Package ‘unitquantreg’

October 16, 2022

Title Parametric Quantile Regression Models for Bounded Data

Version 0.0.5

Maintainer André F. B. Menezes <andrefelipemaringa@gmail.com>

Description
A collection of parametric quantile regression models for bounded data. At present, the package provides 13 parametric quantile regression models. It can specify regression structure for any quantile and shape parameters. It also provides several S3 methods to extract information from fitted model, such as residual analysis, prediction, plotting, and model comparison. For more computation efficient the [dpqr]'s, likelihood, score and hessian functions are written in C++. For further details see Mazucheli et. al (2022) <doi:10.1016/j.cmpb.2022.106816>.

License Apache License (>= 2)

Encoding UTF-8

ByteCompile yes

LazyData true

LinkingTo Rcpp

Imports Rcpp, optimx, stats, quantreg, Formula, MASS, numDeriv

Suggests testthat (>= 3.0.0), rmarkdown, knitr, lmtest, ggplot2, covr

Depends R (>= 3.5.0)

RoxygenNote 7.2.1

NeedsCompilation yes

URL https://andrmenezes.github.io/unitquantreg/

BugReports https://github.com/AndrMenezes/unitquantreg/issues

VignetteBuilder knitr

Author André F. B. Menezes [aut, cre]
(<https://orcid.org/0000-0002-3320-9834>),
Josmar Mazucheli [aut] (<https://orcid.org/0000-0001-6740-0445>)

Repository CRAN

Date/Publication 2022-10-16 07:30:02 UTC

R topics documented:

`unitquantreg-package` *Overview of the* **unitquantreg** *package*

Description

The **unitquantreg** R package provides a collection of parametric quantile regression models for bounded data. At present, the package provides 13 parametric quantile regression models. It also enables several S3 methods to extract information from fitted model, such as residual analysis, prediction, plotting, and model comparison.

Author(s)

André F. B. Menezes <andrefelipemaringa@gmail.com>

Josmar Mazucheli <jmazucheli@gmail.com>

ashw *The arcsecant hyperbolic Weibull distribution*

Description

Density function, distribution function, quantile function and random number generation function for the arcsecant hyperbolic Weibull distribution reparametrized in terms of the τ-th quantile, $\tau \in (0, 1)$.

Usage

```
dashw(x, mu, theta, tau = 0.5, log = FALSE)

pashw(q, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

qashw(p, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

rashw(n, mu, theta, tau = 0.5)
```

Arguments

`x, q`	vector of positive quantiles.
`mu`	location parameter indicating the τ-th quantile, $\tau \in (0, 1)$.
`theta`	shape parameter.
`tau`	the parameter to specify which quantile use in the parametrization.
`log, log.p`	logical; If TRUE, probabilities p are given as log(p).
`lower.tail`	logical; If TRUE, (default), $P(X \leq x)$ are returned, otherwise $P(X > x)$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(y; \alpha, \theta) = \frac{\alpha\theta}{y\sqrt{1-y^2}} \text{arcsech}(y)^{\theta-1} \exp\left[-\alpha \text{arcsech}(y)^{\theta}\right]$$

Cumulative distribution function

$$F(y; \alpha, \theta) = \exp\left[-\alpha \text{arcsech}(y)^{\theta}\right]$$

Quantile function

$$Q(\tau; \alpha, \theta) = \text{sech}\left\{\left[-\alpha^{-1}\log(\tau)\right]^{\frac{1}{\theta}}\right\}$$

Reparameterization

$$\alpha = g^{-1}(\mu) = -\frac{\log(\tau)}{\text{arcsech}(\mu)^{\theta}}$$

where $\theta > 0$ is the shape parameter and $\text{arcsech}(y) = \log\left[\left(1 + \sqrt{1 - y^2}\right)/y\right]$.

Value

`dashw` gives the density, `pashw` gives the distribution function, `qashw` gives the quantile function and `rashw` generates random deviates.

Invalid arguments will return an error message.

Author(s)

Josmar Mazucheli

André F. B. Menezes

References

Korkmaz, M. C., Chesneau, C. and Korkmaz, Z. S., (2021). A new alternative quantile regression model for the bounded response with educational measurements applications of OECD countries. *Journal of Applied Statistics*, 1–25.

Examples

```
set.seed(6969)
x <- rashw(n = 1000, mu = 0.5, theta = 2.5, tau = 0.5)
R <- range(x)
S <- seq(from = R[1L], to = R[2L], by = 0.01)
hist(x, prob = TRUE, main = 'arcsecant hyperbolic Weibull')
lines(S, dashw(x = S, mu = 0.5, theta = 2.5, tau = 0.5), col = 2)
plot(ecdf(x))
lines(S, pashw(q = S, mu = 0.5, theta = 2.5, tau = 0.5), col = 2)
plot(quantile(x, probs = S), type = "l")
lines(qashw(p = S, mu = 0.5, theta = 2.5, tau = 0.5), col = 2)
```

---

bodyfat *Percentage of body fat data set*

Description

The body fat percentage of individuals assisted in a public hospital in Curitiba, Paraná, Brasil.

Usage

```
data(bodyfat, package = "unitquantreg")
```

Format

A `data.frame` with 298 observations and 9 columns:

- `arms`: Arms fat percentage.
- `legs`: Legs fat percentage.
- `body`: Body fat percentage.
- `android`: Android fat percentage.
- `gynecoid`: Ginecoid fat percentage.
- `bmi`: Body mass index - 24.71577.
- `age`: Age - 46.00.
- `sex`: Sex of individual. Female or male.
- `ipaq`: Factor variable indicating the sedentary, insufficiently active or active.

Author(s)

André F. B. Menezes

Josmar Mazucheli

Source

http://www.leg.ufpr.br/doku.php/publications:papercompanions:multquasibeta

References

Petterle, R. R., Bonat, W. H., Scarpin, C. T., Jonasson, T., and Borba, V. Z. C., (2020). Multivariate quasi-beta regression models for continuous bounded data. *The International Journal of Biostatistics*, 1–15, (preprint).

Mazucheli, J., Leiva, V., Alves, B., and Menezes A. F. B., (2021). A new quantile regression for modeling bounded data under a unit Birnbaum-Saunders distribution with applications in medicine and politics. *Symmetry*, **13**(4) 1–21.

---

hnp — *(Half-)Normal probability plots with simulated envelopes for* `unitquantreg` *objects*

Description

Produces a (half-)normal probability plot from a fitted model object of class `unitquantreg`.

Usage

```
hnp(object, ...)

## S3 method for class 'unitquantreg'
hnp(
  object,
  nsim = 99,
  halfnormal = TRUE,
  plot = TRUE,
  output = TRUE,
  level = 0.95,
  resid.type = c("quantile", "cox-snell"),
  ...
)
```

Arguments

`object`	fitted model object of class `unitquantreg`.
`...`	currently not used.
`nsim`	number of simulations used to compute envelope. Default is 99.
`halfnormal`	logical. If `TRUE`, a half-normal plot is produced. If `FALSE`, a normal plot is produced.
`plot`	Should the (half-)normal plot be plotted? Default is `TRUE`.
`output`	Should the output be returned? Default is `TRUE`.
`level`	confidence level of the simulated envelope. Default is 0.95.
`resid.type`	type of residuals to be used. The default is `quantile`. See `residuals.unitquantreg` for further details.

Details

Residuals plots with simulated envelope were proposed by Atkinson (1981) and can be construct as follows:

1. generate sample set of n independent observations from the estimated parameters of the fitted model;
2. fit the model using the generated sample, if `halfnormal` is `TRUE` compute the absolute values of the residuals and arrange them in order;
3. repeat steps (1) and (2) `nsim` number of times;
4. consider the n sets of the `nsim` ordered statistics of the residuals, then for each set compute the quantile `level`/2, the median and the quantile 1 - `level`/2;
5. plot these values and the ordered residuals of the original sample set versus the expected order statistics of a (half)-normal distribution, which is approximated as

$$G^{-1}\left(\frac{i+n-0.125}{2n+0.5}\right)$$

for half-normal plots, i.e., halfnormal=TRUE or

$$G^{-1}\left(\frac{i-0.375}{n+0.25}\right)$$

for normal plots, i.e., halfnormal=FALSE, where $G(\cdot)$ is the the cumulative distribution function of standard Normal distribution for quantile residuals or the standard exponential distribution for the cox-snell residuals.

According to Atkinson (1981), if the model was correctly specified then no more than level100% of the observations are expected to appear outside the envelope bands. Additionally, if a large proportion of the observations lies outside the envelope, thus one has evidence against the adequacy of the fitted model.

Value

A list with the following components in ordered (and absolute if halfnormal is TRUE) values:

obs	the observed residuals.
teo	the theoretical residuals.
lower	lower envelope band.
median	median envelope band.
upper	upper envelope band.
time_elapsed	time elapsed to fit the nsim models.

