Package 'nflplotR'

October 13, 2022

Title NFL Logo Plots in 'ggplot2'

Version 1.1.0

Description A set of functions to visualize National Football League analysis in 'ggplot2'.

License MIT + file LICENSE

URL https://nflplotr.nflverse.com, https://github.com/nflverse/nflplotR

BugReports https://github.com/nflverse/nflplotR/issues

Depends R (>= 3.5.0)

Imports cachem (>= 1.0.0), cli (>= 3.0.0), data.table (>= 1.14.0), ggplot2 (>= 3.3.0), grid, magick (>= 2.7.3), magrittr (>= 2.0.0), memoise (>= 2.0.0), nflreadr (>= 1.1.2), rappdirs (>= 0.3.0), rlang (>= 0.4.11), scales (>= 1.1.0)

Suggests covr, dplyr (>= 1.0.0), ggtext (>= 0.1.1), gridtext (>= 0.1.4), knitr, rmarkdown, rstudioapi (>= 0.13), rsvg (>= 2.0), sjmisc (>= 2.8.7), testthat (>= 3.0.0), vdiffr (>= 1.0.2)

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.2.1

NeedsCompilation no

Author Sebastian Carl [aut, cre]

Maintainer Sebastian Carl <mrcaseb@gmail.com>

Repository CRAN

Date/Publication 2022-08-11 23:10:02 UTC

R topics documented:

`.nflplotR_clear_cache` *Clear nflplotR Cache*

Description

This function clears the memoised cache of all functions memoised by `nflplotR`.

Usage

```
.nflplotR_clear_cache()
```

Value

Invisibly `NULL`

Examples

```
.nflplotR_clear_cache()
```

`element` *Theme Elements for Image Grobs*

Description

In conjunction with the ggplot2::theme system, the following `element_` functions enable images in non-data components of the plot, e.g. axis text.

- `element_nfl_logo()`: draws NFL team logos instead of their abbreviations.
- `element_nfl_wordmark()`: draws NFL team wordmarks instead of their abbreviations.
- `element_nfl_headshot()`: draws NFL player headshots instead of their GSIS IDs.
- `element_path()`: draws images from valid image URLs instead of the URL.

Usage

```
element_nfl_logo(
  alpha = NULL,
  colour = NA,
  hjust = NULL,
  vjust = NULL,
  color = NULL,
  size = 0.5
)

element_nfl_wordmark(
  alpha = NULL,
  colour = NA,
  hjust = NULL,
  vjust = NULL,
  color = NULL,
  size = 0.5
)

element_nfl_headshot(
  alpha = NULL,
  colour = NA,
  hjust = NULL,
  vjust = NULL,
  color = NULL,
  size = 0.5
)

element_path(
  alpha = NULL,
  colour = NA,
  hjust = NULL,
  vjust = NULL,
  color = NULL,
  size = 0.5
)
```

Arguments

`alpha`	The alpha channel, i.e. transparency level, as a numerical value between 0 and 1.
`colour, color`	The image will be colorized with this color. Use the special character `"b/w"` to set it to black and white. For more information on valid color names in ggplot2 see https://ggplot2.tidyverse.org/articles/ggplot2-specs.html?q=colour#colour-and-fill.
`hjust, vjust`	The horizontal and vertical adjustment respectively. Must be a numerical value between 0 and 1.
`size`	The output grob size in `cm` (!).

Details

The elements translate NFL team abbreviations or NFL player GSIS IDs into logo images or player headshots, respectively.

Value

An S3 object of class `element`.

See Also

`geom_nfl_logos()`, `geom_nfl_headshots()`, `geom_nfl_wordmarks()`, and `geom_from_path()` for more information on valid team abbreviations, player IDs, and other parameters.

The examples on https://nflplotr.nflverse.com/articles/nflplotR.html

Examples

```
library(nflplotR)
library(ggplot2)

team_abbr <- valid_team_names()
# remove conference logos from this example
team_abbr <- team_abbr[!team_abbr %in% c("AFC", "NFC", "NFL")]

df <- data.frame(
  random_value = runif(length(team_abbr), 0, 1),
  teams = team_abbr
)

# use logos for x-axis
ggplot(df, aes(x = teams, y = random_value)) +
  geom_col(aes(color = teams, fill = teams), width = 0.5) +
  scale_color_nfl(type = "secondary") +
  scale_fill_nfl(alpha = 0.4) +
  theme_minimal() +
  theme(axis.text.x = element_nfl_logo())

# use logos for y-axis
ggplot(df, aes(y = teams, x = random_value)) +
  geom_col(aes(color = teams, fill = teams), width = 0.5) +
  scale_color_nfl(type = "secondary") +
  scale_fill_nfl(alpha = 0.4) +
  theme_minimal() +
  theme(axis.text.y = element_nfl_logo())

################################################################################
# Headshot Examples
################################################################################
library(nflplotR)
library(ggplot2)
```

```
# Silence an nflreadr message that is irrelevant here
old <- options(nflreadr.cache_warning = FALSE)

dfh <- data.frame(
  random_value = runif(9, 0, 1),
  player_gsis = c("00-0033873",
                  "00-0026498",
                  "00-0035228",
                  "00-0031237",
                  "00-0036355",
                  "00-0019596",
                  "00-0033077",
                  "00-0012345",
                  "00-0031280")
)

# use headshots for x-axis
ggplot(dfh, aes(x = player_gsis, y = random_value)) +
  geom_col(width = 0.5) +
  theme_minimal() +
  theme(axis.text.x = element_nfl_headshot(size = 1))

# use headshots for y-axis
ggplot(dfh, aes(y = player_gsis, x = random_value)) +
  geom_col(width = 0.5) +
  theme_minimal() +
  theme(axis.text.y = element_nfl_headshot(size = 1))

# Restore old options
options(old)

################################################################################
# Wordmarks and other Images
################################################################################

library(ggplot2)

df <- dplyr::mutate(mtcars,
  team = sample(c("LAC", "BUF", "DAL", "ARI"), nrow(mtcars), TRUE),
  player = sample(
  c("00-0033873", "00-0035228", "00-0036355", "00-0019596"),
  nrow(mtcars),
  TRUE
  )
)

ggplot(df, aes(x = mpg, y = disp)) +
  geom_point() +
  facet_wrap(vars(team)) +
  labs(
    title = tools::toTitleCase("These are random teams and data"),
    subtitle = "I just want to show how the nflplotR theme elements work",
    caption = "https://github.com/nflverse/nflseedR/raw/master/man/figures/caption.png"
```

```
) +
theme_minimal() +
theme(
  plot.title.position = "plot",
  plot.title = element_text(face = "bold"),
  axis.title = element_blank(),
  # make wordmarks of team abbreviations
  strip.text = element_nfl_wordmark(size = 1),
  # load image from url in caption
  plot.caption = element_path(hjust = 1, size = 0.4)
)
```

