Package: AAGIThemes (via r-universe)

May 26, 2026

Title AAGI Branding for Graphical and Tabular Outputs

Version 2.1.1

Description Applies Analytics for the Australian Grains Industry ('AAGI') external brand guidelines to graphics. 'AAGI' colours and font guidelines are applied as useful and reasonable to base graphics, 'ggplot2' figures, 'flextable' and 'gt' objects.

License GPL (>= 3)

URL https://github.com/AAGI-AUS/AAGIThemes, https://AAGI-AUS.github.io/AAGIThemes/

BugReports https://github.com/AAGI-AUS/AAGIThemes/issues

Depends R (>= 4.1.0)

Imports AAGIPalettes, cli, flextable, ggplot2 (>= 3.3.5), grDevices, grid, gt, magick, officer, showtext, sysfonts, systemfonts, withr

Suggests dplyr, gapminder, knitr (>= 1.22), ozmaps, RColorBrewer, reshape2, rmarkdown, roxyglobals, rsvg, scales, sf, spelling, testthat (>= 3.0.0), tibble, viridis

VignetteBuilder knitr

Remotes AAGI-AUS/AAGIPalettes

Config/Needs/build moodymudskipper/devtag

Config/roxygen2/version 8.0.0

Config/roxyglobals/filename globals.R

Config/roxyglobals/unique FALSE

Config/testthat/edition 3

Config/testthat/parallel true

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE, roclets = c(``collate", ``rd", ``namespace", ``devtag::dev_roclet"))

X-schema.org-applicationCategory Tools

X-schema.org-isPartOf
https://grdc.com.au/research/partnerships-and-initiatives/strategic-partnerships/aagi

Config/pak/sysreqs libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libmagick++-dev gsfonts libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libxml2-dev libssl-dev libnode-dev zlib1g-dev

Repository https://aagi-aus.r-universe.dev

Date/Publication 2026-05-26 06:11:47 UTC

RemoteUrl https://github.com/AAGI-AUS/AAGIThemes

RemoteRef HEAD

RemoteSha 0ed5aac976e0a1efadc7bae9db98d32c7e3d4e2a

Contents

`add_aagi_logo` *Insert AAGI's Logo Into Any Graphic File*

Description

Insert the AAGI logo in any graphical image. Particularly well- suited for use with files created with **AAGIThemes**. The logo will be inserted in the upper-left of the image with a size of at least 4.6 cm as per AAGI brand guidelines.

Usage

```
add_aagi_logo(file_in, file_out, logo_width = 4.6, overwrite = FALSE)
```

Arguments

file_in	Path to the graphical file to import and add the AAGI logo to apply the logo to.
file_out	File name to create on disk as a .png format image.
logo_width	Size for the logo in centimetres. Defaults to 4.6 cm, the smallest allowed by the AAGI guidelines. Larger sizes may be beneficial for larger graphical images than normal reporting and sharing.
overwrite	A Boolean value that indicates whether to overwrite an existing file or not. Defaults to FALSE and will not overwrite the existing file.

Value

An invisible NULL, called for its side effect of adding the AAGI logo to the plot that's called.

Image Sizing

This function checks the image's DPI values and sets the logo size accordingly. When saving, e.g., from **ggplot**, specify the DPI value along with your height and width and units for optimal performance. 'ggsave(filename = "AAGI.png", plot = p1, path = tempdir(), width = 18, height = 18, units = "cm", dpi = 300)

Author(s)

Adam Sparks, <adam.sparks@curtin.edu.au>

Examples

```
library("ggplot2")

p1 <- ggplot(mtcars) +
  geom_point(aes(
    x = wt,
    y = mpg,
    colour = factor(gear)
  )) +
  facet_wrap(~am) +
  theme_aagi()

ggsave(p1, filename = "AAGI.png", path = tempdir())

add_aagi_logo(
  file_in = file.path(tempdir(), "AAGI.png"),
  file_out = file.path(tempdir(), "AAGI_logo.png")
)

x <- magick::image_read(file.path(tempdir(), "AAGI_logo.png"))
print(x)
```

barplot_aagi *Basic Barplots Using a Unified AAGI Style and Typography*

Description

Basic barplots that follow a standard AAGI style including typography guidelines that uses (hopefully) sensible defaults. All valid `barplot()` options are supported through `...`, for *e.g.*, `col` to set the colour. Defaults to "AAGI Black", a very dark grey colour.

