

Package ‘RobustBetaReg’

August 23, 2021

Version 0.0.1.000

Title Robust of Beta Regression Model

Author Yuri Maluf [aut,cre], Silvia Ferrari [ths]

Maintainer Yuri Maluf <yurimaluf@gmail.com>

Description

Robust estimators for beta regression model, including robust test and graphical diagnostic tools.

Depends R (>= 3.0.0), betareg, nleqslv

Imports Rmpfr, rstudioapi, crayon, pracma, numDeriv, Formula, robustbase, Matrix, parallel

Repository GitHub

URL <https://github.com/yurimaluf/RobustBetaReg.git>

License GPL-3

NeedsCompilation yes

Encoding UTF-8

R topics documented:

| | |
|---------------------------------|----------|
| degbeta | 2 |
| plot | 2 |
| plotenvelope | 3 |
| residuals | 4 |
| robustbetareg | 5 |
| robustbetareg.control | 7 |
| SaddlepointTest | 7 |
| WaldTypeTest | 8 |
| Index | 9 |

| | |
|---------|-----------------------------------|
| degbeta | <i>The EGB of the second type</i> |
|---------|-----------------------------------|

Description

Density and random generation for the exponential generalized beta (EGB) of the second type,

Usage

```
degbeta(y_star, mu, phi, log = FALSE)
```

```
regbeta(n, mu, phi)
```

Arguments

| | |
|--------|---|
| mu | mu parameter. |
| phi | phi parameter. |
| log | a logical value. If TRUE return the log of density function. |
| n | number of observations. |
| y_star | logit transformation of original data ($y \in (0, 1)$ and $y^* \in (-\infty, \infty)$). |

Details

For more details see

Value

Return the value of density function or a random sample.

| | |
|------|--|
| plot | <i>Interactive plots for diagnostic of robust betareg models</i> |
|------|--|

Description

Several types of standard diagnostic plots can be produced interactively, involving various kinds of residuals, influence measures, weights etc.

Usage

```
## S3 method for class 'LSMLE'
plot(object, ask = TRUE, ...)
## S3 method for class 'LMDPDE'
plot(object, ask = TRUE, ...)
```

Arguments

| | |
|--------|--|
| object | fitted model object of class "LSMLE" or "LMDPDE". |
| ask | logical. If TRUE the user is asked before each plot. |
| ... | other parameters to be passed through to plotting functions. |

Examples

```
fit=robustbetareg(I(food/income)~income+persons|1,data=FoodExpenditure,alpha=0.08)
plot(fit)
```

| | |
|--------------|--|
| plotenvelope | <i>Simulated Envelope of Residuals</i> |
|--------------|--|

Description

Plot a simulated envelope of beta residuals, from LSMLE and LMDPDE objects.

Usage

```
plotenvelope(object, n.sim, conf, control = robustbetareg.control(...), ...)
```

Arguments

| | |
|---------|---|
| object | Fitted model object of class "LSMLE" or "LMDPDE" (see robustbetareg). |
| n.sim | the number of simulation sample. Deafault n.sim=50. |
| conf | the confidence level of the envelopes required. The default is to find 95 confidence envelopes. |
| control | a list of control arguments specified via robustbetareg.control . |
| ... | other parameters to be passed through to plotting functions. |

Value

Return a simulated envelope graphic.

Examples

```
fit=robustbetareg(I(food/income)~income+persons|1,data=FoodExpenditure,alpha=0.08)
plotenvelope(fit,n.sim=100)
```

residuals

*Residuals Method for robustbetareg Objects***Description**

Extract various types of residuals from robust beta regression models: Pearson residuals (raw residuals scaled by square root of variance function) and different kinds of weighted residuals suggested by Espinheira et al. (2008) and Espinheira et al. (2017).

