Package: wqrr (via r-universe)

June 8, 2026

Type Package

Title Wavelet Quantile Regression Toolbox

Version 1.0.0

Description A comprehensive toolbox for wavelet-domain quantile analyses of bivariate and multivariate time series. Provides Wavelet Quantile Regression and Multivariate Wavelet Quantile Regression after Adebayo and Ozkan (2024) <doi:10.1016/j.jclepro.2024.140832>, Wavelet Quantile-on-Quantile regression with bootstrap p-values extending Sim and Zhou (2015) <doi:10.1016/j.jbankfin.2015.01.013>, the nonparametric Causality-in-Quantiles test of Balcilar, Gupta and Pierdzioch (2016) <doi:10.1016/j.resourpol.2016.04.004> together with its wavelet variant, Wavelet Quantile Mediation and Moderation, Wavelet Quantile Correlation, and a wavelet-based nonparametric Quantile Density estimator. The Maximal Overlap Discrete Wavelet Transform (MODWT) decomposition is performed via 'waveslim' and Short / Medium / Long band aggregation is supported throughout. For plain Quantile-on-Quantile regression see the companion CRAN package 'QuantileOnQuantile'. All interactive 3D surfaces, heatmaps and contour plots 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), waveslim (>= 1.8), plotly (>= 4.0.0), stats, utils, grDevices

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

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

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

Config/testthat/edition 3

NeedsCompilation no

Author Merwan Roudane [aut, cre, cph], Nicholas Sim [ctb] (Original Quantile-on-Quantile methodology), Hongtao Zhou [ctb] (Original Quantile-on-Quantile methodology), Tomiwa Sunday Adebayo [ctb] (Wavelet Quantile Regression methodology), Mehmet Balcilar [ctb] (Nonparametric quantile causality methodology)

Maintainer Merwan Roudane <merwanroudane920@gmail.com>

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

Date/Publication 2026-06-08 16:33:01 UTC

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

RemoteRef HEAD

RemoteSha 1fdf62437d3f16c12bc188589e664118d51a7cd4

Contents

`wqrr-package` *wqrr: Wavelet Quantile Regression Toolbox*

Description

Eight wavelet-quantile estimators in one consistent interface: Wavelet Quantile Regression, Multivariate WQR, Wavelet Quantile-on-Quantile Regression with p-values, nonparametric Causality-in-Quantiles and its wavelet variant, Wavelet Quantile Mediation and Moderation, Wavelet Quantile Correlation, and a wavelet nonparametric Quantile Density estimator.

Main functions

- wavelet_qr - Wavelet Quantile Regression.
- multivariate_wqr - Multivariate WQR.
- wavelet_qqr - Wavelet QQR with p-values.
- np_quantile_causality - Causality-in-Quantiles.
- wavelet_np_causality - Wavelet variant.
- wavelet_mediation - Mediation / moderation.
- wavelet_quantile_correlation - WQC.
- wavelet_quantile_density - Quantile density.
- parula_colors, wqrr_colorscales - MATLAB-style colour palettes.

See also

For plain Quantile-on-Quantile regression see the companion CRAN package QuantileOnQuantile.

Author(s)

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

---

modwt_mra *MODWT decomposition and band aggregation*

Description

Maximal Overlap Discrete Wavelet Transform (MODWT) MRA via waveslim::mra, plus a helper to combine details into Short / Medium / Long bands.

Usage

```
modwt_mra(x, wavelet = "la8", level = NULL, boundary = "periodic")
aggregate_bands(details, band_spec = NULL)
```

Arguments

x	Numeric vector.
wavelet	Wavelet filter (e.g. "la8", "la16", "d4", "haar").
level	Integer or NULL (auto).
boundary	"periodic" or "reflection".
details	List of detail series from modwt_mra()$details.
band_spec	Named list of integer index vectors.

Value

For `modwt_mra`: a list with elements `details` and `smooth`. For `aggregate_bands`: a named list of aggregated band series.

Examples

```
set.seed(1); x <- cumsum(rnorm(128))
dec <- modwt_mra(x, "la8", level = 4)
bands <- aggregate_bands(dec$details)
sapply(bands, length)
```

`multivariate_wqr` *Multivariate Wavelet Quantile Regression (MWQR)*

Description

MWQR aggregates MODWT details into Short / Medium / Long bands and runs a multiple quantile regression of y_b on all regressors at each quantile within each band.

