

Package: benviplot (via r-universe)

October 21, 2026

Title Color Palettes and 'ggplot2' Extensions for Data Visualization

Version 1.2.2

Description Provides 'ggplot2' scale functions and a custom theme derived from the 'Benvi' visual identity of QuintoAndar Group. Offers qualitative, sequential, and diverging color palettes with both discrete and continuous scale variants.

License MIT + file LICENSE

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports cli, dplyr (>= 1.1.0), ggplot2 (>= 4.0.0), graphics, rlang

Suggests ggfittext, knitr, pkgdown, ragg, rmarkdown, systemfonts, testthat (>= 3.0.0)

Config/testthat/edition 3

LazyData true

VignetteBuilder knitr

URL <https://viniciusoike.github.io/benviplot/>,
<https://github.com/viniciusoike/benviplot>

BugReports <https://github.com/viniciusoike/benviplot/issues>

Repository <https://viniciusoike.r-universe.dev>

Date/Publication 2026-09-20 23:11:20 UTC

RemoteUrl <https://github.com/viniciusoike/benviplot>

RemoteRef HEAD

RemoteSha 3aedb5b08548c4c932491bbd9e24d60411442fc8

Contents

benvi_palette	2
font_status	3
format_num_br	3
ggplot2-scales-continuous	4
ggsave_benvi	5
iqa	6
iqaiw	7
plot_area	8
plot_column	10
plot_histogram	12
plot_line	13
plot_scatter	14
sales_report	16
scale_colour_benvi_d	17
show_palettes	17
theme_benvi	18
Index	20

benvi_palette	<i>Get the colors of a Benvi palette</i>
---------------	--

Description

Returns the hex codes of a named Benvi palette. Use [show_palettes\(\)](#) to see the available palettes. The interface follows the [ghibli](#) package.

Usage

```
benvi_palette(  
  pal_name = "qual_2",  
  n,  
  direction = 1,  
  type = c("discrete", "continuous")  
)
```

Arguments

pal_name	Name of the palette. Defaults to "qual_2".
n	Number of colors to return. Defaults to the number of colors in the selected palette.
direction	Either 1 or -1. Use -1 to reverse the palette.
type	Either "discrete" or "continuous". Continuous palettes interpolate between the available colors.

Value

A palette object containing hexadecimal color values.

Examples

```
# Use default palette
benvi_palette()

# Specify palette name
benvi_palette("greens")
benvi_palette("greens", n = 20, type = "continuous")
benvi_palette("greens", n = 2, type = "discrete")
```

font_status	<i>Report benviplot font status</i>
-------------	-------------------------------------

Description

Reports whether Poppins, systemfonts, and the ragg graphics device are available.

Usage

```
font_status()
```

Value

A list with poppins_available and ragg_available, invisibly.

Examples

```
font_status()
```

format_num_br	<i>Format numbers using Brazilian conventions</i>
---------------	---

Description

Formats numbers with a period as the thousands separator and a comma as the decimal separator. The result is suitable for labels in tables and plots.

Usage

```
format_num_br(x, digits = 1, percent = FALSE)
```

Arguments

x	A numeric vector.
digits	Number of decimal places to include. Can be negative to round to tens, hundreds, etc.
percent	Whether to append a percent sign.

Details

Brazilian number formatting uses . as the thousands separator and , as the decimal separator. For example, 1234567.89 becomes "1.234.567,9" when digits = 1.

Value

A character vector containing the formatted numbers.

Examples

```
# Basic formatting
x <- 1235134.123
format_num_br(x)

# Different decimal places
format_num_br(x, digits = 3)
format_num_br(x, digits = 0)

# With percentage
format_num_br(12.5, digits = 1, percent = TRUE)

# Negative digits round to tens, hundreds, etc.
format_num_br(1234567, digits = -3)

# Works with vectors
format_num_br(c(100, 1000, 10000))
```

ggplot2-scales-continuous

Continuous Benvi color scales

Description

Map continuous values to colors interpolated from a Benvi palette.

Usage

```
scale_colour_benvi_c(pal_name = "benvi_blue", direction = 1, ...)
scale_color_benvi_c(pal_name = "benvi_blue", direction = 1, ...)
scale_fill_benvi_c(pal_name = "benvi_blue", direction = 1, ...)
```

Arguments

pal_name	Name of the palette. Defaults to "benvi_blue".
direction	Either 1 or -1. Use -1 to reverse the palette.
...	Additional arguments passed to <code>ggplot2::scale_colour_gradientn()</code> or <code>ggplot2::scale_fill_gradientn()</code> .

