Package: BFHtheme (via r-universe)

October 31, 2026

Type Package

Title BFH Theme and Color Palettes for ggplot2

Version 0.5.6

Description Provides comprehensive theming support for ggplot2 graphics following Bispebjerg og Frederiksberg Hospital and Region Hovedstaden visual identity guidelines. Includes color palettes for both Hospital and Region H branding, themes, scale functions, and helper utilities for creating consistent and professional visualizations. Uses modern 'systemfonts' package for robust font detection with graceful fallback. Recommends 'ragg', 'svglite', and Cairo devices for high-quality output.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Depends R (>= 3.6.0)

Imports ggplot2 (>= 3.4.0), grid, grDevices, scales, marquee, systemfonts

Suggests knitr, rmarkdown, testthat (>= 3.0.0), withr, showtext, sysfonts, ragg, svglite, png, jpeg, cowplot, magick, patchwork, vdiffr

VignetteBuilder knitr

Config/testthat/edition 3

Config/pak/sysreqs libfontconfig1-dev libfreetype6-dev libfribidi-dev libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev

Repository https://johanreventlow.r-universe.dev

Date/Publication 2026-10-01 12:41:08 UTC

RemoteUrl https://github.com/johanreventlow/BFHtheme

RemoteRef HEAD

RemoteSha 80d2c4dbf772fecfbce14cc9e237ffdce1345499

Contents

add_bfh_footer . . . 2
add_bfh_logo . . . 3
bfh_colors . . . 5
bfh_cols . . . 5
bfh_combine_plots . . . 6
bfh_labs . . . 7
bfh_pal . . . 9
bfh_palettes . . . 10
bfh_save . . . 10
bfh_status_cols . . . 12
bfh_status_values . . . 13
bfh_title_block . . . 14
get_bfh_font . . . 15
get_bfh_logo . . . 16
reset_bfh_defaults . . . 17
scale_bfh . . . 18
scale_fill_bfh_continuous . . . 19
scale_fill_bfh_discrete . . . 20
scale_position_bfh . . . 21
set_bfh_defaults . . . 23
show_bfh_palettes . . . 25
theme_bfh . . . 25
use_bfh_knitr_defaults . . . 26

Index **28**

`add_bfh_footer` *Add BFH Footer to a Plot*

Description

Appends a coloured footer bar with optional text, ideal for credits or contact details in BFH-branded material.

Usage

```
add_bfh_footer(
  plot,
  text = NULL,
  color = bfh_cols("hospital_primary"),
  text_color = "white",
  height = 0.05
)
```

Arguments

plot	A ggplot2 object.
text	Footer text. Defaults to ‘"Bispebjerg og Frederiksberg Hospital"‘.
color	Footer background colour. Defaults to ‘bfh_cols("hospital_primary")‘.
text_color	Text colour. Defaults to ‘"white"‘.
height	Footer height as a fraction of the plotting area. Defaults to ‘0.05‘.

Details

The footer spans the width of the plot. Provide ‘text = NULL‘ to fall back to a default hospital identifier. Adjust ‘height‘ (0–1) to control the bar thickness relative to the plotting area.

Value

ggplot2 object with the footer applied.

See Also

[add_bfh_logo()], [bfh_title_block()]

Other BFH branding: `add_bfh_logo()`, `bfh_title_block()`, `get_bfh_logo()`

Examples

```
## Not run:
library(ggplot2)
p <- ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  theme_bfh()

add_bfh_footer(p, text = "Bispebjerg og Frederiksberg Hospital - 2024")

## End(Not run)
```

---

add_bfh_logo *Add BFH Logo to a Plot*

Description

Places the BFH mark logo at a fixed, brand-compliant position on ggplot2 visualizations. The logo is automatically positioned at the bottom-left corner with standardized sizing (1/15 of plot height, square).

Usage

```
add_bfh_logo(plot, logo_path = NULL, alpha = 1)
```

Arguments

plot	A ggplot2 object.
logo_path	Path to a PNG or JPEG logo file. When 'NULL' (default), uses the packaged BFH mark logo ('bfh_mark.png').
alpha	Transparency of the logo (0–1). Defaults to '1' (fully opaque). A value of '0' produces an invisible logo.

Details

Simplified API (v0.3.0): This function uses fixed positioning to ensure consistent branding across all visualizations. The logo is always placed at the bottom-left corner, flush with the plot edge, at a height and width of 1/15 of the total plot height (square aspect ratio).

Default logo: When 'logo_path = NULL', the function automatically loads 'bfh_mark.png' (full resolution BFH symbol mark). For custom logos, provide a path to a PNG or JPEG file.

Dependencies: Requires the 'cowplot' package for precise logo positioning. Install with: "'r install.packages("cowplot") "'

Path restriction: Restrict logo loading to a specific directory (path-prefix check, not a security sandbox): "'r options(BFHtheme.logo_root = "/approved/logo/directory") "' When set, only logos within this directory (or subdirectories) will be allowed.

Resource limits: Maximum file size and image dimensions can be configured: "'r options(BFHtheme.logo_max_bytes = 5 * 1024^2) # 5 MB default options(BFHtheme.logo_max_dim = 4096) # 4096 px default "'

Performance: Decoded logo images are cached per path and file modification time. Repeated calls with the same unchanged file avoid redundant disk reads.