`geom_from_path` *ggplot2 Layer for Visualizing Images from URLs or Local Paths*

Description

This geom is used to plot NFL images instead of points in a ggplot. It requires x, y aesthetics as well as a path.

Usage

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

Arguments

`mapping`	Set of aesthetic mappings created by `aes()` or `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`	The data to be displayed in this layer. There are three options: If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`. A `data.frame`, 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 `data.frame`, and will be used as the layer data. A `function` can be created from a `formula` (e.g. `~ head(.x, 10)`).

stat	The statistical transformation to use on the data for this layer, as a string.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
...	Other arguments passed on to ggplot2::layer(). These are often aesthetics, used to set an aesthetic to a fixed value. See the below section "Aesthetics" for a full list of possible arguments.
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.
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. borders().

Value

A ggplot2 layer (ggplot2::layer()) that can be added to a plot created with ggplot2::ggplot().

Aesthetics

geom_nfl_logos() understands the following aesthetics (required aesthetics are in bold):

- **x** - The x-coordinate.
- **y** - The y-coordinate.
- **path** - a file path, url, raster object or bitmap array. See magick::image_read() for further information.
- alpha = NULL - The alpha channel, i.e. transparency level, as a numerical value between 0 and 1.
- colour = NULL - The image will be colorized with this colour. Use the special character "b/w" to set it to black and white. For more information on valid colour names in ggplot2 see https://ggplot2.tidyverse.org/articles/ggplot2-specs.html?q=colour#colour-and-fill
- angle = 0 - The angle of the image as a numerical value between 0° and 360°.
- hjust = 0.5 - The horizontal adjustment relative to the given x coordinate. Must be a numerical value between 0 and 1.
- vjust = 0.5 - The vertical adjustment relative to the given y coordinate. Must be a numerical value between 0 and 1.
- width = 1.0 - The desired width of the image in npc (Normalised Parent Coordinates). The default value is set to 1.0 which is *big* but it is necessary because all used values are computed relative to the default. A typical size is width = 0.1 (see below examples).
- height = 1.0 - The desired height of the image in npc (Normalised Parent Coordinates). The default value is set to 1.0 which is *big* but it is necessary because all used values are computed relative to the default. A typical size is height = 0.1 (see below examples).

Examples

```
library(ggplot2)
library(nflplotR)

# create x-y-coordinates of a pentagon and add nflverse logo urls
df <- data.frame(
  a = c(sin(2 * pi * (0:4) / 5), 0),
  b = c(cos(2 * pi * (0:4) / 5), 0),
  url = c(
    "https://github.com/nflverse/nflfastR/raw/master/man/figures/logo.png",
    "https://github.com/nflverse/nflseedR/raw/master/man/figures/logo.png",
    "https://github.com/nflverse/nfl4th/raw/master/man/figures/logo.png",
    "https://github.com/nflverse/nflreadr/raw/main/data-raw/logo.svg",
    "https://github.com/nflverse/nflplotR/raw/main/man/figures/logo.png",
    "https://github.com/nflverse/nflverse/raw/main/man/figures/logo.png"
  )
)

# plot images directly from url
ggplot(df, aes(x = a, y = b)) +
  geom_from_path(aes(path = url), width = 0.15) +
  coord_cartesian(xlim = c(-2, 2), ylim = c(-1.3, 1.5)) +
  theme_void()

# plot images directly from url and apply transparency
ggplot(df, aes(x = a, y = b)) +
  geom_from_path(aes(path = url), width = 0.15, alpha = 0.5) +
  coord_cartesian(xlim = c(-2, 2), ylim = c(-1.3, 1.5)) +
  theme_void()

# It is also possible and recommended to use the underlying Geom inside a
# ggplot2 annotation
ggplot() +
  annotate(
    nflplotR::GeomFromPath,
    x = 0,
    y = 0,
    path = "https://github.com/nflverse/nflplotR/raw/main/man/figures/logo.png",
    width = 0.4
  ) +
  theme_minimal()
```

---

geom_lines *ggplot2 Layer for Horizontal and Vertical Reference Lines*

Description

These geoms can be used to draw horizontal or vertical reference lines in a ggplot. They use the data in the aesthetics `v_var` and `h_var` to compute their `median` or `mean` and draw the as a line.