Value

A continuous ggplot2 scale that can be added to a ggplot object.

Examples

```
library(ggplot2)

ggplot(mtcars, aes(x = wt, y = mpg, color = hp)) +
  geom_point() +
  scale_color_benvi_c("benvi_blue")
```

ggsave_benvi

Save a ggplot with benviplot defaults

Description

A wrapper around `ggplot2::ggsave()` that uses the ragg graphics device for PNG output. Registered fonts such as the bundled Poppins are only visible to systemfonts-aware devices, so ragg is what lets `theme_benvi()` render its font in saved files.

If ragg is not installed, the function uses the default graphics device.

Usage

```
ggsave_benvi(
  filename,
  plot = ggplot2::last_plot(),
  device = NULL,
  width = 7,
  height = 5,
  units = "in",
  dpi = 300,
  ...
)
```

Arguments

filename	File name to create on disk. The file extension determines the graphics device (for example, ".png", ".pdf", or ".svg").
plot	Plot to save. Defaults to the last plot displayed.
device	Graphics device. Uses <code>ragg::agg_png()</code> for PNG files when <code>ragg</code> is installed. Otherwise, uses <code>ggplot2</code> 's default device.
width,height	Plot size. Defaults to 7 by 5 inches.
units	Units for width and height. Choose "in", "cm", "mm", or "px". Defaults to "in".
dpi	DPI to use for raster graphics. Defaults to 300.
...	Additional arguments passed to <code>ggplot2::ggsave()</code> .

Value

filename, invisibly.

See Also

`ggplot2::ggsave()`

Examples

```
p <- ggplot2::ggplot(mtcars, ggplot2::aes(wt, mpg)) + ggplot2::geom_point()
ggsave_benvi(tempfile(fileext = ".png"), plot = p)
```

iqa	<i>QuintoAndar Index (IQA) - Rental Price Index</i>
-----	---

Description

Historical rental price index data from `QuintoAndar`. This is the legacy IQA index, which has been superseded by the IQAIW (see `iqaiw`).

Usage

iqa

Format

- A data frame with 96 observations and 6 variables.
- date** Date of the observation (first day of month)
 - name_muni** Name of the municipality (city)
 - index** Rental price index, normalized to 100 at first observation
 - chg** Monthly percent variation of the index (decimal form)
 - acum12m** 12-month accumulated variation of the index (decimal form)
 - price_m2** Estimated rental price per square meter (R\$/m²)

Source

QuintoAndar

Examples

```
# To visualize the dataset
head(iqa)
str(iqa)

# Plot index over time for all cities
library(ggplot2)
ggplot(iqa, aes(x = date, y = index, color = name_muni)) +
  geom_line() +
  scale_color_benvi_d(pal_name = "qual_9", name = "City") +
  labs(
    title = "IQA: Rental Price Index",
    x = "Date",
    y = "Index (base = 100)"
  ) +
  theme_benvi(base_family = "sans")
```

iqaiw

*QuintoAndar ImovelWeb Rental Index (IQAIW)***Description**

The IQAIW (Índice QuintoAndar ImovelWeb) is a rental index for major Brazilian cities. The index is based on both new rental contracts (managed by QuintoAndar) and online listings published by QuintoAndar and ImovelWeb.

Usage

iqaiw

Format

A data frame with 1,660 observations across six cities and multiple periods.

date Date of the observation (first day of month)

name_muni Name of the municipality. Values include Belo Horizonte, Brasília, Curitiba, Porto Alegre, Rio de Janeiro, and São Paulo

rooms Number of rooms in the property, or "Total" for city-level aggregate

index Rental price index, normalized to 100 at first observation per city

chg Monthly percent variation of the index (decimal form)

acum12m 12-month accumulated variation of the index (decimal form)

price_m2 Estimated rental price per square meter (R\$/m²)

Details

The IQAIW was developed in 2023 and replaced the former IQA index. Given the change in methodology and data sources, the IQAIW is not directly comparable to the IQA index.

Methodology

The index uses double imputation with a hedonic model. A generalized additive model (GAM) controls for property characteristics and location. The sample combines new rental contracts with online listings, so the index does not measure either source in isolation.

