Package 'mqqcause'

June 1, 2026

Type Package

Title Multivariate Quantile-on-Quantile Granger Causality

Version 1.0.0

Description Implements bivariate and Multivariate Quantile-on-Quantile Granger causality tests building on the Quantile-on-Quantile regression framework of Sim and Zhou (2015) <doi:10.1016/j.jbankfin.2015.01.013> and the quantile Granger causality test of Troster (2018) <doi:10.1080/07474938.2016.1172400>. The bivariate test estimates the local-linear slope in the quantile regression of y_t on lagged x_t with lagged y_t as control, using Gaussian kernel weights, and tests it against zero by paired bootstrap. The multivariate (conditional) test additionally conditions on a set of moderators Z and optional x times Z interaction terms, in the spirit of Sinha, Ghosh, Hussain, Nguyen and Das (2023) <doi:10.1016/j.eneco.2023.107021>. A Sup-Wald summary across the quantile grid is also provided. Heatmaps and 3D surfaces default to the 'MATLAB' 'Parula' colour map.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.1

Depends R (>= 3.5.0)

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

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

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

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

Config/testthat/edition 3

NeedsCompilation no

Author Merwan Roudane [aut, cre, cph],
Avik Sinha [ctb] (Original multivariate QQ causality methodology),
Nicholas Sim [ctb] (Original QQ regression methodology),
Hongtao Zhou [ctb] (Original QQ regression methodology)

Maintainer Merwan Roudane <merwanroudane920@gmail.com>

Repository CRAN

Date/Publication 2026-06-01 09:00:31 UTC

Contents

`mqqcause-package` *mqqcause: Multivariate Quantile-on-Quantile Granger Causality*

Description

Implements bivariate Quantile-on-Quantile Granger causality and the multivariate (conditional) version with a vector of moderators and optional interaction terms. A Sup-Wald summary across the (theta, tau) grid is also provided (Troster, 2018).

Author(s)

Dr Merwan Roudane <merwanroudane920@gmail.com>
GitHub: https://github.com/merwanroudane/qqcaus

`parula_colors` *MATLAB-style colour palettes for mqqcause*

Description

Colour palettes used by mqqcause plotting functions. The default scale is MATLAB Parula.

Usage

```
parula_colors(n = 256)
matlab_jet_colors(n = 256)
turbo_colors(n = 256)
bluered_colors(n = 256)
sinha_colors(n = 256)
mqqcause_palette(cols, n_breaks = 32)
resolve_colorscale(name = "Parula", n_breaks = 32)
mqqcause_colorscales(show_preview = TRUE)
```

Arguments

n	Number of colours.
cols	Character vector of hex colours.
n_breaks	Stops for the plotly list.
name	Scale name (Parula by default).
show_preview	Print descriptions.

Value

Character vector or list.

Examples

```
parula_colors(8)
matlab_jet_colors(8)
turbo_colors(8)
bluered_colors(8)
sinha_colors(8)
mqqcause_colorscales(show_preview = FALSE)
```

---

plot_qq_causality_3d *Visualisations for QQ Causality Results*

Description

3D surface, heatmap, contour and significance-only heatmap visualisations of QQ Granger-causality results, defaulting to MATLAB Parula.

Usage

```
plot_qq_causality_3d(qq_result, value = "t_value",
                     colorscale = "Parula", show_contour = TRUE,
                     x_label = "X Quantile (tau)",
                     y_label = "Y Quantile (theta)", title = NULL)

plot_qq_causality_heatmap(qq_result, value = "t_value",
                          colorscale = "Parula", show_stars = TRUE,
                          x_label = "X Quantile (tau)",
                          y_label = "Y Quantile (theta)", title = NULL)

plot_qq_causality_contour(qq_result, value = "t_value",
                          colorscale = "Parula",
                          x_label = "X Quantile (tau)",
                          y_label = "Y Quantile (theta)", title = NULL)

plot_significance_heatmap(qq_result, colorscale = "Parula")
```

Arguments

qq_result	A qq_causality or mqq_causality object.
value	Column to plot.
colorscale	Default "Parula".
show_contour, show_stars, x_label, y_label, title	Plot options.

Value

A plotly object.

