

Package: AAGIPalettes (via r-universe)

May 26, 2026

Title AAGI Colours and Colour Palettes

Version 0.0.4

Description Colour palettes based on the official Analytics for the Australian Grains Industry ('AAGI') comms guide and others designed to work harmoniously with the official 'AAGI' colours while being colour vision deficient (CVD) friendly.

License MIT + file LICENSE

URL <https://github.com/AAGI-AUS/AAGIPalettes>,
<https://AAGI-AUS.github.io/AAGIPalettes/>

BugReports <https://github.com/AAGI-AUS/AAGIPalettes/issues>

Depends R (>= 4.1.0)

Imports cli, grDevices, rlang

Suggests ggplot2, roxygen2, testthat (>= 3.0.0)

Config/Needs/build moodymudskipper/devtag

Config/roxygen2/version 8.0.0

Config/roxygen2/filename globals.R

Config/roxygen2/unique FALSE

Config/testthat/edition 3

Config/testthat/parallel true

Encoding UTF-8

Language en-US

LazyData true

Roxigen list(markdown = TRUE, roclets = c("`collate", "`namespace",
`rd", `roxygen2::global_roclet", `devtag::dev_roclet"))

X-schema.org-applicationCategory Tools

X-schema.org-isPartOf

<https://grdc.com.au/research/partnerships-and-initiatives/strategic-partnerships/aagi>

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

Date/Publication 2026-05-26 06:15:35 UTC

RemoteUrl <https://github.com/AAGI-AUS/AAGIPalettes>

RemoteRef HEAD

RemoteSha 167d3e64f220736880e70876e50a4415e3212532

Contents

aagi_colours	2
aagi_palettes	3
aagi_palettes_list	4
colour_as_hex	5
display_aagi_cols	5
interpolate_aagi_colours	7
Index	9

aagi_colours	<i>Official AAGI Colours From the AAGI Comms Guide</i>
--------------	--

Description

A Vector of the official AAGI Colour Hex Codes. Other colour palettes are provided via [aagi_palettes_list](#) for plotting purposes but may not considered to be "official" AAGI colours.

A Vector of the official AAGI Colour Hex Codes. Other colour palettes are provided via [aagi_palettes_list](#) for plotting purposes but may not considered to be "official" AAGI colours.

Usage

```
aagi_colours
```

```
aagi_colours
```

Format

A vector with eight (8) colours as hexadecimal values:

AAGI Bright Green #B6D438

AAGI Teal #00808B

AAGI Blue #648FD2

AAGI Green #54921E

AAGI Yellow #FFBC42

AAGI Orange #ec8525

AAGI Black #414042

AAGI Grey #F2F2F2

A vector with eight (8) colours as hexadecimal values:

AAGI Bright Green #B6D438

AAGI Teal #00808B

AAGI Blue #648FD2

AAGI Green #54921E

AAGI Yellow #FFBC42

AAGI Orange #ec8525

AAGI Black #414042

AAGI Grey #F2F2F2

See Also

Other data: [aagi_palettes_list](#)

aagi_palettes

Sequential and Diverging Colour Palettes for AAGI Graphical Outputs

Description

Sequential and Diverging Colour Palettes for AAGI Graphical Outputs

Usage

```
aagi_palettes(name, n = 5, direction = 1)
```

Arguments

name	Character. Name of the desired palette.
n	Integer. Number of different colours in the palette. Defaults to 5.
direction	Integer. If 1, default order. If -1, reverse order.

Value

A character vector of hexadecimal colour codes.

aagi_palettes_list *AAGI sequential and diverging colour palettes (hex codes)*

Description

A nested list of sequential and diverging AAGI colour palettes used by `aagi_palettes()`. The top-level names are palette identifiers (e.g., "aagi_RdT1"). Each palette is itself a named list keyed by the number of colour stops ("3", "4", ...). Each entry is a character vector of hex colour codes of length n.

Usage

```
aagi_palettes_list
```

Format

A named list of named lists of character vectors.

Details

This dataset is intended for internal reuse and for users who want direct access to the fixed palette definitions without calling the palette function.

Not all palettes support all values of n. For example, "aagi_RdY1" is defined only for 3–7 stops.

See Also

Other data: [aagi_colours](#)

Examples

```
# List available palettes
names(aagi_palettes_list)

# Available 'n' values for a palette
names(aagi_palettes_list[["aagi_RdT1"]])

# Get the 11-stop RdT1 palette
aagi_palettes_list[["aagi_RdT1"]][["11"]]

# Reverse order (equivalent to direction = -1)
rev(aagi_palettes_list[["aagi_RdT1"]][["11"]])
```

`colour_as_hex`*Extract AAGI Colours as Hex Codes From Names*

Description

Use plain English colour names for AAGI colours to get the hex codes.