Usage

```
geom_median_lines(
  mapping = NULL,
  data = NULL,
  ...,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

geom_mean_lines(
  mapping = NULL,
  data = NULL,
  ...,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

mapping	Set of aesthetic mappings created by aes() or aes_().
data	The data to be displayed in this layer. There are three options: If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot(). A data.frame, 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 data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)).
...	Other arguments passed on to layer(). These are often aesthetics, used to set an aesthetic to a fixed value, like colour = "red" or size = 3. They may also be parameters to the paired geom/stat.
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.
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 behavior from the default plot specification.

Value

A ggplot2 layer (ggplot2::layer()) that can be added to a plot created with ggplot2::ggplot().

Aesthetics

`geom_median_lines()` and `geom_mean_lines()` understand the following aesthetics (at least one of the bold aesthetics is required):

- **v_var** - The variable for which to compute the median/mean that is drawn as vertical line.
- **h_var** - The variable for which to compute the median/mean that is drawn as horizontal line.
- `alpha = NA` - The alpha channel, i.e. transparency level, as a numerical value between 0 and 1.
- `color = "red"` - The color of the drawn lines.
- `linetype = 2` - The linetype of the drawn lines.
- `size = 0.5` - The size of the drawn lines.

See Also

The underlying ggplot2 geoms `geom_hline()` and `geom_vline()`

Examples

```
library(nflplotR)
library(ggplot2)

# inherit top level aesthetics
ggplot(mtcars, aes(x = disp, y = mpg, h_var = mpg, v_var = disp)) +
  geom_point() +
  geom_median_lines() +
  geom_mean_lines(color = "blue") +
  theme_minimal()

# draw horizontal line only
ggplot(mtcars, aes(x = disp, y = mpg, h_var = mpg)) +
  geom_point() +
  geom_median_lines() +
  geom_mean_lines(color = "blue") +
  theme_minimal()

# draw vertical line only
ggplot(mtcars, aes(x = disp, y = mpg, v_var = disp)) +
  geom_point() +
  geom_median_lines() +
  geom_mean_lines(color = "blue") +
  theme_minimal()

# choose your own value
ggplot(mtcars, aes(x = disp, y = mpg)) +
  geom_point() +
  geom_median_lines(v_var = 400, h_var = 15) +
  geom_mean_lines(v_var = 150, h_var = 30, color = "blue") +
  theme_minimal()
```

`geom_nfl_headshots` *ggplot2 Layer for Visualizing NFL Player Headshots*

Description

This geom is used to plot NFL player headshots instead of points in a ggplot. It requires x, y aesthetics as well as a valid NFL player gsis id.

Usage

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

Arguments

`mapping`	Set of aesthetic mappings created by `aes()` or `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`	The data to be displayed in this layer. There are three options: If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`. A `data.frame`, 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 `data.frame`, and will be used as the layer data. A `function` can be created from a `formula` (e.g. `~ head(.x, 10)`).
`stat`	The statistical transformation to use on the data for this layer, as a string.
`position`	Position adjustment, either as a string, or the result of a call to a position adjustment function.
`...`	Other arguments passed on to `ggplot2::layer()`. These are often aesthetics, used to set an aesthetic to a fixed value. See the below section "Aesthetics" for a full list of possible arguments.
`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.
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. borders().

Value

A ggplot2 layer (ggplot2::layer()) that can be added to a plot created with ggplot2::ggplot().

Aesthetics

geom_nfl_headshots() understands the following aesthetics (required aesthetics are in bold):

- **x** - The x-coordinate.
- **y** - The y-coordinate.
- **player_gsis** - The players' NFL gsis id.
- alpha = NULL - The alpha channel, i.e. transparency level, as a numerical value between 0 and 1.
- colour = NULL - The image will be colorized with this colour. Use the special character "b/w" to set it to black and white. For more information on valid colour names in ggplot2 see https://ggplot2.tidyverse.org/articles/ggplot2-specs.html?q=colour#colour-and-fill
- angle = 0 - The angle of the image as a numerical value between 0° and 360°.
- hjust = 0.5 - The horizontal adjustment relative to the given x coordinate. Must be a numerical value between 0 and 1.
- vjust = 0.5 - The vertical adjustment relative to the given y coordinate. Must be a numerical value between 0 and 1.
- width = 1.0 - The desired width of the image in npc (Normalised Parent Coordinates). The default value is set to 1.0 which is *big* but it is necessary because all used values are computed relative to the default. A typical size is width = 0.075 (see below examples).
- height = 1.0 - The desired height of the image in npc (Normalised Parent Coordinates). The default value is set to 1.0 which is *big* but it is necessary because all used values are computed relative to the default. A typical size is height = 0.1 (see below examples).

Examples

```
library(nflplotR)
library(ggplot2)
# Silence an nflreadr message that is irrelevant here
old <- options(nflreadr.cache_warning = FALSE)

df <- data.frame(
  a = rep(1:3, 3),
  b = sort(rep(1:3, 3), decreasing = TRUE),
```

```
  player_gsis = c("00-0033873",
                  "00-0026498",
                  "00-0035228",
                  "00-0031237",
                  "00-0036355",
                  "00-0019596",
                  "00-0033077",
                  "00-0012345",
                  "00-0031280"),
  player_name = c("P.Mahomes",
                  "M.Stafford",
                  "K.Murray",
                  "T.Bridgewater",
                  "J.Herbert",
                  "T.Brady",
                  "D.Prescott",
                  "Non.Match",
                  "D.Carr")
)

# set a custom fill colour for one player
df$colour <- ifelse(df$a == 2 & df$b == 2, NA, "b/w")

# scatterplot of the headshots
ggplot(df, aes(x = a, y = b)) +
  geom_nfl_headshots(aes(player_gsis = player_gsis), height = 0.2) +
  geom_label(aes(label = player_name), nudge_y = -0.35, alpha = 0.5) +
  coord_cartesian(xlim = c(0.75, 3.25), ylim = c(0.7, 3.25)) +
  theme_void()

# apply alpha as constant
ggplot(df, aes(x = a, y = b)) +
  geom_nfl_headshots(aes(player_gsis = player_gsis), height = 0.2, alpha = 0.5) +
  geom_label(aes(label = player_name), nudge_y = -0.35, alpha = 0.5) +
  coord_cartesian(xlim = c(0.75, 3.25), ylim = c(0.7, 3.25)) +
  theme_void()

# apply colour as an aesthetic
ggplot(df, aes(x = a, y = b)) +
  geom_nfl_headshots(aes(player_gsis = player_gsis, colour = colour), height = 0.2) +
  geom_label(aes(label = player_name), nudge_y = -0.35, alpha = 0.5) +
  coord_cartesian(xlim = c(0.75, 3.25), ylim = c(0.7, 3.25)) +
  scale_colour_identity() +
  theme_void()

