Package ‘blockr.ggplot’

December 18, 2025

Title Interactive 'ggplot2' Visualization Blocks

Version 0.1.0

Description Extends 'blockr.core' with interactive blocks for data visualization using 'ggplot2'. Users can build charts through a graphical interface without writing code directly. Includes common chart types (bar charts, line charts, pie charts, scatter plots) as well as statistical plots (boxplots, histograms, density plots, violin plots) with rich customization options and intuitive user interfaces.

URL https://bristolmyerssquibb.github.io/blockr.ggplot/

BugReports https://github.com/BristolMyersSquibb/blockr.ggplot/issues

License GPL (>= 3)

Depends R (>= 4.1.0)

Encoding UTF-8

RoxygenNote 7.3.3

Imports blockr.core (>= 0.1.1), colourpicker, ggplot2, glue, patchwork, shiny, shinyjs, shinyWidgets

Suggests cowplot, ggpubr, ggthemes, knitr, pkgdown, rmarkdown, rlang, shinytest2, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/testthat/parallel true

NeedsCompilation no

Author Christoph Sax [aut, cre] (ORCID:
<https://orcid.org/0000-0002-7192-7044>),
Nicolas Bennett [aut],
David Granjon [aut],
Mike Page [aut],
Bristol Myers Squibb [fnd]

Maintainer Christoph Sax <christoph@cynkra.com>

Repository CRAN

Date/Publication 2025-12-18 14:10:15 UTC

Contents

`block_container_script` *Generate container query script for responsive blocks*

Description

Sets up container queries if supported by the browser.

Usage

```
block_container_script()
```

Value

HTML script tag

`block_responsive_css` *Generate responsive CSS for blockr blocks*

Description

Creates CSS for responsive grid layout using 'block-' prefix. Can be reused across different blockr packages.

Usage

```
block_responsive_css()
```

Value

HTML style tag with responsive CSS

`new_facet_block` *Facet Block*

Description

Applies faceting to a ggplot object using facet_wrap() or facet_grid(). Accepts a single ggplot input and adds faceting based on data columns.

Usage

```
new_facet_block(
  facet_type = "wrap",
  facets = character(),
  rows = character(),
  cols = character(),
  ncol = character(),
  nrow = character(),
  scales = "fixed",
  labeller = "label_value",
  dir = "h",
  space = "fixed",
  ...
)
```

Arguments

`facet_type`	Type of faceting: "wrap" or "grid" (default: "wrap")
`facets`	Column(s) to facet by for facet_wrap (character vector)
`rows`	Column(s) for row facets in facet_grid (character vector)
`cols`	Column(s) for column facets in facet_grid (character vector)
`ncol`	Number of columns for facet_wrap (default: NULL for auto)
`nrow`	Number of rows for facet_wrap (default: NULL for auto)
`scales`	Scale behavior: "fixed", "free", "free_x", "free_y" (default: "fixed")
`labeller`	Labeller function: "label_value", "label_both", "label_parsed" (default: "label_value")
`dir`	Direction for facet_wrap: "h" (horizontal) or "v" (vertical) (default: "h")
`space`	Space behavior for facet_grid: "fixed", "free_x", "free_y" (default: "fixed")
`...`	Forwarded to `new_ggplot_transform_block()`

Value

A ggplot transform block object of class `facet_block`.

Examples

```
# Create a facet wrap block
new_facet_block(facet_type = "wrap", facets = "cyl")

# Create a facet grid block
new_facet_block(facet_type = "grid", rows = "cyl", cols = "gear")

if (interactive()) {
  library(blockr.core)
  # Facet block requires a ggplot input
  serve(new_facet_block())
}
```

new_ggplot_block *Universal ggplot block with selectable visualization types*

Description

A flexible block that allows users to select from various ggplot2 geoms and dynamically shows relevant aesthetics for the selected visualization.

Usage

```
new_ggplot_block(
  type = "point",
  x = character(),
  y = character(),
  color = character(),
  fill = character(),
  size = character(),
  shape = character(),
  linetype = character(),
  group = character(),
  alpha = character(),
  density_alpha = 0.8,
  position = "stack",
  bins = 30,
  donut = FALSE,
  ...
)
```

Arguments

`type`	Initial chart type (default "point"). Options: "point", "bar", "line", "boxplot", "violin", "density", "area", "histogram", "pie"
`x`	Column for x-axis

`y`	Column for y-axis
`color`	Column for color aesthetic
`fill`	Column for fill aesthetic
`size`	Column for size aesthetic
`shape`	Column for shape aesthetic
`linetype`	Column for linetype aesthetic
`group`	Column for group aesthetic
`alpha`	Column for alpha aesthetic (variable transparency)
`density_alpha`	Fixed alpha value for density plots (default 0.8)
`position`	Position adjustment for certain geoms
`bins`	Number of bins for histogram
`donut`	Whether to create donut chart when type is "pie" (default FALSE)
`...`	Forwarded to `new_plot_block`

Value

A plot block object of class `ggplot_block`.