Author(s)

André F. B. Menezes

References

Atkinson, A. C., (1981). Two graphical displays for outlying and influential observations in regression. *Biometrika* **68**(1), 13–20.

See Also

residuals.unitquantreg

---

johnsonsb *The Johnson SB distribution*

Description

Density function, distribution function, quantile function and random number generation function for the Johnson SB distribution reparametrized in terms of the τ-th quantile, $\tau \in (0, 1)$.

Usage

```
djohnsonsb(x, mu, theta, tau = 0.5, log = FALSE)

pjohnsonsb(q, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

qjohnsonsb(p, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

rjohnsonsb(n, mu, theta, tau = 0.5)
```

Arguments

`x, q`	vector of positive quantiles.
`mu`	location parameter indicating the τ-th quantile, $\tau \in (0, 1)$.
`theta`	nonnegative shape parameter.
`tau`	the parameter to specify which quantile is to used.
`log, log.p`	logical; If TRUE, probabilities p are given as log(p).
`lower.tail`	logical; If TRUE, (default), $P(X \leq x)$ are returned, otherwise $P(X > x)$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(y \mid \alpha, \theta) = \frac{\theta}{\sqrt{2\pi}} \frac{1}{y(1-y)} \exp\left\{-\frac{1}{2}\left[\alpha + \theta \log\left(\frac{y}{1-y}\right)\right]^2\right\}$$

Cumulative distribution function

$$F(y \mid \alpha, \theta) = \Phi\left[\alpha + \theta \log\left(\frac{y}{1-y}\right)\right]$$

Quantile function

$$Q(\tau \mid \alpha, \theta) = \frac{\exp\left[\frac{\Phi^{-1}(\tau)-\alpha}{\theta}\right]}{1 + \exp\left[\frac{\Phi^{-1}(\tau)-\alpha}{\theta}\right]}$$

Reparameterization

$$\alpha = g^{-1}(\mu) = \Phi^{-1}(\tau) - \theta \log\left(\frac{\mu}{1-\mu}\right)$$

Value

`djohnsonsb` gives the density, `pjohnsonsb` gives the distribution function, `qjohnsonsb` gives the quantile function and `rjohnsonsb` generates random deviates.

Invalid arguments will return an error message.

Author(s)

Josmar Mazucheli

André F. B. Menezes

References

Lemonte, A. J. and Bazán, J. L., (2015). New class of Johnson SB distributions and its associated regression model for rates and proportions. *Biometrical Journal*, **58**(4), 727–746.

Johnson, N. L., (1949). Systems of frequency curves generated by methods of translation. *Biometrika*, **36**(1), 149–176.

Examples

```
set.seed(123)
x <- rjohnsonsb(n = 1000, mu = 0.5, theta = 1.5, tau = 0.5)
R <- range(x)
S <- seq(from = R[1], to = R[2], by =  0.01)
hist(x, prob = TRUE, main = 'Johnson SB')
lines(S, djohnsonsb(x = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
plot(ecdf(x))
lines(S, pjohnsonsb(q = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
plot(quantile(x, probs = S), type = "l")
lines(qjohnsonsb(p = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
```

---

kum *The Kumaraswamy distribution*

Description

Density function, distribution function, quantile function and random number generation for the Kumaraswamy distribution reparametrized in terms of the τ-th quantile, $\tau \in (0, 1)$.

Usage

```
dkum(x, mu, theta, tau = 0.5, log = FALSE)

pkum(q, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

qkum(p, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

rkum(n, mu, theta, tau = 0.5)
```

Arguments

`x, q`	vector of positive quantiles.
`mu`	location parameter indicating the τ-th quantile, $\tau \in (0, 1)$.
`theta`	nonnegative shape parameter.
`tau`	the parameter to specify which quantile is to used.
`log, log.p`	logical; If TRUE, probabilities p are given as log(p).
`lower.tail`	logical; If TRUE, (default), $P(X \leq x)$ are returned, otherwise $P(X > x)$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(y \mid \alpha, \theta) = \alpha\theta y^{\theta-1}(1 - y^{\theta})^{\alpha-1}$$

Cumulative distribution function

$$F(y \mid \alpha, \theta) = 1 - \left(1 - y^{\theta}\right)^{\alpha}$$

Quantile function

$$Q(\tau \mid \alpha, \theta) = \left[1 - (1 - \tau)^{\frac{1}{\alpha}}\right]^{\frac{1}{\theta}}$$

Reparameterization

$$\alpha = g^{-1}(\mu) = \frac{\log(1 - \tau)}{\log(1 - \mu^{\theta})}$$

Value

`dkum` gives the density, `pkum` gives the distribution function, `qkum` gives the quantile function and `rkum` generates random deviates.

Invalid arguments will return an error message.

Author(s)

Josmar Mazucheli

André F. B. Menezes

References

Kumaraswamy, P., (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, **46**(1), 79–88.

Jones, M. C., (2009). Kumaraswamy's distribution: A beta-type distribution with some tractability advantages. *Statistical Methodology*, **6**(1), 70-81.

Examples

```
set.seed(123)
x <- rkum(n = 1000, mu = 0.5, theta = 1.5, tau = 0.5)
R <- range(x)
S <- seq(from = R[1], to = R[2], by =  0.01)
hist(x, prob = TRUE, main = 'Kumaraswamy')
lines(S, dkum(x = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
plot(ecdf(x))
lines(S, pkum(q = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
plot(quantile(x, probs = S), type = "l")
lines(qkum(p = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
```

`leeg` *The Log-extended exponential-geometric distribution*

Description

Density function, distribution function, quantile function and random number generation function for the Log-extended exponential-geometric distribution reparametrized in terms of the τ-th quantile, $\tau \in (0, 1)$.

Usage

```
dleeg(x, mu, theta, tau = 0.5, log = FALSE)

pleeg(q, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

qleeg(p, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

rleeg(n, mu, theta, tau = 0.5)
```

Arguments

`x, q`	vector of positive quantiles.
`mu`	location parameter indicating the τ-th quantile, $\tau \in (0, 1)$.
`theta`	nonnegative shape parameter.
`tau`	the parameter to specify which quantile is to be used.
`log, log.p`	logical; If TRUE, probabilities p are given as log(p).
`lower.tail`	logical; If TRUE, (default), $P(X \leq x)$ are returned, otherwise $P(X > x)$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n)` > 1, the length is taken to be the number required.

Details

Probability density function

$$f(y \mid \alpha, \theta) = \frac{\theta\,(1+\alpha)\,y^{\theta-1}}{\left(1+\alpha y^{\theta}\right)^2}$$

Cumulative distribution function

$$F(y \mid \alpha, \theta) = \frac{(1+\alpha)\,y^{\theta}}{1+\alpha y^{\theta}}$$

Quantile function

$$Q(\tau \mid \alpha, \theta) = \left[\frac{\tau}{1+\alpha\,(1-\tau)}\right]^{\frac{1}{\theta}}$$

Reparameterization

$$\alpha = g^{-1}(\mu) = -\frac{1-\tau\mu^{\theta}}{(1-\tau)}$$

Value

`dleeg` gives the density, `pleeg` gives the distribution function, `qleeg` gives the quantile function and `rleeg` generates random deviates.

Invalid arguments will return an error message.

Author(s)

Josmar Mazucheli <jmazucheli@gmail.com>

André F. B. Menezes <andrefelipemaringa@gmail.com>

References

Jodrá, P. and Jiménez-Gamero, M. D., (2020). A quantile regression model for bounded responses based on the exponential-geometric distribution. *Revstat - Statistical Journal*, **18**(4), 415–436.

Examples

```
set.seed(123)
x <- rleeg(n = 1000, mu = 0.5, theta = 1.5, tau = 0.5)
R <- range(x)
S <- seq(from = R[1], to = R[2], by =  0.01)
hist(x, prob = TRUE, main = 'Log-extended exponential-geometric')
lines(S, dleeg(x = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
plot(ecdf(x))
lines(S, pleeg(q = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
plot(quantile(x, probs = S), type = "l")
lines(qleeg(p = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
```

likelihood_stats *Likelihood-based statistics of fit for* unitquantreg *objects.*

Description

Computes the likelihood-based statistics (Neg2LogLike, AIC, BIC and HQIC) from unitquantreg objects.

