Package: rquest (via r-universe)

November 23, 2026

Title Hypothesis Tests for Quantiles and Quantile-Based Measures

Version 1.1.1

Description Functions to conduct hypothesis tests and derive confidence intervals for quantiles, linear combinations of quantiles, ratios of dependent linear combinations and differences and ratios of all of the above for comparisons between independent samples. Additionally, quantile-based measures of inequality are also considered.

Encoding UTF-8

License GPL (>= 3)

Imports stats, gld, rlang

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://github.com/shenal-dkumara/rquest

BugReports https://github.com/shenal-dkumara/rquest/issues

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Shenal Dedduwakumara [aut, cre], Luke Prendergast [aut], Robert Staudte [aut]

Maintainer Shenal Dedduwakumara <shenal.dedduwakumara@adelaide.edu.au>

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

Date/Publication 2026-11-23 04:30:02 UTC

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

RemoteRef HEAD

RemoteSha 9b5942b5d1ecc8b2cb1bc1be289c5db719e2ef98

Contents

q.test 2
qcov 6
qden 8
qineq 10
qor 12
qrcov 14
rcv.test 15

Index **19**

`q.test` *Hypothesis Tests and Confidence Intervals for functions of quantiles*

Description

carry out hypothesis tests and obtain associated confidence intervals for linear combinations of quantiles, and ratios of such linear combinations.

Usage

```
q.test(
  x,
  y = NULL,
  measure = "median",
  u = NULL,
  coef = NULL,
  u2 = NULL,
  coef2 = NULL,
  quantile.type = 7,
  var.method = "qor",
  alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95,
  true.q = 0,
  log.transf = FALSE,
  back.transf = FALSE,
  min.q = -Inf,
  p = NULL,
  ...
)
```

Arguments

`x`	a numeric vector of data values.
`y`	an optional second vector of data values for two-sample testing.
`measure`	a character string specifying the quantile measure to be estimated (See details).

u	a numeric vector of probability values in the interval (0,1) specifying the quantiles to be estimated. Note that u must include numeric values between,and not including, 0 and 1 and missing values are not allowed.
coef	a vector or matrix with two rows specifying the coefficients that define the linear combinations (coefficients must match the corresponding probability values in u). If coef is a vector then a single linear combination (LC) is computed. If it is a matrix, then first row defines the numerator LC and the second the denominator LC.
u2	a numeric vector of probability values in the interval (0,1) specifying the quantiles to be estimated. Note that u2 must include numeric values between, and not including, 0 and 1 and missing values are not allowed.
coef2	a vector specifying the coefficients that define the linear combination for the denominator. This can be used as an alternative to defining coef as matrix for ratios of linear combinations.
quantile.type	argument for the quantile function. Default is set to 7 so that output is consistent with default quantile function use and other functions such as IQR (see help file for `quantile()` for more details)
var.method	approach use to estimate the quantile density function. Either `"qor"` (default) or `"density"` (see details).
alternative	a character string for alternative hypothesis equal to one of "two.sided", "greater" or "less".
conf.level	coverage for the estimated confidence interval.
true.q	a numeric value for the true value under the null hypothesis test.
log.transf	boolean indicating whether a log transformation of the measure is to be used (i.e., estimates of the log of the measure are computed).
back.transf	boolean indicating whether the measure and estimates should be back-transformed to the original scale using exp.
min.q	the lower bound for a one-sided confidence interval when alternative argument if "less".
p	optional value in (0, 1) for Bowley's generalized skewness coefficient.
...	additional arguments to be passed to function `qcov()`.

Details

This function `q.test` performs hypothesis tests and calculates confidence intervals for linear combinations of quantiles. The quantile measures that can be estimated are specified in the `measure` argument and are listed below.

- `median`: The default choice.
- `iqr`: The interquartile range.
- `rCViqr`: The robust Coefficient of Variation measure using the IQR (Arachchige et al.,2022)
- `bowley`, `groenR` and `groenL`: These choices are for Bowley's skew coefficient, for the generalized measure, and the right and left skew measures by Groeneveld and Meeden (Groeneveld & Meeden, 1984, 2009). For each of these measures, the user may specify the choice of p using argument `p`. If this is not used, then the default is `p=0.25`.

