Package: pubrplot (via r-universe)

May 15, 2026

Type Package

Title Publication-Ready Plots and Statistical Visualizations

Version 0.0.1

Description Provides functions to create high-quality, publication-ready plots for numeric and categorical data, including bar plots, violin plots, boxplots, line plots, error bars, correlation plots, linear model plots, odds ratio plots, and normality plots.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports ggplot2, dplyr, ggthemes, rlang, broom, tidyr, rstatix, purrr, tibble

Config/pak/sysreqs cmake make libicu-dev

Repository https://umarhussain-git.r-universe.dev

Date/Publication 2025-12-09 09:18:31 UTC

RemoteUrl https://github.com/umarhussain-git/pubrplot

RemoteRef HEAD

RemoteSha e0ad8089a61c6e2d0126b121834a4b5548849437

Contents

`plot_bar` *Bar Plot for Categorical Data with Optional Grouping*

Description

Creates a publication-quality bar plot for a categorical variable, with optional grouping by another variable. Automatically calculates counts and percentages and can display them on the bars. Also performs Chi-square or Fisher exact test if `by` is provided.

Usage

```
plot_bar(
  data,
  var,
  by = NULL,
  vjust = -0.3,
  hjust = 0.5,
  axis.label.angle = 45,
  label = NULL,
  border.color = NULL,
  label.color = "black",
  x.lab = "Group",
  y.lab = "Percentage (%)",
  fill.lab = "Variable",
  text.size = 3,
  color.bar = NULL,
  theme_fun = ggthemes::theme_stata,
  bar.width = 0.8,
  y.expand = 1.12
)
```

Arguments

`data`	A data frame containing the variables to plot.
`var`	The main categorical variable to display on the x-axis (unquoted or quoted).
`by`	Optional grouping variable for stacked/dodged bars (unquoted or quoted). Default is NULL.
`vjust`	Vertical adjustment for text labels. Default is -0.3.
`hjust`	Horizontal adjustment for text labels. Default is 0.5.
`axis.label.angle`	Angle of x-axis labels. Default is 45.
`label`	Optional custom labels for factor levels of `var`.
`border.color`	Optional color for bar borders. Default is NULL.
`label.color`	Color of the text labels on bars. Default is ”black”.

x.lab	Label for x-axis. Default is "Group".
y.lab	Label for y-axis. Default is "Percentage (%)".
fill.lab	Legend title for the fill variable. Default is "Variable".
text.size	Size of the text labels. Default is 3.
color.bar	Optional vector of colors for bars.
theme_fun	Theme function from ggthemes (or ggplot2) for styling. Default is `ggthemes::theme_stata`.
bar.width	Width of the bars. Default is 0.8.
y.expand	Factor to expand the y-axis for space above the highest bar. Default is 1.12.

Value

A `ggplot2` object representing the bar plot.

Examples

```
# Example using CO2 dataset
plot_bar(
  CO2,
  var = "Type",
  by = "Treatment",
  fill.lab = "Plant Type",
  color.bar = c("lightblue","lightgreen"),
  border.color = "black",
  bar.width = 0.5,
  text.size = 3,
  label = c("Quebec","Mississippi")
)

# Example using diamonds dataset
plot_bar(
  ggplot2::diamonds,
  var = "cut",
  by = "color",
  y.lab = "Distribution (%)",
  fill.lab = "Cut",
  text.size = 2,
  bar.width = 0.9,
  color.bar = c("#a465db","steelblue","darkgreen","darkred","#fcba03")
)

# Simple bar plot without grouping
plot_bar(ggplot2::diamonds, var = "cut")
```

`plot_cor` *Correlation Heatmap Plot*

Description

Creates a publication-ready correlation heatmap for numeric variables in a data frame. Each tile shows the correlation coefficient, with optional significance stars.

