Package ‘qbinplots’

February 24, 2025

Title Quantile Binned Plots

Version 0.3.3

Description Create quantile binned and conditional plots for Exploratory Data Analysis. The package provides several plotting functions that are all based on quantile binning. The plots are created with 'ggplot2' and 'patchwork' and can be further adjusted.

License MIT + file LICENSE

URL `https://edwindj.github.io/qbinplots/`

BugReports `https://github.com/edwindj/qbinplots/issues`

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports ggplot2, data.table, patchwork, scales

Suggests palmerpenguins, tinytest

NeedsCompilation no

Author Edwin de Jonge [aut, cre] (<`https://orcid.org/0000-0002-6580-4718`>), Martijn Tennekes [ctb]

Maintainer Edwin de Jonge <`edwindjonge@gmail.com`>

Repository CRAN

Date/Publication 2025-02-24 17:20:02 UTC

Contents

`qbinplots-package` *qbinplots*

Description

This package creates plots using quantile binning.

Details

Quantile binning is an exploratory data analysis tool that helps to see the distribution of the variables in a dataset as a function of the variable that is binned.

A data.frame is quantile binned on a variable `x` using `qbin()` and then plotted with one of the avaible plot functions.

`qbinplots` offers various types of plots:

- `qbin_*` quantile binned plots that show the distribution of the variables in the quantile bins.
- `cond_*` conditional quantile plots that show the distribution of the variables conditional on the `x` variable.

Quantile binned plots

- `qbin_lineplot()` highlights the change in median between qbins, shows the distribution within qbins.
- `qbin_barplot()` shows the size of medians or expected value of qbins.
- `qbin_boxplot()` shows the distribution within qbins.
- `qbin_heatmap()` shows the distribution within the qbins.

Conditional (quantile binned) plots

- `cond_boxplot()` shows the distribution of the variables conditional on the `x` variable.
- `cond_barplot()` shows the expected median/mean of the variables conditional on the `x` variable.
- `funq_plot()` shows a functional view of the data, plotting the median and interquartile range of numerical variables and level frequency of the other variables as a function of the `x` variable using quantile bins.

Author(s)

Maintainer: Edwin de Jonge <`edwindjonge@gmail.com`> (ORCID)

Other contributors:

- Martijn Tennekes <`mtennekes@gmail.com`> [contributor]

See Also

Useful links:

- https://edwindj.github.io/qbinplots/
- Report bugs at https://github.com/edwindj/qbinplots/issues

`cond_barplot` *Conditional quantile barplot*

Description

`cond_barplot()` conditions all variables on `x` by quantile binning and shows the median or mean of the other variables for each `x`.

Usage

```
cond_barplot(
  data,
  x = NULL,
  n = 100,
  min_bin_size = NULL,
  overlap = NULL,
  ncols = NULL,
  fill = "#2f4f4f",
  auto_fill = FALSE,
  show_bins = FALSE,
  type = c("median", "mean"),
  ...
)
```

Arguments

`data`	a `data.frame` to be binned
`x`	`character` variable name used for the quantile binning
`n`	`integer` number of quantile bins.
`min_bin_size`	`integer` minimum number of rows/data points that should be in a quantile bin. If NULL it is initially `sqrt(nrow(data))`
`overlap`	`logical` if `TRUE` the quantile bins will overlap. Default value will be `FALSE`.
`ncols`	The number of column to be used in the layout.
`fill`	The color to use for the bars.
`auto_fill`	If `TRUE`, use a different color for each category
`show_bins`	If `TRUE`, show the bins on the x-axis.
`type`	The type of statistic to use for the bars.
`...`	Additional arguments to pass to the plot functions

Value

A `list` of ggplot objects.