# Restore old options
options(old)
```

`geom_nfl_logos` *ggplot2 Layer for Visualizing NFL Team Logos*

Description

This geom is used to plot NFL team and conference logos instead of points in a ggplot. It requires x, y aesthetics as well as a valid NFL team abbreviation. The latter can be checked with `valid_team_names()`.

Usage

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

Arguments

`mapping`	Set of aesthetic mappings created by `aes()` or `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`	The data to be displayed in this layer. There are three options: If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`. A `data.frame`, 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 `data.frame`, and will be used as the layer data. A `function` can be created from a `formula` (e.g. `~ head(.x, 10)`).
`stat`	The statistical transformation to use on the data for this layer, as a string.
`position`	Position adjustment, either as a string, or the result of a call to a position adjustment function.
`...`	Other arguments passed on to `ggplot2::layer()`. These are often aesthetics, used to set an aesthetic to a fixed value. See the below section "Aesthetics" for a full list of possible arguments.
`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.
`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. `borders()`.

Value

A ggplot2 layer (`ggplot2::layer()`) that can be added to a plot created with `ggplot2::ggplot()`.

Aesthetics

`geom_nfl_logos()` understands the following aesthetics (required aesthetics are in bold):

- **x** - The x-coordinate.
- **y** - The y-coordinate.
- **team_abbr** - The team abbreviation. Should be one of `valid_team_names()`. The function tries to clean team names internally by calling `nflreadr::clean_team_abbrs()`
- `alpha = NULL` - The alpha channel, i.e. transparency level, as a numerical value between 0 and 1.
- `colour = NULL` - The image will be colorized with this colour. Use the special character `"b/w"` to set it to black and white. For more information on valid colour names in ggplot2 see https://ggplot2.tidyverse.org/articles/ggplot2-specs.html?q=colour#colour-and-fill
- `angle = 0` - The angle of the image as a numerical value between 0° and 360°.
- `hjust = 0.5` - The horizontal adjustment relative to the given x coordinate. Must be a numerical value between 0 and 1.
- `vjust = 0.5` - The vertical adjustment relative to the given y coordinate. Must be a numerical value between 0 and 1.
- `width = 1.0` - The desired width of the image in `npc` (Normalised Parent Coordinates). The default value is set to 1.0 which is *big* but it is necessary because all used values are computed relative to the default. A typical size is `width = 0.075` (see below examples).
- `height = 1.0` - The desired height of the image in `npc` (Normalised Parent Coordinates). The default value is set to 1.0 which is *big* but it is necessary because all used values are computed relative to the default. A typical size is `height = 0.1` (see below examples).

Examples

```
library(nflplotR)
library(ggplot2)

team_abbr <- valid_team_names()
# remove conference logos from this example
team_abbr <- team_abbr[!team_abbr %in% c("AFC", "NFC", "NFL")]

df <- data.frame(
  a = rep(1:8, 4),
  b = sort(rep(1:4, 8), decreasing = TRUE),
  teams = team_abbr
)

# keep alpha == 1 for all teams including an "A"
matches <- grepl("A", team_abbr)
```

```
df$alpha <- ifelse(matches, 1, 0.2)
# also set a custom fill colour for the non "A" teams
df$colour <- ifelse(matches, NA, "gray")

# scatterplot of all logos
ggplot(df, aes(x = a, y = b)) +
  geom_nfl_logos(aes(team_abbr = teams), width = 0.075) +
  geom_label(aes(label = teams), nudge_y = -0.35, alpha = 0.5) +
  theme_void()

# apply alpha via an aesthetic from inside the dataset `df`
# please note that you have to add scale_alpha_identity() to use the alpha
# values in your dataset!
ggplot(df, aes(x = a, y = b)) +
  geom_nfl_logos(aes(team_abbr = teams, alpha = alpha), width = 0.075) +
  geom_label(aes(label = teams), nudge_y = -0.35, alpha = 0.5) +
  scale_alpha_identity() +
  theme_void()

# apply alpha and colour via an aesthetic from inside the dataset `df`
# please note that you have to add scale_alpha_identity() as well as
# scale_color_identity() to use the alpha and colour values in your dataset!
ggplot(df, aes(x = a, y = b)) +
  geom_nfl_logos(aes(team_abbr = teams, alpha = alpha, colour = colour), width = 0.075) +
  geom_label(aes(label = teams), nudge_y = -0.35, alpha = 0.5) +
  scale_alpha_identity() +
  scale_color_identity() +
  theme_void()

# apply alpha as constant for all logos
ggplot(df, aes(x = a, y = b)) +
  geom_nfl_logos(aes(team_abbr = teams), width = 0.075, alpha = 0.6) +
  geom_label(aes(label = teams), nudge_y = -0.35, alpha = 0.5) +
  theme_void()