Breaking changes from v0.2.0: The 'position', 'size', and 'padding' parameters have been removed to enforce consistent branding. See NEWS.md for migration guidance.

Value

Modified ggplot2 object with the logo applied.

See Also

[get_bfh_logo()], [add_bfh_footer()]

Other BFH branding: add_bfh_footer(), bfh_title_block(), get_bfh_logo()

Examples

```
## Not run:
library(ggplot2)
p <- ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  theme_bfh()

# Use default BFH mark logo (recommended)
add_bfh_logo(p)

# Use custom logo
```

```
add_bfh_logo(p, logo_path = "/path/to/custom_logo.png")

# Adjust transparency
add_bfh_logo(p, alpha = 0.7)

## End(Not run)
```

`bfh_colors` *BFH Brand Color Definitions*

Description

Master vector of Bispebjerg og Frederiksberg Hospital and Region Hovedstaden brand colors. Values are hex codes that comply with the official visual identity guidelines.

Usage

```
bfh_colors
```

Format

A named character vector where each element is a hex color code.

Details

The object exposes both official color names (e.g. ‘"hospital_primary"‘) and pragmatic aliases (e.g. ‘"primary"‘, ‘"blue"‘) for quick scripting. Use [bfh_cols()] to extract colors safely with validation and helpful error messages.

See Also

[bfh_cols()], [bfh_palettes], [show_bfh_palettes()]

Other BFH colors: `bfh_cols()`, `bfh_pal()`, `bfh_palettes`, `bfh_status_cols()`, `bfh_status_values()`, `clear_bfh_pal_cache()`, `show_bfh_palettes()`

`bfh_cols` *Extract BFH Colors*

Description

Retrieves one or more BFH brand colors by name with validation and friendly error messages.

Usage

```
bfh_cols(...)
```

Arguments

...	Character names present in [bfh_colors]. If omitted, all colors are returned.

Details

The function accepts both official color identifiers (e.g. ‘"hospital_primary"‘) and provided aliases (e.g. ‘"primary"‘). Invalid names trigger an informative error listing available options.

Value

Character vector of hex color codes in the order requested.

See Also

Other BFH colors: `bfh_colors`, `bfh_pal()`, `bfh_palettes`, `bfh_status_cols()`, `bfh_status_values()`, `clear_bfh_pal_cache()`, `show_bfh_palettes()`

Examples

```
# Get a single color
bfh_cols("primary")

# Get multiple colors
bfh_cols("hospital_primary", "hospital_blue")

# Get all colors
bfh_cols()

# Using aliases
bfh_cols("blue", "grey")
```

---

bfh_combine_plots *Create a Multi-Panel Figure with Shared Legend*

Description

Arranges multiple ggplot2 plots into a single layout with an optional shared legend, ideal for BFH report panels and slide decks.

Usage

```
bfh_combine_plots(plots, ncol = NULL, nrow = NULL, legend_position = "bottom")
```

Arguments

plots	List of ggplot2 plot objects.
ncol	Optional number of columns in the layout.
nrow	Optional number of rows in the layout.
legend_position	Legend placement. One of ‘"bottom"‘, ‘"top"‘, ‘"left"‘, ‘"right"‘, or ‘"none"‘. Defaults to ‘"bottom"‘.

Details

The function currently depends on the ‘patchwork‘ package. When unavailable a descriptive error is raised; install ‘patchwork‘ to use this helper.

Value

Combined plot object produced by ‘patchwork‘.

See Also

[patchwork::wrap_plots()], [bfh_save()]

Other BFH helpers: `bfh_labs()`, `bfh_save()`, `get_bfh_dimensions()`

Examples

```
## Not run:
library(ggplot2)
p1 <- ggplot(mtcars, aes(wt, mpg, color = factor(cyl))) + geom_point()
p2 <- ggplot(mtcars, aes(hp, mpg, color = factor(cyl))) + geom_point()
p3 <- ggplot(mtcars, aes(disp, mpg, color = factor(cyl))) + geom_point()

# Combine with shared legend
bfh_combine_plots(list(p1, p2, p3), ncol = 3)

## End(Not run)
```

---

bfh_labs *BFH-Style Plot Labels with Automatic Uppercase*

Description

Wrapper around [ggplot2::labs()] that converts subtitles, captions, axes, and legend titles to uppercase, matching BFH typographic guidance while leaving the main title unchanged.

Usage

```
bfh_labs(...)
```

Arguments

...	Named arguments passed to [ggplot2::labs()]. Character values are uppercased except for 'title'. Common keys include: • 'title': Plot title (kept in natural case). • 'subtitle': Subtitle (uppercased). • 'caption': Caption (uppercased). • 'x', 'y': Axis labels (uppercased). • 'color'/'colour', 'fill': Legend titles (uppercased).

Details

Only the argument named '"title"' is exempt from uppercasing. Supply other character values via '...' and they will be converted using [base::toupper()].

Value

A ggplot2 labels object.