Usage

```
## S3 method for class 'LSMLE'
residuals(object, type = c("sweighted2", "pearson", "weighted",
"swighted", "swighted.gamma", "swighted2.gamma", "combined",
"combined.projection"))

## S3 method for class 'LMDPDE'
residuals(object, type = c("sweighted2", "pearson", "weighted",
"swighted", "swighted.gamma", "swighted2.gamma", "combined",
"combined.projection"))
```

Arguments

| | |
|--------|---|
| object | fitted model object of class "LSMLE" or "LMDPDE". |
| type | character indicating type of residuals. |

Details

The definitions of the first four residuals are provided in Espinheira et al. (2008): Equation 2 for "pearson", Equation 6 for "weighted", Equation 7 for "swighted", and Equation 8 for "swighted2". For the last four residuals the definitions are described in Espinheira et al. (2017): Last equation of Equation 7 and Equation 10 for "swighted.gamma" and "swighted2.gamma" respectively, Equation 9 for "combined", and Equation 11 for "combined.projection".

References

Espinheira, P.L., Ferrari, S.L.P., and Cribari-Neto, F. (2008). On Beta Regression Residuals. *Journal of Applied Statistics*, 35(4), 407–419.

Espinheira, P.L., Santos, E.G. and Cribari-Neto, F. (2017). On nonlinear beta regression residuals. *Biometrical Journal*, 59(3), 445-461.

Examples

```
fit=robustbetareg(I(food/income)~income+persons|1,data=FoodExpenditure,alpha=0.08)
residuals(fit,type="swighted")
```

| | |
|---------------|-------------------------------|
| robustbetareg | <i>Robust Beta Regression</i> |
|---------------|-------------------------------|

Description

Fit robust beta regression models for rates and proportions via LSMLE and LMDPDE using a parametrization with mean (depending through a link function on the covariates) and precision parameter (called phi).

Usage

```
robustbetareg(formula, data, alpha, type = c("LSMLE", "LMDPDE"),
link = c("logit", "probit", "cloglog", "cauchit", "loglog"),
link.phi = NULL, control = robustbetareg.control(...),
model = TRUE, y = TRUE, ...)
```

```
LSMLE.fit(y, x, z, alpha = NULL, link = "logit", link.phi = "log",
control = robustbetareg.control(...), ...)
```

```
LMDPDE.fit(y, x, z, alpha = NULL, link = "logit", link.phi = "log",
control = robustbetareg.control(...), ...)
```

Arguments

| | |
|----------|--|
| formula | symbolic description of the model (of type $y \sim x$ or $y \sim x z$). |
| data | arguments controlling formula. |
| alpha | the tuning value within (0,1), for robust estimation. When alpha is equal to zero is equivalent of MLE. |
| type | character specification of the type of estimator. Currently, LSMLE (default) and LMDPDE. |
| link | character specification of the link function in the mean model (mu). Currently, "logit", "probit", "cloglog", "cauchit", "log", "loglog" are supported |
| link.phi | character specification of the link function in the precision model (phi). Currently, "identity", "log", "sqrt" are supported. The default is "log" unless formula is of type $y \sim x$ where the default is "identity" |
| control | a list of control arguments specified via robustbetareg.control . |
| model, y | logicals for robustbetareg. If TRUE the corresponding components of the fit (model frame, response, model matrix) are returned. For LSMLE.fit and LMDPDE.fit y must be a numeric response vector within (0,1). |
| x, z | numeric regressor matrix for mean and precision model respectively, defaulting to an intercept only. |
| ... | currently not used. |

Details

For more details see:...

Value

robustbetareg returns an object of class "LSMLE" or "LMDPDE" with a list of the following components:

| | |
|------------------|--|
| coefficients | A numeric vector of parameter estimates |
| vcov | the covariance matrix of all parameters in the model |
| converged | logical indicating successful convergence of nleqslv call |
| fitted.values | the vector of predicted values |
| start | the starting values for the parameters estimator |
| weights | the weights generated by robust estimator for each obs. |
| Tuning | the selected tuning parameter |
| residuals | a vector of standardized weighted residual 2 |
| n | number of observations |
| link | mean link function applied |
| link.phi | precision link function applied |
| Optimal.Tuning | logical indicating whether the auto selecting tuning algorithm was selected |
| pseudo.r.squared | pseudo R-squared value (squared correlation of linear predictor and link-transformed response) |
| control | the control arguments passed to auto selecting tuning algorithm and nleqslv call |
| std.error | the standard error of all parameters |
| call | the original function call |
| formula | the original formula |
| model | the full model frame |
| y | the response proportion vector |

robustbetareg.control *Control Parameter for Robust Beta Regression*

Description

Various parameters that control fitting of robust beta regression models using [robustbetareg](#).