Examples

```
# Create a scatter plot block
new_ggplot_block(type = "point", x = "mpg", y = "hp")

# Create a bar chart block
new_ggplot_block(type = "bar", x = "cyl")

if (interactive()) {
  library(blockr.core)
  serve(new_ggplot_block(), list(data = mtcars))
}
```

`new_ggplot_transform_block` *ggplot transform block constructor*

Description

Creates a specialized block for ggplot2-based visualizations. This block returns ggplot objects as data, allowing ggplot blocks to be chained together (e.g., for combining plots with patchwork). Custom output methods ensure plots are displayed properly rather than as data tables.

Usage

```
new_ggplot_transform_block(server, ui, class, ctor = sys.parent(), ...)
```

Arguments

`server`	Server function for the block
`ui`	UI function for the block
`class`	Character vector of CSS classes for the block
`ctor`	Constructor environment (default `sys.parent()`)
`...`	Additional arguments forwarded to `blockr.core::new_block()`

Value

A `ggplot_transform_block` object

Examples

```
# This is a low-level constructor typically used by other block creators
# See new_ggplot_block() for user-facing examples
```

`new_grid_block` *Grid Block*

Description

Combines multiple ggplot objects using patchwork::wrap_plots(). Variadic block that accepts 1 or more ggplot inputs with automatic alignment. Supports layout control (ncol, nrow) and annotations (title, subtitle, auto-tags).

Usage

```
new_grid_block(
  ncol = character(),
  nrow = character(),
  title = character(),
  subtitle = character(),
  caption = character(),
  tag_levels = character(),
  guides = "auto",
  ...
)
```

Arguments

`ncol`	Number of columns in grid layout (default: NULL for auto)
`nrow`	Number of rows in grid layout (default: NULL for auto)
`title`	Overall plot title (default: "")
`subtitle`	Overall plot subtitle (default: "")

caption	Overall plot caption (default: "")
tag_levels	Auto-tagging style: 'A', 'a', '1', 'I', 'i', or NULL (default: NULL)
guides	Legend handling: 'auto', 'collect', or 'keep' (default: 'auto')
...	Forwarded to new_ggplot_transform_block()

Value

A ggplot transform block object of class grid_block.

Examples

```
# Create a grid block with 2 columns
new_grid_block(ncol = "2")

# Create a grid block with title
new_grid_block(title = "My Combined Plots", ncol = "2")

if (interactive()) {
  library(blockr.core)
  # Grid block requires multiple ggplot inputs
  serve(new_grid_block())
}
```

new_theme_block *Theme customization block for ggplot2 plots*

Description

A block that applies advanced theme customizations to ggplot2 objects. Allows fine-grained control over backgrounds, fonts, grid lines, and more. Empty/NULL values will use the base theme's defaults.

Usage

```
new_theme_block(
  panel_bg = "",
  plot_bg = "",
  base_size = NA_real_,
  base_family = "auto",
  show_major_grid = "auto",
  show_minor_grid = "auto",
  grid_color = "",
  show_panel_border = "auto",
  legend_position = "auto",
  base_theme = "auto",
  palette_fill = "auto",
  palette_colour = "auto",
  ...
)
```

Arguments

`panel_bg` Panel background color (default "" uses base theme default)

`plot_bg` Plot background color (default "" uses base theme default)

`base_size` Base font size in points (default NA uses base theme default)

`base_family` Font family: "auto", "sans", "serif", or "mono" (default "auto" preserves upstream font)

`show_major_grid` Show major grid lines: "auto", "show", "hide" (default "auto" uses base theme default)

`show_minor_grid` Show minor grid lines: "auto", "show", "hide" (default "auto" uses base theme default)

`grid_color` Grid line color (default "" uses base theme default)

`show_panel_border` Show panel border: "auto", "show", "hide" (default "auto" uses base theme default)

`legend_position` Legend position: "auto", "right", "left", "top", "bottom", "none" (default "auto" preserves upstream position)

`base_theme` Base ggplot2 theme: "auto", "minimal", "classic", "gray", "bw", etc. (default "auto" preserves upstream theme)

`palette_fill` Color palette for fill aesthetic: "auto" (keep upstream), "viridis", "magma", "plasma", "inferno", "cividis", or "ggplot2" (default "auto" preserves upstream palette)

`palette_colour` Color palette for colour aesthetic: "auto" (keep upstream), "viridis", "magma", "plasma", "inferno", "cividis", or "ggplot2" (default "auto" preserves upstream palette)

`...` Forwarded to `new_transform_block`

Value

A ggplot transform block object of class `theme_block`.

Examples

```
# Create a theme block with classic theme
new_theme_block(base_theme = "classic")

# Create a theme block with custom settings
new_theme_block(
  base_theme = "minimal",
  legend_position = "bottom",
  base_size = 14
)

if (interactive()) {
  library(blockr.core)
  # Theme block requires a ggplot input
  serve(new_theme_block())
}
```

Index