and with CIs if CI == T. If at = NULL (the default), then the data frame will have a number of rows equal to the number of concerned variables. Otherwise, a data list of AMEs of concerned variables, or a data frame

of AMEs if there's only one interested variable, is returned and the number of rows in the data frame for each variable will be a multiple thereof based upon the number of combinations of values specified in at.

Author(s)

Yajie Duan, Birol Emir, Griffith Bell and Javier Cabrera

References

Belotti, F., Deb, P., Manning, W.G. and Norton, E.C. (2015). twopm: Two-part models. *The Stata Journal*, 15(1), pp.3-20.

Leeper, T.J. (2017). Interpreting regression results using average marginal effects with R's margins. *Available at the comprehensive R Archive Network (CRAN)*, pp.1-32.

Leeper, T.J., Arnold, J. and Arel-Bundock, V. (2017). Package "margins". *accessed December*, 5, p.2019.

See Also

twopartm-class, tpm, predict-methods, margin, glm

Examples

```
##data about health expenditures, i.e., non-negative continuous response
data(meps,package = "twopartm")


##fit two-part model with different regressors in both parts, with probit
##regression model for the first part, and glm with Gamma family with log
##link for the second-part model
tpmodel = tpm(formula_part1 = exp_tot~female+age, formula_part2 =
exp_tot~female+age+ed_colplus,data = meps,link_part1 = "logit",
family_part2 = Gamma(link = "log"))

tpmodel

summary(tpmodel)

##AMEs for all variables with standard errors and CIs
AME(tpmodel)

##AMEs for variable "female" with standard errors and CIs at age
##40,and 60 respectively
AME(tpmodel,term = "female",at = list(age = c(40,60)))




##data for count response
data("bioChemists")
```

```
##fit two-part model with the same regressors in both parts, with logistic
##regression model for the first part, and poisson regression model with
##default log link for the second-part model
tpmodel = tpm(art ~ .,data = bioChemists,link_part1 = "logit",
family_part2 = poisson)

tpmodel


##AMEs for variable "phd" if all are women
AME(tpmodel,term = "phd",at = list(fem = "Women"))

##AMEs for variable "ment" when all are women and the numbers
##of children aged 5 or younger are 1,3, with standard errors
##by bootstrap methods, and CIs by bootstrap quantiles
AME(tpmodel,term = "ment",at = list(fem = "Women",kid5 = c(1,3)),
se.method = "bootstrap",CI.boots = TRUE,iter = 15)
```

`bioChemists` *Articles by Graduate Students in Biochemistry Ph.D. Programs*

Description

A sample of 915 biochemistry graduate students.

Usage

```
data("bioChemists")
```

Format

art count of articles produced during last 3 years of Ph.D.

fem factor indicating gender of student, with levels Men and Women

mar factor indicating marital status of student, with levels Single and Married

kid5 number of children aged 5 or younger

phd prestige of Ph.D. department

ment count of articles produced by Ph.D. mentor during last 3 years

Details

This data set is taken from package **pscl** provided by Simon Jackman.

Source

found in Stata format at https://jslsoc.sitehost.iu.edu/stata/spex_data/couart2.dta

References

Long, J. Scott. (1990). The origins of sex difference in science. *Social Forces*, **68**, 1297–1315.

Long, J. Scott. (1997) *Regression Models for Categorical and Limited Dependent Variables*, Thousand Oaks, California: Sage.

`coef-methods` *Method for Function* `coef` *for Two-part Model Objects in Package* **twopartm**

Description

The `coef` method for `twopartm-class` that extracts model coefficients from a fitted two-part regression model object of class `twopartm`.

Usage

```
## S4 method for signature 'twopartm'
coef(object,model = c("tpm","model1","model2"),...)
```

Arguments

`object`	a fitted two-part model object of class `twopartm` as returned by `tpm`.
`model`	character specifying for which part of the model the coefficients should be extracted. It could be either "tpm" for the full two-part model, or "model1", "model2" for the first-part model and the second-part model respectively.The default is "tpm".
`...`	arguments passed to `coef` in the default setup.

Details

The `coef` method for `twopartm-class` by default return a list including two vectors of estimated coefficients for both parts models. By setting the `model` argument, the model coefficients for the corresponding model component can be extracted.

Value

Coefficients extracted from the model object `twopartm`.

With argument `model == "tpm"` this will be a list of two numeric vectors of model coefficients for both parts models. With `model == "model1" | "model2"` it will be a numeric vector of coefficients for the selected part's model.