See Also

[ggplot2::labs()], [bfh_title_block()]

Other BFH helpers: `bfh_combine_plots()`, `bfh_save()`, `get_bfh_dimensions()`

Examples

```
## Not run:
library(ggplot2)

# Only title unchanged, everything else uppercase
ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  theme_bfh() +
  bfh_labs(
    title = "Fuel efficiency analysis",     # → "Fuel efficiency analysis" (unchanged)
    subtitle = "Motor Trend data",          # → "MOTOR TREND DATA"
    x = "weight (tons)",                    # → "WEIGHT (TONS)"
    y = "miles per gallon"                  # → "MILES PER GALLON"
  )

# Time series example
ggplot(time_data, aes(date, value, color = group)) +
  geom_line() +
  theme_bfh() +
  bfh_labs(
    title = "Patient trends over time",     # → "Patient trends over time" (unchanged)
   subtitle = "Bispebjerg og Frederiksberg Hospital 2020-2024",  # → "BISPEBJERG..." (uppercase)
    x = "date",                             # → "DATE"
    y = "number of patients",               # → "NUMBER OF PATIENTS"
    color = "department"                    # → "DEPARTMENT"
  )
```

```
## End(Not run)
```

`bfh_pal` *Interpolate a BFH Color Palette*

Description

Creates a color interpolation function for any named palette in [bfh_palettes], enabling smooth gradients or truncated palette selections.

Usage

```
bfh_pal(palette = "main", reverse = FALSE, ...)
```

Arguments

`palette`	Character name of the palette in [bfh_palettes]. Defaults to '"main"'.
`reverse`	Logical; reverse the palette order. Defaults to 'FALSE'.
`...`	Additional arguments forwarded to [grDevices::colorRampPalette()].

Value

Function that accepts an integer 'n' and returns 'n' colors.

See Also

[bfh_palettes], [scale_color_bfh()], [scale_fill_bfh()]

Other BFH colors: `bfh_colors`, `bfh_cols()`, `bfh_palettes`, `bfh_status_cols()`, `bfh_status_values()`, `clear_bfh_pal_cache()`, `show_bfh_palettes()`

Examples

```
bfh_pal("main")(5)
bfh_pal("blues", reverse = TRUE)(3)
```

bfh_palettes *BFH Color Palettes (Named List)*

Description

Predefined discrete and sequential palettes covering hospital, regional, and legacy BFH use cases. Palettes are sourced from [bfh_colors].

Usage

```
bfh_palettes
```

Format

An object of class `list` of length 17.

Details

The list is structured for use with [bfh_pal()] and the ‘scale_*_bfh*()‘ helpers. Palettes ending in ‘"_seq"‘ are suited for continuous scales while the rest target categorical data and infographics.

See Also

[bfh_pal()], [show_bfh_palettes()], [scale_color_bfh()]

Other BFH colors: `bfh_colors`, `bfh_cols()`, `bfh_pal()`, `bfh_status_cols()`, `bfh_status_values()`, `clear_bfh_pal_cache()`, `show_bfh_palettes()`

bfh_save *Save a Plot with BFH Specifications*

Description

Wrapper around [ggplot2::ggsave()] that applies BFH-recommended dimensions, resolution, and message output for reproducible plot exports.

Usage

```
bfh_save(
  filename,
  plot = ggplot2::last_plot(),
  preset = "report_full",
  width = NULL,
  height = NULL,
  units = "in",
  dpi = 300,
  ...
)
```

Arguments

filename	File name to save the plot. Include an extension such as ‘"plot.png"‘ or ‘"figure.pdf"‘.
plot	Plot to save. Defaults to the last displayed plot.
preset	Character string specifying size presets: ‘"report_full"‘ (7x5 in), ‘"report_half"‘ (3.5x3 in), ‘"presentation"‘ (10x6 in), ‘"presentation_wide"‘ (12x6.75 in), ‘"square"‘ (6x6 in), ‘"poster"‘ (12x9 in).
width	Plot width. Overrides ‘preset‘ when provided.
height	Plot height. Overrides ‘preset‘ when provided.
units	Units for width/height (‘"in"‘, ‘"cm"‘, ‘"mm"‘, ‘"px"‘). Defaults to ‘"in"‘.
dpi	Resolution in dots per inch. Defaults to 300.
...	Additional arguments passed to [ggplot2::ggsave()].

Details

Use the ‘preset‘ argument to quickly match internal communication formats. Width/height arguments override presets when supplied. For more control, pair with [get_bfh_dimensions()] to retrieve structured size guidance.

Value

Invisibly returns ‘filename‘.

See Also

[get_bfh_dimensions()], [theme_bfh()], [bfh_title_block()]

Other BFH helpers: `bfh_combine_plots()`, `bfh_labs()`, `get_bfh_dimensions()`

Examples

```
## Not run:
library(ggplot2)
p <- ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  theme_bfh()

# Save with preset
bfh_save("my_plot.png", p, preset = "report_full")

# Save with custom dimensions
bfh_save("my_plot.pdf", p, width = 8, height = 6, dpi = 600)

## End(Not run)
```

`bfh_status_cols` *Statusfarver: nuancer inden for én statusgruppe*

Description

Semantiske farver til figurer der fordeler forløb eller sager på status, fx udredningsret: udredt inden frist, frist overskredet, endnu ikke afgjort.