- `kelly`: Kelly's quantile skewness measure, based on the 0.1, 0.5, and 0.9 quantiles.
- `moors`: This Moors kurtosis measure (Moors, 1988).
- `lqw` and `rqw`: These are for the robust left and right tail weights (Brys et al., 2006).
- `qrxxyy`: A character string consisting of the first two characters "qr" and followed by four numbers will request a ratio of dependent quantiles (i.e. different quantiles from the same sample).The first two number digits (in place of "xx") will indicate the quantile for the numerator, and the second two numerical digits "yy" for the denominator. For example, `qr9010` will estimate and test the ratio Q(0.9)/Q(0.1).

The default `var.method="qor"` is to estimate the probability density function directly using the lognormal Quantile Optimality Ratio (QOR) for choosing a suitable bandwidth (Prendergast & Staudte,2016). Alternatively, the variances can be estimated by inverting a density estimator evaluated at the quantiles and this can be done using `var.method = "density"`. If `var.method = "density"`, then the function density is used to estimate the probability density function which is needed for the calculation of the covariance matrix using function qcov. If needed, additional arguments can be passed to density (see ?density for details on possible additional arguments).

Additional to using a text string with argument measure to indicate which quantile-based measure is to be used (of those included for the `q.test` function), users can also define and request their own. For example, while the median is the default for a single quantile, other quantiles can also be requested (e.g., `u = 0.25` will request the first quartile). This option is also included for linear combinations of quantiles and ratios of linear combinations, which can be done in two ways. For a single linear combination, argument `u` is the vector of probability values defining the quantiles to be used, and argument `coef` is a vector of coefficients for the linear combination. If a ratio is needed, then `u` and `coef` are used for the numerator and `u2` and `coef2` for the denominator linear combination. Alternatively, a single `u` can be used to identify all quantiles for the ratio, and then `coef` can be a matrix (2 rows) whose first row specifies the coefficients for the numerator and the second row for the denominator. For more information and further examples, see Prendergast, Dedduwakumara & Staudte (2024), and the example code below shows how to obtain results for the robust CVs using all three approaches (to achieve identical results).

Further details on confidence intervals, including coverage properties of, can be found for ratios of dependent quantiles (Prendergast & Staudte, 2017), ratios of independent quantiles and IQRs (Arachchige et al., 2021) and for robust CVs (Arachchige et al., 2022).

Value

hypothesis test results and associated confidence interval (a list with class "htest")

References

Arachchige, C. N., Cairns, M., & Prendergast, L. A. (2021). Interval estimators for ratios of independent quantiles and interquantile ranges. Communications in Statistics-Simulation and Computation, 50(12), 3914-3930.

Arachchige, C.N.P.G., Prendergast, L.A., & Staudte, R.G. (2022). Robust Analogs to the Coefficient of Variation. Journal of Applied Statistics, 49(2), 268–290.

Brys, G., Hubert, M., & Struyf, A. (2006). Robust measures of tail weight. Computational Statistics & Data Analysis, 50(3), 733–759.

Groeneveld, R. A., & Meeden, G. (1984). Measuring skewness and kurtosis. Journal of the Royal Statistical Society Series D: The Statistician, 33(4), 391–399.

Groeneveld, R. A., & Meeden, G. (2009). An improved skewness measure. Metron, 67(3), 325.

Hyndman, R.J., & Fan, Y. (1996). Sample quantiles in statistical packages. The American Statistician, 50(4), 361–365.

Moors, J. J. A. (1988). A quantile alternative for kurtosis. Journal of the Royal Statistical Society: Series D (The Statistician), 37(1), 25–32.

Prendergast, L. A., & Staudte, R. G. (2016). Exploiting the quantile optimality ratio in finding confidence intervals for quantiles. Stat, 5(1), 70-81

Prendergast, L. A., & Staudte, R. G. (2017). When large n is not enough–distribution-free interval estimators for ratios of quantiles. The Journal of Economic Inequality, 15, 277-293.