Author(s)

Yajie Duan, Birol Emir, Griffith Bell and Javier Cabrera

References

Belotti, F., Deb, P., Manning, W.G. and Norton, E.C. (2015). twopm: Two-part models. *The Stata Journal*, 15(1), pp.3-20.

Chambers, J. M. and Hastie, T. J. (1992) *Statistical Models in S*. Wadsworth & Brooks/Cole.

See Also

twopartm-class, glm,coef, tpm

Examples

```
##data about health expenditures, i.e., non-negative continuous response
data(meps,package = "twopartm")


##fit two-part model with the same regressors in both parts, with logistic
##regression model for the first part, and glm with Gamma family with log
##link for the second-part model
tpmodel = tpm(exp_tot~female+age, data = meps,link_part1 = "logit",
family_part2 = Gamma(link = "log"))

tpmodel

##summary information
summary(tpmodel)

##estimated coefficients for both parts
coef(tpmodel)

##estimated coefficients for the first-part model
coef(tpmodel,model = "model1")
```

FiellerRatio *Ratio of two Gaussian random variables with CI by Fieller's theorem*

Description

Calculate ratio of two Gaussian random variables with confidence intervals obtained by Fieller's theorem.

Usage

```
FiellerRatio(xest,yest,V,alpha = 0.05)
```

Arguments

xest	an estimate of one Gaussian random variable as numerator.
yest	an estimate of one Gaussian random variable as denominator.
V	Covariance matrix of two estimates.
alpha	the alpha (significant) level of the confidence interval. The default value is 0.05.

Details

Let X, Y be Gaussian random variables (or normally distributed estimators) with estimates xest and yest, and the ratio of interest E(X)/E(Y). An intuitive point–estimate for the ratio of interest is $xest/yest$. Fieller's theorem allows the calculation of a confidence interval for the ratio of two population means given estimates and covariance matrix.

Value

A numeric vector including the ratio of two estimates, and the bounds of its confidence interval, if the denominator is significantly different from zero. Otherwise, if the denominator is not significantly different from zero but the confidence set is exclusive, a numeric vector including the ratio of two estimates, and the bounds of its exclusive confidence set is returned.

Author(s)

Yajie Duan, Javier Cabrera and Birol Emir

References

Cabrera, J. and McDougall, A. (2002). Statistical consulting. *Springer Science & Business Media*.

Franz, V. H. (2007). Ratios: A short guide to confidence limits and proper use. *arXiv preprint arXiv:0710.2024*.

Fieller, E. C. (1954). Some problems in interval estimation. *Journal of the Royal Statistical Society: Series B (Methodological)*, 16(2), 175-185.

Zerbe, G. O. (1978). On Fieller's theorem and the general linear model. *The American Statistician*, 32(3), 103-105.

Young, D. A., Zerbe, G. O., & Hay Jr, W. W. (1997). Fieller's theorem, Scheffé simultaneous confidence intervals, and ratios of parameters of linear and nonlinear mixed-effects models. *Biometrics*, 838-847.

Examples

```
## example data: bivariate Gaussian random variables
library(MASS)
out <- mvrnorm(1000, mu = c(10,3), Sigma = matrix(c(1,0.2,0.2,1),
ncol = 2))

##ratio with CI between two sample means
FiellerRatio(mean(out[,1]),mean(out[,2]),V = cov(out)/1000)
```

```
##case that the denominator is not significantly different from zero
##but the confidence set is exclusive
out <- mvrnorm(1000, mu = c(3,0.001), Sigma = matrix(c(1,0.2,0.2,1), ncol = 2))
FiellerRatio(mean(out[,1]),mean(out[,2]),V = cov(out)/1000)



##an example of calculating ratio of fitted parameters with CI in regression models
## Dobson (1990) Page 93: Randomized Controlled Trial :
counts <- c(18,17,15,20,10,20,25,13,12)
outcome <- gl(3,1,9)
treatment <- gl(3,3)
data.frame(treatment, outcome, counts) # showing data
glm.D93 <- glm(counts ~ outcome + treatment, family = poisson())
summary(glm.D93)

##obtain estimates and covariance matrix of concerned fitted parameters
xest <- as.numeric(coef(glm.D93)["outcome3"])
yest <- as.numeric(coef(glm.D93)["outcome2"])
V <- vcov(glm.D93)[c("outcome3","outcome2"),c("outcome3","outcome2")]

##ratio with CI between two fitted parameters
FiellerRatio(xest,yest,V)
```

`logLik-methods` *Method for Function* `logLik` *for Two-part Model Objects in Package* **twopartm**

Description

The `logLik` method for `twopartm-class` that extracts log-likelihood from a fitted two-part regression model object of class `twopartm`.