Usage

```
likelihood_stats(..., lt = NULL)

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

Arguments

...	unitquantreg objects separated by commas. Not use in print method.
lt	a list with one or more unitquantreg objects.
x	object of class likelihood_stats obtained from likelihood_stats function.

Details

Neg2LogLike: The log-likelihood is reported as

$$Neg2LogLike = -2\log(L)$$

AIC: The Akaike's information criterion (AIC) is defined as

$$AIC = -2\log(L) + 2p$$

BIC: The Schwarz Bayesian information criterion (BIC) is defined as

$$BIC = -2\log(L) + p\log(n)$$

HQIC: The Hannan and Quinn information criterion (HQIC) is defined as

$$HQIC = -2\log(L) + 2p\log[\log(n)]$$

where L is the likelihood function.

Value

A list with class "likelihood_stats" containing the following components:

call	the matched call.
stats	ordered matrix according AIC value containg the likelihood based statistics.

Author(s)

André F. B. Menezes

Josmar Mazucheli

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transaction on Automatic Control*, **19**(6), 716–723.

Hannan, E. J. and Quinn, B. G. (1979). The determination of the order of an autoregression. *Journal of the Royal Statistical Society, Series B*, **41**(2), 190–195.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, **6**(2), 461–464.

Examples

```
data(sim_bounded, package = "unitquantreg")
sim_bounded_curr <- sim_bounded[sim_bounded$family == "uweibull", ]

models <- c("uweibull", "kum", "ulogistic")
lt_fits <- lapply(models, function(fam) {
  unitquantreg(formula = y1 ~ x, tau = 0.5, data = sim_bounded_curr,
               family = fam)
})

ans <- likelihood_stats(lt = lt_fits)
ans
```

`loglike_unitquantreg` *Log-likelihood, score vector and hessian matrix.*

Description

Internal functions using in `unitquantreg.fit` to compute the negative log-likelihood function, the score vector and the hessian matrix using analytic expressions written in `C++`.

Usage

```
loglike_unitquantreg(par, tau, family, linkobj, linkobj.theta, X, Z, y)
```

Arguments

`par`	vector of regression model coefficients for μ and/or θ.
`tau`	quantile level, value between 0 and 1.
`family`	specify the distribution family name.
`linkobj, linkobj.theta`	a function, usually obtained from `make.link` for link function of μ and θ, respectively.

X	design matrix related to the μ parameter.
Z	design matrix related to the θ parameter.
y	vector of response variable.

---

methods-unitquantreg *Methods for* `unitquantreg` *and* `unitquantregs` *objects*

Description

Methods for extracting information from fitted regression models objects of class `unitquantreg` and `unitquantregs`.

Usage

```
## S3 method for class 'unitquantreg'
print(x, digits = max(4, getOption("digits") - 3), ...)

## S3 method for class 'unitquantreg'
summary(object, correlation = FALSE, ...)

## S3 method for class 'unitquantreg'
coef(object, type = c("full", "quantile", "shape"), ...)

## S3 method for class 'unitquantreg'
vcov(object, ...)

## S3 method for class 'unitquantreg'
logLik(object, ...)

## S3 method for class 'unitquantreg'
confint(object, parm, level = 0.95, ...)

## S3 method for class 'unitquantreg'
fitted(object, type = c("quantile", "shape", "full"), ...)

## S3 method for class 'unitquantreg'
terms(x, type = c("quantile", "shape"), ...)

## S3 method for class 'unitquantreg'
model.frame(formula, ...)

## S3 method for class 'unitquantreg'
model.matrix(object, type = c("quantile", "shape"), ...)

## S3 method for class 'unitquantreg'
update(object, formula., ..., evaluate = TRUE)
```

```
## S3 method for class 'unitquantregs'
print(x, digits = max(3, getOption("digits") - 3), ...)

## S3 method for class 'unitquantregs'
summary(object, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`digits`	minimal number of *significant* digits.
`...`	additional argument(s) for methods. Currently not used.
`object, x`	fitted model object of class `unitquantreg`.
`correlation`	logical; if `TRUE`, the correlation matrix of the estimated parameters is returned and printed. Default is `FALSE`.
`type`	character indicating type of fitted values to return.
`parm`	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
`level`	the confidence level required.
`formula`	an R formula.
`formula.`	Changes to the formula see `update.formula` for details.
`evaluate`	If true evaluate the new call else return the call.

Value

The `summary` method gives Wald tests for the regressions coefficients based on the observed Fisher information matrix. As usual the `summary` method returns a list with relevant model statistics and estimates, which can be printed using the `print` method.

The `coef`, `vcov`, `confint` and `fitted` methods can be use to extract, respectively, the estimated coefficients, the estimated covariance matrix, the Wald confidence intervals, and fitted values.

A `logLik` method is also provide, then the `AIC` function can be use to calculated the Akaike Information Criterion.

The generic methods `terms`, `model.frame`, `model.matrix`, `update` and are also provided.

Author(s)

André F. B. Menezes

Examples

```
data(sim_bounded, package = "unitquantreg")
sim_bounded_curr <- sim_bounded[sim_bounded$family == "uweibull", ]
fit_1 <- unitquantreg(formula = y1 ~ x + z + I(x^2) | z + x,
                      data = sim_bounded_curr,
                      family = "uweibull",
                      tau = 0.5, link.theta = "log")
```

```
fit_1
summary(fit_1)
vcov(fit_1)
coef(fit_1)
confint(fit_1)
terms(fit_1)
model.frame(fit_1)[1:5, ]
model.matrix(fit_1)[1:5, ]
update(fit_1, . ~ . -x)
update(fit_1, . ~ . -z)
update(fit_1, . ~ . -I(x^2))
update(fit_1, . ~ . | . -z)
update(fit_1, . ~ . | . -x)
```

`pairwise.vuong.test` *Pairwise vuong test*

Description

Calculate pairwise comparisons between fitted models performing vuong test for objects of class `unitquantreg`.

Usage

```
pairwise.vuong.test(
  ...,
  lt,
  p.adjust.method = p.adjust.methods,
  alternative = c("two.sided", "less", "greater")
)
```

Arguments

`...`	`unitquantreg` objects separated by commas.
`lt`	a list with one or more `unitquantreg` objects.
`p.adjust.method`	a character string specifying the method for multiple testing adjustment; almost always one of `p.adjust.methods`. Can be abbreviated.
`alternative`	indicates the alternative hypothesis and must be one of `"two.sided"` (default), `"less"`, or `"greater"`. Can be abbreviated.

Value

Object of class `"pairwise.htest"`

See Also

`vuong.test`, `p.adjust`

Examples

```
data(sim_bounded, package = "unitquantreg")
sim_bounded_curr <- sim_bounded[sim_bounded$family == "uweibull", ]

models <- c("uweibull", "kum", "ulogistic")
lt_fits <- lapply(models, function(fam) {
  unitquantreg(formula = y1 ~ x, tau = 0.5, data = sim_bounded_curr,
               family = fam)
})

ans <- pairwise.vuong.test(lt = lt_fits)
ans
```

`plot.unitquantreg` *Plot method for* `unitquantreg` *objects*

Description

Provide diagnostic plots to check model assumptions for fitted model of class `unitquantreg`.

Usage

```
## S3 method for class 'unitquantreg'
plot(
  x,
  which = 1L:4L,
  caption = c("Residuals vs. indices of obs.", "Residuals vs. linear predictor",
    "Working response vs. linear predictor", "Half-normal plot of residuals"),
  sub.caption = paste(deparse(x$call), collapse = "\n"),
  main = "",
  ask = prod(par("mfcol")) < length(which) && dev.interactive(),
  ...,
  add.smooth = getOption("add.smooth"),
  type = "quantile",
  nsim = 99L,
  level = 0.95
)
```

Arguments

`x`	fitted model object of class `unitquantreg`.
`which`	integer. if a subset of the plots is required, specify a subset of the numbers 1 to 4, see below for further details.
`caption`	character. Captions to appear above the plots.
`sub.caption`	character. Common title-above figures if there are multiple.

`main`	character. Title to each plot in addition to the above caption.
`ask`	logical. If `TRUE`, the user is asked before each plot.
`...`	other parameters to be passed through to plotting functions.
`add.smooth`	logical. Indicates if a smoother should be added to most plots
`type`	character. Indicates type of residual to be used, see `residuals.unitquantreg`.
`nsim`	integer. Number of simulations in half-normal plots, see `hnp.unitquantreg`.
`level`	numeric. Confidence level of the simulated envelope, see `hnp.unitquantreg`.

Details

The `plot` method for `unitquantreg` objects produces four types of diagnostic plot.

The `which` argument can be used to select a subset of currently four supported plot, which are: Residuals versus indices of observations (`which = 1`); Residuals versus linear predictor (`which = 2`); Working response versus linear predictor (`which = 3`) to check possible misspecification of link function; Half-normal plot of residuals (`which = 4`) to check distribution assumption.

Value

No return value, called for side effects.

Author(s)

André F. B. Menezes

References

Dunn, P. K. and Smyth, G. K. (2018) Generalized Linear Models With Examples in R, Springer, New York.

See Also

`residuals.unitquantreg`, `hnp.unitquantreg`, `unitquantreg`.

`plot.unitquantregs` *Plot method for* `unitquantregs` *objects*

Description

Provide two type of plots for `unitquantregs` objects.