Prendergast, L. A., Dedduwakumara, D.S. & Staudte, R.G. (2024) rquest: An R package for hypothesis tests and confidence intervals for quantiles and summary measures based on quantiles, preprint, pages 1-13

See Also

qor for quantile optimality ratio values, qcov for covariance estimation of sample quantiles, and qrcov for covariance estimation of ratios of linear combinations of quantiles.

Examples

```
#  Create some data
x <- c(8.43,7.08,8.79,8.88,7.87,5.94,8.79,5.46,8.11,7.08)
y <- c(13.44,13.65,14.77,9.51,14.07,10.92,11.59,13.42,8.93,10.88)

# One sample hypothesis test for the IQR
q.test(x, measure = "iqr")

# Two samples hypothesis test for robust coefficient variations (0.75*IQR/median)
# with log transformation and back-transformation to the ratio scale,.
q.test(x, y, measure = "rCViqr", log.transf = TRUE, back.transf = TRUE)

# The same two samples hypothesis test for robust coefficient variations (0.75*IQR/median)
# by using 'u','u2','coef' and 'coef2' arguments.
u<-c(0.25,0.75)
iqr_weights<-0.75*c(-1,1)
u2<-0.5
coef2<-1
q.test(x,y,u=u,u2=u2,coef=iqr_weights,coef2=coef2,log.transf=TRUE,back.transf=TRUE)

# The same two samples hypothesis test for robust coefficient variations (0.75*IQR/median)
# by using only 'u' and 'coef' arguments.
u<-c(0.25,0.5,0.75)
num <- 0.75*c(-1,0,1)
den <- c(0,1,0)
ratio_weights <- rbind(num, den)
q.test(x,y,u=u,coef=ratio_weights,log.transf=TRUE,back.transf=TRUE)
```

qcov *Approximate Covariance Matrix Estimation for Vectors of Quantile Estimators*

Description

compute a covariance matrix consisting of variances (on the diagonal) for quantile estimates and covariances (off-diagonal) between different quantile estimates

Usage

```
qcov(
  x,
  u,
  method = "qor",
  dist = "gl",
  quantile.type = 7,
  bw.correct = TRUE,
  params = NULL,
  ...
)
```

Arguments

x	a numeric vector of data values.
u	a numeric vector of probability values in the interval $(0, 1)$ specifying the quantiles to be estimated. Missing values are not allowed.
method	the approach used to estimate the quantile density function. Either "qor" or "density".
dist	a character string naming a supported distribution or a quantile function name used by qden.
quantile.type	argument for the quantile function. Default is set to 7 so that output is consistent with default quantile function use and other functions such as IQR (see ?quantile for details).
bw.correct	logical; if TRUE, the bandwidth is corrected near the boundary.
params	a list of parameter values for the chosen distribution. If NULL, parameters are estimated where needed.
...	additional arguments to be passed to qden when method = "qor" and to density when method = "density".

Details

This function computes an approximate covariance matrix for a vector of sample quantile estimators. The covariance structure is based on the asymptotic relationship, for $u_i \leq u_j$ and sample size n,

$$\mathrm{Cov}(\widehat{Q}(u_i), \widehat{Q}(u_j)) \approx \frac{u_i(1-u_j)}{n}\, q(u_i)\, q(u_j),$$

where $q(u) = Q'(u)$ is the quantile density.

The quantile density is estimated by `qden`. By default, `method = "qor"` is used, which estimates the quantile density using the quantile optimality ratio (QOR) approach of Prendergast and Staudte (2016). Alternatively, `method = "density"` estimates the quantile density by fitting a density function and taking its reciprocal at the estimated quantiles.

The argument `dist` specifies the working distribution used by `qden`. It may be one of the built-in distributions supported by `qden`, or a quantile function name when a user-supplied distribution is to be used. If `params` is `NULL`, the required distributional parameters are estimated from the data where appropriate.

If `method = "qor"`, the bandwidth is selected using the QOR approach. The argument `bw.correct` controls the boundary correction used in that bandwidth calculation.

If `method = "density"`, additional arguments in `...` are passed to `density` when density estimation is used directly.

Value

a covariance matrix consisting of variances (on the diagonal) for quantile estimates and covariances (off-diagonal) between different quantile estimates.

References

Prendergast, L. A., & Staudte, R. G. (2016). Exploiting the quantile optimality ratio in finding confidence intervals for quantiles. *Stat*, **5**(1), 70–81.