Usage

```
## S4 method for signature 'twopartm'
logLik(object,...)
```

Arguments

`object`	a fitted two-part model object of class `twopartm` as returned by `tpm`.
`...`	arguments passed to `logLik` in the default setup.

Details

The `logLik` method for `twopartm-class` returns an object of class `logLik`, including the log likelihood value with degree of freedom of a fitted two-part regression model object of class `twopartm`.

Value

Returns an object of class logLik for model object twopartm.This is a number with at least one attribute, "df" (degrees of freedom), giving the number of (estimated) parameters in the two-part model.

Author(s)

Yajie Duan, Birol Emir, Griffith Bell and Javier Cabrera

References

Belotti, F., Deb, P., Manning, W.G. and Norton, E.C. (2015). twopm: Two-part models. *The Stata Journal*, 15(1), pp.3-20.

Harville, D.A. (1974). Bayesian inference for variance components using only error contrasts. *Biometrika*, 61, 383–385. doi: 10.2307/2334370.

See Also

twopartm-class, glm,logLik.lm, tpm

Examples

```
##data about health expenditures, i.e., non-negative continuous response
data(meps,package = "twopartm")


##fit two-part model with the same regressors in both parts, with logistic
##regression model for the first part, and glm with Gamma family with log
##link for the second-part model
tpmodel = tpm(exp_tot~female+age, data = meps,link_part1 = "logit",
family_part2 = Gamma(link = "log"))

tpmodel

##summary information
summary(tpmodel)

##log-likehood
logLik(tpmodel)
```

---

margin *Predictive Margins with CIs for Two-part Model Objects*

Description

Calculate predictive margins with CIs for variables from a fitted two-part regression model object of class twopartm.

Usage

```
## S4 method for signature 'twopartm'
margin(object, newdata = NULL, term = NULL, value = NULL,
se = TRUE, se.method = c("delta","bootstrap"), CI = TRUE, CI.boots = FALSE,
level = 0.95,eps = 1e-7,na.action = na.pass, iter = 50)
```

Arguments

object	a fitted two-part model object of class twopartm as returned by tpm.
newdata	optionally, a data frame in which to look for variables with which to calculate predictive margins. If omitted, the original observations are used.
term	A character vector with the names of variables for which to compute the predictive margins. The default (NULL) returns predictive margins for all variables.
value	A list of one or more named vectors, specifically values at which to calculate the predictive marginal effects. If omitted, for factor or logical variables, predictive margins at all the levels are calculated, and for numeric (and integer) variables, predictive margins at the mean values among observations are calculated. Note: This does not calculate predictive margins for subgroups but rather for whole datasets; to obtain subgroup margins, subset data directly.
se	logical switch indicating if standard errors are required.
se.method	A character string indicating the type of estimation procedure to use for estimating variances of predictive margins. The default ("delta") uses the delta method. The alternative is "bootstrap", which uses bootstrap estimation.
CI	logical switch indicating if confidence intervals are required.
CI.boots	if se.method == "bootstrap", logical switch indicating if confidence intervals are obtained by normal-based or by bootstrap quantiles.
level	A numeric value specifying the confidence level for calculating p-values and confidence intervals.
eps	A numeric value specifying the "step" to use when calculating numerical derivatives.
na.action	function determining what should be done with missing values in newdata. The default is to predict NA.
iter	if se.method == "bootstrap", the number of bootstrap iterations.

Details

Predictive margins are calculated by taking average values of predictive responses at specified levels for factor and logical variables, or specified values for continuous variables. If value = NULL (the default), for factor or logical variables, predictive margins at all the levels are calculated, and for numeric (and integer) variables, predictive margins at the mean values among observations are calculated. Otherwise, predictive margins at values specified in value are calculated.Margins are calculated based on the original observations used in the fitted two-part model, or the new data set that newdata inputs.

The standard errors of predictive margins could be calculated using delta method or bootstrap method. The delta method considers the average Jacobian vectors among observations with respect to both models' parameters, assuming independence between models from two parts. The Jacobian vectors are approximated by numerical differentiations. The bootstrap method generates bootstrap samples to fit two-part models, and get variances and inverted bootstrap quantile CIs or normal-based CIs of predictive margins. If `se == T`, the returned data frames also have columns indicating z-statistics and p-values that are calculated by normal assumption and input `level`, and with CIs if `CI == T`.

If there are two or more values or levels of variables to be concerned for predictive margins, the ratios between calculated predictive margins are calculated. If `se == T` and `CI == T`, CIs at levels specified by `level` of the ratios are calculated by Fieller's theorem using the covariance matrices between predictive margins obtained by delta or bootstrap methods. Fieller's theorem is a statistical method to calculate a confidence interval for the ratio of two means.