See Also

Other conditional quantile plotting functions: `cond_boxplot()`, `cond_heatmap()`, `funq_plot()`

Examples

```
# plots the expected median conditional on Sepal.Width
cond_barplot(iris, "Sepal.Width", n = 12)



  # plots the expected median
  cond_barplot(iris, "Sepal.Width", n = 12, show_bins = TRUE)

  data("diamonds", package="ggplot2")

  cond_barplot(diamonds[c(1:4, 7)], "carat", auto_fill = TRUE)

  if (require(palmerpenguins)) {
    p <- cond_barplot(penguins[1:7], "body_mass_g", auto_fill = TRUE)
    print(p)

    # compare with qbin_boxplot
    p <- cond_boxplot(penguins[1:7], "body_mass_g", auto_fill = TRUE)
    print(p)
  }
```

`cond_boxplot` *Conditional quantile boxplot*

Description

`cond_boxplot()` conditions all variables on `x` by quantile binning and shows the boxplots for the other variables for each value of qbinned `x`.

Usage

```
cond_boxplot(
  data,
  x = NULL,
  n = 100,
  min_bin_size = NULL,
  color = "#002f2f",
  fill = "#2f4f4f",
  auto_fill = FALSE,
  ncols = NULL,
  xmarker = NULL,
  qmarker = NULL,
  show_bins = FALSE,
  xlim = NULL,
  connect = FALSE,
  ...
)
```

Arguments

`data`	a `data.frame` to be binned
`x`	`character` variable name used for the quantile binning
`n`	`integer` number of quantile bins.
`min_bin_size`	`integer` minimum number of rows/data points that should be in a quantile bin. If NULL it is initially `sqrt(nrow(data))`
`color`	The color to use for the line charts
`fill`	The fill color to use for the areas
`auto_fill`	If `TRUE`, use a different color for each category
`ncols`	The number of column to be used in the layout
`xmarker`	`numeric`, the x marker.
`qmarker`	`numeric`, the quantile marker to use that is translated in a x value.
`show_bins`	if `TRUE` a rug is added to the plot
`xlim`	`numeric`, the limits of the x-axis.
`connect`	if `TRUE` subsequent medians are connected.
`...`	Additional arguments to pass to the plot functions

Details

`cond_boxplot` is the same function as `funq_plot()` but with different defaults, namely `connect = FALSE` and `auto_fill = FALSE`. `funq_plot` highlights the functional relationship between x and the y-variables, by connecting the medians of the quantile bins.

`qbin_boxplot()` shows the boxplots of the quantile bins on a quantile scale.

Value

A `list` of ggplot objects.

See Also

Other conditional quantile plotting functions: `cond_barplot()`, `cond_heatmap()`, `funq_plot()`

Examples

```
cond_boxplot(
  iris, x = "Petal.Length"
)
```

cond_heatmap *Conditional heatmap*

Description

`cond_heatmap` shows the conditional distribution of the `y` of variables for each quantile bin of `x`. It is an alternative to `cond_boxplot()`, fine graining the distribution per `qbin()`. `cond_barplot()` highlights the median/mean of the quantile bins, while `funq_plot()` highlights the functional dependency of the median.

Usage

```
cond_heatmap(
  data,
  x = NULL,
  n = 100,
  min_bin_size = NULL,
  overlap = NULL,
  bins = c(n, 25),
  ncols = NULL,
  auto_fill = FALSE,
  show_bins = FALSE,
  fill = "#2f4f4f",
  low = "#eeeeee",
  high = "#2f4f4f",
  ...
)
```

Arguments

`data` a `data.frame` to be binned

`x` `character` variable name used for the quantile binning

`n` `integer` number of quantile bins.

`min_bin_size` `integer` minimum number of rows/data points that should be in a quantile bin. If NULL it is initially `sqrt(nrow(data))`

`overlap` `logical` if `TRUE` the quantile bins will overlap. Default value will be `FALSE`.

`bins` `integer` vector with the number of bins to use for the x and y axis.

`ncols` The number of column to be used in the layout.

`auto_fill` If `TRUE`, use a different color for each category.

`show_bins` If `TRUE`, show the bin boundaries on the x-axis.

`fill` The color used for categorical variables.

`low` The color used for low values in the heatmap.

`high` The color used for high values in the heatmap.

`...` Additional arguments to pass to the plot functions

Value

A `list` of ggplot objects.