Prendergast, L. A., Dedduwakumara, D. S., & Staudte, R. G. (2024). *rquest: An R package for hypothesis tests and confidence intervals for quantiles and summary measures based on quantiles*. Preprint, pages 1–13.

See Also

`qden` for estimating the quantile density function, `qor` for quantile optimality ratio values, and `qrcov` for covariance matrices of ratios of linear combinations of quantiles.

Examples

```
set.seed(1234)
x <- rnorm(100)

# Covariance matrix for sample quartiles
qcov(x, c(0.25, 0.5, 0.75))

# Density-based version
qcov(x, c(0.25, 0.5, 0.75), method = "density")
```

qden *Quantile Density Estimation*

Description

estimate the quantile density function using either a direct estimation approach with a bandwidth selected by the quantile optimality ratio (QOR), or by the density method which uses the inverse of the estimated density at the nominated quantiles

Usage

```
qden(
  x,
  u,
  dist = "gl",
  method = "qor",
  kernel = "epanechnikov",
  params = NULL,
  gl.control = list(method = "Lmom"),
  quantile.type = 7,
  bw.correct = TRUE,
  large.n.switch = TRUE,
  ...
)
```

Arguments

x	a numeric vector of data values.
u	a numeric vector of probability values in the interval $(0, 1)$ specifying where the quantile density is to be estimated. Missing values are not allowed.
dist	a character string naming a distribution or a quantile function name. Supported built-in distributions include "gl", "lnorm", "norm", and "exp".
method	the estimation method to use. The options are "qor" and "density".
kernel	the kernel to use when method = "qor".
params	a list of parameter values. If NULL, parameters are estimated where needed.
gl.control	a list of control arguments passed to fit.fkml when dist = "gl".
quantile.type	argument for the quantile function used when method = "density". Default is 7 so that output is consistent with other functions such as IQR.
bw.correct	logical; if TRUE, the bandwidth is corrected near the boundary.
large.n.switch	logical; if TRUE and dist = "gl" with large n, then "lnorm" is used for bandwidth selection.
...	additional arguments to be passed to density when method = "density" is used, or additional arguments passed to qor when method = "qor" is used.

Details

This function estimates the quantile density function at a vector of probability values u. Two estimation methods are available.

If method = "qor", a bandwidth is selected using the quantile optimality ratio (QOR) and a kernel-based estimator is used. The bandwidth depends on the selected kernel through its variance and roughness constants. Supported kernels are "epanechnikov", "gaussian", "rectangular", "triangular", "biweight", "cosine", and "optcosine". By default, the QOR method uses the flexible four parameter Generalized Lambda Distribution (GLD) to compute the QOR. The method of L-moments is used for GLD parameter estimation, although other estimators can be chosen by passing arguments through the gl.control argument to the fit.fkml function from the **gld** package which is used for estimation. Other included distributions for computing the QOR are the lognormal dist = "lnorm", Gaussian dist = "norm", and exponential "exp". If another character string is passed to dist, then the function will attempt to locate the associated quantile function and so dist needs to follow R's naming convention of prefixing the distribution name with q. E.g., dist = "beta" will use the associated quantile function for the beta distribution, qbeta. The parameter values to be used for this distribution can be specified using the list argument params. If params is NULL, the function will attempt to use default choices for these parameters.

If method = "density", the quantile density is estimated as the reciprocal of a density estimate evaluated at the estimated quantiles. When params is NULL, the density estimate is obtained directly from density. When params is supplied, the corresponding quantile and density functions from argument dist are used. E.g., if dist = "beta", then the quantile density will be computed using the qbeta and dbeta functions.

Value

a vector of estimated quantile density values at u.

References

Prendergast, L. A., Dedduwakumara, D.S. & Staudte, R.G. (2024). *rquest: An R package for hypothesis tests and confidence intervals for quantiles and summary measures based on quantiles*. Preprint, pages 1–13.

See Also

qor for quantile optimality ratio values, qcov for covariance estimation of sample quantiles, and qrcov for covariance estimation of ratios of linear combinations of quantiles.