Tre grupper: * ‘"overholdt"‘ - ved 3 eller flere nuancer fra dyb hospitalsblå ‘#0067a1‘ til lyseblå ‘#d8eef9‘. Én nuance er hospitalsblå, to er hospitalsblå og lyseblå ‘#99d8f6‘. * ‘"ikke_afgjort"‘ - grå fra ‘#333333‘ til ‘#b8b8b8‘. * ‘"overskredet"‘ - accent ‘#c0392b‘, derefter accenten blandet med op til 60

Usage

```
bfh_status_cols(gruppe, n)
```

Arguments

`gruppe`	En af ‘"overholdt"‘, ‘"ikke_afgjort"‘, ‘"overskredet"‘.
`n`	Antal nuancer. Den første er den kraftigste.

Value

Character-vektor med ‘n‘ hex-farver.

See Also

[bfh_status_values()], [bfh_palettes]

Other BFH colors: `bfh_colors`, `bfh_cols()`, `bfh_pal()`, `bfh_palettes`, `bfh_status_values()`, `clear_bfh_pal_cache()`, `show_bfh_palettes()`

Examples

```
bfh_status_cols("overholdt", 5)
bfh_status_cols("overskredet", 3)
```

bfh_status_values *Statusfarver til en hel kategoriliste*

Description

Giver én farve pr. kategori, så kategorier i samme statusgruppe får hver sin nuance fra [bfh_status_cols()]. Beregnet til kategoritabeller, hvor hver kategori er knyttet til en gruppe (fx BFHddl's figurkategorier).

Usage

```
bfh_status_values(grupper, rang = NULL)
```

Arguments

grupper	Character-vektor med en statusgruppe pr. kategori.
rang	Valgfri numerisk vektor af samme længde: intensitet inden for gruppen, hvor lavest er kraftigst. Default er rækkefølgen i 'grupper'.

Value

Character-vektor med hex-farver, samme længde som 'grupper'.

See Also

[bfh_status_cols()]

Other BFH colors: bfh_colors, bfh_cols(), bfh_pal(), bfh_palettes, bfh_status_cols(), clear_bfh_pal_cache(), show_bfh_palettes()

Examples

```
# 04 dekomponering af henvisninger: "inden for 30 dage" kraftigst
bfh_status_values(
  c("ikke_afgjort", "overskredet", "overholdt", "overholdt"),
  rang = c(1, 1, 2, 1)
)
```

`bfh_title_block` *Create a Branded Title Block*

Description

Generates a reusable set of labels for BFH-styled titles, subtitles, and captions.

Usage

```
bfh_title_block(title, subtitle = NULL, caption = NULL)
```

Arguments

`title`	Main title text.
`subtitle`	Optional subtitle text.
`caption`	Optional caption text.

Details

Pair with [bfh_labs()] for automatic uppercase conversion or add the result directly to ggplot objects. Use 'NULL' for optional arguments you want to omit.

Value

A [ggplot2::labs] object.

See Also

[bfh_labs()], [theme_bfh()], [add_bfh_footer()]

Other BFH branding: `add_bfh_footer()`, `add_bfh_logo()`, `get_bfh_logo()`

Examples

```
## Not run:
library(ggplot2)
ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  bfh_title_block(
    title = "Vehicle Weight vs Fuel Efficiency",
    subtitle = "Analysis of mtcars dataset",
    caption = "Source: Motor Trend, 1974"
  ) +
  theme_bfh()

## End(Not run)
```

`get_bfh_font` *Determine BFH Font Family*

Description

Identifies the best available typeface for BFH-branded plots, caching the result to avoid repeated system queries. The priority order is: Mari → Mari Office → Roboto → Arial → sans.

Usage

```
get_bfh_font(check_installed = TRUE, silent = FALSE, force_refresh = FALSE)
```

Arguments

`check_installed`	Logical. If 'TRUE' (default) verify fonts are installed. Set to 'FALSE' to simply return the highest-priority font name.
`silent`	Logical. Suppress informational messages when 'TRUE'. Defaults to 'FALSE'.
`force_refresh`	Logical. When 'TRUE', bypass the cache and re-run the detection. Defaults to 'FALSE'.

Details

Font availability is checked via 'systemfonts::system_fonts()' exact family name matching, which is reliable across platforms.

Results are cached in the package environment; use [clear_bfh_font_cache()] or set 'force_refresh = TRUE' after installing new fonts.

Font availability notes: - **Mari fonts**: Only available on BFH employee computers (proprietary) - **Roboto**: Free open-source font (install via OS or use 'use_bfh_showtext()') - **Arial**: System font on most platforms - **sans**: Universal fallback (system default)

For embedding fonts in outputs: Use modern graphics devices like 'ragg::agg_png()', 'svglite::svglite()', or 'grDevices::cairo_pdf()'. See 'set_bfh_graphics()' for recommended device setup.

Value

Character string containing the selected font family.