Usage

```
multivariate_wqr(y, X_list,
                 quantiles = c(0.05, 0.10, 0.25, 0.50,
                               0.75, 0.90, 0.95),
                 wavelet = "la8", J = 5, bands = TRUE,
                 dep_name = "Y", verbose = TRUE)
```

Arguments

`y`	Numeric response.
`X_list`	Named list of numeric regressors.
`quantiles`	Numeric vector of tau in (0, 1).
`wavelet`, `J`, `bands`, `verbose`	See `wavelet_qr`.
`dep_name`	Character label for the dependent variable.

Value

An object of class `"mwqr_result"`.

Examples

```
set.seed(1); n <- 200
x1 <- cumsum(rnorm(n)); x2 <- cumsum(rnorm(n))
y  <- 0.4 * x1 - 0.2 * x2 + rnorm(n, sd = 0.5)
fit <- multivariate_wqr(y, list(X1 = x1, X2 = x2),
                        quantiles = c(0.25, 0.5, 0.75),
                        wavelet = "la8", J = 4, verbose = FALSE)
```

`np_quantile_causality` *Nonparametric Causality-in-Quantiles (and Wavelet Variant)*

Description

Nonparametric quantile Granger-causality test. Standard critical values are 1.96 (5%) and 1.645 (10%).

Usage

```
np_quantile_causality(x, y,
                      test_type = c("mean", "variance"),
                      q = seq(0.05, 0.95, by = 0.05),
                      bandwidth = NULL)

wavelet_np_causality(x, y,
                     test_type = c("mean", "variance"),
                     q = seq(0.05, 0.95, by = 0.05),
                     wavelet = "la8", J = 5,
                     bands = TRUE, bandwidth = NULL,
                     verbose = TRUE)
```

Arguments

`x`, `y`	Numeric vectors of equal length.
`test_type`	`"mean"` (moment = 1) or `"variance"` (moment = 2).
`q`	Numeric vector of tau in (0, 1).
`bandwidth`	Numeric or `NULL` (Silverman plug-in).
`wavelet`, `J`, `bands`, `verbose`	See `wavelet_qr`.

Value

An object of class `"causality_result"` or `"wavelet_causality_result"`.

References

Balcilar, M., Gupta, R., Pierdzioch, C. (2016). <doi:10.1016/j.resourpol.2016.04.004>.

Examples

```
set.seed(1); n <- 200
x <- rnorm(n); y <- 0.3 * c(0, x[-n]) + rnorm(n, sd = 0.5)
fit <- np_quantile_causality(x, y, q = c(0.25, 0.5, 0.75))
```

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

Description

Colour palettes used by wqrr plots, with MATLAB Parula as the default. The Parula stops are reproduced exactly from MathWorks' 64 RGB values.

Usage

```
parula_colors(n = 256)
matlab_jet_colors(n = 256)
turbo_colors(n = 256)
bluered_colors(n = 256)
sinha_colors(n = 256)
green_orange_red_colors(n = 256)
green_yellow_red_colors(n = 256)
wqrr_palette(cols, n_breaks = 32)
resolve_colorscale(name = "Parula", n_breaks = 32)
wqrr_colorscales(show_preview = TRUE)
```

Arguments

`n`	Number of interpolated 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 plotly colorscale list, depending on the function.

Examples

```
parula_colors(8)
wqrr_colorscales(show_preview = FALSE)
```

plot_wqr_heatmap *Visualisations for wqrr Results*

Description

Interactive 3D surfaces, heatmaps and contour plots for all wqrr estimators. All default to the MATLAB 'Parula' colour map (see parula_colors and wqrr_colorscales).

Usage

```
plot_qq_3d(qq_result, value = "coefficient",
           colorscale = "Parula", show_contour = TRUE,
           x_label = "X Quantile (tau)",
           y_label = "Y Quantile (theta)", title = NULL)

plot_qq_heatmap(qq_result, value = "coefficient",
                colorscale = "Parula", show_stars = FALSE,
                x_label = "X Quantile (tau)",
                y_label = "Y Quantile (theta)", title = NULL)

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

plot_wqr_heatmap(wqr_result, value = "beta",
                 colorscale = "GreenOrangeRed",
                 title = "Wavelet Quantile Regression",
                 x_label = "Quantiles",
                 y_label = "Time Horizons")

plot_wqqr_surface(wqqr_result, colorscale = "Parula", title = NULL)

plot_wqqr_pvalue_heatmap(wqqr_result, alpha = 0.05,
                         colorscale = "Sinha", title = NULL)

plot_wqr_vs_wqqr(wqqr_result, ...)

plot_causality(causality_result, cv = 1.96, title = NULL)

plot_wavelet_causality(wcr, colorscale = "GreenOrangeRed", title = NULL)

plot_correlation_heatmap(x, y,
                         quantiles = seq(0.1, 0.9, by = 0.1),
                         colorscale = "RdBu",
                         title = "Quantile Correlation Heatmap")
```

```
plot_wqc_heatmap(wqc_result, colorscale = "Parula",
                 title = "Wavelet Quantile Correlation")

plot_quantile_density(qd_result, ...)

plot_mediation_panel(med_result, colorscale = "GreenYellowRed")
```