Usage

```
robustbetareg.control(object, start = NULL, alpha.optimal = TRUE,
  tolerance = 1e-3, maxit = 250, L = 0.02, M = 3, ...)
```

Arguments

| | |
|---------------|---|
| object | fitted model object of class "LSMLE" or "LMDPDE". |
| start | a numeric vector with an initial guess of the root of estimation equation. |
| alpha.optimal | a logical value. If TRUE the tuning parameter should be selected automatic. |
| tolerance | numeric tolerance for convergence. |
| maxit | integer specifying the maxit argument of iterations used by the Newton-Raphson algorithm. |
| L | a parameter of auto selecting algorithm of tuning parameter (default L=0.02). |
| M | a integer parameter value of auto selecting algorithm of tuning parameter (default M=3). |
| ... | currently not used. |

Details

For more details see:...

Value

A list with the arguments specified.

SaddlepointTest *Robust Saddlepoint Test*

Description

Saddlepoint tests for both simple and composite hypothesis for independent but non-homogeneous observations, based on LSMLE and LMDPDE.

Usage

```
SaddlepointTest(object, FUN, ..., thrd)
```

Arguments

| | |
|--------|--|
| object | fitted model object of class "LSMLE" or "LMDPDE" |
| FUN | the function representing the null hypothesis to be tested |
| ... | further arguments to be passed |
| thrd | number (integer) of threads to speed up the process. If missing, the value is autodetected by the available number of multi-core processor |

References

Lo, S. N., Ronchetti, E. Robust and accurate inference for generalized linear models. *Journal of Multivariate Analysis*, 100, 2126–2136 (2009)

Examples

```
fit=robustbetareg(I(food/income)~income+persons|1,data=FoodExpenditure,alpha=0.08)
h0=function(theta,B){c(theta[1],B)}#H0: income=persons=0
SaddlepointTest(fit,h0,B=c(0,0))#H0: income=persons=0
```

WaldTypeTest

Robust Wald-type Tests

Description

Wald-type tests for both simple and composite hypothesis for independent but non-homogeneous observations, based on LSMLE and LMDPDE.

Usage

```
WaldTypeTest(object, FUN, ...)
```

Arguments

| | |
|--------|--|
| object | fitted model object of class "LSMLE" or "LMDPDE" |
| FUN | the function representing the null hypothesis to be tested |
| ... | Further arguments to be passed |

References

Basu, A., Ghosh, A., Martin, N. et al. Robust Wald-type tests for non-homogeneous observations based on the minimum density power divergence estimator. *Metrika* 81, 493–522 (2018)

Examples

```
fit=robustbetareg(I(food/income)~income+persons|1,data=FoodExpenditure,alpha=0.08)
h0=function(theta,B){theta[2:3]-B}#H0: income=persons=0
WaldTypeTest(fit,h0,B=c(0,0))#Testing income=persons=0
```


Index

`degbeta`, [2](#)

`LMDPDE.fit (robustbetareg)`, [5](#)

`LSMLE.fit (robustbetareg)`, [5](#)

`plot`, [2](#)

`plotenvelope`, [3](#)

`regbeta (degbeta)`, [2](#)

`residuals`, [4](#)

`robustbetareg`, [3](#), [5](#)

`robustbetareg.`, [7](#)

`robustbetareg.control`, [3](#), [5](#), [7](#)

`SaddlepointTest`, [7](#)

`WaldTypeTest`, [8](#)