Value

A list of data frames of estimated predictive margins for all independent variables in the fitted two-part model or the variables that `term` specifies, if `se == T`, with standard errors of AMEs, z-statistics and p-values that are calculated by normal assumption and input `level`, and with CIs if `CI == T`. If `values = NULL` (the default), for factor or logical variables, predictive margins at all the levels are returned, and for numeric (and integer) variables, predictive margins at the mean values among observations are returned. Otherwise, predictive margins at specified values are returned. If there are two or more values or levels of variables to be concerned for predictive margins, a data frame including ratios between calculated predictive margins is also returned, if `se == T` and `CI == T`, with CIs at levels specified by `level` of the ratios.

Author(s)

Yajie Duan, Birol Emir, Griffith Bell and Javier Cabrera

References

Belotti, F., Deb, P., Manning, W.G. and Norton, E.C. (2015). twopm: Two-part models. *The Stata Journal*, 15(1), pp.3-20.

Leeper, T.J. (2017). Interpreting regression results using average marginal effects with R's margins. *Available at the comprehensive R Archive Network (CRAN)*, pp.1-32.

Leeper, T.J., Arnold, J. and Arel-Bundock, V. (2017). Package 'margins'. *accessed December*, 5, p.2019.

Fieller, E.C. (1954). Some problems in interval estimation. *Journal of the Royal Statistical Society: Series B (Methodological)*, 16(2), pp.175-185.

O'Hagan, A., Stevens, J.W. and Montmartin, J. (2000). Inference for the cost-effectiveness acceptability curve and cost-effectiveness ratio. *Pharmacoeconomics*, 17(4), pp.339-349.

See Also

`twopartm-class`, `tpm`, `predict-methods`, `AME`, `glm`

Examples

```
##data about health expenditures, i.e., non-negative continuous response
data(meps,package = "twopartm")


##fit two-part model with different regressors in both parts, with probit
##regression model for the first part, and glm with Gamma family with log
##link for the second-part model
tpmodel = tpm(formula_part1 = exp_tot~female+age, formula_part2 =
exp_tot~female+age+ed_colplus,data = meps,link_part1 = "logit",
family_part2 = Gamma(link = "log"))

tpmodel

summary(tpmodel)

##Predictive margins and corresponding ratios for all variables with
##standard errors and CIs.
margin(tpmodel)


##Predictive margins and corresponding ratios for female, age at
##20,40,60,80, and more than college education level, resepectively
margin(tpmodel,value = list(female = 1,age = c(50,70),ed_colplus = 1))



##data for count response
data("bioChemists")

##fit two-part model with the same regressors in both parts, with logistic
##regression model for the first part, and poisson regression model with
##default log link for the second-part model
tpmodel = tpm(art ~ .,data = bioChemists,link_part1 = "logit",
family_part2 = poisson)

tpmodel


##Predictive margins and corresponding ratios for variable "kid5"
##at 2,3, with standard errors by bootstrap methods,
##and CIs by bootstrap quantiles
margin(tpmodel,term = "kid5",value = list(kid5 = c(2,3)),
se.method = "bootstrap",CI.boots = TRUE,iter = 20)

##Predictive margins and corresponding ratios for variable "ment" at
##6,7,8, without standard errors and CIs
margin(tpmodel,term = "ment",value = list(ment = c(6,7,8)),se = FALSE)


##Predictive margins and corresponding ratios for all the levels of
##variable "mar", and for variable "phd" at 2.5,3.2, calculated on
```

```
##the first 500 observations, with standard errors and CIs
margin(tpmodel,newdata = bioChemists[1:500,],term = c("phd","mar"),
value = list(phd = c(2.5,3.2)))
```

---

meps *A Sample of Medical Expenditure Panel Survey 2004 data*

Description

A sample of MEPS 2004 data including 19386 observations.

Usage

```
data("meps")
```

Format

duid Dwelling unit id

dupersid Person id (unique)

hieuidx Health insurance eligibility unit id

wtdper Sampling weight for person

age Age

female Female

race_bl Black

race_oth Other race, non-white and non-black

eth_hisp Hispanic

famsize Size of responding annualized family

ed_hs High school education

ed_hsplus Some college education

ed_col College education

ed_colplus More than college education

lninc ln(family income)

reg_midw Midwest region

reg_south South region

reg_west West region

anylim Any disability

mcs12 Mental health component of SF12

pcs12 Physical health component of SF12

ins_mcare Medicare insurance

ins_mcaid Medicaid insurance

ins_unins Uninsured

ins_dent Dental insurance, prorated

exp_tot Total medical care expenses

exp_dent Dental care expenses

exp_self Total expenses paid by self or family

use_disch # hospital discharges

use_los # nights in hospital

use_dent # dental visits

use_rx # prescriptions and refills

Details

This data set is taken from book Health econometrics using Stata (Vol. 3).