Arguments

qq_result, wqr_result, wqqr_result, causality_result, wcr, wqc_result, qd_result, med_result	Result objects from the respective estimators.
x, y, quantiles	Inputs for plot_correlation_heatmap.
value, colorscale, title, x_label, y_label, alpha, cv, show_contour, show_stars, ...	Plot options.

Value

A plotly object (or a named list of plotly objects for plot_mediation_panel).

Examples

```
set.seed(1); n <- 128
x <- cumsum(rnorm(n)); y <- 0.5 * x + rnorm(n, sd = 0.5)
fit <- wavelet_qr(y, x, quantiles = c(0.25, 0.5, 0.75),
                  wavelet = "la8", J = 4, verbose = FALSE)
p <- plot_wqr_heatmap(fit, colorscale = "Parula")
```

---

results_table *Formatted coefficient table with significance stars*

Description

Convenience formatter to insert WQR / MWQR / mediation slope coefficients into a paper-style table.

Usage

```
results_table(x, value = "beta", digits = 3,
              row = "level", variable = NULL)
```

Arguments

x	A data frame, or a wqr_result, mwqr_result, mediation_result.
value	Column to format. Default "beta".
digits	Rounding digits.
row	Pivot row column name ("level" or "band").
variable	Optional filter for MWQR results.

Value

Character data frame with rows = bands and columns = quantiles. Entries look like `"0.345**"` (stars for 1% / 5% / 10%).

Examples

```
set.seed(1); n <- 128
x <- cumsum(rnorm(n)); y <- 0.5 * x + rnorm(n, sd = 0.5)
fit <- wavelet_qr(y, x, quantiles = c(0.25, 0.5, 0.75),
                  wavelet = "la8", J = 4, verbose = FALSE)
results_table(fit)
```

`wavelet_mediation` *Wavelet Quantile Mediation and Moderation*

Description

Five wavelet-band quantile regressions for the triplet (y, x, z): direct effect, x*z moderation (interaction), path a (x -> z), path b (z -> y | x), indirect (a * b).

Usage

```
wavelet_mediation(y, x, z,
                  quantiles = c(0.05, 0.10, 0.25, 0.50,
                                0.75, 0.90, 0.95),
                  wavelet = "la8", J = NULL,
                  dep_name = "Y", main_name = "X",
                  mod_name = "Z", verbose = TRUE)
```

Arguments

`y`, `x`, `z`	Numeric vectors of equal length.
`quantiles`	Numeric vector of tau in (0, 1).
`wavelet`	Wavelet filter.
`J`	Decomposition depth or `NULL` (auto).
`dep_name`, `main_name`, `mod_name`	Character labels for printing.
`verbose`	Logical.

Value

An object of class `"mediation_result"` containing five data frames: `direct`, `interaction`, `path_a`, `path_b`, `indirect`.

Examples

```
set.seed(1); n <- 200
x <- rnorm(n); z <- 0.4 * x + rnorm(n, sd = 0.5)
y <- 0.3 * x + 0.4 * z + rnorm(n, sd = 0.5)
fit <- wavelet_mediation(y, x, z,
                         quantiles = c(0.25, 0.5, 0.75),
                         wavelet = "la8", J = 4, verbose = FALSE)
```

`wavelet_qqr` *Wavelet Quantile-on-Quantile Regression with p-values (WQQR)*

Description

For a selected MODWT frequency band, builds the (theta, tau) coefficient and p-value surface of a kernel-weighted local linear quantile regression of y_b on x_b. Also returns the standard band-level WQR slopes for comparison.