Usage

```
plot_cor(
  data,
  method = "pearson",
  conf.level = 0.95,
  stars = TRUE,
  plot.title = NULL,
  var.labels = NULL
)
```

Arguments

`data`	A data frame containing numeric variables to correlate.
`method`	Correlation method: ”pearson”, ”spearman”, or ”kendall”. Default is ”pearson”.
`conf.level`	Confidence level for correlation confidence intervals. Default is 0.95.
`stars`	Logical. If TRUE, adds significance stars based on p-values. Default is TRUE.
`plot.title`	Character string specifying the plot title. If NULL, a default title is used.
`var.labels`	Optional character vector of variable labels to replace column names in the plot. Must match number of numeric columns.

Value

A ggplot object showing the correlation heatmap with correlation coefficients and significance stars.

Examples

```
plot_cor(mtcars)
plot_cor(mtcars, var.labels = colnames(mtcars))
plot_cor(mtcars, method = "spearman", stars = FALSE)
```

`plot_errorbar` *Plot Mean with Error Bars*

Description

This function creates a line plot with points and customizable error bars (standard deviation, standard error, or confidence interval) for a numeric variable grouped by a categorical variable. Mean values can optionally be displayed above the points.

Usage

```
plot_errorbar(
  data,
  var,
  by,
  error = c("sd", "se", "ci"),
  err.mult = 1.5,
  point.shape = 19,
  point.size = 3,
  line.color = "blue",
  line.size = 1,
  color.point = "black",
  color.error = "black",
  show.mean = TRUE,
  text.size = 3.5,
  err.width = 0.05,
  x.lab = "Group",
  y.lab = NULL,
  title = NULL,
  rotate = FALSE,
  theme_fun = ggthemes::theme_stata
)
```

Arguments

`data`	A data frame containing the variables to plot.
`var`	A numeric variable to be summarized and plotted.
`by`	A grouping (categorical) variable to calculate summary statistics by.
`error`	Type of error to display: ”sd” (standard deviation), ”se” (standard error), or ”ci” (95% confidence interval). Default is ”sd”.
`err.mult`	Numeric multiplier for the error bars. Useful to extend or shrink error bars. Default is 1.5.
`point.shape`	Shape of the points. Default is 19 (solid circle).
`point.size`	Size of the points. Default is 3.
`line.color`	Color of the connecting line. Default is ”blue”.

line.size	Thickness of the connecting line. Default is 1.
color.point	Color of the points. Default is "black".
color.error	Color of the error bars. Default is "black".
show.mean	Logical; if TRUE, mean values are displayed above points. Default is TRUE.
text.size	Size of the mean value text labels. Default is 3.5.
err.width	Width of the error bars (horizontal whiskers). Default is 0.05.
x.lab	Label for the x-axis. Default is "Group".
y.lab	Label for the y-axis. If NULL, uses the name of var.
title	Plot title. Default is NULL.
rotate	Logical; if TRUE, rotates x-axis labels by 45 degrees. Default is FALSE.
theme_fun	ggplot2 theme function to customize the plot appearance. Default is ggthemes::theme_stata.

Value

A ggplot2 object displaying the line plot with points and error bars.

Examples

```
plot_errorbar(
  data = iris,
  var = Sepal.Length,
  by = Species,
  error = "se",
  err.mult = 1,
  point.shape = 19,
  point.size = 3,
  line.color = "red",
  line.size = 0.5,
  color.point = "blue",
  color.error = "blue",
  show.mean = TRUE,
  text.size = 3,
  err.width = 0.05,
  title = "Mean Sepal Length by Species",
  rotate = TRUE
)
```

---

plot_line *Line Plot with Error Bars by Group and Time*

Description

This function creates a line plot showing the mean of a numeric variable over time for different groups, with optional error bars (standard deviation, standard error, or 95% confidence interval). Multiple groups are displayed on the same plot with customizable colors, point shapes, and line thickness.

