Package: waratah (via r-universe)

May 21, 2026

Title Easy data viz - NSW Government colour & typography design

Version 0.1.0

Description This package allows NSW Government R users to create data visualisations in keeping with the NSW Design System. This ensures a consistent approach to data visualisations with appropriate colours and fonts.

License MIT + file LICENSE

Copyright file COPYRIGHTS + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.1)

Imports cli, rlang, ggplot2 (>= 3.5.1), ggtext (>= 0.1.2), scales (>= 1.4.0), systemfonts, extrafont, vctrs, farver

URL https://github.com/digitalnsw/nsw-r-visualisations, https://digitalnsw.github.io/nsw-r-visualisations/

BugReports https://github.com/digitalnsw/nsw-r-visualisations/issues

Suggests dplyr, forcats, ggiraph, reactable (>= 0.4.4), knitr, legendry, magick, palmerpenguins, pkgload, ragg, rmarkdown, svglite, tidyr, testthat (>= 3.0.0), colorBlindness

Config/testthat/edition 3

VignetteBuilder knitr

Config/Needs/website rmarkdown, gdtools, gfonts

Config/pak/sysreqs libfontconfig1-dev libfreetype6-dev libicu-dev libjpeg-dev libpng-dev libxml2-dev libssl-dev

Repository https://cidm-ph.r-universe.dev

Date/Publication 2026-04-21 01:19:18 UTC

RemoteUrl https://github.com/digitalnsw/nsw-r-visualisations

RemoteRef HEAD

RemoteSha c169429ca39d50a38eb020dc980eef9cc023f328

Contents

`waratah-package` *Tools for the New South Wales Design System in R*

Description

This package was inspired by the data viz work of Cara Thompson (https://github.com/cararthompson).

Author(s)

Maintainer: Zoe Baldwin <zoe.baldwin@health.nsw.gov.au> (ORCID)

Authors:

- Carl Suster <carl.suster@sydney.edu.au> (ORCID)

Other contributors:

- Shaun Nielsen <Shaun.Nielsen@health.nsw.gov.au> [contributor]
- Tyson Goddard <tyson.goddard@treasury.nsw.gov.au> [contributor]

See Also

Useful links:

- https://github.com/digitalnsw/nsw-r-visualisations
- https://digitalnsw.github.io/nsw-r-visualisations/
- Report bugs at https://github.com/digitalnsw/nsw-r-visualisations/issues

nsw_colours *Access NSW palette grids by rows and/or columns*

Description

The NSW design system colours work in grids of colour columns and tonal rows.

- `col_nsw()` allows accessing a colour grid like a matrix. For ggplot colour palettes, you'll normally want `pal_nsw()` instead.
- `define_colour_theme()` defines a new colour theme that can be used in any waratah function that accepts a `variant` parameter.

Usage

```
nsw_colours

define_colour_theme(name, parent, colours)

col_nsw(
  hue,
  tone,
  variant = getOption("waratah.colour_theme", default = "base"),
  byrow = FALSE
)
```

Arguments

`name`	name for the new colour theme.
`parent`	name of the parent grid or theme, usually `"base"` or `"aboriginal"`.
`colours`	character vector of colour column names.
`hue`	name or index of the hue - see below.
`tone`	name or index of the tone - see below.
`variant`	name of palette variant. Available options are: base, aboriginal, corporate, treasury.
`byrow`	vary `tone` faster than `hue` if `TRUE`.

Details

`hue` and `tone` work the same way as matrix indexing with `[` in that they can be used to return single or multiple entries from the grid. They can be character vectors, integer vectors, or logicals.

Anchor colours used to create the NSW colour palettes can also be used stand-alone (e.g. `nsw_colours$blue_01` is `"#002664"`).

Value

- for `col_nsw()` a named vector of colours,
- for `define_colour_theme()` nothing: this is called for its side effects.

Colour columns and tonal rows

The `"base"` variant supports:

- **hue**: greys, greens, teals, blues, purples, fucshias, reds, oranges, yellows, browns
- **tone**: dark, normal, light, pale

The `"aboriginal"` variant supports:

- **hue**: reds, oranges, browns, yellows, greens, blues, purples, greys
- **tone**: dark, normal, light, pale

Colour themes support subsets of the hues from one of the main grids in a specific order. These themes are built in:

- `"treasury"`: teals, greys, oranges, greens
- `"corporate"`: blues, reds, greys

The default variant can be specified globally with `options(waratah.colour_theme)`.

Unambiguous shortened forms are accepted, e.g. `pal_nsw(h = "red", v = "a")`.

See Also

`pal_nsw()`

Examples

```
col_nsw(h = "blue", v = "aboriginal")
col_nsw(h = c("teal", "orange"), t = 1:2)
col_nsw(h = c("teal", "orange"), t = 1:2, byrow = TRUE)
```

`pal_interleave` *Construct palette variants*

Description

Depending on the structure of your data you may wish to combine palettes according to some pattern. These helpers may come in handy.

