Package ‘robustbetareg’

September 18, 2022

Version 0.1.0

Title Robust of Beta Regression Model

Author Yuri Maluf [aut,cre], Silvia Ferrari [ths], Francisco Queiroz [cre], Terezinha Ribeiro [aut,cre]

Maintainer
Yuri Maluf <yurimaluf@gmail.com>, Francisco Queiroz <ffelipeq@outlook.com> and Terezinha Ribeiro <terezinha.ribeiro@unb.br>

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

License GPL-3

NeedsCompilation no

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

R topics documented:

degbeta *The EGB of the second type*

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

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

Details

See Kerman and McDonald (2015) for more details.

Value

Return the value of density function or a random sample.

References

Kerman, S. McDonald, J. B. Skewness-kurtosis bounds for EGB1, EGB2, and special cases. Communications in Statistics - Theory and Methods, 44:3857-3864 (2015).

HIC *Health Insurance Coverage (HIC)*

Description

A dataset containing the proportion of the population from several counties of the state of São Paulo in Brazil who have health insurance.

Usage

```
data("HIC",package = "robustbetareg")
```

Format

A data frame with 80 rows and 3 variables:

Urbanization percentage of the total population who lives in the city's urban zone

GDP_percapita per capita gross domestic product

Percent_HIC the health insurance coverage index ...

`plot.robustbetareg` *Interactive plots for diagnostic of robust betareg models*

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 'robustbetareg'
plot(object, ask = TRUE, ...)
```

Arguments

`object`	fitted model object of class `robustbetareg`.
`ask`	logical. If TRUE the user is asked before each plot.
`...`	other parameters to be passed through to plotting functions.

Examples

```
## Not run:
data("HIC", package="robustbetareg")
fit=robustbetareg(Percent_HIC~Urbanization+GDP_percapita|GDP_percapita,data=HIC,alpha=0.06)
plot(fit)
## End(Not run)
```

plotenvelope *Simulated Envelope of Residuals*

Description

Plot a simulated envelope of beta residuals from `robustbetareg` object class.

Usage

```
plotenvelope(object,type=c("sweighted2","pearson","weighted","sweighted",
"sweighted.gamma","sweighted2.gamma","combined","combined.projection","sweighted3"),
conf=0.95,n.sim=100,PrgBar=T,control=robustbetareg.control(...), ...)
```

Arguments

`object`	Fitted model object of class "robustbetareg" (see `robustbetareg`).
`conf`	the confidence level of the envelopes required. The default is to find 95 confidence envelopes.
`n.sim`	the number of simulation sample. Deafault n.sim=100.
`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

```
## Not run: data("HIC", package="robustbetareg")
fit=robustbetareg(Percent_HIC~Urbanization+GDP_percapita|1,data=HIC,alpha=0.06)
plotenvelope(fit,n.sim=500)
## End(Not run)
```

predict *Predict*

Description

Extract various types of predictions from beta regression models: either on the scale of responses in (0, 1) or the scale of the linear predictor, from robustbetareg object class.

Usage

```
predict(
  object,
  newdata = NULL,
  type = c("response", "link", "precision", "variance", "quantile"),
  at = 0.5
)
```

Arguments

`object`	Fitted model object of class "robustbetareg" (see `robustbetareg`).
`newdata`	optionally, a data frame in which to look for variables with which to predict. If omitted, the original observations are used.
`type`	character indicating type of predictions: fitted means of response ("response"), corresponding linear predictor ("link"), fitted precision parameter phi ("precision"), fitted variances of response ("variance"), or fitted quantile(s) of the response distribution ("quantile").
`at`	numeric vector indicating the level(s) at which quantiles should be predicted (only if type = "quantile"), defaulting to the median at = 0.5.

Value

Return a vector of predicted values.

Examples

```
## Not run:
data("HIC", package="robustbetareg")
fit=robustbetareg(Percent_HIC~Urbanization+GDP_percapita|1,data=HIC,alpha=0.04)
cbind(predict(fit,type="response"),predict(fit,type="quantile",at=c(0.25,0.5,0.75)))
## End(Not run)
```

---

residuals.robustbetareg

Residuals Method for robustbetareg Object

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 'robustbetareg'
residuals(
  object,
  type = c("sweighted2", "pearson", "weighted", "sweighted", "sweighted.gamma",
    "sweighted2.gamma", "combined", "combined.projection", "sweighted3")
)
```

