Package ‘sportyR’

April 20, 2021

Title Plot Scaled 'ggplot' Representations of Sports Playing Surfaces
Version 1.0.1
Maintainer Ross Drucker <ross.a.drucker@gmail.com>
Description Create scaled 'ggplot' representations of playing surfaces.
Playing surfaces are drawn pursuant to rule-book specifications.
This package should be used as a baseline plot for displaying player
tracking data.
License MIT + file LICENSE
Encoding UTF-8
RoxygenNote 7.1.1
Imports ggplot2, glue, dplyr
Depends R (>= 3.3)
Suggests testthat (>= 3.0.0), vdiffr (>= 0.3.3), roxygen2
Config/testthat/edition 3
URL https://github.com/rossdrucker/sportyR
BugReports https://github.com/rossdrucker/sportyR/issues
NeedsCompilation no
Author Ross Drucker [aut, cre]
Repository CRAN
Date/Publication 2021-04-20 13:50:02 UTC

R topics documented:

  cani_color_league_features ........................................... 2
  cani_plot_league ....................................................... 2
  cani_plot_sport .......................................................... 3
  convert_units ............................................................. 4
  geom_baseball ............................................................ 4
  geom_basketball .......................................................... 5
  geom_football ............................................................. 6
cani_color_league_features

Check to see what features of a surface can be colored by a user

Description
Check to see what features of a surface can be colored by a user

Usage
cani_color_league_features(league_code, sport_name = NULL)

Arguments
league_code The case-insensitive league code to be plotted
sport_name The name of a sport to use in the event that the league_code supplied has more than one sport associated with it. Default: NULL

Value
Nothing, but a message is sent to the console for the user

Examples
cani_color_league_features('NCAA', 'basketball')

---

cani_plot_league

Check to see if a league can be plotted, and alert the user as to which functions that league will work for

Description
Check to see if a league can be plotted, and alert the user as to which functions that league will work for

Usage
cani_plot_league(league_code)
**cani_plot_sport**

**Arguments**

league_code  The case-insensitive league code to be plotted

**Value**

Nothing, but a message is sent to the console for the user

**Examples**

cani_plot_league('MLB')

---

**cani_plot_sport**  
*Check to see if a sport can be plotted, and alert the user as to which leagues are plottable for the sport*

**Description**

Check to see if a sport can be plotted, and alert the user as to which leagues are plottable for the sport

**Usage**

cani_plot_sport(sport)

**Arguments**

sport  The case-insensitive sport name

**Value**

Nothing, but a message is sent to the console for the user

**Examples**

cani_plot_sport('basketball')
convert_units  
*Function to convert all units, regardless of starting and ending units*

**Description**

Function to convert all units, regardless of starting and ending units

**Usage**

convert_units(meas, from_unit, to_unit, conversion_columns = NULL)

**Arguments**

- `meas`: A measurement in any unit of length
- `from_unit`: A string containing the original unit of measure to be converted
- `to_unit`: A string containing the ending unit of measure
- `conversion_columns`: A vector containing the columns to convert if `meas` is of type `data.frame`

**Value**

The measurement in converted units

**Examples**

```r
convert_units(1, 'in', 'cm')
convert_units(100, 'cm', 'm')
```

gem_baseball  
*Generate a ggplot2 instance containing a regulation baseball field for a specified league*

**Description**

Generate a ggplot2 instance containing a regulation baseball field for a specified league

**Usage**

geom_baseball(league, ...)

**Arguments**

- `league`: The league for which to draw the surface
- `...`: Additional arguments to pass to the function. These should be the colors to pass to the `mlb_features_set_colors()` function, or units with which to draw the plot
**geom_basketball**

**Value**

A ggplot2 instance with a full-surface representation of a basketball field

**Examples**

```r
geom_baseball(league = "MLB")
```

---

**Create a ggplot2 instance of a scale model of a basketball court**

**Description**

Create a ggplot2 instance of a scale model of a basketball court

**Usage**

```r
geom_basketball(
  league,
  full_surf = TRUE,
  rotate = FALSE,
  rotation_dir = "ccw",
  ...
)
```

**Arguments**

- `league` The league for which to draw the surface
- `full_surf` A boolean indicating whether or not to plot a full surface representation of the surface. Default: TRUE
- `rotate` A boolean indicating whether or not this feature needs to be rotated. Default: FALSE
- `rotation_dir` A string indicating which direction to rotate the feature. Default: 'ccw'
- `...` Additional arguments to pass to the function. These should be the colors to pass to the `mlb_features_set_colors()` function, or units with which to draw the plot

**Value**

A ggplot2 instance with a full-surface representation of a basketball court

**Examples**

```r
geom_basketball(league = "NBA")
geom_basketball(league = "NCAA", full_surf = FALSE)
geom_basketball(league = "FIBA", rotate = TRUE, rotation_dir = "ccw")
```
This draws a football field in its standard coordinate system, with (0, 0) being the bottom left corner of the left-most endzone. Each unit on the coordinate system corresponds to 1 yard. Generate a ggplot2 instance containing a regulation football field for a specified league.