# it's also possible to plot conference logos
conf <- data.frame(a = 1:2, b = 0, teams = c("AFC", "NFC"))
ggplot(conf, aes(x = a, y = b)) +
  geom_nfl_logos(aes(team_abbr = teams), width = 0.3) +
  geom_label(aes(label = teams), nudge_y = -0.4, alpha = 0.5) +
  coord_cartesian(xlim = c(0.5,2.5), ylim = c(-0.75,.75)) +
  theme_void()
```

`geom_nfl_wordmarks` *ggplot2 Layer for Visualizing NFL Team Wordmarks*

Description

This geom is used to plot NFL team wordmarks instead of points in a ggplot. It requires x, y aesthetics as well as a valid NFL team abbreviation. The latter can be checked with `valid_team_names()`.

Usage

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

Arguments

mapping	Set of aesthetic mappings created by aes() or 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	The data to be displayed in this layer. There are three options: If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot(). A data.frame, 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 data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)).
stat	The statistical transformation to use on the data for this layer, as a string.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
...	Other arguments passed on to ggplot2::layer(). These are often aesthetics, used to set an aesthetic to a fixed value. See the below section "Aesthetics" for a full list of possible arguments.
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.
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. borders().

Value

A ggplot2 layer (ggplot2::layer()) that can be added to a plot created with ggplot2::ggplot().

Aesthetics

`geom_nfl_logos()` understands the following aesthetics (required aesthetics are in bold):

- **x** - The x-coordinate.
- **y** - The y-coordinate.
- **team_abbr** - The team abbreviation. Should be one of `valid_team_names()`. The function tries to clean team names internally by calling `nflreadr::clean_team_abbrs()`
- `alpha = NULL` - The alpha channel, i.e. transparency level, as a numerical value between 0 and 1.
- `colour = NULL` - The image will be colorized with this colour. Use the special character `"b/w"` to set it to black and white. For more information on valid colour names in ggplot2 see https://ggplot2.tidyverse.org/articles/ggplot2-specs.html?q=colour#colour-and-fill
- `angle = 0` - The angle of the image as a numerical value between 0° and 360°.
- `hjust = 0.5` - The horizontal adjustment relative to the given x coordinate. Must be a numerical value between 0 and 1.
- `vjust = 0.5` - The vertical adjustment relative to the given y coordinate. Must be a numerical value between 0 and 1.
- `width = 1.0` - The desired width of the image in `npc` (Normalised Parent Coordinates). The default value is set to 1.0 which is *big* but it is necessary because all used values are computed relative to the default. A typical size is `width = 0.1` (see below examples).
- `height = 1.0` - The desired height of the image in `npc` (Normalised Parent Coordinates). The default value is set to 1.0 which is *big* but it is necessary because all used values are computed relative to the default. A typical size is `height = 0.1` (see below examples).

Examples

```
library(nflplotR)
library(ggplot2)

team_abbr <- valid_team_names()
# remove conference logos from this example
team_abbr <- team_abbr[!team_abbr %in% c("AFC", "NFC", "NFL")]

df <- data.frame(
  a = rep(1:8, 4),
  b = sort(rep(1:4, 8), decreasing = TRUE),
  teams = team_abbr
)

# keep alpha == 1 for all teams including an "A"
matches <- grepl("A", team_abbr)
df$alpha <- ifelse(matches, 1, 0.2)
# also set a custom fill colour for the non "A" teams
df$colour <- ifelse(matches, NA, "gray")

# scatterplot of all wordmarks
```

```
ggplot(df, aes(x = a, y = b)) +
  geom_nfl_wordmarks(aes(team_abbr = teams), width = 0.12) +
  geom_label(aes(label = teams), nudge_y = -0.20, alpha = 0.5) +
  scale_x_continuous(expand = expansion(add = 0.5)) +
  theme_void()

# apply alpha via an aesthetic from inside the dataset `df`
# please note that you have to add scale_alpha_identity() to use the alpha
# values in your dataset!
ggplot(df, aes(x = a, y = b)) +
  geom_nfl_wordmarks(aes(team_abbr = teams, alpha = alpha), width = 0.12) +
  geom_label(aes(label = teams), nudge_y = -0.20, alpha = 0.5) +
  scale_x_continuous(expand = expansion(add = 0.5)) +
  scale_alpha_identity() +
  theme_void()

# apply alpha and colour via an aesthetic from inside the dataset `df`
# please note that you have to add scale_alpha_identity() as well as
# scale_color_identity() to use the alpha and colour values in your dataset!
ggplot(df, aes(x = a, y = b)) +
 geom_nfl_wordmarks(aes(team_abbr = teams, alpha = alpha, colour = colour), width = 0.12) +
  geom_label(aes(label = teams), nudge_y = -0.20, alpha = 0.5) +
  scale_x_continuous(expand = expansion(add = 0.5)) +
  scale_alpha_identity() +
  scale_color_identity() +
  theme_void()

# apply alpha as constant for all logos
ggplot(df, aes(x = a, y = b)) +
  geom_nfl_wordmarks(aes(team_abbr = teams), width = 0.12, alpha = 0.6) +
  geom_label(aes(label = teams), nudge_y = -0.20, alpha = 0.5) +
  scale_x_continuous(expand = expansion(add = 0.5)) +
  theme_void()
```

`ggpreview` *Preview ggplot in Specified Dimensions*

Description

This function previews a ggplot in its actual dimensions in order to see how it will look when saved. It is also significantly faster than the default preview in RStudio for ggplots created using nflplotR.

Usage

```
ggpreview(
  plot = ggplot2::last_plot(),
  width = NA,
  height = NA,
```

```
  asp = NULL,
  dpi = 300,
  device = "png",
  units = c("in", "cm", "mm", "px"),
  scale = 1,
  limitsize = TRUE,
  bg = NULL,
  ...
)
```

Arguments

plot	Plot to save, defaults to last plot displayed.
width, height, units	Plot size in units ("in", "cm", "mm", or "px"). If not supplied, uses the size of current graphics device.
asp	The aspect ratio of the plot calculated as width / height. If this is a numeric value (and not NULL) the height of the plot will be recalculated to height = width / asp.
dpi	Plot resolution. Also accepts a string input: "retina" (320), "print" (300), or "screen" (72). Applies only to raster output types.
device	Device to use. Can either be a device function (e.g. png), or one of "eps", "ps", "tex" (pictex), "pdf", "jpeg", "tiff", "png", "bmp", "svg" or "wmf" (windows only).
scale	Multiplicative scaling factor.
limitsize	When TRUE (the default), ggsave() will not save images larger than 50x50 inches, to prevent the common error of specifying dimensions in pixels.
bg	Background colour. If NULL, uses the plot.background fill value from the plot theme.
...	Other arguments passed on to the graphics device function, as specified by device.

Value

No return value, called for side effects.

Examples

```
library(nflplotR)
library(ggplot2)

team_abbr <- valid_team_names()
# remove conference logos from this example
team_abbr <- team_abbr[!team_abbr %in% c("AFC", "NFC", "NFL")]

df <- data.frame(
  random_value = runif(length(team_abbr), 0, 1),
  teams = team_abbr
```

```
)