Usage

```
plot_line(
  data,
  var,
  time,
  group,
  error = c("sd", "se", "ci"),
  err.mult = 1.5,
  point.shape = 19,
  point.size = 3,
  line.size = 1,
  color.lines = c("red", "blue"),
  show.mean = FALSE,
  text.size = 3.5,
  err.width = 0.05,
  x.lab = "Time",
  y.lab = NULL,
  title = NULL,
  theme_fun = ggthemes::theme_stata
)
```

Arguments

`data`	A data frame containing the variables to plot.
`var`	A numeric variable to summarize and plot.
`time`	A variable representing time points (x-axis). Converted to factor if not already.
`group`	A grouping variable (color/line grouping) for the plot.
`error`	Type of error to display: ”sd” (standard deviation), ”se” (standard error), or ”ci” (95% confidence interval). Default is ”sd”.
`err.mult`	Numeric multiplier for the error bars. Default is 1.5.
`point.shape`	Shape of the points. Default is 19 (solid circle).
`point.size`	Size of the points. Default is 3.
`line.size`	Thickness of the lines. Default is 1.
`color.lines`	Vector of colors for the lines/groups. Default is c(”red”, ”blue”).
`show.mean`	Logical; if TRUE, mean values can optionally be displayed above points. Default is FALSE.
`text.size`	Size of mean value text labels (if `show.mean = TRUE`). Default is 3.5.
`err.width`	Width of the error bars. Default is 0.05.
`x.lab`	Label for the x-axis. Default is ”Time”.
`y.lab`	Label for the y-axis. If NULL, uses the name of `var`.
`title`	Plot title. Default is NULL.
`theme_fun`	ggplot2 theme function to customize plot appearance. Default is `ggthemes::theme_stata`.

Value

A `ggplot` object displaying the line plot with optional error bars for multiple groups.

Examples

```
set.seed(123)
n_subj <- 10
time_points <- c("T1","T2","T3")
groups <- c("DrugA","DrugB")

df <- expand.grid(
  id = 1:n_subj,
  time = time_points,
  group = groups
)

# Arrange by group, id, time
df <- dplyr::arrange(df, group, id, time)

# Add BMI column
df <- dplyr::mutate(df,
  BMI = dplyr::case_when(
    time == "T1" & group == "DrugA" ~ 29 + stats::rnorm(dplyr::n(), 0, 0.3),
    time == "T2" & group == "DrugA" ~ 26 + stats::rnorm(dplyr::n(), 0, 0.3),
    time == "T3" & group == "DrugA" ~ 22 + stats::rnorm(dplyr::n(), 0, 0.3),
    time == "T1" & group == "DrugB" ~ 28 + stats::rnorm(dplyr::n(), 0, 0.3),
    time == "T2" & group == "DrugB" ~ 25 + stats::rnorm(dplyr::n(), 0, 0.2),
    time == "T3" & group == "DrugB" ~ 21 + stats::rnorm(dplyr::n(), 0, 0.2)
  )
)
```

---

plot_lm *Plot Linear Regression Estimates with Confidence Intervals*

Description

This function fits univariate and multivariate linear regression models for a given outcome and a set of predictors. It returns a ggplot showing point estimates and 95% confidence intervals for each predictor. Reference levels of factors can optionally be added, and univariate and multivariate results are plotted side by side.

Usage

```
plot_lm(
  data,
  outcome,
  predictors,
  label_vjust = -0.8,
  label_hjust = 0.4,
```

```
  label_size = 3.5,
  label_color = "black",
  point_color = c("steelblue", "firebrick"),
  point_shape = 15,
  ref = TRUE
)
```

Arguments

data	A data frame containing the outcome and predictor variables.
outcome	A string specifying the outcome (dependent) variable.
predictors	A character vector of predictor (independent) variables.
label_vjust	Vertical adjustment for text labels. Default is -0.8.
label_hjust	Horizontal adjustment for text labels. Default is 0.4.
label_size	Size of text labels. Default is 3.5.
label_color	Color of text labels. Default is "black".
point_color	Vector of colors for the points. Default is c("steelblue", "firebrick").
point_shape	Shape of the points. Default is 15.
ref	Logical; if TRUE, adds reference levels for factor variables. Default is TRUE.

