Package 'quasar'

November 18, 2025

Type Package

Title Valid Inference on Multiple Quantile Regressions

Version 0.1.0

Date 2025-11-07

Description The approach is based on the closed testing procedure to control familywise error rate in a strong sense.
The local tests implemented are Wald-type and rank-score.
The method is described in De Santis, et al., (2025), <doi:10.48550/arXiv.2511.07999>.

Depends quantreg, Matrix, MASS

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Angela Andreella [aut, cre],
Anna Vesely [ctb]

Maintainer Angela Andreella <angela.andreella@unive.it>

Repository CRAN

Date/Publication 2025-11-18 09:00:02 UTC

Contents

`closedTesting` *Closed testing for quantile regression*

Description

Applies the closed testing procedure to strongly control the familywise error rate (FWER) when testing the effect of a covariate of interest across multiple quantile regression models.

Usage

```
closedTesting(mod, X, tau = NULL, test = "rank-score", ...)
```

Arguments

`mod`	An object of class `rqs` returned by `rq`, representing the fitted quantile regression models.
`X`	A string indicating the covariate of interest.
`tau`	A numeric vector of quantiles of interest used in `mod`. If `NULL` (default), all quantiles from the `mod` object are considered.
`test`	Character. Type of test to be used. Options are `"rank-score"` and `"wald"`.
`...`	Additional arguments, see `rankTest`, `waldTest`.

Details

This procedure requires that the covariate of interest `X` is either numeric or, if categorical, has at most two levels. Multilevel categorical covariates are not supported and will trigger an error.

Value

An object of class `quasar` containing:

- `Quantile`: quantile level
- `Coefficient`: estimated coefficient
- `Statistic`: test statistic
- `p.value`: raw p-value
- `p.value.adjusted`: adjusted p-value from the closed testing procedure

Author(s)

Angela Andreella

References

Marcus, R., Eric, P., & Gabriel, K. R. (1976). On closed testing procedures with special reference to ordered analysis of variance. *Biometrika*, 63(3), 655–660.

Goeman, J. J., Hemerik, J., & Solari, A. (2021). Only closed testing procedures are admissible for controlling false discovery proportions. *The Annals of Statistics*, 49(2), 1218–1238.

See Also

rq, rankTest, waldTest

Examples

```
# Simulate data
set.seed(1234)
D <- simulateData(n = 100, gamma = 0.5, sigma.y = "1 + 2 * pmax(X, 0)")

# Quantile regressions at different levels
tau <- c(0.1, 0.25, 0.5, 0.75, 0.9)
mod <- quantreg::rq(y ~ X + Z1, tau = tau, data=D)

# Closed testing
res <- closedTesting(mod, X = "X")
res

# Summary and plot
summary(res, alpha = 0.1)
plot(res, alpha = 0.1, legend.position = "bottomright")
```

---

plot.quasar *Plot method for quasar objects*

Description

Produces a plot of a quasar object, typically returned by the closedTesting function. It shows the estimated coefficients by quantile level, highlighting statistically significant coefficients based on adjusted p-values.

Usage

```
## S3 method for class 'quasar'
plot(
  x,
  alpha = 0.05,
  legend.position = "topright",
  main = NULL,
  xlab = "Quantile level",
  ylab = "Coefficient",
  col.line = "darkgrey",
  col.sig = "darkred",
  col.nonsig = "darkgrey",
  pch.sig = 19,
  pch.nonsig = 17,
  show.legend = TRUE,
  ...
)
```

Arguments

x	An object of class quasar.
alpha	Significance level.
legend.position	Position of the legend.
main	Main plot title.
xlab	Label for the x-axis.
ylab	Label for the y-axis.
col.line	Color of the connecting line.
col.sig	Color for significant points.
col.nonsig	Color for non-significant points.
pch.sig	Point character for significant points.
pch.nonsig	Point character for non-significant points.
show.legend	Logical; whether to display a legend.
...	Additional graphical parameters passed to plot().

Value

A base R plot.

Author(s)

Anna Vesely

See Also

closedTesting

---

quasar-methods *Print and summary methods for quasar objects*

Description

These methods provide basic information about objects of class quasar, typically returned by the closedTesting function.

Usage

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

## S3 method for class 'quasar'
summary(object, ..., alpha = 0.05)
```

Arguments

x, object	An object of class quasar.
...	Additional arguments passed to other methods.
alpha	Significance level.

Value

The input object invisibly.

Author(s)

Anna Vesely

---

rankTest *Rank-score test for quantile regression*

Description

Performs the rank-score test for the covariate of interest X, at the quantiles defined in tau, using a fitted quantile regression model. The test evaluates the null hypothesis that the coefficient of X is equal to zero against a two-sided alternative, at each specified quantile level. Testing equality to a non-zero value is not yet implemented.

Usage

