

Package: soccerAnimate (via r-universe)

October 19, 2026

Type Package

Title Creates 2D animations of soccer tracking data

Version 0.1.0

Author person(given = ``Ismael", family = ``Gómez", role = c(``aut",
``cre"), email = ``ismaelgomezs@gmail.com")

Maintainer Ismael Gómez Schmidt <ismaelgomezs@gmail.com>

Description This packages allows you to create 2D animations of soccer tracking data in addition to do some player calculations and team possession stats.

License GPL-2

Imports deldir, dplyr, forcats, geosphere, gganimate, ggforce, ggplot2, ggrepel, ggtext, gifski, glue, grDevices, gt, gtExtras, janitor, magrittr, patchwork, png, purrr, readr, rlang, stats, stringr, tidyr, transformr

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Config/pak/sysreqs libabsl-dev cmake libfontconfig1-dev libfreetype6-dev libgdal-dev gdal-bin libgeos-dev make libicu-dev libjpeg-dev libpng-dev libuv1-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libnode-dev libx11-dev libclang-dev

Repository <https://sportsdataverse.r-universe.dev>

Date/Publication 2024-03-12 12:28:35 UTC

RemoteUrl <https://github.com/Dato-Futbol/soccerAnimate>

RemoteRef HEAD

RemoteSha 407aed58e507f0fd3c42eee8fe6fd55d81888706

Contents

events_info	2
get_tidy_data	2

player_animate	3
player_plot	5
players_stats	6
players_stats_graph	6
soccer_animate	7
soccer_plot	8
soccer_plot_poss	9
sprints_info	10
Index	11

events_info	<i>Searches time ocurrence and other information of specific events</i>
-------------	---

Description

Searches time ocurrence and other information of specific events

Usage

events_info(data, events = c("SHOT", "GOAL", "FREE KICK", "CORNER KICK"))

Arguments

- | | |
|--------|--|
| data | event dataframe previously loaded |
| events | one or more events to search (options: SHOT, GOAL, FREE KICK, CORNER KICK) |

Value

dataframe with searched events and their specific info

get_tidy_data	<i>Prepares data</i>
---------------	----------------------

Description

Reads and converts both the home and away team rawdata to tidy data (long format) and join them. Currently only Metrica Sports data format is supported

Usage

```
get_tidy_data(
  home_team_file,
  away_team_file,
  provider = "Metrica",
  add_velocity = T,
  pitch_long = 105,
  pitch_width = 68
)
```

Arguments

`home_team_file` set the filename which contains the data of the home team

`away_team_file` set the filename which contains the data of the away team

`provider` set the name of the tracking data provider which defines the data format

`add_velocity` A boolean variable to set if the players velocities should to be added

`pitch_long` long of the pitch in meters

`pitch_width` width of the pitch in meters

Value

The dataframe with all tidy data joined (long format)

player_animate	<i>Creates 2D soccer animations</i>
----------------	-------------------------------------

Description

Creates 2D animations using soccer tracking data allowing you export it as gif file

Usage

```
player_animate(
  tidy_data,
  target_player,
  ini_frame,
  end_frame,
  player_col = "#ffc300",
  method = "base",
  pitch_long = 105,
  pitch_width = 68,
  pitch_fill = "#74a9cf",
  pitch_lines_col = "lightgrey",
  home_team_col = "white",
  away_team_col = "#dd3497",
)
```

```

    title = "",
    subtitle = "",
    provider = "Metrica",
    show_anim = T,
    export_gif = F,
    gif_name = "animation"
)

```

Arguments

<code>tidy_data</code>	the processed dataframe ready to do visualizations. It could be obtained using either <code>get_tidy_data()</code> or <code>process_catapult()</code> functions.
<code>target_player</code>	player whose actions will be highlighted
<code>ini_frame</code>	the frame to consider as the initial point for the animation
<code>end_frame</code>	the frame to consider as the ending point for the animation
<code>player_col</code>	color to highlight the target player actions
<code>method</code>	four different approaches to visualize: base (default), convexhull, voronoi, delaunay
<code>pitch_long</code>	long of the pitch in meters
<code>pitch_width</code>	width of the pitch in meters
<code>pitch_fill</code>	color used to fill the pitch
<code>pitch_lines_col</code>	color used for lines of the pitch
<code>home_team_col</code>	color used to fill the players of the home team
<code>away_team_col</code>	color used to fill the players of the away team
<code>title</code>	graph title
<code>subtitle</code>	graph subtitle
<code>provider</code>	set the name of the tracking data provider which defines the data format
<code>show_anim</code>	should the animation be shown on the viewer? (T/F)
<code>export_gif</code>	should the animation be exported as gif file? (T/F)
<code>gif_name</code>	the name of the exported gif file