Value

A ggplot object showing regression estimates with 95% confidence intervals for both univariate and multivariate models. @import broom

Examples

```
mtcars2 <- dplyr::mutate(
  mtcars,
  cyl = factor(cyl),
  am = factor(am, labels = c("Automatic", "Manual")),
  gear = factor(gear)
)

plot_lm(
  data = mtcars2,
  outcome = "mpg",
  predictors = c("cyl", "hp", "wt", "am", "gear"),
  point_shape = 18
)

plot_lm(
  data = mtcars2,
  outcome = "mpg",
  predictors = c("cyl", "hp", "wt", "am", "gear"),
  point_shape = 18
)
```

`plot_norm` *Normality Assessment Plot with Shapiro-Wilk and Kolmogorov–Smirnov Tests*

Description

This function visualizes the distribution of multiple numeric variables using boxplots or histograms with overlaid normal distribution curves. It automatically selects the appropriate normality test based on sample size: the Shapiro–Wilk test is applied when sample size is 5000, while the Kolmogorov–Smirnov test is used for larger samples (> 5000). The resulting p-values are displayed directly on the plots.

Usage

```
plot_norm(
  data,
  vars,
  geom = c("box", "hist"),
  color_bar = "#377eb8",
  color_line = "darkred",
  xlab = NULL,
  ylab = NULL,
  bins = 20,
  label_color = "black",
  label_size = 3.5,
  label_vjust = 0,
  label_hjust = 0,
  alpha_bar = 0.5,
  sample_size = 5000,
  label_fraction = 0.05,
  position = NULL,
  p.ypos = NULL
)
```

Arguments

`data`	A data frame containing the variables to be tested and plotted.
`vars`	A character vector of column names (numeric variables) to be assessed for normality.
`geom`	Character string specifying the plot type. Options are `"box"` for boxplots and `"hist"` for histograms with normal curves.
`color_bar`	Fill color for boxplots or histograms.
`color_line`	Color of the normal distribution curve (only used for histograms).
`xlab`	X-axis label.
`ylab`	Y-axis label.

`bins`	Number of bins used in histograms.
`label_color`	Color of the normality test p-value text labels.
`label_size`	Numeric size of the p-value text labels.
`label_vjust`	Vertical justification of the p-value labels.
`label_hjust`	Horizontal justification of the p-value labels.
`alpha_bar`	Transparency level for boxplots or histogram bars.
`sample_size`	Maximum sample size used for the normality test. When the total sample size exceeds 5000, the Kolmogorov–Smirnov test is applied automatically.
`label_fraction`	Fraction of plot height used to automatically position p-value labels.
`position`	Optional named list of manual `(x, y)` positions for p-value placement per variable.
`p.ypos`	Optional numeric value or named list to override automatic y-positions for p-values.

Value

A `ggplot` object displaying the selected normality plots with test p-values.

Examples

```
## Load example dataset safely
data(diamonds, package = "ggplot2")
## Example 1: Boxplots with Shapiro-Wilk test (n <= 5000)
plot_norm(
  data = diamonds[1:4000, ],
  vars = c("carat", "x", "y"),
  geom = "box"
)

## Example 2: Histograms with Shapiro-Wilk test (n <= 5000)
plot_norm(
  data = diamonds[1:4000, ],
  vars = c("carat", "x", "y"),
  geom = "hist",
  bins = 20,
  p.ypos = 0.6
)

## Example 3: Kolmogorov-Smirnov test automatically applied (n > 5000)
plot_norm(
  data = diamonds[1:6000, ],
  vars = c("carat", "x"),
  geom = "hist",
  bins = 25
)

## Example 4: CO2 dataset (base R)
plot_norm(
```

```
  data = CO2,
  vars = c("uptake", "conc"),
  geom = "hist",
  bins = 3
)
```

---

plot_numeric *Publication-Quality Numeric Plot with Optional Grouping and Statistical Tests*

Description

Creates a publication-ready plot for numeric variables, including bar plots, violin plots, boxplots, and combinations (violin + box, violin + jitter, box + jitter). Supports error bars (SD, SE, CI), group comparisons, and automatic or specified statistical tests with optional post-hoc annotations.

