

Package: doestyle (via r-universe)

May 31, 2026

Title NSW Department of Education branding for figures and tables

Version 0.4.0

Description doestyle is an R package containing functions to help produce brand-compliant figures for NSW Department of Education publications.

License MIT + file LICENSE

Imports cli, dplyr, flextable, ggplot2, palettes, officer, purrr, showtext, sysfonts

Encoding UTF-8

Roxygen list(markdown = TRUE)

Suggests fs, knitr, readr, rmarkdown, stringr, testthat (>= 3.0.0), vdiff, xml2

Config/testthat/edition 3

Depends R (>= 4.1)

LazyData true

VignetteBuilder knitr

URL <https://nsw-education.github.io/doestyle/>

BugReports <https://github.com/nsw-education/doestyle/issues>

Config/roxygen2/version 8.0.0

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

Repository <https://nsw-education.r-universe.dev>

Date/Publication 2026-05-31 11:40:37 UTC

RemoteUrl <https://github.com/nsw-education/doestyle>

RemoteRef v0.4.0

RemoteSha a10cc244cceb843964786e9e0e38ad9c1f1703f5

Contents

check_font_families	2
colourdata	3
contrast_ratio	3
doe_palettes	4
geom_downerrorbar	5
geom_uperrorbar	7
get_colours	9
list_doestyle_colours	10
public_schools	10
relative_luminance	11
scale_colour_doe	11
scale_fill_doe	13
secondary_colours	14
show_colours	14
text_colours	15
theme_doe	16
theme_doe_flextable	16

Index	18
--------------	-----------

check_font_families	<i>Activate showtext and check for presence of Public Sans font family</i>
---------------------	--

Description

[theme_doe\(\)](#) uses showtext to allow R graphics devices to render the Public Sans font family. Use [check_font_families\(\)](#) to check that Public Sans is available and to activate showtext. If Public Sans is not installed on the system or user font search paths, doestyle's embedded copy of the files will be used.

Usage

```
check_font_families()
```

Value

[check_font_families\(\)](#) does not have a return value and is called for its side effects.

See Also

[showtext::showtext_auto\(\)](#) and [Public Sans](#).

colourdata	<i>Colours provided by doestyle</i>
------------	-------------------------------------

Description

A dataset containing names, hex and RGB values, and other identifiers for each colour provided in doestyle. Its content can be visualised with [show_colours\(\)](#) or further manipulated.

Usage

```
colourdata
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 51 rows and 13 columns.

Source

- [NSW Design System](#)
- Department of Education [Brand guidelines](#)

See Also

[show_colours\(\)](#)

contrast_ratio	<i>Compute the contrast ratio between pairs of colours</i>
----------------	--

Description

Compute contrast ratios between pairs of colours, using the WCAG 2.1 formula.

Usage

```
contrast_ratio(colour1, colour2)
```

Arguments

```
colour1, colour2
```

Character vectors of hexadecimal colour values.

Details

Contrast ratio is **defined** as $\frac{L_1+0.05}{L_2+0.05}$ where L_1 and L_2 are the **relative luminances** of the lighter and darker colours being compared.

Contrast ratios range from 1 (no colour contrast) to 21 (maximum contrast). To meet **WCAG 2.1 minimum (AA) criteria for accessibility**, the contrast ratios between adjacent colours must be:

- For text, at least 4.5
- For large text (18pt or larger, or 14pt bold), at least 3
- For non-text plot elements (e.g. patterns, lines, points, borders etc.), at least 3

To meet WCAG 2.1 enhanced (AAA) contrast requirements, generally a ratio above 7 is required.

See WCAG 2.1 success criteria [1.4.3 Contrast \(Minimum\)](#), [1.4.6 Contrast \(Enhanced\)](#) and [1.4.11 Non-text Contrast](#).

Value

`contrast_ratio()` returns a numeric vector of contrast ratios, of the same length as `colour1` and `colour2`.

Examples