Source

found in Stata format at https://www.stata-press.com/data/heus.html

References

Deb, P., Norton, E.C. and Manning, W.G. (2017). Health econometrics using Stata (Vol. 3). *College Station, TX: Stata press*.

`plot-methods` *Method for Function* `plot` *for Two-part Model Objects in Package* **twopartm**

Description

The `plot` method for `twopartm-class` that provides plot diagnostics for a fitted two-part regression model object of class `twopartm`.

Usage

```
## S4 method for signature 'twopartm,missing'
plot(x, y, ...)
```

Arguments

`x`	an object of class `twopartm`.
`y`	not used.
`...`	arguments passed to `plot.lm` in the default setup.

Details

The plot method for twopartm-class returns the residual plot for the full two-part model, and also six plots for each part's glm model. Six plots are: a plot of residuals against fitted values, a Scale-Location plot of sqrt(| residuals |) against fitted values, a Normal Q-Q plot, a plot of Cook's distances versus row labels, a plot of residuals against leverages, and a plot of Cook's distances against leverage/(1-leverage). By default, the first three plots and the fifth one of each part's model are provided. The plots for each part's model could be selected by argument which of function plot.lm for glm model object.

Value

Returns residual plot for the full two-part model, and plot diagnostics for each part's model from an object twopartm.

Author(s)

Yajie Duan, Birol Emir, Griffith Bell and Javier Cabrera

References

Belotti, F., Deb, P., Manning, W.G. and Norton, E.C. (2015). twopm: Two-part models. *The Stata Journal*, 15(1), pp.3-20.

Belsley, D. A., Kuh, E. and Welsch, R. E. (1980). *Regression Diagnostics*. New York: Wiley.

Cook, R. D. and Weisberg, S. (1982). *Residuals and Influence in Regression*. London: Chapman and Hall.

See Also

twopartm-class, glm,plot.lm, tpm

Examples

```
##data about health expenditures, i.e., non-negative continuous response
data(meps,package = "twopartm")


##fit two-part model with the same regressors in both parts, with logistic
##regression model for the first part, and glm with Gamma family with log
##link for the second-part model
tpmodel = tpm(exp_tot~female+age, data = meps,link_part1 = "logit",
family_part2 = Gamma(link = "log"))

tpmodel

##summary information
summary(tpmodel)

##plots for two-part model
plot(tpmodel)
```

predict-methods *Method for Function* predict *for Two-part Model Fits in Package* **twopartm**

Description

Obtains predictions and optionally estimates standard errors of those predictions from a fitted two-part model object of class twopartm.

Usage

```
## S4 method for signature 'twopartm'
predict(object,newdata = NULL, se.fit = FALSE,
dispersion_part1 = NULL,dispersion_part2 = NULL,na.action = na.pass)
```

Arguments

object	a fitted two-part model object of class twopartm as returned by tpm.
newdata	optionally, a data frame in which to look for variables with which to predict. If omitted, the original observations are used.
se.fit	logical switch indicating if standard errors are required.
dispersion_part1	the dispersion of the GLM fit to be assumed in computing the standard errors for the first-part model. If omitted, that returned by summary applied to the first-part model object is used.
dispersion_part2	the dispersion of the GLM fit to be assumed in computing the standard errors for the second-part model. If omitted, that returned by summary applied to the second-part model object is used.
na.action	function determining what should be done with missing values in newdata. The default is to predict NA.

Details

The predictive values and corresponding standard errors are on the scales of the response variable not considering the link functions. The predictive responses are calculated by multiplying the predicted probabilities of non-zero responses and the fitted means of non-zero values. The prediction standard errors are calculated using delta method combining prediction standard errors from the models of both parts. If newdata is omitted the predictions are based on the data used for the fit. In that case how cases with missing values in the original fit is determined by the na.action argument of that fit.

Value

If `se.fit = FALSE`, a vector or matrix of predictions. If `se.fit = TRUE`, a list with components

`fit` Predictions, as for `se.fit = FALSE`.

`se.fit` Estimated standard errors.

`residual.scale_part1`
A scalar giving the square root of the dispersion used in computing the standard errors for the first-part model.