```
rankTest(mod, X, tau = NULL, full = FALSE, h = NULL, alpha = 0.05)
```

Arguments

mod	An object of class rqs returned by rq, representing the fitted quantile regression models.
X	A string indicating the covariate of interest.
tau	A numeric vector of quantiles of interest used in mod. If NULL (default), all quantiles from the mod object are considered.
full	Logical. If TRUE, the function returns the test statistics and corresponding p-values for all intersection hypotheses containing tau. If FALSE (default), only the results for the single hypotheses are returned.
h	A numeric value for the bandwidth.
alpha	A numeric value used for bandwidth estimation. Following Koenker (2005), it is typically set equal to the desired significance level.

Details

This procedure requires that the covariate of interest X is either numeric or, if categorical, has at most two levels. Multilevel categorical covariates are not supported and will trigger an error.

Value

A `data.frame` containing:

- `Quantiles.Set`: quantile levels
- `Statistic`: rank-score test statistic
- `p.value`: corresponding unadjusted p-value

Author(s)

Angela Andreella

References

Koenker, R. (2005). *Quantile Regression*. Cambridge University Press.

See Also

`rq`, `waldTest`

Examples

```
set.seed(1234)
D <- simulateData(n = 100, gamma = 0.5, sigma.y = "1 + 2 * pmax(X, 0)")

#Quantile regressions at different levels
tau <- c(0.1, 0.25, 0.5, 0.75, 0.9)
mod <- quantreg::rq(y ~ X + Z1, tau = tau, data=D)

# Rank test
rankTest(mod, X = "X")
```

`simulateData` *Simulate data*

Description

Simulates a main covariate `X`, a vector of additional covariates `Z`, and a response `y` drawn from the chosen distribution.

Usage

```
simulateData(n, beta = 0, gamma = 0, mu = 0, Sigma = NULL,
             sigma.y = 1, distribution = "normal", df = 5, seed = NULL)
```

Arguments

n	Integer. Number of observations.
beta	Numeric scalar. Effect of X.
gamma	Numeric vector. Effects of Z (length p - 1, where p = ncol(Sigma)).
mu	Numeric scalar. Intercept.
Sigma	Numeric p x p symmetric positive-definite covariance matrix for (X, Z). The first column corresponds to X, the remaining columns to Z1, Z2, ...
sigma.y	Either a numeric scalar or a one-sided expression/string (e.g., "0.3 * abs(X) + 0.1") defining the scale of y.
distribution	Character. One of "normal", "t", or "exponential". This is the distribution of y.
df	Numeric scalar > 0. Degrees of freedom for t-distribution.
seed	Numeric scalar > 0. Seed for random number generator.

Details

The response is generated as y = mu + beta * X + Z %*% gamma + error. The error term can be drawn from a normal distribution, scaled Student-t with df degrees of freedom, or a shifted exponential. Its standard deviation is defined by sigma.y: if numeric, a fixed scale is used; if a character expression, the scale can vary with X and/or Z.

Value

A data.frame with columns y, X, and Z1, ..., Zk.

Author(s)

Angela Andreella

Examples

```
set.seed(1)
p <- 3
Sigma <- diag(p)

# Normal
dat_n <- simulateData(n = 200, beta = 0.5, gamma = c(0.2,-0.1),
                      sigma.y = 0.5, distribution = "normal")

# Student-t
dat_t0 <- simulateData(n = 200, beta = 0.5, gamma = c(0.2,-0.1),
                       sigma.y = 0.5, distribution = "t", df = 7)
# Exponential
dat_e <- simulateData(n = 200, beta = 0.5, gamma = c(0.2,-0.1),
                      sigma.y = "0.3 * abs(X) + 0.1", distribution = "exponential")
```

`waldTest` *Wald-type test for quantile regression*

Description

Performs the Wald-type test for the covariate of interest `X`, at the quantiles defined in `tau`, using a fitted quantile regression model. The test evaluates the null hypothesis that the coefficient of `X` is equal to a given value `beta` against a two-sided alternative, at each specified quantile level.

Usage

```
waldTest(mod, X, tau = NULL, full = FALSE, h = NULL, beta = 0, alpha = 0.05)
```

Arguments

`mod`	An object of class `rqs` returned by `rq`, representing the fitted quantile regression models.
`X`	A string indicating the covariate of interest.
`tau`	A numeric vector of quantiles of interest used in `mod`. If `NULL` (default), all quantiles from the `mod` object are considered.
`full`	Logical. If `TRUE`, the function returns the test statistics and corresponding p-values for all intersection hypotheses containing `tau`. If `FALSE` (default), only the results for the single hypotheses are returned.
`h`	A numeric value for the bandwidth.
`beta`	Numeric value of the parameter of interest under the null hypothesis.
`alpha`	A numeric value used for bandwidth estimation. Following Koenker (2005), it is typically set equal to the desired significance level.

Details

This procedure requires that the covariate of interest `X` is either numeric or, if categorical, has at most two levels. Multilevel categorical covariates are not supported and will trigger an error.

Value

A `data.frame` containing:

- `Quantiles.Set`: quantile levels
- `Statistic`: Wald-type test statistic
- `p.value`: corresponding unadjusted p-value

Author(s)

Angela Andreella

References

Koenker, R. (2005). *Quantile Regression*. Cambridge University Press.

See Also

rq, rankTest

Examples

```
set.seed(1234)
D <- simulateData(n = 100, gamma = 0.5, sigma.y = "1 + 2 * pmax(X, 0)")

#Quantile regressions at different levels
tau <- c(0.1, 0.25, 0.5, 0.75, 0.9)
mod <- quantreg::rq(y ~ X + Z1, tau = tau, data=D)

# Wald test
waldTest(mod, X = "X")
```

Index