```
backgrounds <- doe_cols("white", "black", "blue-01", "grey-04", "black")
foregrounds <- doe_cols("black", "white", "blue-03", "grey-03", "black")

contrast_ratio(colour1 = backgrounds, colour2 = foregrounds)
```

<code>doe_palettes</code>	<i>Palettes provided by doestyle</i>
---------------------------	--------------------------------------

Description

A list of palettes, organised by type.

Usage

```
doe_palettes
```

Format

An object of class `palettes_palette` (inherits from `vctrs_list_of`, `vctrs_vctr`, `list`) of length 6.

Details

- Qualitative: designed to show differences between nominal classes
- Sequential: designed to show differences in ordered data that progress in value from low to high (or vice versa)
- Diverging: designed to show differences in ordered data that progress away from a central value to extremes at either end of a scale

Palettes can be accessed by their type and name.

Examples

```
# Examine one palette at a time
doe_palettes$default
doe_palettes$seq.blues

# List all palettes and their colours
print(doe_palettes)

# If you've loaded `palettes`, visualise the available palettes
library(palettes)
plot(doe_palettes)
```

geom_downerrorbar *Add a down errorbar to a ggplot2 plot*

Description

Add the lower portion of an errorbar to a ggplot plot. This allows the halves of the errorbars to be different colours.

Usage

```
geom_downerrorbar(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

mapping	Set of aesthetic mappings created by aes() . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If <code>NULL</code> , the default, the data is inherited from the plot data as specified in the call to ggplot() . A <code>data.frame</code> , or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code> , and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).
stat	The statistical transformation to use on the data for this layer. When using a <code>geom_*()</code> function to construct a layer, the <code>stat</code> argument can be used to override the default coupling between geoms and stats. The <code>stat</code> argument accepts the following: <ul style="list-style-type: none"> • A Stat ggproto subclass, for example <code>StatCount</code>. • A string naming the stat. To give the stat as a string, strip the function name of the <code>stat_</code> prefix. For example, to use <code>stat_count()</code>, give the stat as "count". • For more information and other ways to specify the stat, see the layer stat documentation.
position	A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The <code>position</code> argument accepts the following: <ul style="list-style-type: none"> • The result of calling a position function, such as <code>position_jitter()</code>. This method allows for passing extra arguments to the position. • A string naming the position adjustment. To give the position as a string, strip the function name of the <code>position_</code> prefix. For example, to use <code>position_jitter()</code>, give the position as "jitter". • For more information and other ways to specify the position, see the layer position documentation.
...	Other parameters passed to ggplot2::geom_errorbar() .
na.rm	If <code>FALSE</code> , the default, missing values are removed with a warning. If <code>TRUE</code> , missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? <code>NA</code> , the default, includes if any aesthetics are mapped. <code>FALSE</code> never includes, and <code>TRUE</code> always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use <code>TRUE</code> . If <code>NA</code> , all levels are shown in legend, but unobserved levels are omitted.
inherit.aes	If <code>FALSE</code> , overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. annotation_borders() .

Examples

```
library(doestyle)
library(ggplot2)

df <- data.frame(
  observation = c("A", "B", "C"),
  lower = c(1, 2, 3),
  estimate = c(2, 4, 4),
  upper = c(3, 6, 5)
)

ggplot(df, aes(x = observation,
               y = estimate,
               ymin = lower,
               ymax = upper)) +
  geom_col() +
  geom_uperrorbar(width = 1/10) +
  geom_downerrorbar(width = 1/10,
                   colour = "white")
```

geom_uperrorbar

Add an up errorbar to a ggplot2 plot

Description

Add the upper portion of an errorbar to a ggplot plot. This allows the halves of the errorbars to be different colours.

