Package 'QuantileOnQuantile'

February 8, 2026

Type Package

Title Quantile-on-Quantile Regression Analysis

Version 1.0.3

Description Implements the Quantile-on-Quantile (QQ) regression methodology developed by Sim and Zhou (2015) <doi:10.1016/j.jbankfin.2015.01.013>. QQ regression estimates the effect that quantiles of one variable have on quantiles of another, capturing the dependence between distributions. The package provides functions for QQ regression estimation, 3D surface visualization with 'MATLAB'-
style color schemes ('Jet', 'Viridis', 'Plasma'), heatmaps, contour plots, and quantile correlation analysis. Uses 'quantreg' for quantile regression and 'plotly' for interactive visualizations. Particularly useful for examining relationships between financial variables, oil prices, and stock returns under different market conditions.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.1

Depends R (>= 3.5.0)

Author Merwan Roudane [aut, cre, cph],
Nicholas Sim [ctb] (Original methodology developer),
Hongtao Zhou [ctb] (Original methodology developer)

Maintainer Merwan Roudane <merwanroudane920@gmail.com>

Imports quantreg (>= 5.0), plotly (>= 4.0.0), stats, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0), htmlwidgets

VignetteBuilder knitr

URL https://github.com/merwanroudane/qq

BugReports https://github.com/merwanroudane/qq/issues

Config/testthat/edition 3

NeedsCompilation no

Repository CRAN

Date/Publication 2026-02-08 16:40:02 UTC

Contents

`plot.qq_regression` *Plot Method for qq_regression Objects*

Description

Creates a 3D surface plot of qq_regression results.

Usage

```
## S3 method for class 'qq_regression'
plot(x, type = "coefficient", colorscale = "Jet", ...)
```

Arguments

`x`	An object of class "qq_regression".
`type`	Character string. Type of plot: "coefficient", "rsquared", or "pvalue". Default is "coefficient".
`colorscale`	Character string. Color scale to use. One of "Jet", "BlueRed", "Viridis", or "Plasma". Default is "Jet".
`...`	Additional arguments passed to `plot_qq_3d`.

Value

A plotly object. This function is called for its side effect of creating an interactive 3D visualization.

plot_qq_3d *3D Surface Plot for QQ Regression*

Description

Creates an interactive 3D surface plot of QQ regression results using plotly. Supports multiple color scales including MATLAB-style Jet.

Usage

```
plot_qq_3d(
  qq_result,
  type = c("coefficient", "rsquared", "pvalue"),
  colorscale = c("Jet", "BlueRed", "Viridis", "Plasma"),
  show_contour = TRUE,
  x_label = "X Quantile (tau)",
  y_label = "Y Quantile (theta)",
  title = NULL,
  aspect_ratio = c(1, 1, 0.7),
  camera = list(eye = list(x = 1.4, y = 1.7, z = 1.2))
)
```

Arguments

qq_result	An object of class "qq_regression" or a data frame with columns: y_quantile, x_quantile, coefficient, r_squared, p_value.
type	Character string. Type of values to plot: "coefficient" (default), "rsquared", or "pvalue".
colorscale	Character string. Color scale to use. Options are: • "Jet" - MATLAB-style rainbow (default) • "BlueRed" - Blue to red diverging scale • "Viridis" - Viridis perceptually uniform scale • "Plasma" - Plasma perceptually uniform scale
show_contour	Logical. If TRUE, shows gridlines on the surface. Default is TRUE.
x_label	Character string. Label for x-axis. Default is "X Quantile (tau)".
y_label	Character string. Label for y-axis. Default is "Y Quantile (theta)".
title	Character string. Plot title. If NULL, an automatic title is generated.
aspect_ratio	Numeric vector of length 3. Aspect ratio for x, y, z axes. Default is c(1, 1, 0.7).
camera	List. Camera position settings for plotly. Default provides a good viewing angle.

Details

The function creates an interactive 3D surface plot where:

- X-axis: Quantiles of the independent variable (tau)
- Y-axis: Quantiles of the dependent variable (theta)
- Z-axis: Estimated coefficients, R-squared, or p-values

The Jet colorscale mimics MATLAB's default color scheme for surface plots, transitioning from blue through cyan, green, yellow to red.