Value

animated gif on viewer and/or as output file

player_plot	<i>Creates a 2D static plot marking the selected frames of a specific player</i>
-------------	--

Description

Creates a 2D static plot marking the selected frames of a specific player

Usage

```
player_plot(
  tidy_data,
  target_player,
  start_frames,
  end_frames = NULL,
  pitch_long = 105,
  pitch_width = 68,
  pitch_fill = "#74a9cf",
  pitch_lines_col = "lightgrey",
  action_name = "sprints",
  action_col = "purple",
  label_col = "white",
  export_png = F
)
```

Arguments

tidy_data	the processed dataframe ready to do visualizations. It could be obtained using the function <code>get_tidy_data()</code> .
target_player	the player whose selected frames will be marked
start_frames	vector with the selected frames focused on when specific actions start
end_frames	vector with the selected frames focused on when specific actions end
pitch_long	long of the pitch in meters
pitch_width	width of the pitch in meters
pitch_fill	color used to fill the pitch
pitch_lines_col	color used for lines of the pitch
action_name	action name(s) to be highlighted
action_col	color used for the frames marking related to a specific action
label_col	color used for labels showing the Time when actions start
export_png	should the plot be exported as png file? (T/F)

Value

plot with the selected frames of a player on Plots panel and/or as png output file

players_stats	<i>Calculate multiple players stats based on tracking data. It includes Minutes played, distance, average/max speed in addition to distance splitted by different speed ranges.</i>
---------------	---

Description

Calculate multiple players stats based on tracking data. It includes Minutes played, distance, average/max speed in addition to distance splitted by different speed ranges.

Usage

```
players_stats(data, speed_ranges = c("laurie", "perf"))
```

Arguments

data	dataframe with the already processed tracking data in long format
speed_ranges	character to set what kind of speed ranges has to be considered: a) "laurie": walking, jogging, running, sprinting; b) "perf": low speed activity, high speed running, very high speed running and sprinting

Value

dataframe with the results

players_stats_graph	<i>Create a graph to compare players stats for a specific team</i>
---------------------	--

Description

Create a graph to compare players stats for a specific team

Usage

```
players_stats_graph(
  data,
  team = c("home", "away"),
  var = "ranges",
  export_png = F
)
```

Arguments

data	dataframe with the players stats
team	team to graph "home" or "away"
var	variable to graph, default value = "ranges" for speed ranges
export_png	should the graph be exported as PNG file? (T/F)

Value

dataframe with the results

soccer_animate	<i>Creates 2D soccer animations</i>
----------------	-------------------------------------

Description

Creates 2D animations using soccer tracking data allowing you export it as gif file

Usage

```
soccer_animate(
  tidy_data,
  ini_time,
  end_time,
  method = "base",
  pitch_long = 105,
  pitch_width = 68,
  pitch_fill = "#74a9cf",
  pitch_lines_col = "lightgrey",
  home_team_col = "white",
  away_team_col = "#dd3497",
  title = "",
  subtitle = "",
  provider = "Metrica",
  show_anim = T,
  export_gif = F,
  gif_name = "animation"
)
```

Arguments

tidy_data	the processed dataframe ready to do visualizations. It could be obtained using either <code>get_tidy_data()</code> or <code>process_catapult()</code> functions.
ini_time	the time in seconds of tracking data to consider as initial time of the animation
end_time	the time in seconds of tracking data to consider as ending time of the animation
method	four different approaches to visualize: base (default), convexhull, voronoi, delaunay
pitch_long	long of the pitch in meters
pitch_width	width of the pitch in meters
pitch_fill	colour used to fill the pitch
pitch_lines_col	colour used for lines of the pitch

home_team_col	colour used to fill the players of the home team
away_team_col	colour used to fill the players of the away team
title	graph title
subtitle	graph subtitle
provider	set the name of the tracking data provider which defines the data format
show_anim	should the animation be shown on the viewer? (T/F)
export_gif	should the animation be exported as gif file? (T/F)
gif_name	the name of the exported gif file