Arguments

`object`	fitted model object of class "`robustbetareg`".
`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 "`sweighted`", and Equation (8) for "`sweighted2`". For the last four residuals the definitions are described in Espinheira et al. (2017): Last equation of Equation (7) and Equation (10) for "`sweighted.gamma`" and "`sweighted2.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

```
## Not run:
data("HIC", package="robustbetareg")
fit=robustbetareg(Percent_HIC~Urbanization+GDP_percapita|1,data=HIC,alpha=0.04)
residuals(fit,type="sweighted2")

## End(Not run)
```

`robustbetareg` *Robust Beta Regression*

Description

Fit robust beta regression models for rates and proportions via LSMLE, LMDPDE, SMLE and MDPDE 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", "SMLE", "MDPDE"),
link=c("logit", "probit", "cloglog", "cauchit", "log", "loglog"),
link.phi=NULL, control=robustbetareg.control(...),model = TRUE,y = TRUE)
```

Arguments

`formula`	symbolic description of the model (of type y ~ x or y ~ x | z).
`data`	arguments controlling formula.
`alpha`	the tuning value within (0,1), for robust estimation. When alpha is equal to zero (α=0) is equivalent of MLE. If this argument is supressed, the tuning parameter should be selected automaticaly through the data-driven algorithm proposed by Ribeiro and Ferrari (2022).
`type`	character specification of the type of estimator. Currently, LSMLE (default), LMDPDE, SMLE and MDPDE.
`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 ~ 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, LMDPDE.fit, SMLE.fit and MDPDE.fit `y` must be a numeric response vector within (0,1).

Value

robustbetareg returns an object of class "`robustbetareg`" 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 `optim` 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 `optim` 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,

`data` the dataset used,

`method` estimator type.

References

Ribeiro, K. A. T. Ferrari, S. L. P. Robust estimation in beta regression via maximum Lq-likelihood. Statistical Papers (2022).

Ghosh, A. Robust inference under the beta regression model with application to health care studies. Statistical Methods in Medical Research, 28:271-888 (2019).

`robustbetareg.control` *Control Parameter for Robust Beta Regression*

Description

Several parameters that control fitting of robust beta regression models using `robustbetareg`.

Usage

```
robustbetareg.control(start=NULL,alpha.optimal=TRUE,tolerance=1e-3,maxit=5000,L=0.02,M=3)
```

Arguments

`object`	fitted model object of class `"robustbetareg"`.
`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.
`M`	a integer parameter value of auto selecting algorithm of tuning parameter.
`...`	currently not used.

Details

The arguments `L` and `M` set the parameters of the data-driven algorithm for selecting the α tuning parameter of robust estimator. See Ribeiro and Ferrari (2022) for more details.

Value

A list with the arguments specified.

References

Ribeiro, K. A. T. Ferrari, S. L. P. Robust estimation in beta regression via maximum Lq -likelihood. Statistical Papers (2022).

`set.link` *Create a Link for robust beta regression model*

Description

This function provides several link functions for robust beta regression models

Usage

```
set.link(link.mu = "logit", link.phi = "log")
```

Arguments

`link.mu`	character; one of "logit"(default), "probit", "cauchit", "cloglog" and "loglog"
`link.phi`	character; one of "log"(default) "indentity" and "sqrt"

Value

A structure with link function, inverse link function, derivative $d\eta_\mu/d\mu$ and $d^2\eta_\mu/d\mu^2$ ($d\eta_\phi/d\phi$ and $d^2\eta_\phi/d\phi^2$), for both models: mean and precision, where $\eta_\mu = \mathbf{X}^\top\beta$ and $\eta_\phi = \mathbf{Z}^\top\gamma$.

Examples

```
## Not run:
links=set.link(link.mu="cauchit",link.phi="sqrt")
attributes(links)
## End(Not run)
```

`WaldTypeTest` *Robust Wald-type Tests*

Description

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

Usage

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

Arguments

`object`	fitted model object of class "`robustbetareg`"
`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)

Ribeiro, K. A. T. Ferrari, S. L. P. Robust estimation in beta regression via maximum Lq-likelihood. Statistical Papers (2022).

Examples

```
## Not run:
data("HIC", package="robustbetareg")
fit=robustbetareg(Percent_HIC~Urbanization+GDP_percapita|1,data=HIC,type="LMDPDE",alpha=0.06)
h0=function(theta,B){theta[2:3]-B}#Hiphothesis to be tested
WaldTypeTest(fit,h0,B=c(0,0))#Testing Urbanization=GDP_percapita=0
## End(Not run)
```

Index