Usage

```
plot_numeric(
  data,
  var,
  by,
  geom_type = c("bar", "violin", "box", "violin_box", "violin_jitter", "box_jitter"),
  error = c("sd", "se", "ci"),
  test.type = c("auto", "parametric", "nonparametric"),
  vjust = 0,
  rotate = FALSE,
  x.lab = "Group",
  y.lab = NULL,
  text.size = 3.5,
  color.violin = NULL,
  color.box = NULL,
  box.color = "black",
  color.jitter = "black",
  jitter.size = 1.5,
  ptext.size = 3,
  theme_fun = ggthemes::theme_stata,
  bar.width = 0.85,
  box.width = 0.2,
  show.posthoc = TRUE,
  err.mult = 1.5,
  position.p = NULL,
  jitter.width = 0.1
)
```

Arguments

`data`	A data frame containing the variables to plot.
`var`	Numeric variable to plot (unquoted).
`by`	Optional grouping variable (unquoted) to create separate groups.
`geom_type`	Type of plot: "bar", "violin", "box", "violin_box", "violin_jitter", "box_jitter".
`error`	Type of error to display for bar plots: "sd", "se", or "ci".
`test.type`	Statistical test type: "auto", "parametric", or "nonparametric".
`vjust`	Vertical adjustment for text labels. Default is 0.
`rotate`	Logical, whether to rotate x-axis labels. Default is FALSE.
`x.lab`	Label for x-axis. Default is "Group".
`y.lab`	Label for y-axis. Defaults to variable name.
`text.size`	Size of labels above bars or violin/box plots. Default is 3.5.
`color.violin`	Fill color for violin plots. Can be a vector of colors per group.
`color.box`	Fill color for boxplots inside violins. Can be a vector of colors per group.
`box.color`	Outline color for boxplots. Default is "black".
`color.jitter`	Color of jittered points. Default is "black".
`jitter.size`	Size of jittered points. Default is 1.5.
`ptext.size`	Size of text for post-hoc annotations. Default is 3.
`theme_fun`	Theme function from `ggthemes` or `ggplot2` for styling. Default is `ggthemes::theme_stata`.
`bar.width`	Width of bars for bar plots. Default is 0.85.
`box.width`	Width of boxplots inside violin. Default is 0.2.
`show.posthoc`	Logical, whether to display post-hoc test results. Default is TRUE.
`err.mult`	Multiplier for error bars (SD/SE/CI). Default is 1.5.
`position.p`	Optional vector `c(x, y)` to place post-hoc text manually.
`jitter.width`	Width of jitter for points in violin_jitter or box_jitter plots. Default is 0.1.

Value

A `ggplot2` object representing the numeric variable plot.

Examples

```
# Violin + Box plot for iris dataset
plot_numeric(
  data = iris,
  var = Sepal.Length,
  by = Species,
  geom_type = "violin_box",
  box.width = 0.1,
  color.violin = c("#377eb8", "#ff7f00", "#4daf4a"),
  color.box = c("darkgreen", "#a65628", "#f781bf"),
```

```
  box.color = "black",
  color.jitter = "red",
  position.p = c(1,9),
  jitter.size = 2,
  ptext.size = 4,
  show.posthoc = TRUE
)

# Simple bar plot with error bars
plot_numeric(
  data = iris,
  var = Sepal.Length,
  by = Species,
  geom_type = "bar",
  error = "se"
)

# Violin plot with jitter points
plot_numeric(
  data = iris,
  var = Sepal.Length,
  by = Species,
  geom_type = "violin_jitter"
)
```

---

plot_or *Plot Odds Ratios from Logistic Regression*

Description

This function fits univariate and multivariate logistic regression models and plots odds ratios with 95% confidence intervals. Reference levels can optionally be displayed.