See Also

[check_bfh_fonts()], [set_bfh_fonts()], [set_bfh_defaults()]

Other BFH fonts: `check_bfh_fonts()`, `clear_bfh_font_cache()`, `install_roboto_font()`, `set_bfh_fonts()`, `set_bfh_graphics()`, `setup_bfh_fonts()`, `use_bfh_showtext()`

Examples

```
# Get best available font (cached after first call)
font <- get_bfh_font()

# Use in theme
theme_bfh(base_family = get_bfh_font())

# Force refresh cache if fonts change
font <- get_bfh_font(force_refresh = TRUE)

# For external users without Mari fonts:
# Install showtext for automatic Roboto loading
## Not run:
install.packages("showtext")
library(ggplot2)
ggplot(mtcars, aes(wt, mpg)) + geom_point() + theme_bfh()
# Roboto will be auto-loaded via Google Fonts if not installed

## End(Not run)
```

`get_bfh_logo` *Get Path to the Packaged BFH Logo*

Description

Retrieves the file path to BFH logo assets bundled with the package. Variants include full colour, greyscale, and symbol-only marks across multiple resolutions.

Usage

```
get_bfh_logo(size = "web", variant = "color")
```

Arguments

`size`	Logo size: ‘"full"‘ (print resolution), ‘"web"‘ (800 px), or ‘"small"‘ (400 px). Defaults to ‘"web"‘.
`variant`	Logo variant: ‘"color"‘ (default), ‘"grey"‘ (greyscale), or ‘"mark"‘ (symbol only).

Value

Character string containing the absolute path, or ‘NULL‘ if the asset is unavailable.

See Also

[add_bfh_logo()]

Other BFH branding: `add_bfh_footer()`, `add_bfh_logo()`, `bfh_title_block()`

Examples

```
## Not run:
library(ggplot2)
p <- ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  theme_bfh()

# Add BFH logo to plot
add_bfh_logo(p, get_bfh_logo())

# Use high-resolution version for print
add_bfh_logo(p, get_bfh_logo("full"))

# Use greyscale logo
add_bfh_logo(p, get_bfh_logo(variant = "grey"))

# Use mark (symbol only)
add_bfh_logo(p, get_bfh_logo(variant = "mark"))

## End(Not run)
```

---

reset_bfh_defaults *Reset ggplot2 Defaults*

Description

Restores the theme and geom defaults that were active before [set_bfh_defaults()] was last called.

Usage

```
reset_bfh_defaults()
```

Details

If [set_bfh_defaults()] has not been called in the current session, falls back to ‘theme_gray()‘ and ggplot2’s hardcoded geom defaults, and emits a message indicating no saved state was found.

Value

Invisibly returns ‘TRUE‘ once defaults are reset.

See Also

[set_bfh_defaults()]

Other BFH defaults: `set_bfh_defaults()`, `use_bfh_knitr_defaults()`

Examples

```
## Not run:
# Set BFH defaults
set_bfh_defaults()

# ... create some plots ...

# Reset to previous state
reset_bfh_defaults()

## End(Not run)
```

---

scale_bfh *BFH Colour Scales for ggplot2*

Description

Convenient wrappers around ggplot2 scales that apply BFH brand palettes to colour or fill aesthetics.

Usage

```
scale_color_bfh(palette = "main", discrete = TRUE, reverse = FALSE, ...)

scale_colour_bfh(palette = "main", discrete = TRUE, reverse = FALSE, ...)

scale_fill_bfh(palette = "main", discrete = TRUE, reverse = FALSE, ...)
```

Arguments

palette	Character name of a palette in [bfh_palettes()]. Defaults to ‘"main"‘.
discrete	Logical; treat the mapped variable as discrete? Defaults to ‘TRUE‘.
reverse	Logical; reverse the palette order? Defaults to ‘FALSE‘.
...	Additional arguments passed to the underlying ggplot2 scale functions.

Details

When ‘discrete = TRUE‘ the functions delegate to [ggplot2::discrete_scale()]. For continuous aesthetics they build a gradient using [bfh_pal()] and [ggplot2::scale_color_gradientn()] / [ggplot2::scale_fill_gradientn()]. Set ‘reverse = TRUE‘ to invert palette order, and pass additional arguments via ‘...‘ to fine-tune scale behaviour.

Value

A ggplot2 scale object.

See Also

[bfh_pal()], [bfh_palettes], [scale_fill_bfh_continuous()]

Other BFH scales: scale_fill_bfh_continuous(), scale_fill_bfh_discrete(), scale_position_bfh

Examples

```
## Not run:
library(ggplot2)
ggplot(mtcars, aes(x = wt, y = mpg, color = factor(cyl))) +
  geom_point() +
  scale_color_bfh()

## End(Not run)
## Not run:
library(ggplot2)
ggplot(mtcars, aes(x = factor(cyl), fill = factor(cyl))) +
  geom_bar() +
  scale_fill_bfh()

## End(Not run)
```

---

scale_fill_bfh_continuous

Continuous colour scales using specific BFH palettes

Description

Continuous colour scales using specific BFH palettes

Usage

```
scale_fill_bfh_continuous(palette = "blues", reverse = FALSE, ...)

scale_color_bfh_continuous(palette = "blues", reverse = FALSE, ...)

scale_colour_bfh_continuous(palette = "blues", reverse = FALSE, ...)
```

Arguments

palette	Character name of a palette in [bfh_palettes()].
reverse	Logical; reverse palette direction? Defaults to 'FALSE'.
...	Additional arguments passed to [ggplot2::scale_color_gradientn()].