Usage

```
barplot_aagi(height, ...)
```

Arguments

`height`	Either a vector or matrix of values describing the bars which make up the plot. If height is a vector, the plot consists of a sequence of rectangular bars with heights given by the values in the vector. If height is a matrix and beside is `FALSE` then each bar of the plot corresponds to a column of height, with the values in the column giving the heights of stacked sub-bars making up the bar. If height is a matrix and beside is `TRUE`, then the values in each column are juxtaposed rather than stacked.
`...`	Arguments to be passed to methods, such as graphical parameters (see `graphics::par()`). The most commonly used argument would be `y` for the factor to use for the y-axis, `type` describing what type of plot should be drawn, or `col` for point colour (defaults to AAGI Black), a very dark grey.

Value

A `barplot` object, returned invisibly (see `graphics::barplot()`).

Author(s)

Adam Sparks, <adam.sparks@curtin.edu.au>

See Also

- `graphics::plot()` for full documentation of the basic plotting capabilities.
- plot_aagi, boxplot_aagi, hist_aagi

Other Baseplots: `boxplot_aagi()`, `hist_aagi()`, `plot_aagi()`

Examples

```
barplot_aagi(islands)
barplot_aagi(islands, col = "AAGI Orange")
```

boxplot_aagi *Basic Boxplots Using a Unified AAGI Style and Typography*

Description

Basic boxplots that follow a standard AAGI style including typography guidelines that uses (hopefully) sensible defaults. All valid `boxplot()` options are supported through `...`, for *e.g.*, `col` to set the colour. Defaults to "AAGI Black", a very dark grey colour.

Usage

```
boxplot_aagi(x, main = "", sub = "", xlab = "", ylab = "", pch = 16, ...)
```

Arguments

`x`	for specifying data from which the boxplots are to be produced. Either a numeric vector, or a single list containing such vectors. Additional unnamed arguments specify further data as separate vectors (each corresponding to a component boxplot). `NA`s are allowed in the data.
`main`	Main title. Optional, if not supplied it will be blank.
`sub`	Sub title below x-axis label. Optional, if not supplied it will be blank.
`xlab`	X-axis label. Optional.
`ylab`	Y-axis label. Optional.
`pch`	plotting 'character', *i.e.*, symbol to use.
`...`	Arguments to be passed to methods, such as graphical parameters (see `graphics::par()`). The most commonly used argument would be `y` for the factor to use for the y-axis, `type` describing what type of plot should be drawn, or `col` for point colour (defaults to AAGI Black), a very dark grey.

Value

A `boxplot` object, returned invisibly (see `graphics::boxplot()`).

Author(s)

Adam Sparks, <adam.sparks@curtin.edu.au>

See Also

- `graphics::boxplot()` for full documentation of the basic boxplot capabilities, and
- `graphics::par()` for full documentation of `pch`.
- `graphics::boxplot()` for full documentation of the basic plotting capabilities.
- barplot_aagi, hist_aagi, plot_aagi

Other Baseplots: `barplot_aagi()`, `hist_aagi()`, `plot_aagi()`

Examples

```
boxplot_aagi(decrease ~ treatment,
  data = OrchardSprays,
  xlab = "treatment",
  ylab = "decrease"
)
```

`hist_aagi` *Basic Histograms Using a Unified AAGI Style and Typography*

Description

Basic histograms that follow a standard AAGI style including typography guidelines that uses (hopefully) sensible defaults. All valid `hist()` options are supported through `...`, for *e.g.*, `col` to set the colour. Defaults to "AAGI Black", a very dark grey colour.

Usage

```
hist_aagi(
  x,
  main = "",
  sub = "",
  xlab = "",
  ylab = "Count",
  breaks = "scott",
  ...
)
```

Arguments

`x`	A vector of values for which the histogram is desired.
`main`	Main title. Optional, if not supplied it will be blank.
`sub`	Sub title below x-axis label. Optional, if not supplied it will be blank.
`xlab`	X-axis label. Optional, if not supplied will be an empty string.
`ylab`	Y-axis label. Optional, if not supplied will default to "Count".
`breaks`	One of "scott" (default), "fd", "sturges" (R's default), or "exact" (fixed bin width of 1 for integer. counts).
`...`	Arguments to be passed to methods, such as graphical parameters (see `graphics::par()`). The most commonly used argument would be `y` for the factor to use for the y-axis, `type` describing what type of plot should be drawn, or `col` for point colour (defaults to AAGI Black), a very dark grey.

Value

A `histogram` object, returned invisibly (see `graphics::hist()`).