Value

A plotly object representing the 3D surface plot.

Examples

```
# Generate example data
set.seed(42)
n <- 200
x <- rnorm(n)
y <- 0.5 * x + rnorm(n, sd = 0.5)

# Run QQ regression
result <- qq_regression(y, x, verbose = FALSE)


# Create 3D surface plot of coefficients
plot_qq_3d(result, type = "coefficient", colorscale = "Jet")

# Create plot of R-squared values
plot_qq_3d(result, type = "rsquared", colorscale = "Viridis")
```

---

plot_qq_contour *Contour Plot for QQ Regression*

Description

Creates an interactive contour plot of QQ regression coefficients using plotly.

Usage

```
plot_qq_contour(
  qq_result,
  type = c("coefficient", "rsquared", "pvalue"),
  colorscale = "Jet",
  show_labels = TRUE,
```

```
  x_label = "X Variable Quantiles",
  y_label = "Y Variable Quantiles",
  title = NULL
)
```

Arguments

qq_result	An object of class "qq_regression" or a data frame with columns: y_quantile, x_quantile, coefficient.
type	Character string. Type of values to plot: "coefficient" (default), "rsquared", or "pvalue".
colorscale	Character string. Color scale to use. Default is "Jet".
show_labels	Logical. If TRUE, shows contour labels. Default is TRUE.
x_label	Character string. Label for x-axis. Default is "X Variable Quantiles".
y_label	Character string. Label for y-axis. Default is "Y Variable Quantiles".
title	Character string. Plot title. If NULL, an automatic title is generated.

Value

A plotly object representing the contour plot.

Examples

```
# Generate example data
set.seed(42)
n <- 200
x <- rnorm(n)
y <- 0.5 * x + rnorm(n, sd = 0.5)

# Run QQ regression
result <- qq_regression(y, x, verbose = FALSE)


# Create contour plot
plot_qq_contour(result)
```

---

plot_qq_correlation *Quantile Correlation Heatmap*

Description

Creates an interactive heatmap showing the correlation between quantiles of two variables. Uses a blue-red diverging color scale similar to Python's seaborn style.

Usage

```
plot_qq_correlation(
  y,
  x,
  quantiles = seq(0.1, 0.9, by = 0.1),
  x_label = "X Variable Quantiles",
  y_label = "Y Variable Quantiles",
  title = "Quantile Correlation Heatmap",
  show_annotations = TRUE
)
```

Arguments

y	Numeric vector. The first variable.
x	Numeric vector. The second variable.
quantiles	Numeric vector. Quantiles to compute correlations for. Default is `seq(0.1, 0.9, by = 0.1)`.
x_label	Character string. Label for x-axis. Default is "X Variable Quantiles".
y_label	Character string. Label for y-axis. Default is "Y Variable Quantiles".
title	Character string. Plot title. Default is "Quantile Correlation Heatmap".
show_annotations	Logical. If `TRUE`, shows correlation values as text annotations. Default is `TRUE`.

Details

This function computes correlations between binary indicators of whether observations fall below each quantile threshold. The resulting heatmap shows how the relationship between variables varies across different parts of their distributions.

Value

A plotly object representing the correlation heatmap.

Examples

```
# Generate example data
set.seed(42)
n <- 200
x <- rnorm(n)
y <- 0.5 * x + rnorm(n, sd = 0.5)


# Create correlation heatmap
plot_qq_correlation(y, x)
```

`plot_qq_heatmap` *Heatmap for QQ Regression*

Description

Creates an interactive heatmap of QQ regression results using plotly.

Usage

```
plot_qq_heatmap(
  qq_result,
  type = c("coefficient", "rsquared", "pvalue"),
  colorscale = NULL,
  x_label = "X Variable Quantiles",
  y_label = "Y Variable Quantiles",
  title = NULL,
  zmin = NULL,
  zmax = NULL
)
```

Arguments

`qq_result`	An object of class "qq_regression" or a data frame with columns: y_quantile, x_quantile, coefficient, r_squared, p_value.
`type`	Character string. Type of values to plot: "coefficient" (default), "rsquared", or "pvalue".
`colorscale`	Character string. Color scale to use. Options are: • "Viridis" - Viridis scale (default for coefficient) • "Plasma" - Plasma scale (default for rsquared) • "Jet" - MATLAB-style rainbow (default for pvalue) • "BlueRed" - Blue to red diverging scale
`x_label`	Character string. Label for x-axis. Default is "X Variable Quantiles".
`y_label`	Character string. Label for y-axis. Default is "Y Variable Quantiles".
`title`	Character string. Plot title. If `NULL`, an automatic title is generated.
`zmin`	Numeric. Minimum value for color scale. If `NULL`, automatically determined.
`zmax`	Numeric. Maximum value for color scale. If `NULL`, automatically determined.