Usage

```
wavelet_qqr(y, x, quantile_step = 0.05,
            wavelet = "la8", J = 5,
            band = "long", bandwidth = 1.0,
            verbose = TRUE)
```

Arguments

`y`, `x`	Numeric vectors of equal length.
`quantile_step`	Step of the tau-grid.
`wavelet`, `J`, `verbose`	See `wavelet_qr`.
`band`	One of `"short"`, `"medium"`, `"long"`, `"all"` or `"D1"`, `"D2"`, ...
`bandwidth`	Gaussian-kernel bandwidth for the local linear QR.

Value

An object of class `"wqqr_result"` with coefficient matrix, p-value matrix, the standard WQR slopes (`qr_coef`) and their p-values.

Examples

```
set.seed(1); n <- 128
x <- cumsum(rnorm(n)); y <- 0.4 * x + rnorm(n, sd = 0.5)
fit <- wavelet_qqr(y, x, quantile_step = 0.25,
                   wavelet = "la8", J = 4, band = "long",
                   verbose = FALSE)
```

wavelet_qr *Wavelet Quantile Regression (WQR)*

Description

Decompose y and x by MODWT, then run quantile regression of y_b on x_b at each quantile in each band b.

Usage

```
wavelet_qr(y, x,
           quantiles = seq(0.05, 0.95, by = 0.05),
           wavelet = "la8", J = 5,
           bands = TRUE, verbose = TRUE)

wqr_to_matrix(wqr, value = "beta")
```

Arguments

y, x	Numeric vectors of equal length.
quantiles	Numeric vector of tau in (0, 1).
wavelet	Wavelet filter; e.g. "la8" (sym4), "la16" (sym8), "d4" (db2), "haar".
J	Decomposition depth.
bands	Logical. TRUE aggregates details to Short / Medium / Long.
verbose	Logical.
wqr	A wqr_result object.
value	Column to pivot: "beta" (default), "se", "p_value".

Value

An object of class "wqr_result" (or numeric matrix).

References

Adebayo, T.S., Ozkan, O. (2024) <doi:10.1016/j.jclepro.2024.140832>.

Examples

```
set.seed(1); n <- 128
x <- cumsum(rnorm(n)); y <- 0.5 * x + rnorm(n, sd = 0.5)
fit <- wavelet_qr(y, x, quantiles = c(0.25, 0.5, 0.75),
                  wavelet = "la8", J = 4, verbose = FALSE)
wqr_to_matrix(fit)
```

`wavelet_quantile_correlation` *Wavelet Quantile Correlation (WQC)*

Description

Quantile correlation between MODWT detail series of two time series, with parametric-bootstrap 95% confidence intervals.

Usage

```
wavelet_quantile_correlation(x, y,
                             quantiles = c(0.05, 0.25, 0.50,
                                           0.75, 0.95),
                             wavelet = "la8", J = 8,
                             n_sim = 200, verbose = TRUE)
```

Arguments

`x`, `y`	Numeric vectors of equal length.
`quantiles`	Numeric vector of tau in (0, 1).
`wavelet`, `J`, `verbose`	See `wavelet_qr`.
`n_sim`	Bootstrap replicates for the 95% confidence intervals.

Value

An object of class `"wqc_result"`.

Examples

```
set.seed(1); n <- 256
x <- cumsum(rnorm(n)); y <- 0.4 * x + rnorm(n, sd = 0.5)
fit <- wavelet_quantile_correlation(
  x, y, quantiles = c(0.25, 0.5, 0.75),
  J = 4, n_sim = 20, verbose = FALSE)
```

`wavelet_quantile_density` *Wavelet Quantile Density Estimation*

Description

Three nonparametric estimators of the quantile density q(p) = 1 / f(F^-1(p)): linear wavelet, hard-thresholded, and local-linear smoothed.

Usage

```
wavelet_quantile_density(y, j0 = 5, bandwidth = 0.15,
                         wavelet = "haar",
                         gld_params = NULL)
```

Arguments

`y`	Numeric sample.
`j0`	Coarsest decomposition level. Default 5.
`bandwidth`	Gaussian-kernel bandwidth for local linear smoothing.
`wavelet`	Wavelet filter for the thresholding step.
`gld_params`	Length-4 `c(l1, l2, l3, l4)` of the GLD or `NULL`; when provided the true quantile density is computed for ISE diagnostics.

Value

An object of class `"quantile_density_result"`.

Examples

```
set.seed(1); y <- rnorm(256)
qd <- wavelet_quantile_density(y, j0 = 4)
```

Index