Usage

```
## S3 method for class 'unitquantregs'
plot(
  x,
  which = c("coef", "conddist"),
  output_df = FALSE,
  parm = NULL,
  level = 0.95,
  mean_effect = FALSE,
  mfrow = NULL,
  mar = NULL,
  ylim = NULL,
  main = NULL,
  col = gray(c(0, 0.75)),
  border = NULL,
  cex = 1,
  pch = 20,
  type = "b",
  xlab = bquote("Quantile level (" * tau * ")"),
  ylab = "Estimate effect",
  dist_type = c("density", "cdf"),
  at_avg = TRUE,
  at_obs = NULL,
  legend_position = "topleft",
  ...
)
```

Arguments

`x`	fitted model object of class `unitquantregs`.
`which`	character. Indicate the type of plot. Currently supported are `"coef"` which provide the estimated coefficients for several quantiles and `"conddist"` which provide the conditional distribution (cdf or pdf) at specific values of covariates.
`output_df`	logical. Should `data.frame` used to plot be returned?
`parm`	a specification of which parameters are to be plotted, either a vector of numbers or a vector of names. By default, all parameters are considered.
`level`	level of significance for the confidence interval of parameters.
`mean_effect`	logical. Should a line for the mean effect coefficients be added?
`mfrow, mar, ylim, main, col, border, cex, pch, type, xlab, ylab`	graphical parameters.
`dist_type`	character. Which conditional distribution should be plotted? The options are `"density"` or `"cdf"`.
`at_avg`	logical. Should consider the conditional distribution at average values of covariates?
`at_obs`	list. List with name and values for each covariate.

legend_position
: character. The legend position argument used in legend function.

...
: other parameters to be passed through to plotting functions.

Details

The plot method for unitquantregs objects is inspired in PROC QUANTREG of SAS/STAT. This plot method provide two type of visualizations.

If which = "coef" plot the estimated coefficients for several quantiles.

If which = "conddist" plot the conditional distribution at specific values of covariates. The conditional distribution could be the cumulative distribution function if dist_type = "cdf" or the probability density function if dist_type = "pdf".

Value

If output_df = TRUE then returns a data.frame used to plot. Otherwise, no return value, called for side effects.

Author(s)

André F. B. Menezes

See Also

plot.unitquantreg.

---

predict.unitquantreg *Prediction method for* unitquantreg *class*

Description

Extract various types of predictions from unit quantile regression models.

Usage

```
## S3 method for class 'unitquantreg'
predict(
  object,
  newdata,
  type = c("link", "quantile", "shape", "terms"),
  interval = c("none", "confidence"),
  level = 0.95,
  se.fit = FALSE,
  ...
)
```

Arguments

`object`	fitted model object of class `unitquantreg`.
`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. The options are `link`, `quantile`, `shape` and `terms`.
`interval`	type of interval desired. The options are `none` and `confidence`. The `"terms"` option returns a matrix giving the fitted values of each term in the model formula on the linear predictor scale.
`level`	coverage probability for the confidence intervals. Default is `0.95`.
`se.fit`	logical. If `TRUE` return the asymptotic standard errors.
`...`	currently not used.

Value

If `se.fit = FALSE` then returns a `data.frame` with predict values and confidence interval if `interval = TRUE`.

If `se.fit = TRUE` returns a list with components:

`fit`	Predictions, as for `se.fit = FALSE`.
`se.fit`	Estimated standard errors.

For `type = "terms"` the output is a `data.frame` with a columns per term.

Author(s)

André F. B. Menezes

---

residuals.unitquantreg

Residuals method for `unitquantreg` *objects*

Description

Extract various types of residuals from unit quantile regression models.

Usage

```
## S3 method for class 'unitquantreg'
residuals(object, type = c("quantile", "cox-snell", "working", "partial"), ...)
```

Arguments

`object`	fitted model object of class `unitquantreg`.
`type`	character indicating type of residuals. The options are `"quantile"`, `"cox-snell"`, `"working"` and `"partial"`.
`...`	currently not used.

Details

The residuals method can compute quantile and Cox-Snell residuals. These residuals are defined, respectively, by

$$r_Q = \Phi^{-1}\left[F(y_i \mid \widehat{\mu}_i, \widehat{\theta}_i)\right]$$

and

$$r_{CS} = -\log\left[1 - F(y_i \mid \widehat{\mu}_i, \widehat{\theta}_i)\right]$$

where $\widehat{\mu}_i$ and $\widehat{\theta}_i$ are the fitted values of parameters μ and θ, $F(\cdot \mid \cdot, \cdot)$ is the cumulative distribution function (c.d.f.) and $\Phi(\cdot)$ is the c.d.f. of standard Normal distribution.

Apart from the variability due the estimates of parameters,if the fitted regression model is correctly specified then the quantile residuals, r_Q, follow a standard Normal distribution and the Cox-Snell residuals, r_{CS}, follow a standard exponential distribution.

Value

Numeric vector of residuals extract from an object of class unitquantreg.

Author(s)

André F. B. Menezes

References

Cox, D. R. and Snell E. J., (1968). A general definition of residuals. *Journal of the Royal Statistical Society - Series B*, **30**(2), 248–265.

Dunn, P. K. and Smyth, G. K., (1996). Randomized quantile residuals. *Journal of Computational and Graphical Statistics*, **5**(3), 236–244.

`sim_bounded` *Simulated data set*

Description

This data set was simulated from all families of distributions available in unitquantreg package considering the median, i.e., $\tau = 0.5$.

Usage

```
data(sim_bounded, package = "unitquantreg")
```

Format

data.frame with 1300 observations and 5 columns:

- y1: simulated response variable with constant shape parameter, $\theta = 2$.
- y2: simulated response variable with regression structure in the shape parameter, $\theta_i = \exp(\zeta_i)$, where $\zeta_i = \mathbf{z}_i^\top \boldsymbol{\gamma}$.
- x: covariate related to μ_i, i.e., the median.
- z: covariate related to θ_i, i.e., the shape parameter.
- family: string indicating the family of distribution.

Details

There are two response variable, namely y1 and y2. The former was simulated considering a regression structure for μ and one covariate simulated from a standard uniform distribution, where the true vector of coefficients for μ is $\boldsymbol{\beta} = (1, 2)$ and $\theta = 2$. The latter was simulated assuming a regression structure for both μ and θ (shape parameter) and only one independent covariates simulated from two standard uniform distributions. The true vectors of coefficients for μ and θ are $\boldsymbol{\beta} = (1, 2)$ and $\boldsymbol{\gamma} = (-1, 1)$, respectively.

Author(s)

André F. B. Menezes

---

ubs *The unit-Birnbaum-Saunders distribution*

Description

Density function, distribution function, quantile function and random number generation function for the unit-Birnbaum-Saunders distribution reparametrized in terms of the τ-th quantile, $\tau \in (0, 1)$.

Usage

```
dubs(x, mu, theta, tau = 0.5, log = FALSE)

pubs(q, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

qubs(p, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

rubs(n, mu, theta, tau = 0.5)
```

Arguments

`x, q`	vector of positive quantiles.
`mu`	location parameter indicating the τ-th quantile, $\tau \in (0, 1)$.
`theta`	nonnegative shape parameter.
`tau`	the parameter to specify which quantile is to be used.
`log, log.p`	logical; If TRUE, probabilities p are given as log(p).
`lower.tail`	logical; If TRUE, (default), $P(X \leq x)$ are returned, otherwise $P(X > x)$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(y \mid \alpha, \theta) = \frac{1}{2y\alpha\theta\sqrt{2\pi}} \left[\left(-\frac{\alpha}{\log(y)}\right)^{\frac{1}{2}} + \left(-\frac{\alpha}{\log(y)}\right)^{\frac{3}{2}} \right] \exp\left[\frac{1}{2\theta^2}\left(2 + \frac{\log(y)}{\alpha} + \frac{\alpha}{\log(y)}\right)\right]$$

Cumulative distribution function

$$F(y \mid \alpha, \theta) = 1 - \Phi\left\{\frac{1}{\theta}\left[\left(-\frac{\log(y)}{\alpha}\right)^{\frac{1}{2}} - \left(-\frac{\alpha}{\log(y)}\right)^{\frac{1}{2}}\right]\right\}$$

Quantile function

$$Q(\tau \mid \alpha, \theta) = \exp\left\{-\frac{2\alpha}{2 + \left[\theta\Phi^{-1}(1-\tau)\right]^2 - \theta\Phi^{-1}(1-\tau)\sqrt{4 + \left[\theta\Phi^{-1}(1-\tau)\right]^2}}\right\}$$

Reparameterization

$$\alpha = g^{-1}(\mu) = \log(\mu)\, g(\theta, \tau)$$

where $g(\theta, \tau) = -\frac{1}{2}\left\{2 + \left[\theta\Phi^{-1}(1-\tau)\right]^2 - \theta\Phi^{-1}(1-\tau)\sqrt{4 + \theta\Phi^{-1}(1-\tau)}\right\}$.

Value

`dubs` gives the density, `pubs` gives the distribution function, `qubs` gives the quantile function and `rubs` generates random deviates.

Invalid arguments will return an error message.