**Usage**

```r
geom_football(
  league,
  full_surf = TRUE,
  rotate = FALSE,
  rotation_dir = "ccw",
  ...
)
```

**Arguments**

- `league`: The league for which to draw the surface.
- `full_surf`: A boolean indicating whether or not to draw a full-surface representation of the playing surface. Default: `TRUE`
- `rotate`: A boolean indicating whether or not this feature needs to be rotated. Default: `FALSE`
- `rotation_dir`: A string indicating which direction to rotate the feature. Default: `'ccw'`
- `...`: Additional arguments to pass to the function. These should be the colors to pass to the `{league}_features_set_colors()` function, (although the colors are defined in the rule book) or units with which to draw the plot.

**Value**

A ggplot2 instance with a full-surface representation of a football field.

**Examples**

```r
geom_football(league = "NFL")
geom_football(league = "NCAA", rotate = TRUE, rotation_dir = "ccw")
```
**Description**

Generate a ggplot2 instance containing an ice rink for a specified league

**Usage**

```
geom_hockey(
  league,
  full_surf = TRUE,
  rotate = FALSE,
  rotation_dir = "ccw",
  ...
)
```

**Arguments**

- `league`: The league for which to draw the surface
- `full_surf`: A boolean indicating whether or not to plot a full surface representation of the surface. Default: TRUE
- `rotate`: A boolean indicating whether or not the final rink plot needs to be rotated. Default: FALSE
- `rotation_dir`: A string indicating which direction to rotate the final rink plot. Default: 'ccw'
- `...`: Additional arguments to pass to the function. These should be the colors to pass to the `{league}_features_set_colors()` function, (although the colors are defined in the rule book) or units with which to draw the plot

**Value**

A ggplot2 instance with a full-surface representation of an ice hockey rink

**Examples**

```
geom_hockey(league = "NHL")
geom_hockey(league = "IIHF", full_surf = FALSE)
geom_hockey(league = "NCAA", rotate = TRUE, rotation_dir = "ccw")
```
**geom_soccer**

Generate a ggplot2 instance containing a soccer pitch for a specified league

**Description**

Generate a ggplot2 instance containing a soccer pitch for a specified league

**Usage**

```r
geom_soccer(
  league,
  touchline_length = 120,
  goal_line_length = 90,
  full_surf = TRUE,
  rotate = FALSE,
  rotation_dir = "ccw",
  ...
)
```

**Arguments**

- `league` The league for which to draw the surface
- `touchline_length` The length of the touchline. This should be the entire length (both halves) of the pitch. Default: 120
- `goal_line_length` The length of the goal line. Default: 90
- `full_surf` A boolean indicating whether or not to plot a full surface representation of the surface. Default: TRUE
- `rotate` A boolean indicating whether or not this feature needs to be rotated. Default: FALSE
- `rotation_dir` A string indicating which direction to rotate the feature. Default: 'ccw'
- `...` Additional arguments to pass to the function. These should be the colors to pass to the `mlb_features_set_colors()` function, or units with which to draw the plot

**Value**

A ggplot2 instance with a full-surface representation of a soccer pitch

**Examples**

```r
geom_soccer(league = "MLS")
geom_soccer(league = "PREMIER", rotate = TRUE, rotation_dir = "ccw")
```
**reflect**

Perform a mathematical reflection of coordinates over a specified axis

**Usage**

```r
reflect(df, over_x = FALSE, over_y = TRUE)
```

**Arguments**

- `df` The data frame to reflect. It must have `x` and `y` columns
- `over_x` A boolean indicating whether or not to reflect over the x axis. Default: FALSE
- `over_y` A boolean indicating whether or not to reflect over the y axis. Default: TRUE

**Value**

The reflected data frame

**Examples**

```r
reflect(data.frame(x = 1, y = 0))
```

---

**rotate_coords**

Perform a mathematical rotation about (0, 0) of coordinates. This rotation is given as

\[
x' = x \cos(\theta) - y \sin(\theta)
\]

\[
y' = x \sin(\theta) + y \cos(\theta)
\]

**Description**

Perform a mathematical rotation about (0, 0) of coordinates. This rotation is given as 

\[
x' = x \cos(\theta) - y \sin(\theta)
\]

\[
y' = x \sin(\theta) + y \cos(\theta)
\]

**Usage**

```r
rotate_coords(df, rotation_dir = "ccw", angle = 0.5)
```

**Arguments**

- `df` The data frame to rotate. It must have `x` and `y` columns
- `rotation_dir` The direction in which to rotate the coordinates. `ccw` corresponds to counterclockwise
- `angle` the angle (in radians, divided by pi) through which to rotate the coordinates
Value

The rotated data frame

Examples

rotate_coords(data.frame(x = 0, y = 1))

---

**translate**  
*Perform a mathematical translation of coordinates*

Description

Perform a mathematical translation of coordinates

Usage

translate(df, translate_x = 0, translate_y = 0)

Arguments

df  
The data frame to translate It must have x and y columns

translate_x  
The number of units (in the input data frame’s units) to translate the points in the +x direction. Default: 0

translate_y  
The number of units (in the input data frame’s units) to translate the points in the +y direction. Default: 0

Value

The translated data frame

Examples

translate(data.frame(x = 0, y = 1), translate_x = 1)  
translate(data.frame(x = 0, y = 1), translate_y = 1)
Index

cani_color_league_features, 2
cani_plot_league, 2
cani_plot_sport, 3
convert_units, 4

geom_baseball, 4
g geom_basketball, 5
g geom_football, 6
g geom_hockey, 7
g geom_soccer, 8

reflect, 9
rotate_coords, 9

translate, 10