`residual.scale_part2`
A scalar giving the square root of the dispersion used in computing the standard errors for the second-part model.

Author(s)

Yajie Duan, Birol Emir, Griffith Bell and Javier Cabrera

References

Belotti, F., Deb, P., Manning, W.G. and Norton, E.C. (2015). twopm: Two-part models. *The Stata Journal*, 15(1), pp.3-20.

Chambers, J. M. and Hastie, T. J. (1992) *Statistical Models in S*. Wadsworth & Brooks/Cole.

See Also

`twopartm-class`, `tpm`, `AME`, `margin`, `glm`, `predict.glm`

Examples

```
##data about health expenditures, i.e., non-negative continuous response
data(meps,package = "twopartm")


##fit two-part model with the same regressors in both parts, with logistic
##regression model for the first part, and glm with Gamma family with log
##link for the second-part model
tpmodel = tpm(exp_tot~female+age, data = meps,link_part1 = "logit",
family_part2 = Gamma(link = "log"))

tpmodel

##get prediction results with standard errors for the
##first 500 observations in the dataset
predict(tpmodel,newdata = meps[1:500,],se.fit = TRUE)


##data for count response
data("bioChemists")

##fit two-part model with the same regressors in both parts, with logistic
##regression model for the first part, and poisson regression model with
```

```
##default log link for the second-part model
tpmodel = tpm(art ~ .,data = bioChemists,link_part1 = "logit",
family_part2 = poisson)

tpmodel

##get predictive counts
predict(tpmodel)
```

`residuals-methods` *Method for Function* `residuals` *for Two-part Model Objects in Package* **twopartm**

Description

The `residuals` method for `twopartm-class` that extracts model residuals from a fitted two-part regression model object of class `twopartm`.

Usage

```
## S4 method for signature 'twopartm'
residuals(object,model = c("tpm","model1","model2"),
type = c("deviance", "pearson", "working","response", "partial"))
```

Arguments

`object`	a fitted two-part model object of class `twopartm` as returned by `tpm`.
`model`	character specifying for which part of the model the residuals should be extracted. It could be either "tpm" for the full two-part model, or "model1", "model2" for the first-part model and the second-part model respectively.The default is "tpm".
`type`	if `model == "model1" | "model2"`, the type of residuals which should be returned. The alternatives are: "response" (default), "pearson", "working", "deviance", and "partial". Can be abbreviated.

Details

The `residuals` method for `twopartm-class` can compute raw response residuals (observed - fitted) for the full two-part model, or different types of residues from both parts models respectively. The references define the types of residuals: Davison & Snell is a good reference for the usages of each. The partial residuals are a matrix of working residuals, with each column formed by omitting a term from the model.

Value

Returns a numerical vector of residuals, either for the full two-part model, or two separate part models from an object `twopartm`.

Author(s)

Yajie Duan, Birol Emir, Griffith Bell and Javier Cabrera

References

Belotti, F., Deb, P., Manning, W.G. and Norton, E.C. (2015). twopm: Two-part models. *The Stata Journal*, 15(1), pp.3-20.

Davison, A. C. and Snell, E. J. (1991). Residuals and diagnostics. *Statistical Theory and Modeling*. In Honour of Sir David Cox, FRS, eds. Hinkley, D. V., Reid, N. and Snell, E. J., Chapman and Hall.

See Also

twopartm-class, glm,residuals.glm, tpm, predict-methods

Examples

```
##data about health expenditures, i.e., non-negative continuous response
data(meps,package = "twopartm")


##fit two-part model with the same regressors in both parts, with logistic
##regression model for the first part, and glm with Gamma family with log
##link for the second-part model
tpmodel = tpm(exp_tot~female+age, data = meps,link_part1 = "logit",
family_part2 = Gamma(link = "log"))

tpmodel

##summary information
summary(tpmodel)

##response residues from the full two-part model
residuals(tpmodel)

##response residues from the first-part model
residuals(tpmodel,model = "model1")

##deviance residues from the second-part model
residuals(tpmodel,model = "model2",type = "deviance")
```

---

tpm *Fit Two-part Regression Models for Zero-inflated Data*

Description

Fit two-part regression models for zero-inflated data. The first-model is a binomial regression model for indicators about any non-zero responses. The second-model is a generalized linear regression model for non-zero response values.