# use logos for x-axis
# note that the plot is assigned to the object "p"
p <- ggplot(df, aes(x = teams, y = random_value)) +
  geom_col(aes(color = teams, fill = teams), width = 0.5) +
  scale_color_nfl(type = "secondary") +
  scale_fill_nfl(alpha = 0.4) +
  theme_minimal() +
  theme(axis.text.x = element_nfl_logo())

# preview p with defined width and aspect ratio (only available in RStudio)
if (rstudioapi::isAvailable()){
  ggpreview(p, width = 5, asp = 16/9)
}
```

`nfl_team_factor` *Create Ordered NFL Team Name Factor*

Description

Create Ordered NFL Team Name Factor

Usage

```
nfl_team_factor(teams, ...)
```

Arguments

`teams`	A vector of NFL team abbreviations that should be included in `valid_team_names()`. The function tries to clean team names internally by calling `nflreadr::clean_team_abbrs()`.
`...`	One or more unquoted column names of `nflreadr::load_teams()` to sort by. If empty, the function will sort by division and nick name in ascending order. This is intended to be used for faceted plots where team wordmarks are used in strip texts, i.e. `element_nfl_wordmark()`. See examples for more details.

Value

Object of class `"factor"`

Examples

```
# unsorted vector including NFL team abbreviations
teams <- c("LAC", "LV", "CLE", "BAL", "DEN", "PIT", "CIN", "KC")

# defaults to sort by division and nick name in ascending order
nfl_team_factor(teams)
```

```
# can sort by every column in nflreadr::load_teams()
nfl_team_factor(teams, team_abbr)

# descending order by using base::rev()
nfl_team_factor(teams, rev(team_abbr))

######### HOW TO USE IN PRACTICE #########

library(ggplot2)
library(magrittr)
# load some sample data from the ggplot2 package
plot_data <- mpg
# add a new column by randomly sampling the above defined teams vector
plot_data$team <- sample(teams, nrow(mpg), replace = TRUE)

# Now we plot the data and facet by team abbreviation. ggplot automatically
# converts the team names to a factor and sorts it alphabetically
ggplot(plot_data, aes(displ, hwy)) +
  geom_point() +
  facet_wrap(~team, ncol = 4) +
  theme_minimal()

# We'll change the order of facets by making another team name column and
# converting it to an ordered factor. Again, this defaults to sort by division
# and nick name in ascending order.
plot_data$ordered_team <- sample(teams, nrow(mpg), replace = TRUE) %>%
  nfl_team_factor()

# Let's check how the facets are ordered now.
ggplot(plot_data, aes(displ, hwy)) +
  geom_point() +
  facet_wrap(~ordered_team, ncol = 4) +
  theme_minimal()

# The facet order looks weird because the defaults is meant to be used with
# NFL team wordmarks. So let's use the actual wordmarks and look at the result.
ggplot(plot_data, aes(displ, hwy)) +
  geom_point() +
  facet_wrap(~ordered_team, ncol = 4) +
  theme_minimal() +
  theme(strip.text = element_nfl_wordmark())
```

`nfl_team_tiers` *Create NFL Team Tiers*

Description

This function sets up a ggplot to visualize NFL team tiers.

Usage

```
nfl_team_tiers(
  data,
  title = "NFL Team Tiers, 2021 as of Week 4",
  subtitle = "created with the #nflplotR Tiermaker",
  caption = NULL,
 tier_desc = c(`1` = "Super Bowl", `2` = "Very Good", `3` = "Medium", `4` = "Bad", `5` =
    "What are they doing?", `6` = "", `7` = ""),
  presort = FALSE,
  alpha = 0.8,
  width = 0.075,
  no_line_below_tier = NULL,
  devel = FALSE
)
```

Arguments

data	A data frame that has to include the variables tier_no (the number of the tier starting from the top tier no. 1) and team_abbr (the team abbreviation). team_abbr should be one of valid_team_names() and the function tries to clean team names internally by calling nflreadr::clean_team_abbrs(). If data includes the variable tier_rank, these ranks will be used within each tier. Otherwise, if presort = FALSE, the function will assume that data is already sorted and if presort = TRUE, teams will be sorted alphabetically within tiers.
title	The title of the plot. If NULL, it will be omitted.
subtitle	The subtitle of the plot. If NULL, it will be omitted.
caption	The caption of the plot. If NULL, it will be omitted.
tier_desc	A named vector consisting of the tier descriptions. The names must equal the tier numbers from tier_no
presort	If FALSE (the default) the function assumes that the teams are already sorted within the tiers. Will otherwise sort alphabetically.
alpha	The alpha channel of the logos, i.e. transparency level, as a numerical value between 0 and 1.
width	The desired width of the logo in npc (Normalised Parent Coordinates).
no_line_below_tier	Vector of tier numbers. The function won't draw tier separation lines below these tiers. This is intended to be used for tiers that shall be combined (see examples).
devel	Determines if logos shall be rendered. If FALSE (the default), logos will be rendered on each run. If TRUE the team abbreviations will be plotted instead of the logos. This is much faster and helps with the plot development.

Value

A plot object created with ggplot2::ggplot().

Examples

```
library(ggplot2)
library(dplyr, warn.conflicts = FALSE)
teams <- valid_team_names()
# remove conference logos from this example
teams <- teams[!teams %in% c("AFC", "NFC", "NFL")]

# Build the team tiers data frame
# This is completely random!
df <- data.frame(
  tier_no = sample(1:5, length(teams), replace = TRUE),
  team_abbr = teams
) %>%
  dplyr::group_by(tier_no) %>%
  dplyr::mutate(tier_rank = sample(1:n(), n()))

# Plot team tiers
nfl_team_tiers(df)

# Create a combined tier which is useful for tiers with lots of teams that
# should be split up in two or more rows. This is done by setting an empty
# string for the tier 5 description and removing the tier separation line
# below tier number 4.
# This example also shows how to turn off the subtitle and add a caption
nfl_team_tiers(df,
               subtitle = NULL,
               caption = "This is the caption",
               tier_desc = c("1" = "Super Bowl",
                             "2" = "Very Good",
                             "3" = "Medium",
                             "4" = "A Combined Tier",
                             "5" = ""),
               no_line_below_tier = 4)