Author(s)

Josmar Mazucheli <`jmazucheli@gmail.com`>

André F. B. Menezes <`andrefelipemaringa@gmail.com`>

References

Birnbaum, Z. W. and Saunders, S. C., (1969). A new family of life distributions. *Journal of Applied Probability*, **6**(2), 637–652. Mazucheli, J., Menezes, A. F. B. and Dey, S., (2018). The unit-Birnbaum-Saunders distribution with applications. *Chilean Journal of Statistics*, **9**(1), 47–57.

Mazucheli, J., Alves, B. and Menezes, A. F. B., (2021). A new quantile regression for modeling bounded data under a unit Birnbaum-Saunders distribution with applications. *Simmetry*, (), 1–28.

Examples

```
set.seed(123)
x <- rubs(n = 1000, mu = 0.5, theta = 1.5, tau = 0.5)
R <- range(x)
S <- seq(from = R[1], to = R[2], by =  0.01)
hist(x, prob = TRUE, main = 'unit-Birnbaum-Saunders')
lines(S, dubs(x = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
plot(ecdf(x))
lines(S, pubs(q = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
plot(quantile(x, probs = S), type = "l")
lines(qubs(p = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
```

---

uburrxii *The unit-Burr-XII distribution*

Description

Density function, distribution function, quantile function and random number generation function for the unit-Burr-XII distribution reparametrized in terms of the τ-th quantile, $\tau \in (0, 1)$.

Usage

```
duburrxii(x, mu, theta, tau = 0.5, log = FALSE)

puburrxii(q, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

quburrxii(p, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

ruburrxii(n, mu, theta, tau = 0.5)
```

Arguments

`x, q`	vector of positive quantiles.
`mu`	location parameter indicating the τ-th quantile, $\tau \in (0, 1)$.
`theta`	nonnegative shape parameter.
`tau`	the parameter to specify which quantile is to used.
`log, log.p`	logical; If TRUE, probabilities p are given as log(p).

lower.tail	logical; If TRUE, (default), $P(X \leq x)$ are returned, otherwise $P(X > x)$.
p	vector of probabilities.
n	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(y \mid \alpha, \theta) = \frac{\alpha\theta}{y} \left[-\log(y)\right]^{\theta-1} \left\{1 + \left[-\log(y)\right]^{\theta}\right\}^{-\alpha-1}$$

Cumulative distribution function

$$F(y \mid \alpha, \theta) = \left\{1 + \left[-\log(y)\right]^{\theta}\right\}^{-\alpha}$$

Quantile function

$$Q(\tau \mid \alpha, \theta) = \exp\left[-\left(\tau^{-\frac{1}{\alpha}} - 1\right)^{\frac{1}{\theta}}\right]$$

Reparameterization

$$\alpha = g^{-1}(\mu) = \frac{\log\left(\tau^{-1}\right)}{\log\left[1 + \log\left(\frac{1}{\mu}\right)^{\theta}\right]}$$

Value

`duburrxii` gives the density, `puburrxii` gives the distribution function, `quburrxii` gives the quantile function and `ruburrxii` generates random deviates.

Invalid arguments will return an error message.

Author(s)

Josmar Mazucheli <jmazucheli@gmail.com>

André F. B. Menezes <andrefelipemaringa@gmail.com>

References

Korkmaz M. C. and Chesneau, C., (2021). On the unit Burr-XII distribution with the quantile regression modeling and applications. *Computational and Applied Mathematics*, **40**(29), 1–26.

Examples

```
set.seed(123)
x <- ruburrxii(n = 1000, mu = 0.5, theta = 1.5, tau = 0.5)
R <- range(x)
S <- seq(from = R[1], to = R[2], by =  0.01)
hist(x, prob = TRUE, main = 'unit-Burr-XII')
lines(S, duburrxii(x = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
```

```
plot(ecdf(x))
lines(S, puburrxii(q = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
plot(quantile(x, probs = S), type = "l")
lines(quburrxii(p = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
```

uchen *The unit-Chen distribution*

Description

Density function, distribution function, quantile function and random number generation function for the unit-Chen distribution reparametrized in terms of the τ-th quantile, $\tau \in (0, 1)$.

Usage

```
duchen(x, mu, theta, tau = 0.5, log = FALSE)

puchen(q, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

quchen(p, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

ruchen(n, mu, theta, tau = 0.5)
```

Arguments

x, q	vector of positive quantiles.
mu	location parameter indicating the τ-th quantile, $\tau \in (0, 1)$.
theta	nonnegative shape parameter.
tau	the parameter to specify which quantile is to be used.
log, log.p	logical; If TRUE, probabilities p are given as log(p).
lower.tail	logical; If TRUE, (default), $P(X \leq x)$ are returned, otherwise $P(X > x)$.
p	vector of probabilities.
n	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(y \mid \alpha, \theta) = \frac{\alpha\theta}{y} \left[-\log(y)\right]^{\theta-1} \exp\left\{\left[-\log\left(y\right)\right]^{\theta}\right\} \exp\left\{\alpha\left\{1 - \exp\left[\left(-\log(y)\right)^{\theta}\right]\right\}\right\}$$

Cumulative distribution function

$$F(y \mid \alpha, \theta) = \exp\left\{\alpha\left\{1 - \exp\left[\left(-\log(y)\right)^{\theta}\right]\right\}\right\}$$

Quantile function

$$Q(\tau \mid \alpha, \theta) = \exp\left\{-\left[\log\left(1 - \frac{\log(\tau)}{\alpha}\right)\right]^{\frac{1}{\theta}}\right\}$$

Reparameterization

$$\alpha = g^{-1}(\mu) = \frac{\log(\tau)}{1 - \exp\left[(-\log(\mu))^{\theta}\right]}$$

Value

duchen gives the density, puchen gives the distribution function, quchen gives the quantile function and ruchen generates random deviates.

Invalid arguments will return an error message.

Author(s)

Josmar Mazucheli <jmazucheli@gmail.com>

André F. B. Menezes <andrefelipemaringa@gmail.com>

References

Korkmaz, M. C., Emrah, A., Chesneau, C. and Yousof, H. M., (2020). On the unit-Chen distribution with associated quantile regression and applications. *Journal of Applied Statistics*, **44**(1) 1–22.

Examples

```
set.seed(123)
x <- ruchen(n = 1000, mu = 0.5, theta = 1.5, tau = 0.5)
R <- range(x)
S <- seq(from = R[1], to = R[2], by =  0.01)
hist(x, prob = TRUE, main = 'unit-Chen')
lines(S, duchen(x = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
plot(ecdf(x))
lines(S, puchen(q = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
plot(quantile(x, probs = S), type = "l")
lines(quchen(p = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
```

---

ughne *The unit-Half-Normal-E distribution*

Description

Density function, distribution function, quantile function and random number generation function for the unit-Half-Normal-E distribution reparametrized in terms of the τ-th quantile, $\tau \in (0, 1)$.

Usage

```
dughne(x, mu, theta, tau = 0.5, log = FALSE)

pughne(q, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

qughne(p, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

rughne(n, mu, theta, tau = 0.5)
```

Arguments

`x, q`	vector of positive quantiles.
`mu`	location parameter indicating the τ-th quantile, $\tau \in (0, 1)$.
`theta`	nonnegative shape parameter.
`tau`	the parameter to specify which quantile is to be used.
`log, log.p`	logical; If TRUE, probabilities p are given as log(p).
`lower.tail`	logical; If TRUE, (default), $P(X \leq x)$ are returned, otherwise $P(X > x)$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(y \mid \alpha, \theta) = \sqrt{\frac{2}{\pi}} \frac{\theta}{y\left[-\log(y)\right]} \left(-\frac{\log(y)}{\alpha}\right)^{\theta} \exp\left\{-\frac{1}{2}\left[-\frac{\log(y)}{\alpha}\right]^{2\theta}\right\}$$

Cumulative distribution function

$$F(y \mid \alpha, \theta) = 2\Phi\left[-\left(-\frac{\log(y)}{\alpha}\right)^{\theta}\right]$$

Quantile function

$$Q(\tau \mid \alpha, \theta) = \exp\left\{-\alpha\left[-\Phi^{-1}\left(\frac{\tau}{2}\right)\right]^{\frac{1}{\theta}}\right\}$$

Reparameterization

$$\alpha = g^{-1}(\mu) = -\log(\mu)\left[-\Phi^{-1}\left(\frac{\tau}{2}\right)\right]^{-\frac{1}{\theta}}$$

Value

`dughne` gives the density, `pughne` gives the distribution function, `qughne` gives the quantile function and `rughne` generates random deviates.

Invalid arguments will return an error message.

Author(s)

Josmar Mazucheli <jmazucheli@gmail.com>

André F. B. Menezes <andrefelipemaringa@gmail.com>

References

Korkmaz, M. C., (2020). The unit generalized half normal distribution: A new bounded distribution with inference and application. *University Politehnica of Bucharest Scientific*, **82**(2), 133–140.