Usage

```
geom_uperrorbar(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

mapping Set of aesthetic mappings created by `aes()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.

data	<p>The data to be displayed in this layer. There are three options:</p> <p>If NULL, the default, the data is inherited from the plot data as specified in the call to <code>ggplot()</code>.</p> <p>A <code>data.frame</code>, or other object, will override the plot data. All objects will be fortified to produce a data frame. See <code>fortify()</code> for which variables will be created.</p> <p>A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code>, and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).</p>
stat	<p>The statistical transformation to use on the data for this layer. When using a <code>geom_*()</code> function to construct a layer, the <code>stat</code> argument can be used to override the default coupling between geoms and stats. The <code>stat</code> argument accepts the following:</p> <ul style="list-style-type: none"> • A Stat ggproto subclass, for example <code>StatCount</code>. • A string naming the stat. To give the stat as a string, strip the function name of the <code>stat_</code> prefix. For example, to use <code>stat_count()</code>, give the stat as "count". • For more information and other ways to specify the stat, see the layer stat documentation.
position	<p>A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The <code>position</code> argument accepts the following:</p> <ul style="list-style-type: none"> • The result of calling a position function, such as <code>position_jitter()</code>. This method allows for passing extra arguments to the position. • A string naming the position adjustment. To give the position as a string, strip the function name of the <code>position_</code> prefix. For example, to use <code>position_jitter()</code>, give the position as "jitter". • For more information and other ways to specify the position, see the layer position documentation.
...	Other parameters passed to <code>ggplot2::geom_errorbar()</code> .
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use TRUE. If NA, all levels are shown in legend, but unobserved levels are omitted.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. <code>annotation_borders()</code> .

Examples

```
library(doestyle)
library(ggplot2)
```

```
df <- data.frame(
  observation = c("A", "B", "C"),
  lower = c(1, 2, 3),
  estimate = c(2, 4, 4),
  upper = c(3, 6, 5)
)

ggplot(df, aes(x = observation,
               y = estimate,
               ymin = lower,
               ymax = upper)) +
  geom_col() +
  geom_uperrorbar(width = 1/10) +
  geom_downerrorbar(width = 1/10,
                    colour = "white")
```

get_colours

Get brand colours by name

Description

Get brand colours by name

Usage

```
get_colours(...)
```

```
doe_colours(...)
```

```
doe_cols(...)
```

Arguments

... Colour name(s). May be supplied as a character vector or as individual arguments.

Details

get_colours() will only return hex values for valid NSW and Department of Education colour names. To list available colours, use [list_doestyle_colours\(\)](#). To plot available colours, use [show_colours\(\)](#).

Value

get_colours() returns a character vector containing the hex values for each requested brand colour. If the input contained named arguments, their names will be used for the outputs.

Examples

```
get_colours("blue-01", "red-02", "blue-04")
get_colours(c("red-01", "red-02", "red-03", "red-04"))
```

```
list_doestyle_colours
```

List valid NSW or DoE colours and their hex values

Description

List valid NSW or DoE colours and their hex values

Usage

```
list_doestyle_colours()
```

Value

A tibble containing the names of valid colours and their hex values.

Examples

```
list_doestyle_colours()
list_doestyle_colours() |> print(n = 50)
```

```
public_schools
```

The New South Wales Education Public Schools Master Dataset

Description

The master dataset contains comprehensive information for all government schools in NSW. Data items include school locations, latitude and longitude coordinates, school type, student enrolment numbers, electorate information, contact details and more. This dataset is publicly available through the [Data NSW website](https://data.nsw.gov.au/data/dataset/nsw-education-nsw-public-schools-master-dataset).

Usage

```
public_schools
```

Format

An object of class `spec_tbl_df` (inherits from `tbl_df`, `tbl`, `data.frame`) with 2219 rows and 45 columns.

Source

<https://data.nsw.gov.au/data/dataset/nsw-education-nsw-public-schools-master-dataset>

relative_luminance *Compute the relative luminance of an sRGB colour*

Description

Compute the relative luminance of sRGB colour(s). See [WCAG 2.1 definition: relative luminance](#).

Usage

```
relative_luminance(hex_values)
```

Arguments

hex_values A character vector of hex value(s).

Value

relative_luminance() returns a numeric vector of luminance values.

Examples

```
relative_luminance(c("#ffffff", "#000000", "#002664", "#d7153a"))
relative_luminance(doe_cols("white", "black", "blue-01", "red-02"))
```

scale_colour_doe *Apply a DoE palette to the colour aesthetic*

Description

Apply a DoE palette to the colour aesthetic

Usage