Source

https://publicfiles.data.quintoandar.com.br/indice_quintoandar_imovelweb/index_quintoandar_imovelweb_serie.csv

Examples

```
# To visualize the dataset
head(iqaiw)
str(iqaiw)

# Plot index over time for all cities
library(ggplot2)

iqaiw_rooms <- subset(iqaiw, rooms != "Total" & !is.na(acum12m))

ggplot(iqaiw_rooms, aes(x = date, y = acum12m, color = rooms)) +
  geom_line(lwd = 0.5) +
  geom_hline(yintercept = 0) +
  scale_color_benvi_d(pal_name = "qual_6", name = "Rooms") +
  facet_wrap(vars(name_muni), ncol = 3, scales = "free") +
  labs(
    title = "IQAIW: 12-month rental price change",
    x = "Date",
    y = "12-month change (decimal)"
  ) +
  theme_benvi(base_family = "sans")
```

plot_area

Plot an area chart

Description

Plot an area chart

Usage

```
plot_area(
  data,
  x,
  y,
  fill = NULL,
  zero = TRUE,
  order = TRUE,
  pal_name = "qual_benvi",
  scale_name = "",
  scale_label = ggplot2::waiver(),
  position = "stack"
)
```

Arguments

data	A data frame.
x	<data-masked> Variable mapped to the x-axis.
y	<data-masked> Variable mapped to the y-axis.
fill	Fill color for the area. Either a color string (e.g., "blue", "#021841") for a single static color, or a bare column name (without quotes) to map a grouping variable to fill color.
zero	Whether to draw a horizontal line at $y = 0$.
order	Whether to order stacked areas by size. The default, TRUE, places the largest groups on top.
pal_name	Name of the palette.
scale_name	Fill legend title.
scale_label	Fill legend labels.
position	Position adjustment passed to <code>ggplot2::geom_area()</code> .

Value

A ggplot object.

Examples

```
# Simple area chart
sao_paulo <- subset(iqa, name_muni == "São Paulo")
plot_area(data = sao_paulo, x = date, y = index)

# Stacked area chart with fill mapping
total <- subset(iqaiw, rooms == "Total")
plot_area(data = total, x = date, y = index, fill = name_muni)
```

plot_column

*Plot a column chart***Description**

Plot a column chart

Usage

```
plot_column(
  data,
  x,
  y,
  fill = NULL,
  zero = TRUE,
  text = FALSE,
  text_inside = FALSE,
  text_place = NULL,
  text_padding = NULL,
  pal_name = "qual_benvi",
  scale_name = "",
  scale_label = ggplot2::waiver(),
  digits = 0,
  percent = FALSE,
  text_color = "gray20",
  text_family = getOption("theme_benvi.font_family", "sans"),
  text_size = 3,
  position_col = "stack",
  position_text = position_col,
  ...
)
```

Arguments

data	A data frame.
x	<data-masked> Variable mapped to the x-axis.
y	<data-masked> Variable mapped to the y-axis.
fill	Fill color for the columns. Either a color string (e.g., "blue", "#021841") for a single static color, or a bare column name (without quotes) to map a grouping variable to fill color.
zero	Whether to draw a horizontal line at $y = 0$.
text	Whether to add value labels to the columns.
text_inside	Whether to place labels inside the columns with <code>ggfittext</code> . When FALSE (the default), labels use a fixed size and appear above or beside the columns.

text_place	Placement of labels inside the columns. Choose "top", "bottom", "left", "right", "centre", or "center". Defaults to "centre" and applies only when text_inside = TRUE.
text_padding	Padding around inside labels, supplied as a <code>grid::unit()</code> object. Defaults to 1 mm and applies only when text_inside = TRUE.
pal_name	Name of the palette.
scale_name	Fill legend title.
scale_label	Fill legend labels.
digits	Number of digits to show in text labels.
percent	Whether to append a percent sign to text labels.
text_color	Color of the text label. Default is "gray20".
text_family	Font family for the text label. Defaults to <code>getOption("theme_benvi.font_family", "sans")</code> .
text_size	Size of the text label. Default is 3.
position_col	Position adjustment passed to <code>ggplot2::geom_col()</code> .
position_text	Position adjustment passed to <code>ggplot2::geom_text()</code> .
...	Additional arguments passed to <code>ggplot2::geom_col()</code> or <code>ggplot2::geom_text()</code> .

Value

A ggplot object.