Examples

```
## Small toy example -- auto-tested.
set.seed(1); n <- 80
x <- rnorm(n); y <- 0.3 * c(0, x[-n]) + rnorm(n, sd = 0.4)
fit <- qq_causality(x, y,
                    y_quantiles = c(0.25, 0.5, 0.75),
                    x_quantiles = c(0.25, 0.5, 0.75),
                    n_boot = 10, verbose = FALSE)
p1 <- plot_qq_causality_heatmap(fit, value = "t_value", show_stars = TRUE)
p2 <- plot_qq_causality_3d(fit, value = "t_value")
p3 <- plot_qq_causality_contour(fit, value = "t_value")
p4 <- plot_significance_heatmap(fit)
```

---

qq_causality *Quantile-on-Quantile Granger Causality (Bivariate and Multivariate)*

Description

Tests whether the tau-quantile of x_{t-1} Granger-causes the theta-quantile of y_t. The multivariate variant conditions on a set of moderators Z and optional $x \cdot Z$ interaction terms (the conditional-causality test that under-pins Sinha et al., 2024).

Usage

```
qq_causality(x, y,
             y_quantiles = seq(0.05, 0.95, by = 0.05),
             x_quantiles = seq(0.05, 0.95, by = 0.05),
             bandwidth = 0.05, n_boot = 200,
             cdf_based_kernel = TRUE,
             cause_name = "X", effect_name = "Y",
             verbose = TRUE, seed = 42)

mqq_causality(x, y, moderators = list(),
              y_quantiles = seq(0.05, 0.95, by = 0.05),
              x_quantiles = seq(0.05, 0.95, by = 0.05),
```

```
                bandwidth = 0.05, n_boot = 200, interactions = TRUE,
                cdf_based_kernel = TRUE,
                cause_name = "X", effect_name = "Y",
                verbose = TRUE, seed = 42)

sup_wald(qq_result, alpha = 0.05)
qq_causality_to_matrix(qq_result, value = "t_value")
```

Arguments

x, y	Numeric vectors (cause and effect).
moderators	Named list of numeric vectors (the conditioning set Z).
y_quantiles, x_quantiles	Numeric vectors of quantile levels in (0, 1).
bandwidth	Kernel bandwidth on the empirical-CDF scale.
n_boot	Bootstrap replicates.
interactions	Include x*Z cross-terms (mqq_causality).
cdf_based_kernel	Use CDF-distance kernel.
cause_name, effect_name	Variable names for printing.
verbose	Print progress.
seed	RNG seed.
qq_result	A qq_causality / mqq_causality object.
value	Column to pivot.
alpha	Significance threshold.

Value

For qq_causality and mqq_causality, an object of class "qq_causality" or "mqq_causality". sup_wald returns a list; qq_causality_to_matrix returns a numeric matrix.

References

Sim, N., Zhou, H. (2015). *Journal of Banking and Finance*, 55, 1-12.

Troster, V. (2018). Testing for Granger-causality in Quantiles. *Econometric Reviews*, 37(8), 850-866.

Examples

```
set.seed(1); n <- 200
x <- rnorm(n); z <- rnorm(n)
y <- 0.3 * c(0, x[-n]) + 0.2 * c(0, z[-n]) + rnorm(n, sd = 0.4)
fit <- mqq_causality(x, y, list(Z = z),
                     y_quantiles = c(0.25, 0.5, 0.75),
                     x_quantiles = c(0.25, 0.5, 0.75),
```

```
                      n_boot = 30, verbose = FALSE)
print(fit)
sup_wald(fit)
```

---

qq_weights *Numerical building blocks*

Description

Gaussian kernel, QQ weights, and weighted quantile regression via `quantreg::rq.wfit`. Exposed for advanced users.

Usage

```
qq_weights(x, tau, h = 0.05, cdf_based = TRUE)
gaussian_kernel(u)
weighted_qr(y, X, tau, weights = NULL)
```

Arguments

x, y, u	Numeric vectors.
X	Numeric design matrix.
tau	Quantile in (0, 1).
h	Bandwidth.
cdf_based	Use empirical-CDF distance kernel.
weights	Optional numeric weights.

Value

Numeric vector or list.

Examples

```
set.seed(1)
x <- rnorm(50)
w <- qq_weights(x, tau = 0.5)
sum(w)            # weights sum to length(x)
k <- gaussian_kernel(seq(-3, 3, by = 1))
X <- cbind(1, x)
y <- 0.5 * x + rnorm(50, sd = 0.3)
fit <- weighted_qr(y, X, tau = 0.5, weights = w)
fit$coef
```

Index