```
scale_colour_doe(
  palette = doe_palettes$default,
  discrete = TRUE,
  direction = 1,
  ...
)
```

Arguments

palette	An object of class <code>palettes_palette</code> . Several useful palettes are ready to use in the list <code>doe_palettes</code> , which is included with the <code>doestyle</code> package. The default is <code>doe_palettes\$qualitative\$default</code> , which uses all of the on-brand colours defined in the NSW Department of Education Brand Guidelines.
discrete	Logical indicating whether the desired scale should be discrete. The default is <code>TRUE</code> . If <code>FALSE</code> , a continuous scale with interpolated colours will be constructed, instead.
direction	Sets the order of colours in the scale. If 1, the default, colours are ordered from first to last as they appear in the palette. If -1, the order of colours is reversed.
...	Additional arguments passed to <code>ggplot2::discrete_scale()</code> or <code>ggplot2::continuous_scale()</code> .

Value

A scale function that can be added to a `ggplot2` plot.

See Also

- `doe_palettes` for the palettes available in `doestyle`.
- `palettes::scale_colour_palette_d()`
- `palettes::scale_colour_palette_c()`

Examples

```
library(dplyr)
library(ggplot2)
library(doestyle)

# Plot growth in primary and secondary schools over time
public_schools |>
  filter(Level_of_schooling %in% c("Primary School", "Secondary School")) |>
  arrange(Date_1st_teacher) |>
  mutate(
    count = 1,
    cumulative_schools = cumsum(count),
    .by = Level_of_schooling) |>
  ggplot(aes(x = Date_1st_teacher,
             y = cumulative_schools,
             colour = Level_of_schooling)) +
  geom_step() +
  # Add a default DoE colour scale
  scale_colour_doe()
```

scale_fill_doe	<i>Apply a DoE palette to the fill aesthetic</i>
----------------	--

Description

Apply a DoE palette to the fill aesthetic

Usage

```
scale_fill_doe(  
  palette = doe_palettes$default,  
  discrete = TRUE,  
  direction = 1,  
  ...  
)
```

Arguments

palette	An object of class <code>palettes_palette</code> . Several useful palettes are ready to use in the list <code>doe_palettes</code> , which is included with the <code>doestyle</code> package. The default is <code>doe_palettes\$qualitative\$default</code> , which uses all of the on-brand colours defined in the NSW Department of Education Brand Guidelines.
discrete	Logical indicating whether the desired scale should be discrete. The default is <code>TRUE</code> . If <code>FALSE</code> , a continuous scale with interpolated colours will be constructed, instead.
direction	Sets the order of colours in the scale. If 1, the default, colours are ordered from first to last as they appear in the palette. If -1, the order of colours is reversed.
...	Additional arguments passed to <code>ggplot2::discrete_scale()</code> or <code>ggplot2::continuous_scale()</code> .

Value

A scale function that can be added to a `ggplot2` plot.

See Also

- `doe_palettes` for the palettes available in `doestyle`.
- `palettes::scale_fill_palette_d()`
- `palettes::scale_fill_palette_c()`

Examples

```
library(doestyle)  
library(dplyr)  
library(ggplot2)  
  
public_schools |>
```

```

filter(grepl("Connected Communities", Principal_network)) |>
ggplot(aes(x = Principal_network, fill = Level_of_schooling)) +
geom_bar(colour = "black", position = position_dodge(preserve = "single")) +
theme(legend.position = "bottom") +
labs(y = "Schools") +
# Add a default department scale to the `fill` aesthetic.
scale_fill_doe()

```

secondary_colours *Get secondary colours for named brand colours*

Description

Secondary colours are on-brand colours that can be used for pattern fills or other visual elements that need contrast against an on-brand fill colour. Each NSW Design System colour has a corresponding secondary colour available via `secondary_colours()`.

Usage

```
secondary_colours(...)
```

Arguments

... Colour name(s). May be supplied as a character vector or as individual arguments.

Value

`secondary_colours()` returns the hex value of an appropriate secondary colour for each on-brand colour name supplied in its arguments.

Examples