Examples

```
# Column chart by city at the latest date
latest <- subset(iqa, date == max(iqa$date))
plot_column(data = latest, x = name_muni, y = index)

# With text labels above bars
plot_column(data = latest, x = name_muni, y = index, text = TRUE)

# With text labels inside bars
latest <- subset(iqa, date == max(iqa$date))
plot_column(data = latest, x = name_muni, y = index, text = TRUE, text_inside = TRUE)
```

plot_histogram	<i>Plot a histogram chart</i>
----------------	-------------------------------

Description

Plot a histogram chart

Usage

```
plot_histogram(
  data,
  x,
  color = "#FFFFFF",
  fill = NULL,
  pal_name = "qual_benvi",
  scale_name = "",
  zero = TRUE,
  bins = NULL,
  method = "fd",
  density = FALSE,
  facet = NULL,
  ...
)
```

Arguments

data	A data frame.
x	<data-masked> Numeric variable mapped to the x-axis.
color	Color of the column border. Defaults to "#FFFFFF" (white).
fill	Fill color for the columns. Either a color string (e.g., "blue", "#021841") for a single static color, or a bare column name (without quotes) to map a grouping variable to fill color.
pal_name	Name of the palette used when fill maps a variable.
scale_name	Fill legend title.
zero	Whether to draw a horizontal line at $y = 0$.
bins	Number of bins. When specified, overrides method.
method	Binning method. Choose "fd" (the default), "FD", "Scott", "Sturges", "Rice", or "sqrt". See Details. Ignored when bins is specified.
density	Whether to map density to the y-axis.
facet	<data-masked> Optional variable to facet the plot. NULL (the default) draws a single panel.
...	Additional arguments passed to ggplot2::facet_wrap() .

Details**Binning methods:**

The method parameter controls how the function computes the bin width.

"fd" or "FD" Freedman-Diaconis rule. Uses the interquartile range: $2 * IQR / n^{1/3}$.

"Scott" Scott's rule. The formula uses the standard deviation, $3 * sd / n^{1/3}$.

"Sturges" Sturges' formula. Uses $k = \lceil \log_2(n) \rceil$ bins.

"Rice" Rice rule. Uses $k = \lceil 2n^{1/3} \rceil$ bins.

"sqrt" Square-root rule. Uses $k = \lceil \sqrt{n} \rceil$ bins.

Value

A ggplot object.

Examples

```
# Default parameters use Freedman-Diaconis
plot_histogram(data = mtcars, x = mpg)
# Use bins to manually choose number of bins
plot_histogram(data = mtcars, x = mpg, bins = 10)
# Example of alternative methods: square root and Rice
plot_histogram(data = mtcars, x = mpg, method = "sqrt")
plot_histogram(data = mtcars, x = mpg, method = "Rice")

# Facet by rooms category
spo <- subset(iqaiw, name_muni == "São Paulo" & rooms != "Total")
plot_histogram(data = spo, x = index, facet = rooms)
```

plot_line

Plot a line chart

Description

Plot a line chart

Usage

```
plot_line(
  data,
  x,
  y,
  color = NULL,
  zero = TRUE,
  point = FALSE,
  pal_name,
  scale_name = "",
  scale_label = ggplot2::waiver(),
  ...
)
```

Arguments

data	A data frame.
x	<data-masked> Variable mapped to the x-axis.
y	<data-masked> Variable mapped to the y-axis.
color	Color of the line. Either a color string (e.g., "blue", "#021841") for a single static color, or a bare column name (without quotes) to map a grouping variable to color.
zero	Whether to draw a horizontal line at $y = 0$.
point	Whether to draw points over the line.
pal_name	Name of the color palette.
scale_name	Color legend title.
scale_label	Color legend labels.
...	Additional arguments passed to <code>ggplot2::labs()</code> .

Value

A ggplot object.

Examples

```
# Single series
sao_paulo <- subset(iqa, name_muni == "S\u00e3o Paulo")
plot_line(data = sao_paulo, x = date, y = index)

# Multiple series with color mapping
total <- subset(iqaiw, rooms == "Total")
plot_line(data = total, x = date, y = index, color = name_muni)
```

plot_scatter	<i>Plot a scatter chart</i>
--------------	-----------------------------

Description

Plot a scatter chart

Usage

```
plot_scatter(
  data,
  x,
  y,
  color = NULL,
  fit = FALSE,
  fit_variable = FALSE,
```

```

    fit_method = "auto",
    fit_formula = NULL,
    fit_color = NULL,
    fit_ci = FALSE,
    zero = "none",
    pal_name = "qual_benvi",
    scale_name = "",
    scale_label = ggplot2::waiver(),
    ...
  )