Usage

```
plot_or(
  data,
  outcome,
  predictors,
  label_vjust = -0.8,
  label_hjust = 0.5,
  label_size = 3.5,
  label_color = "black",
  point_color = c("steelblue", "firebrick"),
  ref = TRUE
)
```

Arguments

`data`	A data frame containing the outcome and predictors.
`outcome`	Name of the binary outcome variable (as string).
`predictors`	Vector of predictor variable names (as strings).
`label_vjust`	Vertical adjustment for labels (default -0.8).
`label_hjust`	Horizontal adjustment for labels (default 0.5).
`label_size`	Size of the text labels (default 3.5).
`label_color`	Color of the text labels (default "black").
`point_color`	Colors for points corresponding to univariate and multivariate models (default c("steelblue", "firebrick")).
`ref`	Logical, whether to show reference levels (default TRUE).

Value

A ggplot object showing odds ratios with confidence intervals.

Examples

```
# Load built-in infertility dataset
infert1 <- datasets::infert
infert$case <- factor(infert$case, levels = c(0,1), labels = c("Control","Infertile"))
infert$induced <- factor(infert$induced, levels = c(0,1), labels = c("No","Yes"))
infert$spontaneous <- factor(infert$spontaneous, levels = c(0,1), labels = c("No","Yes"))

# Plot with reference levels
plot_or(
  data = infert1,
  outcome = "case",
  predictors = c("parity","induced","spontaneous","age"),
  ref = TRUE
)

# Plot without reference levels
plot_or(
  data = infert1,
  outcome = "case",
  predictors = c("parity","induced","spontaneous","age"),
  ref = FALSE
)
```

`plot_scatter` *Scatter Plot with Linear Regression and Equation Annotation*

Description

This function creates a scatter plot of a numeric outcome against a numeric predictor, optionally grouped by a factor (`by`). A linear regression line is added with optional standard error (SE) shading, and the regression equation and R^2 value are displayed on the plot.

Usage

```
plot_scatter(
  data,
  outcome,
  predictor,
  by = NULL,
  point_color = "#377eb8",
  line_color = "#e41a1c",
  se_fill = "#e41a1c55",
  line_size = 1,
  se = TRUE,
  facet_scales = "free",
  eq_position = c(0.05, 0.95),
  ncol_by = NULL
)
```

Arguments

`data`	A data frame containing the variables to plot.
`outcome`	Character string. Name of the numeric outcome variable.
`predictor`	Character string. Name of the numeric predictor variable.
`by`	Character string, optional. Name of a factor variable for grouping/faceting.
`point_color`	Color for the scatter plot points (default: "#377eb8").
`line_color`	Color for the regression line (default: "#e41a1c").
`se_fill`	Fill color for the confidence interval shading around the regression line (default: "#e41a1c55").
`line_size`	Numeric. Line width for the regression line (default: 1).
`se`	Logical. Whether to display the standard error shading around the regression line (default: TRUE).
`facet_scales`	Character. Scales argument for `facet_wrap` ("free", "fixed", "free_x", "free_y") (default: "free").
`eq_position`	Numeric vector of length 2. Relative position of regression equation on the plot: `c(x_pos, y_pos)` (default: c(0.05, 0.95)).
`ncol_by`	Numeric. Number of columns for faceting (passed to `facet_wrap`) (default: NULL, automatic).

Value

A ggplot2 object of the scatter plot with regression line and annotated equation.

Examples

```
# Basic scatter plot with regression line and equation
plot_scatter(mtcars, "mpg", "wt")

# Scatter plot grouped by cylinder
```

```
plot_scatter(mtcars, "mpg", "wt", by = "cyl",
             point_color = "blue",
             line_color = "red",
             se_fill = "#ff000055",
             line_size = 0.9,
             se = TRUE,
             eq_position = c(0.5, 0.95),
             ncol_by = 2)
```

Index