Author(s)

Adam Sparks, <adam.sparks@curtin.edu.au>

References

Scott, D.W. (1979) On optimal and data-based histograms. *Biometrika*, 66, 605–610.

Hyndman, R.J., The Problem with Sturges' Rule for Constructing Histograms, 1995, Monash University, https://www.robjhyndman.com/papers/sturges.pdf.

See Also

- graphics::hist() for full documentation of the basic histogram capabilities.
- barplot_aagi, boxplot_aagi, plot_aagi

Other Baseplots: barplot_aagi(), boxplot_aagi(), plot_aagi()

Examples

```
hist_aagi(islands)
```

---

plot_aagi *Basic X-Y Plotting Using a Unified AAGI Style and Typography*

Description

Basic plot that follow a standard AAGI style including typography guidelines that uses (hopefully) sensible defaults. All valid plot() options are supported through ..., for *e.g.*, col to set the colour. Defaults to "AAGI Black", a very dark grey colour.

Usage

```
plot_aagi(x, y = NULL, ...)
```

Arguments

x	the coordinates of points in the plot. Alternatively, a single plotting structure, function or any R *object with a plot method* can be provided.
y	The y coordinates of points in the plot, optional. If y is NULL, x is assumed to describe the y values, and an index is used for x.
...	Arguments to be passed to methods, such as graphical parameters (see graphics::par()). The most commonly used argument would be y for the factor to use for the y-axis, type describing what type of plot should be drawn, or col for point colour (defaults to AAGI Black), a very dark grey.

Value

A plot object, returned invisibly (see graphics::plot()).

Author(s)

Adam Sparks, <adam.sparks@curtin.edu.au>

See Also

- graphics::plot() for full documentation of the basic plotting capabilities.
- barplot_aagi, boxplot_aagi, hist_aagi

Other Baseplots: barplot_aagi(), boxplot_aagi(), hist_aagi()

Examples

```
library(datasets)

plot_aagi(airquality$Ozone,
  main = "Air Quality",
  xlab = "Ozone"
)

plot_aagi(pressure)
```

`scale_colour_aagi` *Scale Constructors for AAGI Colour Palettes to be Used in ggplot2 Objects*

Description

Helpers for applying AAGI colour palettes to **ggplot2** colour and fill aesthetics.

Usage

```
scale_colour_aagi(
  palette = NULL,
  discrete = TRUE,
  colours = NULL,
  values = NULL,
  ...
)

scale_fill_aagi(
  palette = NULL,
  discrete = TRUE,
  colours = NULL,
  values = NULL,
  ...
)
```

Arguments

palette	Character name of a palette supported by AAGIPalettes::aagi_palettes(). If NULL and discrete = TRUE, use the official colours in AAGIPalettes::aagi_colours.
discrete	Logical; should the scale be discrete? Defaults to TRUE.
colours	Character vector of official AAGI colour names to interpolate for continuous scales. If NULL, defaults used by AAGIPalettes::interpolate_aagi_colours() are applied.
values	Optional character vector of colour values to use for discrete scales. This can be a vector of hex colours or values returned by AAGIPalettes::colour_as_hex(). If supplied, values takes precedence over palette and the default ordering of AAGIPalettes::aagi_colours.
...	Additional arguments passed to the underlying **ggplot2** scale functions.

Details

By default, discrete scales use the official colours in AAGIPalettes::aagi_colours. If palette is supplied, discrete scales use a named palette from AAGIPalettes::aagi_palettes().

Continuous scales use interpolated colours via AAGIPalettes::interpolate_aagi_colours().

Functions

- scale_colour_aagi(): For colour scales
- scale_fill_aagi(): For fill scales

---

showtext_aagi_off *Disable Global Showtext Auto Rendering (AAGIThemes)*

Description

theme_aagi(showtext_auto = TRUE) enables showtext::showtext_auto(), which affects subsequent plots globally. Call this function to turn it off.

Usage

```
showtext_aagi_off()
```

Value

Invisibly returns NULL.

theme_aagi *A ggplot2 Theme Using a Unified AAGI Style and Typography for Plots*

Description

This theme follows a standard AAGI style including typography guidelines that uses (hopefully) sensible defaults for **ggplot2** graphics.