See Also

Other conditional quantile plotting functions: `cond_barplot()`, `cond_boxplot()`, `funq_plot()`

Examples

```
cond_heatmap(
  iris,
  x = "Petal.Length",
  n = 12
)



  data("diamonds", package="ggplot2")

  cond_heatmap(
    diamonds,
    x = "carat",
    bins <- c(100,100)
  )[6:8]
```

`funq_plot` *Functional quantile plot*

Description

`funq_plot()` conditions on variable `x` with quantile binning and plots the median and interquartile range of numerical variables and level frequency of the other variables as a function the `x` variable.

Usage

```
funq_plot(
  data,
  x = NULL,
  n = 100,
  min_bin_size = NULL,
  overlap = NULL,
  color = "#002f2f",
  fill = "#2f4f4f",
  auto_fill = TRUE,
  ncols = NULL,
  xmarker = NULL,
  qmarker = NULL,
  show_bins = FALSE,
  xlim = NULL,
  connect = TRUE,
  ...
)
```

Arguments

data a data.frame to be binned

x character variable name used for the quantile binning

n integer number of quantile bins.

min_bin_size integer minimum number of rows/data points that should be in a quantile bin. If NULL it is initially sqrt(nrow(data))

overlap logical if TRUE the quantile bins will overlap. Default value will be FALSE.

color The color to use for the line charts

fill The fill color to use for the areas

auto_fill If TRUE, use a different color for each category

ncols The number of column to be used in the layout

xmarker numeric, the x marker.

qmarker numeric, the quantile marker to use that is translated in a x value.

show_bins if TRUE a rug is added to the plot

xlim numeric, the limits of the x-axis.

connect if TRUE subsequent medians are connected.

... Additional arguments to pass to the plot functions

Details

By highlighting and connecting the median values it creates a functional view of the data. What is the (expected) median given a certain value of x?

It qbins the x variable and plots the medians of the qbins vs the other variables, thereby creating a functional view of x to the rest of the data, calculating the statistics for each bin, hence the name funq_plot.

Value

A ggplot object with the plots

See Also

Other conditional quantile plotting functions: `cond_barplot()`, `cond_boxplot()`, `cond_heatmap()`

Examples

```
funq_plot(iris, "Sepal.Length", xmarker=5.5)



  funq_plot(
    iris,
    x = "Sepal.Length",
    xmarker=5.5,
    overlap = TRUE
  )


  data("diamonds", package="ggplot2")
  funq_plot(diamonds[1:7], "carat", xlim=c(0,2))

  if (require(palmerpenguins)){
    funq_plot(
      penguins[1:7],
      x = "body_mass_g",
      xmarker=4650,
      ncol = 3
    )
  }
```

`qbin` *Bin a data.frame into quantile bins*

Description

Bins a data.frame into quantile bins for variable `x` in `data`.

Usage

```
qbin(data, x = NULL, n = 100, min_bin_size = NULL, overlap = NULL, ...)
```

Arguments

`data` a `data.frame` to be binned

`x` `character` variable name used for the quantile binning

`n` `integer` number of quantile bins.

min_bin_size integer minimum number of rows/data points that should be in a quantile bin. If NULL it is initially sqrt(nrow(data))

overlap logical if TRUE the quantile bins will overlap. Default value will be FALSE.

... reserved for future use

Details

Each numeric variable in the data.frame is binned into n quantile bins, for which the fivenum() and mean() is calculated.

When n/nrow(data) is less than min_bin_size, qbin gives a warning and n is adjusted to nrow(data)/min_bin_size. Each categorical variable is binned into n quantile bins, for which the level frequency is calculated.

Value

a qbin object with:

- $x the variable name used for binning
- $bin a vector of bin numbers
- $n the number of bins
- $num_cols a vector of numeric column names
- $cat_cols a vector of categorical column names
- $data a list of data.tables with the collected information

qbin_barplot *Quantile binned bar plot*

Description

qbin_barplot() shows the median or mean for each quantile bin, thereby focusing on the expected value per qbin(). For a conditional plot, see cond_barplot().