See Also

[scale_color_bfh()], [bfh_pal()]

Other BFH scales: scale_bfh, scale_fill_bfh_discrete(), scale_position_bfh

Examples

```
## Not run:
library(ggplot2)
ggplot(faithfuld, aes(waiting, eruptions, fill = density)) +
  geom_tile() +
  scale_fill_bfh_continuous(palette = "blues")

## End(Not run)
```

`scale_fill_bfh_discrete` *Discrete colour scales using specific BFH palettes*

Description

Discrete colour scales using specific BFH palettes

Usage

```
scale_fill_bfh_discrete(palette = "main", reverse = FALSE, ...)

scale_color_bfh_discrete(palette = "main", reverse = FALSE, ...)

scale_colour_bfh_discrete(palette = "main", reverse = FALSE, ...)
```

Arguments

`palette`	Character name of a palette in [bfh_palettes()].
`reverse`	Logical; reverse palette direction? Defaults to 'FALSE'.
`...`	Additional arguments passed to [ggplot2::discrete_scale()].

See Also

[scale_color_bfh()], [bfh_pal()], [bfh_palettes]

Other BFH scales: `scale_bfh`, `scale_fill_bfh_continuous()`, `scale_position_bfh`

Examples

```
## Not run:
library(ggplot2)
ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +
  geom_point() +
  scale_color_bfh_discrete(palette = "primary")

## End(Not run)
```

scale_position_bfh *BFH Position Scales with Uppercase Labels*

Description

Position scale functions that automatically convert axis tick labels to uppercase, following BFH typography guidelines. These functions wrap ggplot2's standard position scales and apply uppercase transformation to all axis labels.

Usage

```
scale_x_continuous_bfh(..., labels = toupper)

scale_y_continuous_bfh(..., labels = toupper)

scale_x_discrete_bfh(..., labels = toupper)

scale_y_discrete_bfh(..., labels = toupper)

scale_x_date_bfh(..., date_labels = waiver(), labels = waiver())

scale_y_date_bfh(..., date_labels = waiver(), labels = waiver())

scale_x_datetime_bfh(..., date_labels = waiver(), labels = waiver())

scale_y_datetime_bfh(..., date_labels = waiver(), labels = waiver())
```

Arguments

...	Additional arguments passed to the underlying ggplot2 scale function. Common arguments include ‘breaks‘, ‘limits‘, ‘expand‘, etc.
labels	Label formatting function. For date/datetime scales, defaults to ‘waiver()‘ which uses ‘scales::label_date_short()‘ for compact, hierarchical labels. For other scales, defaults to ‘toupper‘. You can pass any ‘scales::label_*()‘function (e.g., ‘label_date(" and output will be uppercased automatically. Set to ‘waiver()‘ to use the ‘date_labels‘ parameter for date/datetime scales.
date_labels	Date format string using standard strftime format codes (e.g., ‘" scales. When ‘labels = waiver()‘, this parameter controls the date format via ‘scales::label_date(format = date_labels)‘. Output is automatically uppercased. If ‘labels‘ is explicitly specified, ‘date_labels‘ is ignored. Defaults to ‘waiver()‘ which uses ‘label_date_short()‘.

Details

These functions are particularly useful for maintaining consistent typographic style across plots. The uppercase transformation is applied via the ‘labels‘ argument, which accepts a function that formats the axis breaks.

For date and datetime scales, **the default is [scales::label_date_short()]** which creates compact, hierarchical labels that only show what has changed (e.g., year only appears when it changes). This minimizes horizontal space on the x-axis. All output is automatically converted to uppercase.

Default Behavior (label_date_short)

By default, date/datetime scales use 'label_date_short()' which produces: "' JAN. | FEB. | MAR. | ... | DEC. | JAN. | FEB. | ... 2023 2024 "' Instead of the repetitive: "' JAN. 2023 | FEB. 2023 | MAR. 2023 | ... | JAN. 2024 | FEB. 2024 | ... "'

Integration with scales package

The date/datetime scales integrate seamlessly with the 'scales' package. You can override the default or use 'breaks_pretty()' for intelligent break positioning:

"'r # Use default (label_date_short with smart breaks) scale_x_date_bfh()

Add custom breaks scale_x_date_bfh(breaks = scales::breaks_pretty(n = 6))

Override with custom label format scale_x_date_bfh(labels = scales::label_date("

Combine custom breaks and labels scale_x_date_bfh(breaks = scales::breaks_pretty(n = 8), labels = scales::label_date(") "'

Value

A ggplot2 scale object.