# For the development of the tiers, it can be useful to turn off logo image
# rendering as this can take quite a long time. By setting `devel = TRUE`, the
# logo images are replaced by team abbreviations which is much faster
nfl_team_tiers(df,
               tier_desc = c("1" = "Super Bowl",
                             "2" = "Very Good",
                             "3" = "",
                             "4" = "A Combined Tier",
                             "5" = ""),
               no_line_below_tier = c(2, 4),
               devel = TRUE)
```

`scale_axes_nfl` *Axis Scales for NFL Team Logos*

Description

These functions map NFL team names to their team logos and make them available as axis labels

Usage

```
scale_x_nfl(
  ...,
  expand = ggplot2::waiver(),
  guide = ggplot2::waiver(),
  position = "bottom",
  size = 12
)

scale_y_nfl(
  ...,
  expand = ggplot2::waiver(),
  guide = ggplot2::waiver(),
  position = "left",
  size = 12
)

scale_x_nfl_headshots(
  ...,
  expand = ggplot2::waiver(),
  guide = ggplot2::waiver(),
  position = "bottom",
  size = 20
)

scale_y_nfl_headshots(
  ...,
  expand = ggplot2::waiver(),
  guide = ggplot2::waiver(),
  position = "left",
  size = 30
)
```

Arguments

`...` Arguments passed on to `discrete_scale`

`palette` A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., `scales::hue_pal()`).

`breaks` One of:

- `NULL` for no breaks
- `waiver()` for the default breaks (the scale limits)
- A character vector of breaks

- A function that takes the limits as input and returns breaks as output. Also accepts rlang lambda function notation.

`limits` One of:

- `NULL` to use the default scale values
- A character vector that defines possible values of the scale and their order
- A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation.

`drop` Should unused factor levels be omitted from the scale? The default, `TRUE`, uses the levels that appear in the data; `FALSE` uses all the levels in the factor.

`na.translate` Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify `na.translate = FALSE`.

`na.value` If `na.translate = TRUE`, what aesthetic value should the missing values be displayed as? Does not apply to position scales where `NA` is always placed at the far right.

`aesthetics` The names of the aesthetics that this scale works with.

`scale_name` The name of the scale that should be used for error messages associated with this scale.

`name` The name of the scale. Used as the axis or legend title. If `waiver()`, the default, the name of the scale is taken from the first mapping used for that aesthetic. If `NULL`, the legend title will be omitted.

`labels` One of:

- `NULL` for no labels
- `waiver()` for the default labels computed by the transformation object
- A character vector giving labels (must be same length as `breaks`)
- A function that takes the breaks as input and returns labels as output. Also accepts rlang lambda function notation.

`super` The super class to use for the constructed scale

`expand` For position scales, a vector of range expansion constants used to add some padding around the data to ensure that they are placed some distance away from the axes. Use the convenience function `expansion()` to generate the values for the expand argument. The defaults are to expand the scale by 5% on each side for continuous variables, and by 0.6 units on each side for discrete variables.

`guide` A function used to create a guide or its name. See `guides()` for more information.

`position` For position scales, The position of the axis. `left` or `right` for y axes, `top` or `bottom` for x axes.

`size` The logo size in pixels. It is applied as height for an x-scale and as width for an y-scale.

Details

The scale translates the NFL team abbreviations into raw image html and places the html as axis labels. Because of the way ggplots are constructed, it is necessary to adjust the `theme()` after

calling this scale. This can be done by calling theme_x_nfl() or theme_y_nfl() or alternatively by manually changing the relevant axis.text to ggtext::element_markdown(). However, this will only work if an underlying dependency, "gridtext", is installed with a newer version than 0.1.4

Value

A discrete ggplot2 scale created with ggplot2::scale_x_discrete() or ggplot2::scale_y_discrete().

See Also

theme_x_nfl(), theme_y_nfl()

Examples

```
library(nflplotR)
library(ggplot2)

team_abbr <- valid_team_names()
# remove conference logos from this example
team_abbr <- team_abbr[!team_abbr %in% c("AFC", "NFC")]

df <- data.frame(
  random_value = runif(length(team_abbr), 0, 1),
  teams = team_abbr
)

if (utils::packageVersion("gridtext") > "0.1.4"){
  # use logos for x-axis
  ggplot(df, aes(x = teams, y = random_value)) +
    geom_col(aes(color = teams, fill = teams), width = 0.5) +
    scale_color_nfl(type = "secondary") +
    scale_fill_nfl(alpha = 0.4) +
    scale_x_nfl() +
    theme_minimal() +
    # theme_*_nfl requires gridtext version > 0.1.4
    theme_x_nfl()

  # use logos for y-axis
  ggplot(df, aes(y = teams, x = random_value)) +
    geom_col(aes(color = teams, fill = teams), width = 0.5) +
    scale_color_nfl(type = "secondary") +
    scale_fill_nfl(alpha = 0.4) +
    scale_y_nfl() +
    theme_minimal() +
    # theme_*_nfl requires gridtext version > 0.1.4
    theme_y_nfl()
}
################################################################################
# Headshot Examples
################################################################################
library(nflplotR)
```

```
library(ggplot2)

# Silence an nflreadr message that is irrelevant here
old <- options(nflreadr.cache_warning = FALSE)

dfh <- data.frame(
  random_value = runif(9, 0, 1),
  player_gsis = c("00-0033873",
                  "00-0026498",
                  "00-0035228",
                  "00-0031237",
                  "00-0036355",
                  "00-0019596",
                  "00-0033077",
                  "00-0012345",
                  "00-0031280")
)

if (utils::packageVersion("gridtext") > "0.1.4"){
  # use headshots for x-axis
  ggplot(dfh, aes(x = player_gsis, y = random_value)) +
    geom_col(width = 0.5) +
    scale_x_nfl_headshots() +
    theme_minimal() +
    # theme_*_nfl requires gridtext version > 0.1.4
    theme_x_nfl()