Value

A plotly object representing the heatmap.

Examples

```
# Generate example data
set.seed(42)
n <- 200
x <- rnorm(n)
y <- 0.5 * x + rnorm(n, sd = 0.5)

# Run QQ regression
result <- qq_regression(y, x, verbose = FALSE)


# Create coefficient heatmap
plot_qq_heatmap(result, type = "coefficient")

# Create R-squared heatmap
plot_qq_heatmap(result, type = "rsquared", colorscale = "Plasma")

# Create p-value heatmap
plot_qq_heatmap(result, type = "pvalue", colorscale = "Jet")
```

---

print.qq_regression *Print Method for qq_regression Objects*

Description

Prints a summary of a qq_regression object.

Usage

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

Arguments

x	An object of class "qq_regression".
...	Additional arguments (currently ignored).

Value

The input object x is returned invisibly. This function is called for its side effect of printing the QQ regression summary to the console.

`qq_colorscales` *Available Color Scales for QQ Regression Plots*

Description

Returns information about the available color scales for 3D surface plots and heatmaps in the QuantileOnQuantile package.

Usage

```
qq_colorscales(show_preview = TRUE)
```

Arguments

`show_preview`	Logical. If `TRUE`, prints information about each color scale. Default is `TRUE`.

Details

The following color scales are available:

Jet MATLAB-style rainbow colorscale. Transitions from blue through cyan, green, yellow to red. Best for general visualization where the full range of values is important.

BlueRed Diverging colorscale from blue to red. Also known as "COVID style" in some contexts. Best for data with a meaningful center point (e.g., zero for coefficients).

Viridis Perceptually uniform colorscale. Designed to be colorblind-friendly and print well in grayscale.

Plasma Another perceptually uniform colorscale. Similar benefits to Viridis but with a different color palette.

Value

A character vector of available color scale names, invisibly.

Examples

```
# Show available color scales
qq_colorscales()

# Get scales without printing
scales <- qq_colorscales(show_preview = FALSE)
print(scales)
```

qq_export *Export QQ Results to CSV*

Description

Exports QQ regression results to a CSV file.

Usage

```
qq_export(qq_result, file, digits = 4)
```

Arguments

qq_result	An object of class "qq_regression".
file	Character string. Path to the output file.
digits	Integer. Number of decimal places for rounding. Default is 4.

Value

Invisible NULL. Called for its side effect of writing a CSV file.

Examples

```
# Generate example data
set.seed(42)
n <- 200
x <- rnorm(n)
y <- 0.5 * x + rnorm(n, sd = 0.5)

# Run QQ regression
result <- qq_regression(y, x, verbose = FALSE)

# Export to CSV
qq_export(result, file.path(tempdir(), "qq_results.csv"))
```

qq_regression *Quantile-on-Quantile Regression*

Description

Performs Quantile-on-Quantile (QQ) regression analysis as described in Sim and Zhou (2015). This approach estimates the effect that quantiles of an independent variable have on quantiles of a dependent variable, capturing the dependence between their distributions.

Usage

```
qq_regression(
  y,
  x,
  y_quantiles = seq(0.05, 0.95, by = 0.05),
  x_quantiles = seq(0.05, 0.95, by = 0.05),
  min_obs = 10,
  se_method = "boot",
  verbose = TRUE
)
```

Arguments

y	Numeric vector. The dependent variable.
x	Numeric vector. The independent variable.
y_quantiles	Numeric vector. Quantiles of the dependent variable to estimate. Default is seq(0.05, 0.95, by = 0.05).
x_quantiles	Numeric vector. Quantiles of the independent variable to condition on. Default is seq(0.05, 0.95, by = 0.05).
min_obs	Integer. Minimum number of observations required for quantile regression. Default is 10.
se_method	Character string. Method for computing standard errors in quantile regression. One of "boot" (bootstrap), "nid" (no i.i.d. assumption), "iid", or "ker" (kernel). Default is "boot".
verbose	Logical. If TRUE, prints progress messages. Default is TRUE.