See Also

[ggplot2::scale_x_continuous()], [ggplot2::scale_x_discrete()], [ggplot2::scale_x_date()], [bfh_labs()], [scales::label_date()], [scales::label_date_short()], [scales::breaks_pretty()]

Other BFH scales: scale_bfh, scale_fill_bfh_continuous(), scale_fill_bfh_discrete()

Examples

```
## Not run:
library(ggplot2)

# Continuous scale with uppercase labels
ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  scale_x_continuous_bfh() +
  theme_bfh()

# Discrete scale with uppercase labels
ggplot(mtcars, aes(x = factor(cyl), y = mpg)) +
  geom_boxplot() +
  scale_x_discrete_bfh() +
  theme_bfh()

# Date scale with default label_date_short() (compact hierarchical labels)
df <- data.frame(
  date = seq.Date(as.Date("2023-01-01"), as.Date("2024-12-31"), by = "month"),
  value = rnorm(24)
)
```

```
ggplot(df, aes(date, value)) +
  geom_line() +
  scale_x_date_bfh() +  # Uses label_date_short() by default!
  theme_bfh()

# Add custom breaks for better positioning
ggplot(df, aes(date, value)) +
  geom_line() +
  scale_x_date_bfh(breaks = scales::breaks_pretty(n = 8)) +
  theme_bfh()

# Override with custom label format using labels parameter
ggplot(df, aes(date, value)) +
  geom_line() +
  scale_x_date_bfh(labels = scales::label_date("%B %Y")) +  # Full month names
  theme_bfh()

# Or use date_labels parameter for simple format strings
ggplot(df, aes(date, value)) +
  geom_line() +
  scale_x_date_bfh(date_labels = "%Y") +  # Year only (uppercased)
  theme_bfh()

# Another date_labels example: custom format
ggplot(df, aes(date, value)) +
  geom_line() +
  scale_x_date_bfh(date_labels = "%Y-%m-%d") +  # ISO date format
  theme_bfh()

## End(Not run)
```

`set_bfh_defaults` *Set BFH Defaults for ggplot2*

Description

Applies a BFH theme globally and updates default geom colours for the current R session. Ideal for scripts or reports where every plot should inherit BFH styling without repeated boilerplate.

Usage

```
set_bfh_defaults(
  theme = "bfh",
  palette = "main",
  base_size = 12,
  base_family = NULL
)
```

Arguments

theme	Character name of the BFH theme to adopt. Only ‘"bfh"‘ is supported (other theme variants removed in v0.2.0). Defaults to ‘"bfh"‘.
palette	Character name of the palette from [bfh_palettes] used to seed geom defaults. Defaults to ‘"main"‘.
base_size	Base font size for the theme. Defaults to 12.
base_family	Base font family. Use ‘NULL‘ (default) to auto-detect with [get_bfh_font()].

Details

The function calls [ggplot2::theme_set()] with the selected BFH theme and updates default aesthetics for commonly used geoms (points, lines, bars, and more). Colour/fill scales are not altered automatically—continue to add ‘scale_*_bfh()‘ when mapping aesthetics.

Previous global state (active theme and geom defaults) is saved automatically and can be restored with [reset_bfh_defaults()].

Value

Invisibly returns ‘TRUE‘ after the defaults have been applied.

See Also

[reset_bfh_defaults()], [set_bfh_fonts()]

Other BFH defaults: reset_bfh_defaults(), use_bfh_knitr_defaults()

Examples

```
## Not run:
# Set BFH defaults at the start of your script
set_bfh_defaults()

# Plots without color mapping use BFH colors automatically
library(ggplot2)
ggplot(mtcars, aes(wt, mpg)) +
  geom_point()  # Uses BFH primary color

# For color mappings, add scale manually
ggplot(mtcars, aes(wt, mpg, color = factor(cyl))) +
  geom_point() +
  scale_color_bfh()  # Required for BFH colors

# Use a different palette
set_bfh_defaults(palette = "blues")

## End(Not run)
```

show_bfh_palettes *Visualise BFH Palettes*

Description

Displays every palette defined in [bfh_palettes] using ‘scales::show_col()‘ to assist with palette selection.

Usage

```
show_bfh_palettes(n = NULL)
```

Arguments

n	Optional integer specifying how many colors to preview from each palette. When ‘NULL‘ (default) all palette entries are shown.

See Also

Other BFH colors: bfh_colors, bfh_cols(), bfh_pal(), bfh_palettes, bfh_status_cols(), bfh_status_values(), clear_bfh_pal_cache()

Examples

```
## Not run:
show_bfh_palettes()

## End(Not run)
```

theme_bfh *BFH ggplot2 Theme*

Description

Primary ggplot2 theme aligned with Bispebjerg og Frederiksberg Hospital’s visual identity guidelines. Applies typography, spacing, and layout defaults that mirror official templates.

Usage

```
theme_bfh(
  base_size = 12,
  base_family = NULL,
  base_line_size = base_size/22,
  base_rect_size = base_size/22
)
```

Arguments

base_size	Base font size in points. Default is 12.
base_family	Base font family. Set to 'NULL' (default) to auto-detect a BFH approved font via [get_bfh_font()].
base_line_size	Base line size. Default is 'base_size / 22'.
base_rect_size	Base rectangle size. Default is 'base_size / 22'.

Details

The theme builds on 'ggplot2::theme_minimal()' and augments it with BFH-specific typography via 'marquee::element_marquee', custom axis styling, and legend placement. When 'base_family = NULL' (the default) the best available BFH font is detected automatically using [get_bfh_font()]. Use the 'base_*' arguments to fine-tune typography while preserving the brand aesthetic.

Value

A [ggplot2::theme] object that can be added to ggplot graphs.