Examples

```
set.seed(123)
x <- rughne(n = 1000, mu = 0.5, theta = 2, tau = 0.5)
R <- range(x)
S <- seq(from = R[1], to = R[2], by =  0.01)
hist(x, prob = TRUE, main = 'unit-Half-Normal-E')
lines(S, dughne(x = S, mu = 0.5, theta = 2, tau = 0.5), col = 2)
plot(ecdf(x))
lines(S, pughne(q = S, mu = 0.5, theta = 2, tau = 0.5), col = 2)
plot(quantile(x, probs = S), type = "l")
lines(qughne(p = S, mu = 0.5, theta = 2, tau = 0.5), col = 2)
```

---

ughnx — *The unit-Half-Normal-X distribution*

Description

Density function, distribution function, quantile function and random number generation function for the unit-Half-Normal-X distribution reparametrized in terms of the τ-th quantile, $\tau \in (0, 1)$.

Usage

```
dughnx(x, mu, theta, tau = 0.5, log = FALSE)

pughnx(q, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

qughnx(p, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

rughnx(n, mu, theta, tau = 0.5)
```

Arguments

`x, q`	vector of positive quantiles.
`mu`	location parameter indicating the τ-th quantile, $\tau \in (0, 1)$.
`theta`	nonnegative shape parameter.
`tau`	the parameter to specify which quantile is to be used.

log, log.p	logical; If TRUE, probabilities p are given as log(p).
lower.tail	logical; If TRUE, (default), $P(X \leq x)$ are returned, otherwise $P(X > x)$.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.

Details

Probability density function

$$f(y \mid \alpha, \theta) = \sqrt{\frac{2}{\pi}} \frac{\theta}{y\,(1-y)} \left(\frac{y}{\alpha\,(1-y)}\right)^{\theta} \exp\left\{-\frac{1}{2}\left[\frac{y}{\alpha\,(1-y)}\right]^{2\theta}\right\}$$

Cumulative density function

$$F(y \mid \alpha, \theta) = 2\Phi\left[\left(\frac{y}{\alpha\,(1-y)}\right)^{\theta}\right] - 1$$

Quantile Function

$$Q(\tau \mid \alpha) = \frac{\alpha\left[\Phi^{-1}\left(\frac{\tau+1}{2}\right)\right]^{\frac{1}{\theta}}}{1 + \alpha\left[\Phi^{-1}\left(\frac{\tau+1}{2}\right)\right]^{\frac{1}{\theta}}}$$

Reparametrization

$$\alpha = g^{-1}(\mu) = \frac{\mu}{(1-\mu)\left[\Phi^{-1}\left(\frac{\tau+1}{2}\right)\right]^{\frac{1}{\theta}}}$$

Value

dughnx gives the density, pughnx gives the distribution function, qughnx gives the quantile function and rughnx generates random deviates.

Invalid arguments will return an error message.

Author(s)

Josmar Mazucheli <jmazucheli@gmail.com>

André F. B. Menezes <andrefelipemaringa@gmail.com>

References

Bakouch, H. S., Nik, A. S., Asgharzadeh, A. and Salinas, H. S., (2021). A flexible probability model for proportion data: Unit-Half-Normal distribution. *Communications in Statistics: CaseStudies, Data Analysis and Applications*, **0**(0), 1–18.

Examples

```
set.seed(123)
x <- rughnx(n = 1000, mu = 0.5, theta = 2, tau = 0.5)
R <- range(x)
S <- seq(from = R[1], to = R[2], by =  0.01)
hist(x, prob = TRUE, main = 'unit-Half-Normal-X')
lines(S, dughnx(x = S, mu = 0.5, theta = 2, tau = 0.5), col = 2)
plot(ecdf(x))
lines(S, pughnx(q = S, mu = 0.5, theta = 2, tau = 0.5), col = 2)
plot(quantile(x, probs = S), type = "l")
lines(qughnx(p = S, mu = 0.5, theta = 2, tau = 0.5), col = 2)
```

---

ugompertz — *The unit-Gompertz distribution*

Description

Density function, distribution function, quantile function and random number deviates for the unit-Gompertz distribution reparametrized in terms of the τ-th quantile, $\tau \in (0, 1)$.

Usage

```
dugompertz(x, mu, theta, tau = 0.5, log = FALSE)

pugompertz(q, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

qugompertz(p, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

rugompertz(n, mu, theta, tau = 0.5)
```

Arguments

`x, q`	vector of positive quantiles.
`mu`	location parameter indicating the τ-th quantile, $\tau \in (0, 1)$.
`theta`	nonnegative shape parameter.
`tau`	the parameter to specify which quantile is to be used.
`log, log.p`	logical; If TRUE, probabilities p are given as log(p).
`lower.tail`	logical; If TRUE, (default), $P(X \leq x)$ are returned, otherwise $P(X > x)$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n)` > 1, the length is taken to be the number required.

Details

Probability density function

$$f(y \mid \alpha, \theta) = \frac{\alpha\theta}{x} \exp\left\{\alpha - \theta \log(y) - \alpha \exp\left[-\theta \log(y)\right]\right\}$$

Cumulative density function

$$F(y \mid \alpha, \theta) = \exp\left[\alpha\left(1 - y^{\theta}\right)\right]$$

Quantile Function

$$Q(\tau \mid \alpha, \theta) = \left[\frac{\alpha - \log(\tau)}{\alpha}\right]^{-\frac{1}{\theta}}$$

Reparameterization

$$\alpha = g^{-1}(\mu) = \frac{\log(\tau)}{1 - \mu^{\theta}}$$

Value

`dugompertz` gives the density, `pugompertz` gives the distribution function, `qugompertz` gives the quantile function and `rugompertz` generates random deviates.

Invalid arguments will return an error message.

Author(s)

Josmar Mazucheli <jmazucheli@gmail.com>

André F. B. Menezes <andrefelipemaringa@gmail.com>

References

Mazucheli, J., Menezes, A. F. and Dey, S., (2019). Unit-Gompertz Distribution with Applications. *Statistica*, **79**(1), 25-43.

Examples

```
set.seed(123)
x <- rugompertz(n = 1000, mu = 0.5, theta = 2, tau = 0.5)
R <- range(x)
S <- seq(from = R[1], to = R[2], by =  0.01)
hist(x, prob = TRUE, main = 'unit-Gompertz')
lines(S, dugompertz(x = S, mu = 0.5, theta = 2, tau = 0.5), col = 2)
plot(ecdf(x))
lines(S, pugompertz(q = S, mu = 0.5, theta = 2, tau = 0.5), col = 2)
plot(quantile(x, probs = S), type = "l")
lines(qugompertz(p = S, mu = 0.5, theta = 2, tau = 0.5), col = 2)
```

ugumbel *The unit-Gumbel distribution*

Description

Density function, distribution function, quantile function and random number generation function for the unit-Gumbel distribution reparametrized in terms of the τ-th quantile, $\tau \in (0, 1)$.

Usage

```
dugumbel(x, mu, theta, tau = 0.5, log = FALSE)

pugumbel(q, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

qugumbel(p, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

rugumbel(n, mu, theta, tau = 0.5)
```

Arguments

`x, q`	vector of positive quantiles.
`mu`	location parameter indicating the τ-th quantile, $\tau \in (0, 1)$.
`theta`	nonnegative shape parameter.
`tau`	the parameter to specify which quantile use in the parametrization.
`log, log.p`	logical; If TRUE, probabilities p are given as log(p).
`lower.tail`	logical; If TRUE, (default), $P(X \leq x)$ are returned, otherwise $P(X > x)$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(y \mid \alpha, \theta) = \frac{\theta}{y(1-y)} \exp\left\{-\alpha - \theta \log\left(\frac{y}{1-y}\right) - \exp\left[-\alpha - \theta \log\left(\frac{y}{1-y}\right)\right]\right\}$$

Cumulative distribution function

$$F(y \mid \alpha, \theta) = \exp\left[-\exp\left(-\alpha\right)\left(\frac{1-y}{y}\right)^{\theta}\right]$$

Quantile function

$$Q(\tau \mid \alpha, \theta) = \frac{\left[-\frac{1}{\log(\tau)}\right]^{\frac{1}{\theta}}}{\exp\left(\frac{\alpha}{\theta}\right) + \left[-\frac{1}{\log(\tau)}\right]^{\frac{1}{\theta}}}$$

Reparameterization

$$\alpha = g^{-1}(\mu) = \theta \log\left(\frac{1-\mu}{\mu}\right) + \log\left(-\frac{1}{\log(\tau)}\right)$$

where $0 < y < 1$ and $\theta > 0$ is the shape parameter.

Value

`dugumbel` gives the density, `pugumbel` gives the distribution function, `qugumbel` gives the quantile function and `rugumbel` generates random deviates.

Invalid arguments will return an error message.

Author(s)

Josmar Mazucheli

Andre F. B. Menezes

References

Mazucheli, J. and Alves, B., (2021). The unit-Gumbel Quantile Regression Model for Proportion Data. *Under Review*.