- `pal_c()` concatenates multiple discrtee palettes.
- `pal_interleave()` interleaves colours from multiple palettes of the same size. It helps when your data are grouped and you need more flexibility than the colour grid.
- `pal_stretch()` interpolates a palette into a new discrete palette. It's useful for stretching the 4 tones into 5 or 6, at the expense of straying from the NSW grid.
- `col_contrasting()` chooses colours based on the given background colours. It helps when drawing text on top of a mapped (i.e. variable) fill aesthetic

Usage

```
pal_interleave(...)

pal_c(...)

pal_stretch(pal)

col_contrasting(colour, light = "white", dark = "black")
```

Arguments

...	two or more vectors of colours.
pal	palette object.
colour	vector of colours.
light, dark	colours to output when colour is dark or light respectively.

Value

- for col_contrasting() a vector of colours the same length as colour,
- for pal_*() a palette object.

---

pal_nsw *NSW Design System colour palettes*

Description

Palettes created using the NSW Design System. To use the Aboriginal colour grid, specify variant = "aboriginal".

Usage

```
pal_nsw(
  palette = waiver(),
  hue = NA,
  tone = NA,
  variant = getOption("waratah.colour_theme", default = "base"),
  direction = 1
)

pal_nsw_manual(colours)
```

Arguments

palette	name of a predefined palette: default, brand_default, core.
hue	name or index of the hue - see below. Ignored if palette is specified.
tone	name or index of the tone - see below. Ignored if palette is specified.
variant	name of palette variant. Available options are: base, aboriginal, corporate, treasury. Ignored unless hue or tone is specified.
direction	set to -1 to reverse the order of colours in the palette, or 1 for the original order.
colours	vector of colour names corresponding to nsw_colours.

Details

To use palettes based on the NSW Design System colour grids, either specify hue and allow the tone to vary, or specify tone to allow the hue to vary. The recommendation is to use the first two tonal rows going one colour at a time from a set of colours; this can be achieved by specifying tone = 1:2.

There are several named palettes which can be specified with palette. To create custom combinations of named colours from the design system, use pal_nsw_manual().

Value

A palette object (see palette constructors)

Colour columns and tonal rows

The "base" variant supports:

- **hue**: greys, greens, teals, blues, purples, fucshias, reds, oranges, yellows, browns
- **tone**: dark, normal, light, pale

The "aboriginal" variant supports:

- **hue**: reds, oranges, browns, yellows, greens, blues, purples, greys
- **tone**: dark, normal, light, pale

Colour themes support subsets of the hues from one of the main grids in a specific order. These themes are built in:

- "treasury": teals, greys, oranges, greens
- "corporate": blues, reds, greys

The default variant can be specified globally with options(waratah.colour_theme).

Unambiguous shortened forms are accepted, e.g. pal_nsw(h = "red", v = "a").

See Also

col_nsw()

Other palettes: pal_waratah()

Examples

```
library(scales)

pal_nsw() |> show_col()
pal_nsw(hue = "blues") |> show_col()
pal_nsw(tone = 1:2, variant = "corporate") |> show_col()
pal_nsw(tone = "light") |> show_col()
pal_nsw(tone = "normal", variant = "aboriginal") |> show_col()
pal_nsw_manual(c("blue_02", "red_01", "green_03")) |> show_col()

# you can interpolate colours by converting to a continuous scale
pal_nsw(hue = "blues") |> as_continuous_pal() |> show_col(labels = FALSE)
```

---

pal_waratah *Flexible colour palettes*

Description

Unlike pal_nsw(), pal_waratah() is not based strictly on the NSW colour palette grids. It tries to choose colours that are perceptually distinct, optionally taking into account colour-blindness when doing so.

Usage

```
pal_waratah(
  type = c("qual", "seq", "div", "pairs", "triples"),
  hue = 1,
  cvd = getOption("waratah.cvd", default = FALSE),
  variant = getOption("waratah.colour_theme", default = "base"),
  direction = 1
)
```

Arguments

type	type of palette that should be generated: • "qual" discrete qualitative palette • "seq" continuous sequential palette: a gradient spanning extreme tones of hue • "div" continuous diverging palette: a gradient spanning extreme tones of hue and a second distinct colour. • "pairs", "triples" like ‘"qual"‘ but in sets of 2 or 2 tones of each hue.
hue	main hue, either by name of index. See col_nsw() for supported values. Only used when type is "seq" or "div".
cvd	when TRUE account for colour vision disorders. Requires the colorBlindness package.
variant	name of palette variant. Available options are: base, aboriginal, corporate, treasury. Ignored unless hue or tone is specified.
direction	set to -1 to reverse the order of colours in the palette, or 1 for the original order.