Usage

```
tpm(formula_part1, formula_part2 = NULL,data, link_part1 = c("logit",
"probit", "cloglog", "cauchit", "log"), family_part2 = gaussian(), weights = NULL, ...)

## S4 method for signature 'twopartm'
summary(object,...)
```

Arguments

formula_part1	formula specifying the dependent variable and the regressors used for the first-part model, i.e., the binomial model for probabilities of non-zero responses. If formula_part2 is NULL, the same regressors specified here are employed in both parts.
formula_part2	formula specifying the dependent variable and the regressors used for the second-part model, i.e., the glm model for non-zero responses. If it's NULL, the same regressors specified in formula_part1 are employed in the second-part model.
data	a data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the models for both parts.
link_part1	character string specifying the link function of the first-part model, i.e., the binomial model for probabilities of non-zero responses. It could be logit, probit, cauchit, (corresponding to logistic, normal and Cauchy CDFs respectively) log or cloglog (complementary log-log).
family_part2	a description of the error distribution and link function to be used in the second-part model, i.e., the glm model for non-zero responses. This can be a character string naming a family function, a family function or the result of a call to a family function.
weights	an optional numeric vector of weights to be used in the fitting process for both parts. Should be NULL or a numeric vector.
object	a fitted two-part model object of class twopartm as returned by tpm.
...	arguments passed to glm or summary.glm in the default setup.

Details

Two-part models are two-component models for zero-inflated data, one modeling indicators about any non-zero responses and another modeling non-zero response values. It models the zeros and non-zeros as two separate processes. For instance, in explaining individual annual health expenditure, the event is represented by a specific disease. If the illness occurs, then some not-for-free treatment will be needed, and a positive expense will be observed. In these situations, a two-part model allows the censoring mechanism and the outcome to be modeled to use separate processes. In other words, it permits the zeros and nonzeros to be generated by different densities as a special type of mixture model.

In function tpm, the zeros are handled using the first-model, specifically a glm with binomial family and specified link function for the probability of a non-zero outcome. The second-model is a glm with specified family function with link for non-zero values. The regressors for both parts could be

different and specified separately. The two components of the model are estimated separately using glm calls, with iterated reweighted least-squares (IRLS) optimization.

The returned fitted model object is of class twopartm.A set of standard extractor functions for fitted model objects is available for objects of class twopartm, including methods to the generic functions print, summary, plot, coef, logLik, residuals, and predict.See predict-methods for more details on prediction method.

The summary method lists result summaries of two fitted glm models for each part respectively.

Value

tpm returns an object of class twopartm.

summary returns a list with two objects of class summary.glm for first-part model and second-part model respectively.

Author(s)

Yajie Duan, Birol Emir, Griffith Bell and Javier Cabrera

References

Belotti, F., Deb, P., Manning, W.G. and Norton, E.C. (2015). twopm: Two-part models. *The Stata Journal*, 15(1), pp.3-20.

Hay, J. W., and R. J. Olsen. (1984). Let them eat cake: A note on comparing alternative models of the demand for medical care. *Journal of Business and Economic Statistics* 2: 279–282.

Leung, S. F., and S. Yu. (1996). On the choice between sample selection and two-part models. *Journal of Econometrics* 72: 197–229

Mihaylova, B., A. Briggs, A. O'Hagan, and S. G. Thompson. (2011). Review of statistical methods for analyzing healthcare resources and costs. *Health Economics* 20: 897–916.

See Also

twopartm-class, glm, summary.glm, predict-methods

Examples

```
##data about health expenditures, i.e., non-negative continuous response
data(meps,package = "twopartm")


##fit two-part model with the same regressors in both parts, with logistic
##regression model for the first part, and glm with Gamma family with log
##link for the second-part model
tpmodel = tpm(exp_tot~female+age, data = meps,link_part1 = "logit",
family_part2 = Gamma(link = "log"))

tpmodel

summary(tpmodel)
```

```
##fit two-part model with different regressors in both parts, with probit
##regression model for the first part, and glm with Gamma family with log
##link for the second-part model
tpmodel = tpm(formula_part1 = exp_tot~female+age, formula_part2 =
exp_tot~female+age+ed_colplus,data = meps,link_part1 = "probit",
family_part2 = Gamma(link = "log"))

tpmodel

summary(tpmodel)

##fit two-part model with transformed regressors and randomly assigned weights
meps$weights = sample(1:30,nrow(meps),replace = TRUE)

tpmodel = tpm(formula_part1 = exp_tot~female+age, formula_part2 =
exp_tot~female+I(age^2)+ed_colplus,data = meps,link_part1 = "logit",
family_part2 = Gamma(link = "log"),weights = meps$weights)

tpmodel

summary(tpmodel)


##data for count response
data("bioChemists")

##fit two-part model with the same regressors in both parts, with logistic
##regression model for the first part, and poisson regression model with
##default log link for the second-part model
tpmodel = tpm(art ~ .,data = bioChemists,link_part1 = "logit",
family_part2 = poisson)

tpmodel

summary(tpmodel)
```

`twopartm-class` *Class* `twopartm`

Description

A fitted two-part regression model by `tpm`.