Gumbel, E. J., (1941). The return period of flood flows. *The Annals of Mathematical Statistics*, **12**(2), 163–190.

Examples

```
set.seed(6969)
x <- rugumbel(n = 1000, mu = 0.5, theta = 1.5, tau = 0.5)
R <- range(x)
S <- seq(from = R[1], to = R[2], by = 0.01)
hist(x, prob = TRUE, main = 'unit-Gumbel')
lines(S, dugumbel(x = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
plot(ecdf(x))
lines(S, pugumbel(q = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
plot(quantile(x, probs = S), type = "l")
lines(qugumbel(p = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
```

`ulogistic` *The unit-Logistic distribution*

Description

Density function, distribution function, quantile function and random number generation for the unit-Logistic distribution reparametrized in terms of the τ-th quantile, $\tau \in (0, 1)$.

Usage

```
dulogistic(x, mu, theta, tau = 0.5, log = FALSE)

pulogistic(q, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

qulogistic(p, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

rulogistic(n, mu, theta, tau = 0.5)
```

Arguments

`x, q`	vector of positive quantiles.
`mu`	location parameter indicating the τ-th quantile, $\tau \in (0,1)$.
`theta`	nonnegative shape parameter.
`tau`	the parameter to specify which quantile is to used.
`log, log.p`	logical; If TRUE, probabilities p are given as log(p).
`lower.tail`	logical; If TRUE, (default), $P(X \leq x)$ are returned, otherwise $P(X > x)$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(y \mid \alpha, \theta) = \frac{\theta \exp(\alpha) \left(\frac{y}{1-y}\right)^{\theta-1}}{\left[1 + \exp(\alpha) \left(\frac{y}{1-y}\right)^{\theta}\right]^2}$$

Cumulative distribution function

$$F(y \mid \alpha, \theta) = \frac{\exp(\alpha) \left(\frac{y}{1-y}\right)^{\theta}}{1 + \exp(\alpha) \left(\frac{y}{1-y}\right)^{\theta}}$$

Quantile function

$$Q(\tau \mid \alpha, \theta) = \frac{\exp\left(-\frac{\alpha}{\theta}\right) \left(\frac{\tau}{1-\tau}\right)^{\frac{1}{\theta}}}{1 + \exp\left(-\frac{\alpha}{\theta}\right) \left(\frac{\tau}{1-\tau}\right)^{\frac{1}{\theta}}}$$

Reparameterization

$$\alpha = g^{-1}(\mu) = \log\left(\frac{\tau}{1-\tau}\right) - \theta \log\left(\frac{\mu}{1-\mu}\right)$$

Value

`dulogistic` gives the density, `pulogistic` gives the distribution function, `qulogistic` gives the quantile function and `rulogistic` generates random deviates.

Invalid arguments will return an error message.

Author(s)

Josmar Mazucheli <jmazucheli@gmail.com>

André F. B. Menezes <andrefelipemaringa@gmail.com>

References

Paz, R. F., Balakrishnan, N. and Bazán, J. L., 2019. L-Logistic regression models: Prior sensitivity analysis, robustness to outliers and applications. *Brazilian Journal of Probability and Statistics*, **33**(3), 455–479.

Examples

```
set.seed(123)
x <- rulogistic(n = 1000, mu = 0.5, theta = 1.5, tau = 0.5)
R <- range(x)
S <- seq(from = R[1], to = R[2], by =  0.01)
hist(x, prob = TRUE, main = 'unit-Logistic')
lines(S, dulogistic(x = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
plot(ecdf(x))
lines(S, pulogistic(q = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
plot(quantile(x, probs = S), type = "l")
lines(qulogistic(p = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
```

`unitquantreg` *Parametric unit quantile regression models*

Description

Fit a collection of parametric unit quantile regression model by maximum likelihood using the log-likelihood function, the score vector and the hessian matrix implemented in `C++`.

Usage

```
unitquantreg(
  formula,
  data,
  subset,
  na.action,
  tau,
  family,
  link = c("logit", "probit", "cloglog", "cauchit"),
```

```
  link.theta = c("identity", "log", "sqrt"),
  start = NULL,
  control = unitquantreg.control(),
  model = TRUE,
  x = FALSE,
  y = TRUE
)

unitquantreg.fit(
  y,
  X,
  Z = NULL,
  tau,
  family,
  link,
  link.theta,
  start = NULL,
  control = unitquantreg.control()
)
```

Arguments

formula	symbolic description of the quantile model like `y ~ x` or `y ~ x | z`. See below for details.
data	data.frame contain the variables in the model.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
na.action	a function which indicates what should happen when the data contain `NA`s.
tau	numeric vector. The quantile(s) to be estimated, i.e., number between 0 and 1. If just one quantile is specified an object of class `unitquantreg` is returned. If a numeric vector of values between 0 and 1 is specified an object of class `unitquantregs` is returned. See below for details.
family	character. Specify the distribution family.
link	character. Specify the link function in the quantile model. Currently supported are `logit`, `probit`, `cloglog` and `cauchit`. Default is `logit`.
link.theta	character. Specify the link function in the shape model. Currently supported are `identity`, `log` and `sqrt`. Default is `log`.
start	numeric vector. An optional vector with starting values for all parameters.
control	list. Control arguments specified via `unitquantreg.control`.
model	logical. Indicates whether model frame should be included as a component of the returned value.
x, y	logical. If `TRUE` the corresponding components of the fit (model frame, response, model matrix) are returned. For `unitquantreg.fit` `y` should be the numeric response vector with values in (0,1).
X, Z	numeric matrix. Regressor matrix for the quantile and shape model, respectively. Default is constant shape model, i.e., `Z` is matrix with column of ones.

Details

The parameter estimation and inference are performed under the frequentist paradigm. The optimx R package is use, since allows different optimization technique to maximize the log-likelihood function. The analytical score function are use in the maximization and the standard errors are computed using the analytical hessian matrix, both are implemented in efficient away using C++.

Value

unitquantreg can return an object of class unitquantreg if tau is a scalar, i.e., a list with the following components.

family	the distribution family name.
coefficients	a list with elements "quantile" and "shape" containing the coefficients from the respective models.
fitted.values	a list with elements "quantile" and "shape" containing the fitted parameters from the respective models.
linear.predictors	a list with elements "quantile" and "shape" containing the fitted linear predictors from the respective models.
link	a list with elements "quantile" and "shape" containing the link objects from the respective models.
tau	the quantile specify.
loglik	log-likelihood of the fitted model.
gradient	gradient evaluate at maximum likelihood estimates.
vcov	covariance matrix of all parameters in the model.
nobs	number of observations.
npar	number of parameters.
df.residual	residual degrees of freedom in the fitted model.
theta_const	logical indicating if the θ parameter was treated as nuisance parameter.
control	the control parameters used to fit the model.
iterations	number of iterations of optimization method.
converged	logical, if TRUE indicates successful convergence.
kkt	a list of logical kkt1 and kkt2 provide check on Kuhn-Karush-Tucker conditions, first-order KKT test (kkt1) checks whether the gradient at the final parameters estimates is "small" and the second-order KKT test (kkte) checks whether the Hessian at the final parameters estimates is positive definite.
elapsed_time	time elapsed to fit the model.
call	the original function call.
formula	the original model formula.
terms	a list with elements "quantile", "shape" and "full" containing the terms objects for the respective models.
model	the full model frame, if model = TRUE.

`y`	the response vector, if `y = TRUE`.
`x`	a list with elements `"quantile"` and `"shape"` containing the model matrices from the respective models, if `x = TRUE`.

While `unitquantreg.fit` returns an unclassed list with components up to `elapsed_time`.

If `tau` is a numeric vector with length greater than one an object of class `unitquantregs` is returned, which consist of list of objects of class `unitquantreg` for each specified quantiles.

Author(s)

André F. B. Menezes

`unitquantreg.control` *Control parameters for unit quantile regression*

Description

Auxiliary function that control fitting of unit quantile regression models using `unitquantreg`.

Usage

```
unitquantreg.control(
  method = "BFGS",
  hessian = FALSE,
  gradient = TRUE,
  maxit = 5000,
  factr = 1e+07,
  reltol = sqrt(.Machine$double.eps),
  trace = 0L,
  starttests = FALSE,
  dowarn = FALSE,
  ...
)
```

Arguments

`method`	string. Specify the method argument passed to `optimx`.
`hessian`	logical. Should use the numerically Hessian matrix to compute variance-covariance? Default is `FALSE`, i.e., use the analytic Hessian.
`gradient`	logical. Should use the analytic gradient? Default is `TRUE`.
`maxit`	integer. Specify the maximal number of iterations passed to `optimx`.
`factr`	numeric.Controls the convergence of the `"L-BFGS-B"` method.
`reltol`	numeric. Relative convergence tolerance passed to `optimx`.
`trace`	non-negative integer. If positive, tracing information on the progress of the optimization is produced.

starttests	logical. Should optimx run tests of the functions and parameters? Default is FALSE.
dowarn	logical. Show warnings generated by optimx? Default is FALSE.
...	arguments passed to optimx.