Value

animated gif on viewer and/or as output file

soccer_plot	<i>Creates a 2D static plot</i>
-------------	---------------------------------

Description

Creates a 2D static plot of a unique frame from soccer tracking data

Usage

```
soccer_plot(
  tidy_data,
  target_frame,
  method = "base",
  pitch_long = 105,
  pitch_width = 68,
  pitch_fill = "#74a9cf",
  pitch_lines_col = "lightgrey",
  home_team_col = "white",
  away_team_col = "#dd3497",
  provider = "Metrica",
  export_png = F,
  png_name = "plot",
  title = "",
  subtitle = ""
)
```

Arguments

tidy_data	the processed dataframe ready to do visualizations. It could be obtained using either get_tidy_data() or process_catapult() functions.
target_frame	the unique frame of the tracking data to visualize
method	four different approaches to visualize: base, convexhull, voronoi, delaunay

pitch_long	long of the pitch in meters
pitch_width	width of the pitch in meters
pitch_fill	colour used to fill the pitch
pitch_lines_col	colour used for lines of the pitch
home_team_col	colour used to fill the players of the home team
away_team_col	colour used to fill the players of the away team
provider	set the name of the tracking data provider which defines the data format
export_png	should the plot be exported as png file? (T/F)
png_name	the name of the exported gif file
title	graph title
subtitle	graph subtitle

Value

soccer plot on Plots panel and/or as output file

soccer_plot_poss	<i>Creates a 2D team visualization on a team level with avg. positioning splitting by ON/OFF possession</i>
------------------	---

Description

Creates a 2D team visualization on a team level with avg. positioning splitting by ON/OFF possession

Usage

```
soccer_plot_poss(
  tidy_data,
  events_path,
  team = c("Home", "Away"),
  pitch_long = 105,
  pitch_width = 68,
  pitch_fill = "#74a9cf",
  pitch_lines_col = "lightgrey",
  target_team_col = "#7209b7",
  opp_team_col = "#06d6a0",
  on_ball_col = "white",
  off_ball_col = "#dd3497",
  export_png = F,
  png_name = "plot",
  annotation = "NA"
)
```

Arguments

tidy_data	the processed dataframe ready to do visualizations. It could be obtained using either <code>get_tidy_data()</code> or <code>process_catapult()</code> functions.
events_path	path where the CSV file with eventing data is located
team	target team
pitch_long	long of the pitch in meters
pitch_width	width of the pitch in meters
pitch_fill	colour used to fill the pitch
pitch_lines_col	colour used for lines of the pitch
target_team_col	colour used to fill the border of players circles of the target team (top position on the graph)
opp_team_col	colour used to fill the border of players circles of the opponent team (bottom position on the graph)
on_ball_col	colour used to fill the players when their team was ON possession
off_ball_col	colour used to fill the players when their team was OFF possession
export_png	should the plot be exported as png file? (T/F)
png_name	the name of the exported gif file
annotation	'NA' by default. If "depth" is setted, vertical lines with lowest defender and highest forward will be added

Value

soccer plot on 'Plots' panel and/or as output PNG file

sprints_info	<i>Get the start/end time of the sprints that every player made</i>
--------------	---

Description

Get the start/end time of the sprints that every player made

Usage

```
sprints_info(data, team = c("home", "away"))
```

Arguments

data	dataframe with the already processed tracking data in long format
team	team to graph "home" or "away"

Value

dataframe with the results

Index

events_info, [2](#)

get_tidy_data, [2](#)

player_animate, [3](#)

player_plot, [5](#)

players_stats, [6](#)

players_stats_graph, [6](#)

soccer_animate, [7](#)

soccer_plot, [8](#)

soccer_plot_poss, [9](#)

sprints_info, [10](#)