Examples

```
set.seed(1234)
x <- rnorm(100)

# QOR-based quantile density estimation using the flexible GLD distribution (default method)
qden(x, c(0.25, 0.5, 0.75), method = "qor")

# QOR-based quantile density estimation using the normal distribution
qden(x, c(0.25, 0.5, 0.75), dist = "norm", method = "qor")
```

```
# Density-based quantile density estimation
qden(x, c(0.25, 0.5, 0.75), method = "density")
```

`qineq` *Hypothesis Tests and Confidence Intervals for Quantile-based Inequality Measures*

Description

carry out hypothesis tests and obtain associated confidence intervals for quantile based inequality measures

Usage

```
qineq(
  x,
  y = NULL,
  J = 100,
  measure = "QRI",
  alternative = c("two.sided", "less", "greater"),
  quantile.type = 7,
  var.method = "qor",
  conf.level = 0.95,
  true.ineq = 0.5,
  ...
)
```

Arguments

`x`	a numeric vector of data values.
`y`	an optional second vector of data values for two-sample testing.
`J`	number of grid points
`measure`	character string of quantile based inequality measure to be estimated or a list of parameter choices or formula for user defined measures (see details for available measures and user defined measures).
`alternative`	a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less".
`quantile.type`	argument for the quantile function. Default is set to 7 so that output is consistent with default quantile function use and other functions such as IQR (see help file for `quantile()` for more details)
`var.method`	approach use to estimate the quantile density function. Either "qor"(default) or "density".(See details).
`conf.level`	coverage for the estimated confidence interval.
`true.ineq`	the specified hypothesized value of the inequality measure or the difference of the inequality depending on whether it was a one-sample test or a two-sample test.
`...`	additional arguments to be passed to function `qcov()`.

Details

This function performs hypothesis testing and calculates the corresponding confidence intervals for inequality measures based on quantiles. Let $l_1(p)$ and $l_2(p)$ be functions of $p \in [0,1]$ that defined linear combinations of quantiles. For example, if $x_{0.4}$ and $x_{0.6}$ are the 0.4 and 0.6 quantiles respectively, and the ratio $x_{0.4}/(x_{0.4}+x_{0.6})$ is of interest, then possible choices are $l_1(p) = x_p$ and $l_2(p) = x_p + x_{1-p}$ for $p = 0.4$. The inequality measures supported are of the form

$$I = 1 - \int_0^1 \frac{l_1(p)}{l_2(p)} dp$$

and where $I \in [0,1]$.

Estimation of I is done numerically over a grid of points determined by integer J (default 100), where $p_j = (j - 1/2)/J$, and the estimate to I is

$$\widehat{I} = 1 - \frac{1}{J}\sum_{j=1}^{J} \frac{\hat{l}_1(p_j)}{\hat{l}_2(p_j)}$$

where $\hat{l}_1(p_j)$ and $\hat{l}_2(p_j)$ are the estimated linear combinations of quantiles at p_j.

The available options for quantile based measures in argument `measure` are shown below.

- `"QRI"` : Quantile Ratio Index from Prendergast & Staudte (2018). This is the default choice.
- `"G1"`, `"G2"`, `"G3"` : Quantile variants of the Gini index from Prendergast & Staudte (2016a).
- `"S1"`, `"S2"` : 'Strategy 1' and 'Strategy 2' income inequalities from Brazauskas et al. (2024).

The default `var.method="qor"` is to estimate the probability density function directly using the lognormal Quantile Optimality Ratio (QOR) for choosing a suitable bandwidth (Prendergast & Staudte,2016b). Alternatively, the variances can be estimated by inverting a density estimator evaluated at the quantiles and this can be done using `var.method = "density"`. If `var.method = "density"`, then the function density is used to estimate the probability density function which is needed for the calculation of the covariance matrix using function qcov. If needed, additional arguments can be passed to density (see ?density for details on possible additional arguments).

It is also possible for users to define their own inequality measures either using a formula or by creating a list. The formula approach is the most straightforward of the two and can be defined in terms `Q` to specify a quantile function, `p` and then specifying the formula which must be a ratio of linear combinations of quantiles, and where the coefficients may also contain `p`. As an example, the QRI measure would be `measure = ~ Q(p/2)/Q(1 - p/2)` while G2 would require `measure = ~ 2*Q(p/2)/Q(1 - p/2)`. More details and examples can be found in Prendergast et al. (2024).