```

Arguments

<code>data</code>	A data frame.
<code>x</code>	<code><data-masked></code> Variable mapped to the x-axis.
<code>y</code>	<code><data-masked></code> Variable mapped to the y-axis.
<code>color</code>	Color of the points. Either a color string (e.g., "blue", "#021841") for a single static color, or a bare column name (without quotes) to map a grouping variable to color. Continuous numeric variables automatically use a continuous color scale.
<code>fit</code>	Whether to draw a fitted line over the points.
<code>fit_variable</code>	Whether to fit separate lines for groups mapped to color. Defaults to FALSE.
<code>fit_method</code>	Smoothing method passed to <code>ggplot2::geom_smooth()</code> . Defaults to "auto".
<code>fit_formula</code>	Formula passed to <code>ggplot2::geom_smooth()</code> .
<code>fit_color</code>	Color of the fitted regression line. Only applied when <code>fit_variable = FALSE</code> . When NULL (default), uses automatic color selection. When <code>fit_variable = TRUE</code> , the fit line colors are inherited from the grouping variable and this parameter is ignored.
<code>fit_ci</code>	Whether to draw the confidence interval around fitted lines. Defaults to FALSE.
<code>zero</code>	Axis lines to draw. Choose "x", "y", "both", or "none" (the default).
<code>pal_name</code>	Name of the palette.
<code>scale_name</code>	Fill legend title.
<code>scale_label</code>	Fill legend labels.
<code>...</code>	Additional arguments passed to <code>ggplot2::geom_point()</code> .

Value

A ggplot object.

Examples

```

plot_scatter(data = mtcars, x = wt, y = mpg)

# With regression line
plot_scatter(data = mtcars, x = wt, y = mpg, fit = TRUE)

```

`sales_report`*QuintoAndar sales report*

Description

Contract prices per square meter at the zone level for major Brazilian cities.

Usage`sales_report`**Format**

A data frame with 272 observations across multiple cities and zones.

date Date of the observation (first day of month)

name_muni Name of the municipality. Values include Belo Horizonte, Rio de Janeiro, and São Paulo

name_zone Name of the zone within the city

price_m2 Median contract price per square meter (R\$/m²)

Source

QuintoAndar (Sales Report 2020-Q1/2023-Q3). https://publicfiles.data.quintoandar.com.br/sale_report/RelatorioCV_4T_2022.pdf

Examples

```
# Compare contract prices across zones
library(ggplot2)

bhe_sales <- subset(sales_report, name_muni == "Belo Horizonte" & date == max(date))

bhe_sales$name_zone <- factor(
  bhe_sales$name_zone,
  levels = bhe_sales$name_zone[order(bhe_sales$price_m2)]
)

ggplot(bhe_sales, aes(x = price_m2, y = name_zone)) +
  geom_col(fill = benvi_palette("benvi_blue")[3]) +
  labs(x = "Median price per m2 (R$)", y = NULL) +
  theme_benvi(base_family = "sans")
```

scale_colour_benvi_d *Discrete Benvi color scales*

Description

Map discrete values to colors from a Benvi palette.

Usage

```
scale_colour_benvi_d(pal_name = "qual_benvi", direction = 1, ...)
scale_color_benvi_d(pal_name = "qual_benvi", direction = 1, ...)
scale_fill_benvi_d(pal_name = "qual_benvi", direction = 1, ...)
```

Arguments

pal_name Name of the palette. Defaults to "qual_benvi".
direction Either 1 or -1. Use -1 to reverse the palette.
... Additional arguments passed to [ggplot2::discrete_scale\(\)](#).

Value

A discrete ggplot2 scale that can be added to a ggplot object.

Examples

```
library(ggplot2)
# Discrete color scale with rental index data
iqaiw_total <- subset(iqaiw, rooms == "Total")
ggplot(iqaiw_total, aes(x = date, y = index, colour = name_muni)) +
  geom_line() +
  scale_color_benvi_d("qual_benvi")
```

show_palettes *Display all Benvi palettes*

Description

Draws the available Benvi palettes in a grid. The layout follows [RColorBrewer::display.brewer.all\(\)](#).

Usage

```
show_palettes(type = "all", n = NULL)
```

Arguments

type	Palette type to display. Choose "all" (the default), "theme", "sequential", "qualitative", "diverging", "city", or "brand".
n	Number of colors to display from each palette. If NULL (default), displays every color in each palette.

Value

NULL, invisibly. This function is called for its plotting side effect.