Details

The control argument of unitquantreg uses the arguments of unitquantreg.control. In particular, the parameters in unitquantreg are estimated by maximum likelihood using the optimx, which is a general-purpose optimization wrapper function that calls other R tools for optimization, including the existing optim function. The main advantage of optimx is to unify the tools allowing a number of different optimization methods and provide sanity checks.

Value

A list with components named as the arguments.

Author(s)

André F. B. Menezes

References

Nash, J. C. and Varadhan, R. (2011). Unifying Optimization Algorithms to Aid Software System Users: optimx for R., *Journal of Statistical Software*, **43**(9), 1–14.

See Also

optimx for more details about control parameters and unitquantreg.fit the fitting procedure used by unitquantreg.

Examples

```
data(sim_bounded, package = "unitquantreg")
sim_bounded_curr <- sim_bounded[sim_bounded$family == "uweibull", ]

# Fitting using the analytical gradient
fit_gradient <- unitquantreg(formula = y1 ~ x,
                             data = sim_bounded_curr, tau = 0.5,
                             family = "uweibull",
                             control = unitquantreg.control(gradient = TRUE,
                                                            trace = 1))

# Fitting without using the analytical gradient
fit_nogradient <- unitquantreg(formula = y1 ~ x,
                             data = sim_bounded_curr, tau = 0.5,
                             family = "uweibull",
                             control = unitquantreg.control(gradient = FALSE,
                                                            trace = 1))
# Compare estimated coefficients
cbind(gradient = coef(fit_gradient), no_gradient = coef(fit_nogradient))
```

`uweibull` *The unit-Weibull distribution*

Description

Density function, distribution function, quantile function and random number generation function for the unit-Weibull distribution reparametrized in terms of the τ-th quantile, $\tau \in (0, 1)$.

Usage

```
duweibull(x, mu, theta, tau = 0.5, log = FALSE)

puweibull(q, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

quweibull(p, mu, theta, tau = 0.5, lower.tail = TRUE, log.p = FALSE)

ruweibull(n, mu, theta, tau = 0.5)
```

Arguments

`x, q`	vector of positive quantiles.
`mu`	location parameter indicating the τ-th quantile, $\tau \in (0, 1)$.
`theta`	nonnegative shape parameter.
`tau`	the parameter to specify which quantile use in the parametrization.
`log, log.p`	logical; If TRUE, probabilities p are given as log(p).
`lower.tail`	logical; If TRUE, (default), $P(X \leq x)$ are returned, otherwise $P(X > x)$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

Probability density function

$$f(y \mid \alpha, \theta) = \frac{\alpha\theta}{y} \left[-\log(y)\right]^{\theta-1} \exp\left\{-\alpha \left[-\log(y)\right]^{\theta}\right\}$$

Cumulative distribution function

$$F(y \mid \alpha, \theta) = \exp\left\{-\alpha \left[-\log(y)\right]^{\theta}\right\}$$

Quantile function

$$Q\left(\tau \mid \alpha, \theta\right) = \exp\left\{-\left[-\frac{\log(\tau)}{\alpha}\right]^{\frac{1}{\theta}}\right\}$$

Reparameterization

$$\alpha = g^{-1}(\mu) = -\frac{\log(\tau)}{\left[-\log(\mu)\right]^{\theta}}$$

Value

duweibull gives the density, puweibull gives the distribution function, quweibull gives the quantile function and ruweibull generates random deviates.

Invalid arguments will return an error message.

Author(s)

Josmar Mazucheli

André F. B. Menezes

References

Mazucheli, J., Menezes, A. F. B and Ghitany, M. E., (2018). The unit-Weibull distribution and associated inference. *Journal of Applied Probability and Statistics*, **13**(2), 1–22.

Mazucheli, J., Menezes, A. F. B., Fernandes, L. B., Oliveira, R. P. and Ghitany, M. E., (2020). The unit-Weibull distribution as an alternative to the Kumaraswamy distribution for the modeling of quantiles conditional on covariates. *Journal of Applied Statistics*, **47**(6), 954–974.

Mazucheli, J., Menezes, A. F. B., Alqallaf, F. and Ghitany, M. E., (2021). Bias-Corrected Maximum Likelihood Estimators of the Parameters of the Unit-Weibull Distribution. *Austrian Journal of Statistics*, **50**(3), 41–53.

Examples

```
set.seed(6969)
x <- ruweibull(n = 1000, mu = 0.5, theta = 1.5, tau = 0.5)
R <- range(x)
S <- seq(from = R[1], to = R[2], by = 0.01)
hist(x, prob = TRUE, main = 'unit-Weibull')
lines(S, duweibull(x = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
plot(ecdf(x))
lines(S, puweibull(q = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
plot(quantile(x, probs = S), type = "l")
lines(quweibull(p = S, mu = 0.5, theta = 1.5, tau = 0.5), col = 2)
```

---

vuong.test *Vuong test*

Description

Performs Vuong test between two fitted objects of class unitquantreg

Usage

```
vuong.test(object1, object2, alternative = c("two.sided", "less", "greater"))
```

Arguments

`object1, object2`
: objects of class `unitquantreg` containing the fitted models.

`alternative`
: indicates the alternative hypothesis and must be one of `"two.sided"` (default), `"less"`, or `"greater"`. You can specify just the initial letter of the value, but the argument name must be given in full. See 'Details' for the meanings of the possible values.

Details

The statistic of Vuong likelihood ratio test for compare two non-nested regression models is defined by

$$T = \frac{1}{\widehat{\omega}^2 \sqrt{n}} \sum_{i=1}^{n} \log \frac{f(y_i \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\theta}})}{g(y_i \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\gamma}})}$$

where

$$\widehat{\omega}^2 = \frac{1}{n} \sum_{i=1}^{n} \left(\log \frac{f(y_i \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\theta}})}{g(y_i \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\gamma}})} \right)^2 - \left[\frac{1}{n} \sum_{i=1}^{n} \left(\log \frac{f(y_i \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\theta}})}{g(y_i \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\gamma}})} \right) \right]^2$$

is an estimator for the variance of $\frac{1}{\sqrt{n}} \sum_{i=1}^{n} \log \frac{f(y_i \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\theta}})}{g(y_i \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\gamma}})}$, $f(y_i \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\theta}})$ and $g(y_i \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\gamma}})$ are the corresponding rival densities evaluated at the maximum likelihood estimates.

When $n \to \infty$ we have that $T \to N(0, 1)$ in distribution. Therefore, at $\alpha\%$ level of significance the null hypothesis of the equivalence of the competing models is rejected if $|T| > z_{\alpha/2}$, where $z_{\alpha/2}$ is the $\alpha/2$ quantile of standard normal distribution.

In practical terms, $f(y_i \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\theta}})$ is better (worse) than $g(y_i \mid \boldsymbol{x}_i, \widehat{\boldsymbol{\gamma}})$ if $T > z_{\alpha/2}$ (or $T < -z_{\alpha/2}$).

Value

A list with class `"htest"` containing the following components:

`statistic`
: the value of the test statistic.

`p.value`
: the p-value of the test.

`alternative`
: a character string describing the alternative hypothesis.

`method`
: a character string with the method used.

`data.name`
: a character string ginven the name of families models under comparison.

Author(s)

André F. B. Menezes

Josmar Mazucheli

References

Vuong, Q. (1989). Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica*, **57**(2), 307–333.

Examples

```
data(sim_bounded, package = "unitquantreg")
sim_bounded_curr <- sim_bounded[sim_bounded$family == "uweibull", ]

fit_uweibull <- unitquantreg(formula = y1 ~ x, tau = 0.5,
                             data = sim_bounded_curr,
                             family = "uweibull")
fit_kum <- unitquantreg(formula = y1 ~ x, tau = 0.5,
                             data = sim_bounded_curr,
                             family = "kum")

ans <- vuong.test(object1 = fit_uweibull, object2 = fit_kum)
ans
str(ans)
```

---

water *Access to piped water supply data set*

Description

The access of people in households with piped water supply in the cities of Brazil from the Southeast and Northeast regions. Information obtained during the census of 2010.

Usage

```
data(water, package = "unitquantreg")
```

Format

`data.frame` with 3457 observations and 5 columns:

- `phpws`: the proportion of households with piped water supply.
- `mhdi`: municipal human development index.
- `incpc`: per capita income.
- `region`: 0 for Southeast, 1 for Northeast.
- `pop`: total population.

Author(s)

André F. B. Menezes

References

Mazucheli, J., Menezes, A. F. B., Fernandes, L. B., Oliveira, R. P. and Ghitany, M. E., (2020). The unit-Weibull distribution as an alternative to the Kumaraswamy distribution for the modeling of quantiles conditional on covariates. *Jounal of Applied Statistics*, **47**(6), 954–974.

Index