If `measure` is a list, it must be of the form `list(u=...,coef1=...,coef2)` where `u` is a numeric vector indicating the probabilities for every quantile estimate needed for $\hat{I}$, `coef1` and `coef2` are numeric matrices with `J` rows and number of columns equal to the length of `u`. Each row of `coef1` stores the coefficients for the numerator linear combination, and ditto `coef2` for the denominator.

For more information and further examples, including user defined measures, see Prendergast, Deddduwakumara & Staudte (2024).

Value

hypothesis test results and associated confidence interval (a list with class "htest")

References

Prendergast, L.A., & Staudte, R.G. (2016a). Quantile versions of the Lorenz curve. Electronic Journal of Statistics, 10(2), 1896 – 1926.

Prendergast, L. A., & Staudte, R. G. (2016b). Exploiting the quantile optimality ratio in finding confidence intervals for quantiles. Stat, 5(1), 70-81

Prendergast, L. A., & Staudte, R. G. (2018). A simple and effective inequality measure. The American Statistician, 72(4), 328-343.

Prendergast, L. A., Dedduwakumara, D.S. & Staudte, R.G. (2024) rquest: An R package for hypothesis tests and confidence intervals for quantiles and summary measures based on quantiles, preprint, pages 1-13

Brazauskas, V., Greselin, F., & Zitikis, R. (2024). Measuring income inequality via percentile relativities. Quality & Quantity, 58(5), 4859-4896.

Examples

```
# Create some data
x <- c(8.43,7.08,8.79,8.88,7.87,5.94,8.79,5.46,8.11,7.08)
y <- c(13.44,13.65,14.77,9.51,14.07,10.92,11.59,13.42,8.93,10.88)

# Test of equality of QRI measure between two groups
qineq(x, y, measure = "QRI")

# Another way by defining the formula for the ratio in QRI
qineq(x, y, measure = ~ Q(p/2)/Q(1 - p/2))

# Another way by defining the quantiles and coefficients directly in a list
J <- 100
p <- (1:J - 0.5)/J
u <- sort(c(p/2, 1 - p/2))
num <- cbind(diag(rep(1, J)), matrix(0, J, J))
den <- cbind(matrix(0, J, J), diag(J)[, J:1])
qineq(x, y, measure = list(u = u, coef1 = num, coef2 = den))
```

---

qor *Quantile Optimality Ratio*

Description

compute the quantile optimality ratio (QOR) for a vector of probabilities from either a named distribution or a user-supplied quantile function

Usage

```
qor(
  u,
  dist = "gl",
```

```
  x = NULL,
  params = NULL,
  gl.control = list(method = "Lmom")
)
```

Arguments

u	a numeric vector of probability values in the interval $(0, 1)$. Missing values are not allowed.
dist	a character string naming a supported distribution, or a function giving a quantile function. Supported character values currently include "gl", "lnorm", "norm", and "exp".
x	a numeric vector of data values. This is used to estimate distribution parameters when params is NULL for supported built-in distributions.
params	a list of parameter values to use. If NULL and dist is a supported built-in distribution, the parameters are estimated from x where needed. If NULL and dist is a function, the default arguments of that quantile function are used with a warning.
gl.control	a list of control arguments passed to fit.fkml when dist = "gl".

Details

This function evaluates the quantile optimality ratio (QOR, Prendergast & Staudte, 2016) for probabilities in u. For a quantile function Q, let $q(u) = Q'(u)$ denote the quantile density. The QOR is

$$\mathrm{QOR}(u) = \frac{q(u)}{q''(u)} = \frac{Q'(u)}{Q'''(u)}.$$

If dist is a character string, the QOR is computed using a closed-form expression whenever one is available. For "norm" and "exp", the QOR does not depend on the distribution parameters and therefore x is not required. For "lnorm", the required parameter is estimated from x when params is NULL. For "gl", the FKML generalized lambda distribution is fitted using fit.fkml and the options supplied in gl.control, unless params is supplied directly.