```
secondary_colours("red-02", "blue-01", "blue-04")
```

show_colours *Show swatches for doestyle colours*

Description

`show_colours()` produces a plot that shows and names the colours available in doestyle.

Usage

```
show_colours(swatches = c("brand", "primary", "nsw", "all"), linewidth = 1)
```

Arguments

swatches	The name of a set of colours to display. "brand" displays the department's brand colours only; "primary" shows the department's primary brand colours only; "nsw" shows all NSW Government brand colours; and "all" shows every colour included in doestyle, including additional off-brand colours for UX element specified in the department's Brand guidelines document.
linewidth	The thickness of each swatch's coloured border.

Value

show_colours() returns a ggplot2 gg object, which can be further manipulated with user-specified themes etc.

Examples

```
show_colours()
```

text_colours	<i>Get text colours for named brand colours</i>
--------------	---

Description

Text colours are designed for plotting text over an on-brand fill colour. Each NSW Design System colour has a corresponding text colour available via text_colours().

Usage

```
text_colours(...)
```

Arguments

... Colour name(s). May be supplied as a character vector or as individual arguments.

Value

secondary_colours() returns the hex value of an appropriate text colour for each on-brand colour name supplied in its arguments.

Examples

```
text_colours("red-02", "blue-01", "blue-04")
```

theme_doe	<i>Apply the DoE theme to a ggplot2 plot</i>
-----------	--

Description

Apply a minimalist DoE theme. By default, sets the font family to Public Sans and enables `showtext::showtext_auto()` to allow easy font display in plots.

Usage

```
theme_doe(base_size = 16, base_family = "Public Sans", ...)
```

Arguments

base_size	Base font size, given in pts.
base_family	Base font family. Defaults to the NSW Government on-brand font family, Public Sans .
...	Other parameters passed to <code>ggplot2::theme()</code> .

Examples

```
library(doestyle)
library(ggplot2)

public_schools |>
  ggplot(aes(y = Operational_directorate)) +
  geom_bar() +
  # Apply theme_doe(): minimal decoration and Public Sans typeface
  theme_doe()
```

theme_doe_flextable	<i>The DoE flextable theme</i>
---------------------	--------------------------------

Description

Apply a DoE brand-compliant theme to a flextable object.

Usage

```
theme_doe_flextable(x)

apply_DoE_zebra_format(x)
```

Arguments

x	a flextable object
---	--------------------

Examples

```
head(mtcars) |>  
  flextable::flextable() |>  
  theme_doe_flextable()
```

Index

- * **datasets**
 - colourdata, 3
 - doe_palettes, 4
 - public_schools, 10
- aes(), 6, 7
- annotation_borders(), 6, 8
- apply_DoE_zebra_format
 - (theme_doe_flextable), 16
- check_font_families, 2
- colourdata, 3
- contrast_ratio, 3
- doe_colours (get_colours), 9
- doe_cols (get_colours), 9
- doe_palettes, 4, 12, 13
- fortify(), 6, 8
- geom_downerrorbar, 5
- geom_uperrorbar, 7
- get_colours, 9
- ggplot(), 6, 8
- ggplot2, 12, 13
- ggplot2::continuous_scale(), 12, 13
- ggplot2::discrete_scale(), 12, 13
- ggplot2::geom_errorbar(), 6, 8
- ggplot2::theme(), 16
- layer position, 6, 8
- layer stat, 6, 8
- list_doestyle_colours, 10
- list_doestyle_colours(), 9
- palettes::scale_colour_palette_c(), 12
- palettes::scale_colour_palette_d(), 12
- palettes::scale_fill_palette_c(), 13
- palettes::scale_fill_palette_d(), 13
- palettes_palette, 12, 13
- public_schools, 10
- relative_luminance, 11
- scale_colour_doe, 11
- scale_fill_doe, 13
- secondary_colours, 14
- show_colours, 14
- show_colours(), 3, 9
- showtext::showtext_auto(), 2, 16
- text_colours, 15
- theme_doe, 16
- theme_doe(), 2
- theme_doe_flextable, 16