Examples

```
# Display all palettes
show_palettes()

# Display only theme palettes
show_palettes("theme")

# Display sequential palettes
show_palettes("sequential")
```

 theme_benvi

A theme for Benvi-styled plots

Description

A ggplot2 base theme for Benvi-styled plots.

The Poppins font is bundled with the package and registered automatically on load when systemfonts is installed. When both systemfonts and ragg are available, the theme uses Poppins by default. Otherwise it falls back to the system's default sans-serif font.

Registered fonts only work with systemfonts-aware devices such as `ragg::agg_png()`. Base R devices (PDF and PostScript) cannot render them. If you see font warnings when saving to PDF, pass `base_family = "sans"` or `set_options(theme_benvi.font_family = "sans")`. See `font_status()` to check your setup.

Usage

```
theme_benvi(
  base_family = getOption("theme_benvi.font_family", default_font_family()),
  base_size = 10,
  background = FALSE
)
```

Arguments

base_family	Argument passed to <code>ggplot2::theme_minimal()</code> . Defaults to "Poppins" when the bundled font is registered and ragg is available, "sans" otherwise. Override globally with <code>options(theme_benvi.font_family = ...)</code> .
base_size	Argument passed to <code>ggplot2::theme_minimal()</code> . Defaults to 10.
background	Whether to use an off-white background.

Value

A ggplot2 theme object.

Examples

```
library(ggplot2)
series <- subset(iqaiw, name_muni == "São Paulo" & rooms == "Total")

# Base theme (using "sans" for portability)
ggplot(series, aes(date, index)) +
  geom_line(color = benvi_palette("benvi_blue")[1], lwd = 1) +
  labs(x = NULL, y = "Index (base = 100)", title = "IQAIW") +
  theme_benvi(base_family = "sans")

# Optional offwhite (creme) background
ggplot(series, aes(date, index)) +
  geom_line(color = benvi_palette("benvi_blue")[1], lwd = 1) +
  labs(x = NULL, y = "Index (base = 100)", title = "IQAIW") +
  theme_benvi(base_family = "sans", background = TRUE)

## Not run:
# Use Poppins with a ragg device (requires systemfonts and ragg)
ggplot(series, aes(date, index)) +
  geom_line(color = benvi_palette("benvi_blue")[1], lwd = 1) +
  labs(x = NULL, y = "Index (base = 100)", title = "IQAIW") +
  theme_benvi(base_family = "Poppins")

## End(Not run)
```

Index

- * **datasets**
 - iqa, [6](#)
 - iqaiw, [7](#)
 - sales_report, [16](#)
- benvi_palette, [2](#)
- font_status, [3](#)
- font_status(), [18](#)
- format_num_br, [3](#)
- ggplot2-scales-continuous, [4](#)
- ggplot2::discrete_scale(), [17](#)
- ggplot2::facet_wrap(), [12](#)
- ggplot2::geom_area(), [9](#)
- ggplot2::geom_col(), [11](#)
- ggplot2::geom_point(), [15](#)
- ggplot2::geom_smooth(), [15](#)
- ggplot2::geom_text(), [11](#)
- ggplot2::ggsave(), [5](#), [6](#)
- ggplot2::labs(), [14](#)
- ggplot2::scale_colour_gradientn(), [5](#)
- ggplot2::scale_fill_gradientn(), [5](#)
- ggplot2::theme_minimal(), [19](#)
- ggsave_benvi, [5](#)
- grid::unit(), [11](#)
- iqa, [6](#)
- iqaiw, [6](#), [7](#)
- plot_area, [8](#)
- plot_column, [10](#)
- plot_histogram, [12](#)
- plot_line, [13](#)
- plot_scatter, [14](#)
- ragg::agg_png(), [6](#), [18](#)
- RColorBrewer::display.brewer.all(), [17](#)
- sales_report, [16](#)
- scale_color_benvi_c
 - (ggplot2-scales-continuous), [4](#)
- scale_color_benvi_d
 - (scale_colour_benvi_d), [17](#)
- scale_colour_benvi_c
 - (ggplot2-scales-continuous), [4](#)
- scale_colour_benvi_d, [17](#)
- scale_fill_benvi_c
 - (ggplot2-scales-continuous), [4](#)
- scale_fill_benvi_d
 - (scale_colour_benvi_d), [17](#)
- show_palettes, [17](#)
- show_palettes(), [2](#)
- theme_benvi, [18](#)