Usage

```
colour_as_hex(  
  name = c("AAGI Black", "AAGI Bright Green", "AAGI Blue", "AAGI Teal", "AAGI Green",  
          "AAGI Yellow", "AAGI Orange", "AAGI Grey")  
)
```

Arguments

`name` Character names of AAGI colours. Multiple values are allowed.

Value

A string value that consists of the corresponding colour hex code that was requested.

Examples

```
# return AAGI Black as a hex code  
colour_as_hex(name = "AAGI Black")  
  
# return AAGI Black, Bright Green and Orange as hex codes  
colour_as_hex(name = c("AAGI Black", "AAGI Bright Green", "AAGI Orange"))
```

`display_aagi_cols`*AAGI Palette Plots*

Description

A quick and dirty way to show AAGI colours in a palette plot.

Usage

```
display_aagi_cols(name = "aagi_colours", n = 5)
```

Arguments

name	Either aagi_colours (default) or a palette name from the lists below.
n	Numeric Number of different colours in the palette, minimum 3, maximum depends on the palette. Defaults to 5. Ignored if name is aagi_colours. There are two (2) types of palettes provided, sequential and diverging. <ol style="list-style-type: none"> 1. Sequential palettes are suited to ordered data that progress from low to high. Lightness steps dominate the look of these schemes, with light colours for low data values to dark colours for high data values. 2. Diverging palettes put equal emphasis on mid-range critical values and extremes at both ends of the data range. The critical class or break in the middle of the legend is emphasized with light colours and low and high extremes are emphasized with dark colours that have contrasting hues.

The sequential palettes names are:

- “aagi_BrYl”
- “aagi_blues”,
- “aagi_bright_greens”,
- “aagi_greens”,
- “aagi_greys”,
- “aagi_reds”,
- “aagi_oranges”,
- “aagi_RdYl”,
- “aagi_teals”,
- “aagi_TlGnYl”,
- “aagi_yellows”

All the sequential palettes are available in variations from 3 different values up to 11 different values.

The diverging palettes are:

- “aagi_BuOr”,
- “aagi_BuYl”,
- “aagi_GnYl”,
- “aagi_RdBu”,
- “aagi_RdYl”,
- “aagi_TlGn”,
- “aagi_TlYl”

All the diverging palettes are available in variations from 3 different values up to 11 different values.

For qualitative palettes, the lowest number of distinct values available always is 3, but the largest number is different for different palettes. It is given together with the palette names in the following table.

Value

Called for its side-effects, a plot of the given palette.

Examples

```
# display the official AAGI colours
display_aagi_cols(name = "aagi_colours")

# display five colours from the Blue Orange diverging palette
display_aagi_cols(name = "aagi_BuOr")
```

interpolate_aagi_colours

Interpolate an AAGI Colour Palette to Expand the Colour Values

Description

Takes a AAGI colour palette and generates more colours from it, so that there are enough to make your chart. The interpolation method is set to `spline` (the default is `linear`) in an attempt to reduce the number of undesirable colours that get produced when generating many colours.

Usage

```
interpolate_aagi_colours(
  colours = c("AAGI Orange", "AAGI Yellow", "AAGI Bright Green", "AAGI Teal",
    "AAGI Blue"),
  direction = 1,
  ...
)
```

Arguments

<code>colours</code>	Character; given names of AAGI colours to use in the interpolation. Defaults to "AAGI Orange", "AAGI Yellow", "AAGI Bright Green", "AAGI Teal", "AAGI Blue" in that order.
<code>direction</code>	Integer Sets the order of colours in the scale. If 1, the default order is used. If -1, the order of colours is reversed.
<code>...</code>	Additional arguments to pass to <code>grDevices::colorRampPalette()</code> , see details here grDevices::colorRamp() .

Value

A function that takes a single value and makes that many colours.

See Also

`aagi_palettes`

Examples

```
library(ggplot2)

x <- interpolate_aagi_colours()
wt_vals <- x(length(unique(round(mtcars$wt, 1))))

ggplot(mtcars, aes(x = mpg, y = hp, colour = as.factor(round(wt, 1)))) +
  geom_point() +
  scale_colour_manual("Weight", values = wt_vals)
```

Index

* datasets

aagi_colours, 2
aagi_palettes_list, 4

* data

aagi_colours, 2
aagi_palettes_list, 4

aagi_colours, 2, 4
aagi_palettes, 3
aagi_palettes(), 4
aagi_palettes_list, 2, 3, 4

colour_as_hex, 5

display_aagi_cols, 5

grDevices::colorRamp(), 7
grDevices::colorRampPalette(), 7

interpolate_aagi_colours, 7