

# Package ‘sgplot’

May 27, 2025

**Title** Graphic Styles and Colours for Scottish Government Plots

**Version** 0.4.0

**Description** A 'ggplot2' theme and colour palettes to create accessible data visualisations in the Scottish Government.

**License** MIT + file LICENSE

**URL** <https://github.com/ScotGovAnalysis/sgplot/>,  
<https://ScotGovAnalysis.github.io/sgplot/>

**BugReports** <https://github.com/ScotGovAnalysis/sgplot/issues>

**Depends** R (>= 2.10)

**Imports** cli, ggplot2, lifecycle, rlang, scales

**Suggests** dplyr, gapminder, glue, gt, knitr, plotly, purrr, rmarkdown,  
stringr, testthat (>= 2.1.0), tibble, tidyr

**VignetteBuilder** knitr

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Scottish Government [cph, fnd],  
Alice Hannah [aut, cre]

**Maintainer** Alice Hannah <[alice.hannah@gov.scot](mailto:alice.hannah@gov.scot)>

**Repository** CRAN

**Date/Publication** 2025-05-27 08:10:02 UTC

## Contents

af_colour_values . . . . .	2
available_palettes . . . . .	2
check_palette . . . . .	3
mm_to_inch . . . . .	4

scale_colour_continuous_sg . . . . .	4
scale_colour_discrete_sg . . . . .	5
sg_colour_values . . . . .	7
sss_colour_values . . . . .	7
theme_sg . . . . .	8
use_sgplot . . . . .	9

<b>Index</b>	<b>10</b>
--------------	-----------

---

af_colour_values	<i>Government Analysis Function colours and palettes</i>
------------------	--

---

**Description**

- af\_colour\_values is a vector containing colour names and their corresponding hex codes.
- af\_colour\_palettes is a list grouping colours into palettes.

**Usage**

```
af_colour_values

af_colour_palettes
```

**Format**

An object of class character of length 9.

An object of class list of length 5.

**Source**

[Government Analysis Function Colours Guidance](#)

---

available_palettes	<i>Available palettes and palette types</i>
--------------------	---

---

**Description**

Available palettes and palette types

**Usage**

```
available_palettes(palette_type, error_call = rlang::caller_env())

available_palette_types()
```

**Arguments**

palette\_type     String.  
error\_call       Environment to reference in error messages.

**Value**

A character vector of available palettes or palette types.

**Examples**

```
available_palette_types()  
available_palettes("sg")
```

---

check_palette	<i>Check palettes and palette types</i>
---------------	---

---

**Description**

Checks values against those available in available\_palette\_types() and available\_palettes().

**Usage**

```
check_palette(palette_type, palette, error_call = rlang::caller_env())  
check_palette_type(palette_type, error_call = rlang::caller_env())
```

**Arguments**

palette\_type, palette  
                    String.  
error\_call       Environment to reference in error messages.

**Value**

The value being checked is returned invisibly if the check is successful. Otherwise the function will return an error.

**Examples**

```
check_palette_type("sg")  
check_palette("sg", "main")
```

`mm_to_inch`*Convert millimetres to inches*

---

**Description**

Convert millimetres to inches

**Usage**

```
mm_to_inch(x)
```

**Arguments**

`x`                      Numeric value in millimetres

**Value**

A numerical value in inches

**Examples**

```
mm_to_inch(100)
```

---

`scale_colour_continuous_sg`*Continuous colour/fill scales for Scottish Government plots*

---

**Description**

Continuous colour/fill scales for Scottish Government plots

**Usage**

```
scale_colour_continuous_sg(  
  palette = "sequential",  
  palette_type = "sg",  
  reverse = FALSE,  
  na_colour = "grey50",  
  guide = "colourbar",  
  ...  
)
```

```
scale_fill_continuous_sg(  
  palette = "sequential",  
  palette_type = "sg",
```

```

    reverse = FALSE,
    na_colour = "grey50",
    guide = "colourbar",
    ...
  )

```

### Arguments

palette	Name of palette to use; e.g. "main", "sequential", "focus". Default value is "sequential".
palette_type	Name of palette type to use. Defaults to "sg". For all available palette types, run <code>available_palette_types()</code> .
reverse	Boolean value to indicate whether the palette should be reversed.
na_colour	Colour to set for missing values.
guide	A name or function used to create guide. Default is "colourbar".
...	Additional arguments passed to scale type.

### Value

ggplot2 continuous colour/fill scale

### Examples

```

library(ggplot2)

ggplot(mtcars, aes(x = mpg, y = wt, colour = cyl)) +
  geom_point() +
  scale_colour_continuous_sg()

ggplot(faithfuld, aes(x = waiting, y = eruptions, fill = density)) +
  geom_raster() +
  scale_fill_continuous_sg()

```

---

scale\_colour\_discrete\_sg

*Discrete colour/fill scales for Scottish Government plots*

---

### Description

Discrete colour/fill scales for Scottish Government plots

**Usage**

```
scale_colour_discrete_sg(
  palette = "main",
  palette_type = "sg",
  reverse = FALSE,
  ...
)

scale_fill_discrete_sg(
  palette = "main",
  palette_type = "sg",
  reverse = FALSE,
  ...
)
```

**Arguments**

<code>palette</code>	Name of palette to use; e.g. "main", "sequential", "focus". Default value is "main".
<code>palette_type</code>	Name of palette type to use. Defaults to "sg". For all available palette types, run <code>available_palette_types()</code> .
<code>reverse</code>	Boolean value to indicate whether the palette should be reversed.
<code>...</code>	Additional arguments passed to scale type.

