Package 'qqkrls'

June 1, 2026

Type Package

Title Quantile-on-Quantile Kernel Regularized Least Squares

Version 1.0.0

Description Implements Quantile-on-Quantile Kernel-Based Regularized Least Squares (QQKRLS) as in Adebayo, Ozkan and Eweade (2024) <doi:10.1016/j.jclepro.2024.140832>. Combines Kernel-Based Regularized Least Squares (KRLS) of Hainmueller and Hazlett (2014) <doi:10.1093/pan/mpt019> with the Quantile-on-Quantile regression of Sim and Zhou (2015) <doi:10.1016/j.jbankfin.2015.01.013>: for each quantile theta of the independent variable the response is fit by KRLS on the corresponding sub-sample and the tau-quantile of the resulting pointwise marginal effects yields beta(theta, tau). Standard errors come from a paired bootstrap. Visualisations use the 'MATLAB' 'Parula' colour map by default.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.1

Depends R (>= 3.5.0)

Imports KRLS (>= 1.0-0), plotly (>= 4.0.0), stats, utils, grDevices

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

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

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

Config/testthat/edition 3

NeedsCompilation no

Author Merwan Roudane [aut, cre, cph],
Tomiwa Sunday Adebayo [ctb] (Original QQKRLS methodology),
Jens Hainmueller [ctb] (Original KRLS methodology),
Chad Hazlett [ctb] (Original KRLS methodology)

Maintainer Merwan Roudane <merwanroudane920@gmail.com>

Repository CRAN

Date/Publication 2026-06-01 09:00:14 UTC

Contents

`qqkrls-package` *qqkrls: Quantile-on-Quantile Kernel Regularized Least Squares*

Description

Implements QQKRLS (Adebayo et al., 2024): for each x-quantile theta, fit KRLS of y on x on the corresponding sub-sample and take the tau-quantile of the pointwise marginal effects as beta(theta, tau).

Author(s)

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

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

Description

Colour palettes used by QQKRLS plots. The default scale is MATLAB Parula.

Usage

```
parula_colors(n = 256)
matlab_jet_colors(n = 256)
turbo_colors(n = 256)
bluered_colors(n = 256)
qqkrls_palette(cols, n_breaks = 32)
resolve_colorscale(name = "Parula", n_breaks = 32)
qqkrls_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.
`show_preview`	Print descriptions.

Value

Character vector or list.

Examples

```
parula_colors(8)
matlab_jet_colors(8)
turbo_colors(8)
bluered_colors(8)
qqkrls_colorscales(show_preview = FALSE)
```

---

plot_qqkrls_3d *Visualisations for QQKRLS Results*

Description

3D surface, heatmap and contour plots for QQKRLS results, defaulting to MATLAB Parula.

Usage

```
plot_qqkrls_3d(qqkrls_result, value = "coefficient",
               colorscale = "Parula", show_contour = TRUE,
               x_label = "X Quantile (theta)",
               y_label = "Y Quantile (tau)", title = NULL)

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

plot_qqkrls_contour(qqkrls_result, value = "coefficient",
                    colorscale = "Parula",
                    x_label = "X Quantile (theta)",
                    y_label = "Y Quantile (tau)", title = NULL)
```

Arguments

`qqkrls_result`	A `qqkrls` object.
`value`	Column to plot.
`colorscale`	Default `"Parula"`.
`show_contour`, `show_stars`, `x_label`, `y_label`, `title`	See details.

Value

A plotly object.

Examples