If dist is a function, it is treated as a quantile function. In that case, params supplies the arguments passed to the function. If params is NULL, the default parameter values of the supplied quantile function are used and a warning is issued. For user-supplied quantile functions, the QOR is evaluated numerically using finite differences.

Value

a list with components qor (the QOR values at u) and params (the parameter values used).

References

Prendergast, L. A., & Staudte, R. G. (2016). Exploiting the quantile optimality ratio in finding confidence intervals for quantiles. Stat, 5(1), 70-81

See Also

qcov for estimating the covariance matrix of sample quantiles, qrcov for estimating covariance matrices of ratios of linear combinations of quantiles, and fit.fkml from package **gld** for fitting the FKML generalized lambda distribution.

Examples

```
set.seed(1234)
x <- rlnorm(100)

# QOR using the flexible FKML GLD distribution (default)
qor(seq(0.1, 0.9, by = 0.2), x = x)

# QOR for the lognormal distribution
qor(seq(0.1, 0.9, by = 0.2), dist = "lnorm", x = x)

# QOR for the exponential distribution
qor(seq(0.1, 0.9, by = 0.2), dist = "exp")

# QOR for a user-supplied quantile function
qor(c(0.2, 0.5, 0.8),
    dist = qbeta,
    params = list(shape1 = 2, shape2 = 5))
```

---

qrcov *Approximate Covariance Matrix Estimation for Ratios of Linear Combinations of Quantile Estimators*

Description

compute a covariance matrix consisting of variances (on the diagonal) for ratios of linear combinations of quantile estimates and covariances (off-diagonal) between different ratios

Usage

```
qrcov(x, u, coef1, coef2, quantile.type = 7, ...)
```

Arguments

x	a numeric vector of data values.
u	a numeric vector of probability values in the interval (0,1) specifying the quantiles to be estimated. Note that u must include numeric values between, and not including, 0 and 1 and missing values are not allowed.
coef1	a numeric matrix whose ith row contains the coefficients for the numerator linear combination for the ith ratio.
coef2	a numeric matrix whose ith row contains the coefficients for the denominator linear combination for the ith ratio.

quantile.type	argument for the quantile function. Default is set to 7 so that output is consistent with default quantile function use and other functions such as IQR (see help file for quantile() for more details)
...	additional arguments to be passed to function qcov().

Details

This function computes a sample covariance matrix for ratios of linear combinations of quantile estimates from a single sample. This is done via a sample covariance matrix for the quantiles used in the linear combinations which is obtained using function qcov. Let $\mathbf{q}$ denote a vector of quantile estimators, $\mathbf{S}$ denote the covariance matrix for $\mathbf{q}$, and $\mathbf{c}_{1i}$ and $\mathbf{c}_{2i}$ $i = 1, \ldots, r$ be the numerator and denominator vectors of coefficients that define the linear combinations used in each of the r ratios (e.g. the ith ratio is $R_i = (\mathbf{c}_{1i}^\top \mathbf{q})/(\mathbf{c}_{2i}^\top \mathbf{q})$). Then, using the Delta method, an approximate covariance matrix for $[R_1, \ldots, R_r]^\top$ is

$$\mathbf{JSJ}^\top$$

where $\mathbf{J}$ is the Jacobian matrix whose ith row is the gradient vector of R_i.

Value

a list consisting of ratios (the ratio estimates) and cov (the estimated covariance matrix).

References

Prendergast, L. A., Dedduwakumara, D.S. & Staudte, R.G. (2024) rquest: An R package for hypothesis tests and confidence intervals for quantiles and summary measures based on quantiles, preprint, pages 1-13

Examples

```
# Create some data
set.seed(1234)
x <- rnorm(100)

# Compute the variance-covariance matrix for the ratio of third
# quartile and the median and the ratio of first quartile and the median.
coef1 <- matrix(c(0, 0, 1, 1, 0, 0), nrow = 2, byrow = TRUE)
coef2 <- matrix(c(0, 1, 0, 0, 1, 0), nrow = 2, byrow = TRUE)
qrcov(x, c(0.25, 0.5, 0.75), coef1 = coef1, coef2 = coef2)
```

---

rcv.test *Hypothesis Tests and Confidence Intervals for the Robust Coefficient of Variation*