Value

A palette object (see palette constructors)

See Also

Other palettes: `pal_nsw()`

Examples

```
library(scales)

pal_waratah("qual") |> show_col()
pal_waratah("pairs") |> show_col()
pal_waratah("seq", hue = "red") |> show_col(labels = FALSE)
pal_waratah("div", hue = "yellow", variant = "aboriginal") |>
  show_col(labels = FALSE)
```

---

reactable_theme *Theme for reactable interactive tables*

Description

If using reactable, this helper constructs a theme.

Usage

```
reactable_theme(
  colour = "blue_01",
  text_colour = colour,
  borderColor = colour,
  backgroundColor = "white",
  base_family = "Public Sans",
  title_family = "Public Sans",
  base_text_size = 10,
  stripedColor = NA,
  ...
)
```

Arguments

`colour`	starting colour used only to set a common default for other colour parameters.
`text_colour`	colour of all text.
`borderColor`	colour for the horizontal lines between rows.
`backgroundColor`	background colour.
`base_family`	font family for normal text.

`title_family`	font family for title text.
`base_text_size`	base text size in pt. The header size is relative to the base text size.
`stripedColor`	fill colour for alternate rows. When `NA`, this is set to a blend of `borderColor` and `backgroundColor`.
`...`	other parameters to pass to `reactable::reactableTheme()`. For examples of what can be done with reactable, see https://glin.github.io/reactable/articles/cookbook/cookbook.html.

Details

Colours can be specified as named NSW colours as in `nsw_colours`.

As normal with HTML elements, you must make sure that the final document loads the necessary fonts. See `vignette("waratah")` for instructions.

Value

A reactable theme object

Examples

```
library(reactable)

# The default theme
head(palmerpenguins::penguins, 10) |>
  reactable(theme = reactableTheme())

# Adding waratah style
head(palmerpenguins::penguins, 10) |>
  reactable(theme = reactable_theme())

# More customised styling
head(palmerpenguins::penguins, 10) |>
  reactable(
    theme = reactable_theme(
      colour = "blue_01",
      base_family = "Arial",
      text_colour = "black",
      backgroundColor = NULL
    ),
    striped = TRUE
  )
```

theme_waratah *Theme plots using NSW colours and fonts*

Description

A 'ggplot2' theme compatible with the NSW design system. It sets default colour scales, fonts, and some other styles.

Usage

```
theme_waratah(
  base_size = 11,
  base_family = "Public Sans",
  header_family = "Public Sans",
  base_line_size = base_size/22,
  base_rect_size = base_size/22,
  ink = "black",
  paper = "white",
  geom_ink = "blue_01",
  accent = "blue_02",
  variant = getOption("waratah.colour_theme", default = "base"),
  show_grid_lines = TRUE,
  void = FALSE
)
```

Arguments

base_size	base font size, given in pts.
base_family	base font family
header_family	font family for titles and headers. The default, NULL, uses theme inheritance to set the font. This setting affects axis titles, legend titles, the plot title and tag text.
base_line_size	base size for line elements
base_rect_size	base size for rect elements
ink, paper, accent	colour for foreground, background, and accented elements respectively.
geom_ink	default ink colour used by geoms for points, lines and fills.
variant	name of palette variant. Available options are: base, aboriginal, corporate, treasury.
show_grid_lines	whether to show grid lines. If FALSE, all grid lines are removed but the axis text is retained. Ignored when void is TRUE.
void	whether to hide grid lines and axes. If TRUE, all grid lines and axes are removed. This is useful when creating pie/donut charts.

Value

ggplot theme specification to add to a plot

Examples

```
library(ggplot2)
set_theme(theme_waratah())

ggplot(palmerpenguins::penguins) +
  geom_point(aes(
    x = bill_length_mm,
    y = flipper_length_mm,
    colour = species,
    size = body_mass_g
  )) +
  labs(
    caption = "Data from {palmerpenguins}",
    dictionary = c(
      bill_length_mm = "Bill length (mm)",
      flipper_length_mm = "Flipper length (mm)",
      species = "Species",
      body_mass_g = "Body mass (g)"
    )
  )
```

---

tooltip_css *Style interactive plot tooltips*

Description

CSS code required to add waratah-styled tooltips to interactive graphs created using ggiraph.

Usage

```
tooltip_css(
  background_colour = "grey_01",
  text_colour = "off_white",
  font_family = "\"Public Sans\", Arial, sans",
  font_size = "11pt"
)
```

Arguments

background_colour	tooltip background colour
text_colour	tooltip text colour
font_family	font family for text in tooltips
font_size	font size for text in tooltips

Details

As normal with HTML elements, you must make sure that the final document loads the necessary fonts. See `vignette("waratah")` for instructions.

Value

A character vector containing CSS rules

Examples

```
tooltip_css()
```

Index