  # use headshots for y-axis
  ggplot(dfh, aes(y = player_gsis, x = random_value)) +
    geom_col(width = 0.5) +
    scale_y_nfl_headshots() +
    theme_minimal() +
    # theme_*_nfl requires gridtext version > 0.1.4
    theme_y_nfl()
}
# Restore old options
options(old)
```

`scale_nfl` *Scales for NFL Team Colors*

Description

These functions map NFL team names to their team colors in color and fill aesthetics

Usage

```
scale_color_nfl(
  type = c("primary", "secondary"),
```

```
  values = NULL,
  ...,
  aesthetics = "colour",
  breaks = ggplot2::waiver(),
  na.value = "grey50",
  guide = NULL,
  alpha = NA
)

scale_colour_nfl(
  type = c("primary", "secondary"),
  values = NULL,
  ...,
  aesthetics = "colour",
  breaks = ggplot2::waiver(),
  na.value = "grey50",
  guide = NULL,
  alpha = NA
)

scale_fill_nfl(
  type = c("primary", "secondary"),
  values = NULL,
  ...,
  aesthetics = "fill",
  breaks = ggplot2::waiver(),
  na.value = "grey50",
  guide = NULL,
  alpha = NA
)
```

Arguments

type	One of "primary" or "secondary" to decide which colortype to use.
values	If NULL (the default) use the internal team color vectors. Otherwise a set of aesthetic values to map data values to. The values will be matched in order (usually alphabetical) with the limits of the scale, or with breaks if provided. If this is a named vector, then the values will be matched based on the names instead. Data values that don't match will be given na.value.
...	Arguments passed on to discrete_scale palette A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., scales::hue_pal()). limits One of: • NULL to use the default scale values • A character vector that defines possible values of the scale and their order

- A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation.

`drop` Should unused factor levels be omitted from the scale? The default, `TRUE`, uses the levels that appear in the data; `FALSE` uses all the levels in the factor.

`na.translate` Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify `na.translate = FALSE`.

`scale_name` The name of the scale that should be used for error messages associated with this scale.

`name` The name of the scale. Used as the axis or legend title. If `waiver()`, the default, the name of the scale is taken from the first mapping used for that aesthetic. If `NULL`, the legend title will be omitted.

`labels` One of:

- `NULL` for no labels
- `waiver()` for the default labels computed by the transformation object
- A character vector giving labels (must be same length as `breaks`)
- A function that takes the breaks as input and returns labels as output. Also accepts rlang lambda function notation.

`guide` A function used to create a guide or its name. See `guides()` for more information.

`super` The super class to use for the constructed scale

`aesthetics` Character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with. This can be useful, for example, to apply colour settings to the `colour` and `fill` aesthetics at the same time, via `aesthetics = c("colour", "fill")`.

`breaks` One of:

- `NULL` for no breaks
- `waiver()` for the default breaks (the scale limits)
- A character vector of breaks
- A function that takes the limits as input and returns breaks as output

`na.value` The aesthetic value to use for missing (`NA`) values

`guide` A function used to create a guide or its name. If `NULL` (the default) no guide will be plotted for this scale. See `ggplot2::guides()` for more information.

`alpha` Factor to modify color transparency via a call to `scales::alpha()`. If `NA` (the default) no transparency will be applied. Can also be a vector of alphas. All alpha levels must be in range `[0,1]`.

Examples

```
library(nflplotR)
library(ggplot2)

team_abbr <- valid_team_names()
# remove conference logos from this example
```

```
team_abbr <- team_abbr[!team_abbr %in% c("AFC", "NFC", "NFL")]

df <- data.frame(
  random_value = runif(length(team_abbr), 0, 1),
  teams = team_abbr
)
ggplot(df, aes(x = teams, y = random_value)) +
  geom_col(aes(color = teams, fill = teams), width = 0.5) +
  scale_color_nfl(type = "secondary") +
  scale_fill_nfl(alpha = 0.4) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

`theme_nfl` *Theme for NFL Team Logos*

Description

These functions are convenience wrappers around a theme call that activates markdown in x-axis and y-axis labels using `ggtext::element_markdown()`.

Usage

```
theme_x_nfl()

theme_y_nfl()
```

Details

These functions are a wrapper around the function calls `ggplot2::theme(axis.text.x = ggtext::element_markdown())` as well as `ggplot2::theme(axis.text.y = ggtext::element_markdown())`. They are made to be used in conjunction with `scale_x_nfl()` and `scale_y_nfl()` respectively.

Value

A ggplot2 theme created with `ggplot2::theme()`.

See Also

`theme_x_nfl()`, `theme_y_nfl()`

Examples

```
library(nflplotR)
library(ggplot2)

team_abbr <- valid_team_names()
```

```
# remove conference logos from this example
team_abbr <- team_abbr[!team_abbr %in% c("AFC", "NFC", "NFL")]

df <- data.frame(
  random_value = runif(length(team_abbr), 0, 1),
  teams = team_abbr
)
if (utils::packageVersion("gridtext") > "0.1.4"){
  ggplot(df, aes(x = teams, y = random_value)) +
    geom_col(aes(color = teams, fill = teams), width = 0.5) +
    scale_color_nfl(type = "secondary") +
    scale_fill_nfl(alpha = 0.4) +
    scale_x_nfl() +
    theme_minimal() +
    # theme_*_nfl requires gridtext version > 0.1.4
    theme_x_nfl()
}
```

`valid_team_names` *Output Valid NFL Team Abbreviations*

Description

Output Valid NFL Team Abbreviations

Usage

```
valid_team_names(exclude_duplicates = TRUE)
```

Arguments

`exclude_duplicates`
: If `TRUE` (the default) the list of valid team abbreviations will exclude duplicates related to franchises that have been moved

Value

A vector of type `"character"`.

Examples

```
# List valid team abbreviations excluding duplicates
valid_team_names()

# List valid team abbreviations excluding duplicates
valid_team_names(exclude_duplicates = FALSE)
```

Index