Examples

```
## Not run:
library(ggplot2)
ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  theme_bfh()

## End(Not run)
```

---

use_bfh_knitr_defaults

Enable BFH knitr Defaults (Opt-in)

Description

Sets knitr chunk options to use ragg-based rendering at 300 DPI — the recommended output device for BFH-styled graphics in R Markdown and Quarto documents.

Usage

```
use_bfh_knitr_defaults(dpi = 300)
```

Arguments

dpi	Output resolution in dots per inch. Defaults to '300'.

Details

This function must be called explicitly. ‘library(BFHtheme)‘ does NOT modify knitr options automatically. This is intentional: modifying global session state on package load would silently override any knitr configuration the user has already set.

Call ‘use_bfh_knitr_defaults()‘ once in your setup chunk:

```
library(BFHtheme)
use_bfh_knitr_defaults()
```

Requires the ‘ragg‘ package. Install with ‘install.packages("ragg")‘.

Value

Invisibly returns ‘NULL‘. Called for its side effect of setting knitr chunk options.

See Also

[set_bfh_defaults()]

Other BFH defaults: `reset_bfh_defaults()`, `set_bfh_defaults()`

Examples

```
## Not run:
library(BFHtheme)
use_bfh_knitr_defaults()
# knitr::opts_chunk$get("dev") is now "ragg_png"

## End(Not run)
```

Index

∗ **BFH branding**
add_bfh_footer, 2
add_bfh_logo, 3
bfh_title_block, 14
get_bfh_logo, 16
∗ **BFH colors**
bfh_colors, 5
bfh_cols, 5
bfh_pal, 9
bfh_palettes, 10
bfh_status_cols, 12
bfh_status_values, 13
show_bfh_palettes, 25
∗ **BFH defaults**
reset_bfh_defaults, 17
set_bfh_defaults, 23
use_bfh_knitr_defaults, 26
∗ **BFH fonts**
get_bfh_font, 15
∗ **BFH helpers**
bfh_combine_plots, 6
bfh_labs, 7
bfh_save, 10
∗ **BFH scales**
scale_bfh, 18
scale_fill_bfh_continuous, 19
scale_fill_bfh_discrete, 20
scale_position_bfh, 21
∗ **BFH themes**
theme_bfh, 25
∗ **datasets**
bfh_colors, 5
bfh_palettes, 10

add_bfh_footer, 2, *4*, *14*, *16*
add_bfh_logo, 3, *3*, *14*, *16*

bfh_colors, 5, *6*, *9, 10*, *12, 13*, *25*
bfh_cols, 5, *5*, *9, 10*, *12, 13*, *25*
bfh_combine_plots, 6, *8*, *11*
bfh_labs, 7, *7*, *11*
bfh_pal, *5, 6*, 9, *10*, *12, 13*, *25*
bfh_palettes, *5, 6*, *9*, 10, *12, 13*, *25*
bfh_save, *7, 8*, 10
bfh_status_cols, *5, 6*, *9, 10*, 12, *13*, *25*
bfh_status_values, *5, 6*, *9, 10*, *12*, 13, *25*
bfh_title_block, *3, 4*, 14, *16*

check_bfh_fonts, *15*
clear_bfh_font_cache, *15*
clear_bfh_pal_cache, *5, 6*, *9, 10*, *12, 13*, *25*

get_bfh_dimensions, *7, 8*, *11*
get_bfh_font, 15
get_bfh_logo, *3, 4*, *14*, 16

install_roboto_font, *15*

reset_bfh_defaults, 17, *24*, *27*

scale_bfh, 18, *19, 20*, *22*
scale_color_bfh *(*scale_bfh*)*, 18
scale_color_bfh_continuous *(*scale_fill_bfh_continuous*)*, 19
scale_color_bfh_discrete *(*scale_fill_bfh_discrete*)*, 20
scale_colour_bfh *(*scale_bfh*)*, 18
scale_colour_bfh_continuous *(*scale_fill_bfh_continuous*)*, 19
scale_colour_bfh_discrete *(*scale_fill_bfh_discrete*)*, 20
scale_fill_bfh *(*scale_bfh*)*, 18
scale_fill_bfh_continuous, 19, *19, 20*, *22*
scale_fill_bfh_discrete, *19*, 20, *22*
scale_position_bfh, *19, 20*, 21
scale_x_continuous_bfh *(*scale_position_bfh*)*, 21
scale_x_date_bfh *(*scale_position_bfh*)*, 21
scale_x_datetime_bfh *(*scale_position_bfh*)*, 21

scale_x_discrete_bfh
(scale_position_bfh), 21
scale_y_continuous_bfh
(scale_position_bfh), 21
scale_y_date_bfh (scale_position_bfh), 21
scale_y_datetime_bfh
(scale_position_bfh), 21
scale_y_discrete_bfh
(scale_position_bfh), 21
set_bfh_defaults, *17*, 23, *27*
set_bfh_fonts, *15*
set_bfh_graphics, *15*
setup_bfh_fonts, *15*
show_bfh_palettes, *5, 6*, *9, 10*, *12, 13*, 25

theme_bfh, 25

use_bfh_knitr_defaults, *17*, *24*, 26
use_bfh_showtext, *15*