```
## Small toy example -- auto-tested. Plot objects are constructed but
## not rendered when run non-interactively.
set.seed(1); n <- 30
x <- rnorm(n); y <- 0.5 * x + rnorm(n, sd = 0.3)
fit <- qqkrls(y, x,
              y_quantiles = c(0.25, 0.5),
              x_quantiles = c(0.5, 0.75),
              n_boot = 5, verbose = FALSE)
p1 <- plot_qqkrls_3d(fit, colorscale = "Parula")
p2 <- plot_qqkrls_heatmap(fit, show_stars = TRUE)
p3 <- plot_qqkrls_contour(fit)
```

`qqkrls` *Quantile-on-Quantile Kernel Regularized Least Squares*

Description

Implements QQKRLS (Adebayo et al., 2024). For each x-quantile theta, fit KRLS of y on x on the sub-sample where x <= Q_x(theta), then take the tau-quantile of the pointwise marginal effects as the coefficient beta(theta, tau).

Usage

```
qqkrls(y, x,
       y_quantiles = seq(0.05, 0.95, by = 0.05),
       x_quantiles = seq(0.05, 0.95, by = 0.05),
       subset_col = 1, deriv_index = 1,
       sigma = NULL, lambda = NULL,
       min_obs = 15, n_boot = 200,
       verbose = TRUE, seed = 42)
```

Arguments

`y`	Numeric response.
`x`	Numeric vector or matrix.
`y_quantiles`	Quantiles of y (tau) in (0, 1).
`x_quantiles`	Quantiles of x (theta) in (0, 1).
`subset_col`	Column of x used to subset by quantile.
`deriv_index`	Column of x summarised across pointwise marginal effects.
`sigma`	KRLS Gaussian bandwidth or NULL.
`lambda`	KRLS regularisation parameter or NULL.
`min_obs`	Minimum observations in a subset.
`n_boot`	Bootstrap replicates.
`verbose`	Print progress.
`seed`	RNG seed.

Value

An object of class "qqkrls".

References

Adebayo, T.S., Ozkan, O., Eweade, B.S. (2024). Do energy efficiency R&D investments and ICT promote environmental sustainability in Sweden? A QQKRLS investigation. *Journal of Cleaner Production*, 440, 140832. doi:10.1016/j.jclepro.2024.140832

Hainmueller, J., Hazlett, C. (2014). Kernel Regularized Least Squares. *Political Analysis*, 22(2), 143-168. doi:10.1093/pan/mpt019

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

Examples

```
## Small toy example -- auto-tested by R CMD check, runs in well under 5 s.
set.seed(1); n <- 30
x <- rnorm(n); y <- 0.5 * x + rnorm(n, sd = 0.3)
fit <- qqkrls(y, x,
              y_quantiles = c(0.25, 0.5),
              x_quantiles = c(0.5, 0.75),
              n_boot = 5, verbose = FALSE)
print(fit)
head(fit$results)


## Realistic example with the default 19 x 19 quantile grid and 200
## bootstrap replicates (slow; for interactive use).
set.seed(1); n <- 150
xb <- rnorm(n); yb <- 0.5 * sin(xb) + rnorm(n, sd = 0.3)
fit_full <- qqkrls(yb, xb, n_boot = 100, verbose = FALSE)
print(fit_full)
```

---

qqkrls_to_matrix *Helpers for QQKRLS results*

Description

Pivot QQKRLS results into a matrix or export them to CSV.

Usage

```
qqkrls_to_matrix(qqkrls_result, value = "coefficient")
qqkrls_export(qqkrls_result, file, digits = 4)
```

Arguments

qqkrls_result	A qqkrls object.
value	Column to pivot: "coefficient" (default), "std_error", "t_value", "p_value".
file	Output file path for CSV export.
digits	Rounding digits for CSV export.

Value

Numeric matrix or NULL (invisible).

Examples

```
## Small toy example -- auto-tested.
set.seed(1); n <- 30
x <- rnorm(n); y <- 0.5 * x + rnorm(n, sd = 0.3)
fit <- qqkrls(y, x,
              y_quantiles = c(0.25, 0.5),
              x_quantiles = c(0.5, 0.75),
              n_boot = 5, verbose = FALSE)
M <- qqkrls_to_matrix(fit, "coefficient")
print(M)
qqkrls_export(fit, tempfile(fileext = ".csv"))
```

Index