Usage

```
theme_aagi(
  base_size = 12,
  major_grid = FALSE,
  minor_grid = FALSE,
  border = FALSE,
  showtext_auto = TRUE,
  ...
)
```

Arguments

base_size	Numeric. The figure's base size, set to 12 by default. For presentations using a larger base_size will make the fonts larger and more suitable for presentation slides, *e.g.*, base_size = 24.
major_grid	Boolean. Include major gridlines in the panel. Defaults to FALSE.
minor_grid	Boolean. Include minor gridlines in the panel. Defaults to FALSE.
border	Boolean. Include a border around the figure on all four sides. Defaults to FALSE.
showtext_auto	Boolean. If TRUE, enable showtext::showtext_auto() that provides full AAGI font support. Defaults to TRUE. **Important:** showtext::showtext_auto() changes global rendering behaviour for the entire R session/device. This can affect subsequent plots (including non-**AAGIThemes** plots) until it is disabled. However, since it is anticipated that this theme will only ever be used for official AAGI work, this seems unlikely to be an issue. To undo, call showtext_aagi_off().
...	Other arguments as passed along to ggplot2::theme().

Author(s)

Adam Sparks, <adam.sparks@curtin.edu.au>

Examples

```
library("ggplot2")

p <- ggplot(mtcars) +
  geom_point(aes(wt, mpg)) +
  theme_aagi(showtext_auto = TRUE)

p

# Disable global showtext auto afterwards (recommended)
showtext_aagi_off()
```

`theme_ft_aagi` *Apply AAGI Theme to a flextable Object*

Description

Apply theme AAGI to a **flextable**. An AAGI formatted table body is grey with a teal header and white header font. Header text is bold, text columns are left aligned, other columns are right aligned.

Usage

```
theme_ft_aagi(x)
```

Arguments

x	a **flextable** object

Value

a formatted **flextable** object

Behaviour

Theme functions are not like **ggplot2** themes. They are applied to the existing table **immediately**. If you add a row in the footer, the new row is not formatted with the theme. The theme function applies the theme only to existing elements when the function is called.

That is why theme functions should be applied after all elements of the table have been added (mainly additional header or footer rows).

If you want to automatically apply a theme function to each **flextable** object, you can use the `theme_fun` argument of flextable::set_flextable_defaults; be aware that this theme function is applied as the last instruction when calling `flextable::flextable()` – so if you add headers or footers to the array, they will not be formatted with the theme.

You can also use the `post_process_html` argument of flextable::set_flextable_defaults (or `post_process_pdf`, `post_process_docx`, `post_process_pptx`) to specify a theme to be applied systematically before the flextable::flextable is printed; in this case, don't forget to take care that the theme doesn't override any formatting done before the print statement.

Author(s)

Adam H. Sparks, <adam.sparks@curtin.edu.au>

See Also

Other tables: theme_gt_aagi()

Examples

```
library(flextable)
library(dplyr)
ft <- flextable(head(airquality) |> mutate(`Month Name` = "May"))
ft <- theme_ft_aagi(ft)
ft
```

---

theme_gt_aagi *Apply the AAGI Theme to a gt Object*

Description

Apply the AAGI theme to a **gt**. An AAGI formatted table body is grey with a teal header and white header font. Header text is bold, text columns are left aligned, other columns are right aligned.

Usage

```
theme_gt_aagi(x)
```

Arguments

x	a **gt** object

Value

a formatted **gt** object

Author(s)

Adam H. Sparks, <adam.sparks@curtin.edu.au>

See Also

Other tables: theme_ft_aagi()

Examples

```
library(gt)
library(dplyr)
gt <- head(airquality) |>
  mutate(`Month Name` = "May") |>
  gt()
gt <- theme_gt_aagi(gt)
gt
```

`watermark` *Add a Watermark Annotation Layer to a ggplot2 Object*

Description

Add a Watermark Annotation Layer to a ggplot2 Object

Usage

```
watermark(
  watermark,
  fontsize = 120,
  colour = "grey90",
  alpha = 0.1,
  fontface = "bold",
  angle = 22
)
```

Arguments

`watermark`	String to be added as watermark
`fontsize`	Font size
`colour`	Font colour
`alpha`	Alpha (transparency; lower number = more transparent)
`fontface`	Font face ("bold" by default)
`angle`	Angle of the watermark

Value

A ggplot2 object with a watermark added to the plot that's called.

Author(s)

Adam H. Sparks, <adam.sparks@curtin.edu.au> based on Matt Cowgill's and Will Mackey's work.

Examples

```
library(ggplot2)
# First, start with a plot:
p <- ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  theme_aagi()

# Then add a watermark:

p + watermark("DRAFT")
```

Index