Slots

`formula_part1` Formula specified for the first-part model, i.e., the binomial model for indicators about any non-zero responses.

`formula_part2` Formula specified for the second-part model, i.e., the glm model for non-zero responses.

`data` Data set used to fit the two-part model. It's the same data set as the `data` argument in tpm.

`n:` Number of observations used in the two-part model (with weights > 0).

`n_part1` Number of of observations used in the first-part model (with weights > 0), i.e., the binomial model for indicators about any non-zero responses.

`n_part2` Number of of observations used in the second-part model (with weights > 0), i.e., the glm model for non-zero responses.

`data_model1` The model frame for the first-part model, i.e., the binomial model for indicators about any non-zero responses.

`data_model2` The model frame for the second-part model, i.e., the glm model for non-zero responses.

`model_part1` An object of class `glm` of the fitted first-part model, i.e., the binomial model for indicators about any non-zero responses.

`model_part2` An object of class `glm` of the fitted second-part model, i.e., the glm model for non-zero responses.

`link_part1` Character string describing the link function of the first-part model, i.e., the binomial model for indicators about any non-zero responses.

`family_part2` The family object used in the second-part model, i.e., the glm model for non-zero responses.

`weights` A vector of weights used in the two-part model fitting, or NULL if no weights used.

`fitted` Fitted mean values by the two-part model, obtained by multiplying the fitted probabilities of non-zero responses and the fitted means of non-zero responses.

`residuals` A vector of raw residuals (observed - fitted).

`loglik` Log-likelihood values of the fitted two-part model.

`y` The response vector.

Author(s)

Yajie Duan, Birol Emir, Griffith Bell and Javier Cabrera

See Also

tpm, AME, margin, glm

Examples

```
##data about health expenditures, i.e., non-negative continuous response
data(meps,package = "twopartm")


##fit two-part model with the same regressors in both parts, with logistic regression model
##for the first part, and glm with Gamma family with log link for the second-part model
tpmodel = tpm(formula_part1 = exp_tot~female+age, formula_part2 = exp_tot~female+age+ed_colplus,
data = meps,link_part1 = "logit",family_part2 = Gamma(link = "log"))
```

```
##get the formula specified for the first-part model
tpmodel@formula_part1

##get the formula specified for the second-part model
tpmodel@formula_part2

##get the log-likelihood for the fitted two-part model
tpmodel@loglik

##get the fitted glm model for the first part
tpmodel@model_part1
##get the fitted glm model for the second part
tpmodel@model_part2
```

Index

∗ **classes**
twopartm-class, 23
∗ **datasets**
bioChemists, 5
meps, 14
∗ **methods**
AME, 2
coef-methods, 6
FiellerRatio, 7
logLik-methods, 9
margin, 10
plot-methods, 15
predict-methods, 17
residuals-methods, 19
∗ **models**
tpm, 20
∗ **regression**
tpm, 20

AME, 2, *12*, *18*, *24*
AME,twopartm-method *(*AME*)*, 2

bioChemists, 5

coef, *6*, *7*
coef,twopartm-method *(*coef-methods*)*, 6
coef-methods, 6

FiellerRatio, 7
FiellerRatio,numeric-method *(*FiellerRatio*)*, 7

glm, *4*, *7*, *10*, *12*, *16*, *18*, *20–22*, *24*

logLik, *9*
logLik,twopartm-method *(*logLik-methods*)*, 9
logLik-methods, 9
logLik.lm, *10*

margin, *4*, 10, *18*, *24*
margin,twopartm-method *(*margin*)*, 10
meps, 14

plot,twopartm,missing-method *(*plot-methods*)*, 15
plot-methods, 15
plot.lm, *15*, *16*
predict,twopartm-method *(*predict-methods*)*, 17
predict-methods, 17
predict.glm, *18*
print.twopartm *(*tpm*)*, 20

residuals,twopartm-method *(*residuals-methods*)*, 19
residuals-methods, 19
residuals.glm, *20*

show,twopartm-method *(*tpm*)*, 20
summary,twopartm-method *(*tpm*)*, 20
summary.glm, *21*, *22*

tpm, *2*, *4*, *6*, *7*, *9–12*, *16–19*, 20, *20*, *23*, *24*
tpm,formula-method *(*tpm*)*, 20
twopartm-class, 23