Description

carry out hypothesis tests and obtain associated confidence intervals for robust versions of the coefficient of variation

Usage

```
rcv.test(
  x,
  y = NULL,
  numerator = "mad",
  quantile.type = 7,
  var.method = "qor",
  alternative = c("two.sided", "less", "greater"),
  conf.level = 0.95,
  true.rcv = 0,
  log.transf = TRUE,
  back.transf = TRUE,
  ...
)
```

Arguments

x	a numeric vector of data values.
y	an optional second numeric vector of data values for two-sample testing.
numerator	a character string specifying the numerator used in the robust coefficient of variation. Must be either "mad" or "iqr".
quantile.type	argument for the quantile function. Default is 7 so that output is consistent with the default quantile function use and with other functions such as IQR (see ?quantile for details).
var.method	method used to estimate the quantile density function when numerator = "iqr". Either "qor" (default) or "density".
alternative	a character string specifying the alternative hypothesis; must be one of "two.sided" (default), "less", or "greater".
conf.level	coverage level for the confidence interval.
true.rcv	the hypothesized value of the robust coefficient of variation, or of the difference/ratio in the two-sample case, depending on the transformation used.
log.transf	logical; if TRUE, the test and interval are computed on the log scale.
back.transf	logical; if TRUE, estimates and confidence intervals are back-transformed to the original rCV scale using exp.
...	additional arguments to be passed to density when numerator = "mad" and to q.test when numerator = "iqr".

Details

This function performs hypothesis tests and constructs confidence intervals for robust versions of the coefficient of variation (rCV). It can be used for either one-sample or two-sample inference.

The type of robust coefficient of variation is determined by the argument numerator. Two choices are available (Arachchige & Prendergast, 2022):

- "mad": based on the median absolute deviation (MAD). This is the default and most widely used choice.

- `"iqr"`: based on the interquartile range (IQR).

Let m denote the sample median. Then the MAD-based robust coefficient of variation is

$$\widehat{\mathrm{rCV}} = 1.4826 \times \frac{\mathrm{MAD}}{m}.$$

The multiplier 1.4826 makes the estimator comparable to the usual coefficient of variation under a Gaussian model.

If the IQR is preferred, then

$$\widehat{\mathrm{rCV}} = 0.75 \times \frac{\mathrm{IQR}}{m}.$$

The multiplier 0.75 is the corresponding scaling constant for the IQR-based version.

For `numerator = "mad"`, Wald-type confidence intervals are computed using an asymptotic variance estimate for the rCV. The default is to perform the calculations on the log scale (`log.transf = TRUE`) and then, when `back.transf = TRUE`, exponentiate the estimate and interval endpoints back to the original rCV scale.

For `numerator = "iqr"`, the rCV is treated as a ratio of linear combinations of quantiles and the function `q.test` is used to carry out the hypothesis test and compute the associated interval estimate. In this case, `var.method` controls how the quantile density is estimated.

Additional arguments in `...` are passed to `density` when `numerator = "mad"` and to `q.test` when `numerator = "iqr"`.

Value

an object of class `"htest"` containing the test result, estimate, null value, p-value, and confidence interval.

References

Arachchige, C. N. P. G., & Prendergast, L. A. (2022). Robust analogs to the coefficient of variation. *Journal of Applied Statistics*, **49**(2), 268–290.

See Also

`qcov` for covariance estimation of sample quantiles, `qrcov` for covariance estimation of ratios of linear combinations of quantiles, and `q.test` for hypothesis tests and confidence intervals based on ratios of linear combinations of quantiles.

Examples

```
set.seed(123)
x1 <- rnorm(100, 8)
x2 <- rnorm(120, 10)

# One-sample tests
rcv.test(x1)
rcv.test(x1, numerator = "iqr")

# Two-sample tests
```

```
rcv.test(x1, x2)
rcv.test(x1, x2, numerator = "iqr")
```

Index

fit.fkml, *14*

q.test, 2, *17*
qcov, *5*, 6, *9*, *14*, *17*
qden, *7*, 8
qineq, 10
qor, *5*, *7*, *9*, 12
qrcov, *5*, *7*, *9*, 14, *14*, *17*

rcv.test, 15