Details

The QQ approach combines quantile regression with local linear regression to estimate the effect of the τ-quantile of x on the θ-quantile of y.

The method proceeds as follows:

1. For each x_quantile (τ), subset the data where x <= quantile(x, τ)
2. For each y_quantile (θ), perform quantile regression of y on x
3. Extract coefficients, standard errors, t-values, p-values
4. Compute pseudo R-squared based on check function

If the subset has fewer than min_obs observations, a correlation-based approach is used instead.

Value

An object of class "qq_regression" containing:

- results - Data frame with columns: y_quantile, x_quantile, coefficient, std_error, t_value, p_value, r_squared
- y_quantiles - Quantiles of y used

- `x_quantiles` - Quantiles of x used
- `n_obs` - Number of complete observations
- `call` - The matched call
- `method` - Method used ("quantile_regression" or "correlation")

References

Sim, N. and Zhou, H. (2015). Oil Prices, US Stock Return, and the Dependence Between Their Quantiles. *Journal of Banking & Finance*, 55, 1-12. doi:10.1016/j.jbankfin.2015.01.013

Examples

```
# Generate example data
set.seed(42)
n <- 200
x <- rnorm(n)
y <- 0.5 * x + rnorm(n, sd = 0.5)

# Run QQ regression with default quantiles
result <- qq_regression(y, x, verbose = FALSE)

# Print summary
print(result)

# View first few results
head(result$results)


# Run with custom quantiles
result2 <- qq_regression(y, x,
                         y_quantiles = seq(0.1, 0.9, by = 0.1),
                         x_quantiles = seq(0.1, 0.9, by = 0.1),
                         verbose = FALSE)
```

---

qq_statistics *Summary Statistics for QQ Regression*

Description

Computes comprehensive summary statistics for QQ regression results.

Usage

```
qq_statistics(qq_result, alpha = 0.05)
```

Arguments

qq_result	An object of class "qq_regression".
alpha	Numeric. Significance level for counting significant results. Default is 0.05.

Value

A data frame with summary statistics including:

- Mean, median, min, max of coefficients
- Mean, median, min, max of R-squared
- Number of significant results at specified alpha
- Total number of results

Examples

```
# Generate example data
set.seed(42)
n <- 200
x <- rnorm(n)
y <- 0.5 * x + rnorm(n, sd = 0.5)

# Run QQ regression
result <- qq_regression(y, x, verbose = FALSE)

# Get summary statistics
stats <- qq_statistics(result)
print(stats)
```

---

qq_to_matrix *Convert QQ Results to Matrix*

Description

Converts QQ regression results to a matrix format suitable for base R plotting or export.

Usage

```
qq_to_matrix(qq_result, type = c("coefficient", "rsquared", "pvalue"))
```

Arguments

qq_result	An object of class "qq_regression" or a data frame.
type	Character string. Type of values: "coefficient", "rsquared", or "pvalue". Default is "coefficient".

Value

A matrix with y_quantiles as rows and x_quantiles as columns.

Examples

```
# Generate example data
set.seed(42)
n <- 200
x <- rnorm(n)
y <- 0.5 * x + rnorm(n, sd = 0.5)

# Run QQ regression
result <- qq_regression(y, x, verbose = FALSE)

# Convert to matrix
coef_matrix <- qq_to_matrix(result, type = "coefficient")
print(dim(coef_matrix))
```

---

summary.qq_regression *Summary Method for qq_regression Objects*

Description

Provides a detailed summary of a qq_regression object.

Usage

```
## S3 method for class 'qq_regression'
summary(object, ...)
```

Arguments

object	An object of class "qq_regression".
...	Additional arguments (currently ignored).

Value

A list containing summary statistics, returned invisibly. The function is called for its side effect of printing a detailed summary to the console.

Index