Usage

```
qbin_barplot(
  data,
  x = NULL,
  n = 100,
  min_bin_size = NULL,
  overlap = NULL,
  ncols = NULL,
  fill = "#2f4f4f",
  type = c("median", "mean"),
  ...
)

table_plot(data, x = NULL, n = 100, ncols = ncol(data), fill = "#555555", ...)
```

Arguments

`data` a `data.frame` to be binned

`x` `character` variable name used for the quantile binning

`n` `integer` number of quantile bins.

`min_bin_size` `integer` minimum number of rows/data points that should be in a quantile bin. If NULL it is initially `sqrt(nrow(data))`

`overlap` `logical` if `TRUE` the quantile bins will overlap. Default value will be `FALSE`.

`ncols` The number of column to be used in the layout.

`fill` The color to use for the bars.

`type` The type of statistic to use for the bars.

`...` Additional arguments to pass to the plot functions

Details

The `table_plot` is a specific form of `qbin_barplot` with `ncols` set to `ncol(data)`.

Value

A `list` of ggplot objects.

See Also

Other qbin plotting functions: `qbin_boxplot()`, `qbin_heatmap()`, `qbin_lineplot()`

Examples

```
data("diamonds", package="ggplot2")

table_plot(diamonds[c(1:4, 7)], "carat")

qbin_barplot(iris, "Sepal.Length", n = 12)

table_plot(iris, "Sepal.Length", n=12)
table_plot(
  iris,
  x = "Sepal.Length",
  min_bin_size=20,
  overlap=TRUE
)

if (require(palmerpenguins)) {
  table_plot(penguins[1:7], "body_mass_g", 19)
}
```

qbin_boxplot *Quantile binned boxplot*

Description

qbin_boxplot creates quantile binned boxplots from data using x as the binning variable. It focuses on the change of median between qbins. It is a complement to qbin_heatmap() which focuses on the distribution within the qbins.

Usage

```
qbin_boxplot(
  data,
  x = NULL,
  n = 100,
  min_bin_size = NULL,
  ncols = NULL,
  overlap = NULL,
  connect = FALSE,
  color = "#002f2f",
  fill = "#2f4f4f",
  auto_fill = FALSE,
  qmarker = NULL,
  xmarker = NULL,
  ...
)
```

Arguments

data	a data.frame to be binned
x	character variable name used for the quantile binning
n	integer number of quantile bins.
min_bin_size	integer minimum number of rows/data points that should be in a quantile bin. If NULL it is initially sqrt(nrow(data))
ncols	The number of column to be used in the layout
overlap	logical if TRUE the quantile bins will overlap. Default value will be FALSE.
connect	if TRUE subsequent boxplots are connected
color	The color to use for the lines
fill	The color to use for the bars
auto_fill	If TRUE, use a different color for each category
qmarker	numeric, the quantile marker to use.
xmarker	numeric the x marker, i.e. the value for x that is translated into a q value.
...	Additional arguments to pass to the plot functions

Details

The data is binned by the x and a boxplot is created for each bin. The median of the subsequent boxplots are connected to highlight jumps in the data. It hints at the dependecy of the variable on the binning variable.

Value

A list of ggplot objects.

See Also

Other qbin plotting functions: qbin_barplot(), qbin_heatmap(), qbin_lineplot()

Examples

```
qbin_boxplot(
  iris,
  x = "Sepal.Length",
)


  qbin_boxplot(
    iris,
    x = "Sepal.Length",
    connect = TRUE,
    overlap = TRUE
  )

  qbin_boxplot(
    iris,
    x = "Sepal.Length",
    connect = TRUE,
    xmarker = 5.5,
    auto_fill = TRUE
  )

  data("diamonds", package="ggplot2")

  qbin_boxplot(
    diamonds[1:7],
    "carat",
    auto_fill = TRUE
  )

  qbin_boxplot(
    diamonds[1:7],
    "price",
    auto_fill = TRUE,
  )
```

qbin_heatmap *Quantile binned heatmap*

Description

qbin_heatmap shows the distribution of the y of variables for each quantile bin of x. It is an alternative to qbin_boxplot(), fine graining the distribution per qbin(). qbin_barplot() highlights the median/mean of the quantile bins, while

Usage

```
qbin_heatmap(
  data,
  x = NULL,
  n = 25,
  min_bin_size = NULL,
  overlap = NULL,
  bins = c(n),
  type = c("gradient", "size"),
  ncols = NULL,
  auto_fill = FALSE,
  fill = "#2f4f4f",
  low = "#eeeeee",
  high = "#2f4f4f",
  ...
)
```

Arguments

`data` a `data.frame` to be binned

`x` `character` variable name used for the quantile binning

`n` `integer` number of quantile bins.