**Value**

ggplot2 discrete colour/fill scale

**Examples**

```
library(ggplot2)
library(dplyr)

economics_long %>%
  filter(variable %in% c("psavert", "uempmed")) %>%
  ggplot(aes(x = date, y = value, colour = variable)) +
  geom_line(linewidth = 1) +
  scale_colour_discrete_sg()

d <- subset(mpg, manufacturer == "ford")

ggplot(d, aes(x = class, fill = class)) +
  geom_bar() +
  scale_fill_discrete_sg()
```

---

sg_colour_values	<i>Scottish Government colours and palettes</i>
------------------	---

---

**Description**

- sg\_colour\_values is a vector containing colour names and their corresponding hex codes.
- sg\_colour\_palettes is a list grouping colours into palettes.

**Usage**

```
sg_colour_values
```

```
sg_colour_palettes
```

**Format**

An object of class character of length 11.

An object of class list of length 4.

**Source**

[Scottish Government Design System](#)

---

sss_colour_values	<i>Social Security Scotland colours and palettes</i>
-------------------	--

---

**Description**

- sss\_colour\_values is a vector containing colour names and their corresponding hex codes.
- sss\_colour\_palettes is a list grouping colours into palettes.

**Usage**

```
sss_colour_values
```

```
sss_colour_palettes
```

**Format**

An object of class character of length 12.

An object of class list of length 4.

**Source**

Contact the [Social Security Scotland Statistics mailbox](#) with any queries about these colours and palettes.

---

 theme\_sg

*Scottish Government theme for ggplot2 charts.*


---

## Description

ggplot2 theme for Scottish Government plots.

## Usage

```
theme_sg(
  base_size = 12,
  base_line_size = base_size/24,
  base_rect_size = base_size/24,
  grid = c("y", "x", "xy", "none"),
  axis = c("x", "y", "xy", "none"),
  ticks = c("xy", "x", "y", "none"),
  legend = c("right", "left", "top", "bottom", "none")
)
```

## Arguments

base_size	base font size, given in pts.
base_line_size	base size for line elements.
base_rect_size	base size for rect elements.
grid, axis, ticks	'x', 'y', 'xy' or 'none' to determine for which axes the attribute should be drawn. Grid defaults to 'y', axis to 'x', and ticks to 'xy'.
legend	'right', 'left', 'top', 'bottom', or 'none' to determine the position of the legend. Defaults to 'right'.

## Value

ggplot2 theme

## Examples

```
library(ggplot2)

p <- ggplot(mpg, aes(x = class)) + geom_bar()

p
p + theme_sg()
```



---

use_sgplot	<i>Use sgplot defaults.</i>
------------	-----------------------------

---

## Description

Set sgplot theme, colour palette and geom aesthetic defaults for ggplot2 charts.

## Usage

```
use_sgplot(palette_type = "sg", ..., default_colour = deprecated())
```

## Arguments

palette_type	Name of palette type to use. Defaults to "sg". For all available palette types, run <code>available_palette_types()</code> .
...	Arguments passed to <code>sgplot::theme_sg()</code> .
default_colour	<b>[Deprecated]</b> Use the <code>palette_type</code> argument instead.

## Value

NULL. Function is used for side effects of setting default ggplot2 theme, colour palette and geom aesthetics.

## Examples

```
library(ggplot2)

d <- subset(mpg, manufacturer == "ford")

# ggplot2 defaults
ggplot(d, aes(x = model)) + geom_bar()
ggplot(d, aes(x = model, fill = class)) + geom_bar()

# Use sgplot defaults
use_sgplot()
ggplot(d, aes(x = model)) + geom_bar()
ggplot(d, aes(x = model, fill = class, colour = class)) + geom_bar()

# Use sgplot defaults and Social Security Scotland colour palettes
use_sgplot(palette_type = "sss")
ggplot(d, aes(x = model)) + geom_bar()
ggplot(d, aes(x = model, fill = class, colour = class)) + geom_bar()
```

# Index

## \* datasets

- af\_colour\_values, [2](#)
- sg\_colour\_values, [7](#)
- sss\_colour\_values, [7](#)

af\_colour\_palettes (af\_colour\_values), [2](#)

af\_colour\_values, [2](#)

available\_palette\_types  
(available\_palettes), [2](#)

available\_palettes, [2](#)

check\_palette, [3](#)

check\_palette\_type (check\_palette), [3](#)

mm\_to\_inch, [4](#)

scale\_colour\_continuous\_sg, [4](#)

scale\_colour\_discrete\_sg, [5](#)

scale\_fill\_continuous\_sg  
(scale\_colour\_continuous\_sg), [4](#)

scale\_fill\_discrete\_sg  
(scale\_colour\_discrete\_sg), [5](#)

sg\_colour\_palettes (sg\_colour\_values), [7](#)

sg\_colour\_values, [7](#)

sss\_colour\_palettes  
(sss\_colour\_values), [7](#)

sss\_colour\_values, [7](#)

theme\_sg, [8](#)

use\_sgplot, [9](#)