`min_bin_size` `integer` minimum number of rows/data points that should be in a quantile bin. If NULL it is initially `sqrt(nrow(data))`

`overlap` `logical` if `TRUE` the quantile bins will overlap. Default value will be `FALSE`.

`bins` `integer` vector with the number of bins to use for the x and y axis.

`type` The type of heatmap to use. Either "gradient" or "size".

`ncols` The number of column to be used in the layout.

`auto_fill` If `TRUE`, use a different color for each category.

`fill` The color used for categorical variables.

`low` The color used for low values in the heatmap.

`high` The color used for high values in the heatmap.

`...` Additional arguments to pass to the plot functions

Value

A `list` of ggplot objects.

See Also

Other qbin plotting functions: `qbin_barplot()`, `qbin_boxplot()`, `qbin_lineplot()`

Examples

```
qbin_heatmap(
  iris,
  "Sepal.Length",
  auto_fill = TRUE
)

qbin_heatmap(
  iris,
  "Sepal.Length",
  auto_fill = TRUE,
  type = "size"
)

qbin_heatmap(
  iris,
  "Sepal.Length",
  overlap = TRUE,
  auto_fill = TRUE
)

data("diamonds", package="ggplot2")

qbin_heatmap(
  diamonds[c(1,7:9)],
  x = "price",
  n = 150
)
```

`qbin_lineplot` *Quantile binned lineplot*

Description

`qbin_lineplot` creates quantile binned boxplots from `data` using `x` as the binning variable and connects the medians: it focuses on the change of median between qbins.

Usage

```
qbin_lineplot(
  data,
  x = NULL,
  n = 100,
  min_bin_size = NULL,
  ncols = NULL,
  connect = TRUE,
  color = "#002f2f",
  fill = "#2f4f4f",
  auto_fill = FALSE,
  qmarker = NULL,
  xmarker = NULL,
  ...
)
```

Arguments

`data`	a `data.frame` to be binned
`x`	`character` variable name used for the quantile binning
`n`	`integer` number of quantile bins.
`min_bin_size`	`integer` minimum number of rows/data points that should be in a quantile bin. If NULL it is initially `sqrt(nrow(data))`
`ncols`	The number of column to be used in the layout
`connect`	if `TRUE` subsequent boxplots are connected
`color`	The color to use for the lines
`fill`	The color to use for the bars
`auto_fill`	If `TRUE`, use a different color for each category
`qmarker`	`numeric`, the quantile marker to use.
`xmarker`	`numeric` the x marker, i.e. the value for x that is translated into a q value.
`...`	Additional arguments to pass to the plot functions

Details

The data is binned by the `x` and a boxplot is created for each bin. The median of the subsequent boxplots are connected to highlight jumps in the data. It hints at the dependecy of the variable on the binning variable.

Value

A `list` of ggplot objects.

See Also

Other qbin plotting functions: `qbin_barplot()`, `qbin_boxplot()`, `qbin_heatmap()`

Examples

```
qbin_lineplot(
  iris,
  x = "Sepal.Length",
)


  qbin_lineplot(
    iris,
    x = "Sepal.Length",
    xmarker = 5.5,
    auto_fill = TRUE
  )

  qbin_lineplot(
    iris,
    x = "Sepal.Length",
    overlap=TRUE,
    xmarker = 5.5,
    auto_fill = TRUE
  )

  data("diamonds", package="ggplot2")

  qbin_lineplot(
    diamonds[1:7],
    "carat",
    auto_fill = TRUE
  )

  qbin_lineplot(
    diamonds[1:7],
    "price",
    auto_fill = TRUE,
  )
```

Index