Package 'soccermatics'

October 22, 2018

Version 0.9.0

Title Visualise tracking and event data from soccer matches

Description Provides tools to visualise x,y-coordinates of soccer players and event data (passes, shots). Uses ggplot to draw soccer pitch and overplot expected goal maps, pass maps, average player positions, player heatmaps, individual player paths, player flow fields, and more.

Depends R (>= 3.4.1)

Imports dplyr, magrittr, ggplot2, ggforce, zoo

License GPL (>=3.0) Note: Use of the name 'soccermatics' was kindly permitted by David Sumpter and is protected from commercial use under EU copyright law.

Encoding UTF-8

LazyData true

Collate 'soccerFlipDirection.R' 'soccerPitchFG.R' 'soccerHeatmap.R' 'soccerPath.R' 'soccerPitch.R' 'soccerPositionMap.R' 'soccerResample.R' 'soccerShortenName.R' 'soccerShotmap.R' 'soccerSpokes.R' 'soccerTransform.R' 'soccerVelocity.R' 'statsbomb.R' 'tromso.R' 'tromso_extra.R'

RoxygenNote 6.0.1

Author Joe Gallagher [aut, cre]

Maintainer Joe Gallagher <joedgallagher@gmail.com>

RemoteType github

RemoteHost https://api.github.com

RemoteRepo soccermatics

RemoteUsername jogall

RemoteRef master

RemoteSha 92345464cb123208dee02eea5fcbbefd9426a998

GithubRepo soccermatics

GithubUsername jogall

GithubRef master

GithubSHA1 92345464cb123208dee02eea5fcbbefd9426a998

R topics documented:

`soccerFlipDirection` *Flips x,y-coordinates horizontally in one half to account for changing sides at half-time*

Description

Normalises direction of attack in both halves of both teams by flipping x,y-coordinates horizontally in either the first or second half; i.e. teams attack in the same direction all game despite changing sides at half-time.

Usage

```
soccerFlipDirection(dat, pitchLength = 105, pitchWidth = 68,
  period = "period", periodToFlip = 1, x = "x", y = "y")
```

Arguments

`dat`	= dataframe containing unnormalised x,y-coordinates
`pitchLength, pitchWidth`	= length, width of pitch in metres
`period`	= name of variable containing period labels
`periodToFlip`	= identity of period to flip
`x, y`	= name of variables containing x,y-coordinates

Value

a dataframe

Examples

```
# fake period data for tromso dataset, and flip direction of '2nd half'
tromso %>%
  mutate(period = if_else(t > as.POSIXct("2013-11-07 21:14:00 GMT"), 1, 2))
  soccerFlipDirection(pitchLength = 120, pitchWidth = 80, periodToFlip = 2)
```

`soccerHeatmap` *Draw a heatmap on a soccer pitch.*

Description

Draws a heatmap showing player position frequency in each area of the pitch and adds soccer pitch outlines.

Usage

```
soccerHeatmap(df, lengthPitch = 105, widthPitch = 68, xBins = 10,
  yBins = NULL, arrow = c("none", "r", "l"), colPitch = "black",
  colLow = "white", colHigh = "red", title = NULL, subtitle = NULL,
  x = "x", y = "y")
```

Arguments

`df`	dataframe containing x,y-coordinates of player position
`lengthPitch, widthPitch`	numeric, length and width of pitch in metres.
`xBins, yBins`	integer, the number of horizontal (length-wise) and vertical (width-wise) bins the soccer pitch is to be divided up into. If no value for `yBins` is provided, it will take the value of `xBins`.
`arrow`	optional, adds arrow showing team attack direction as right (`'r'`) or left (`'l'`)
`colPitch`	pitch line colour
`colLow, colHigh`	character, colours for the low and high ends of the heatmap gradient.
`title, subtitle`	optional, adds title and subtitle to plot
`x, y`	= name of variables containing x,y-coordinates

Details

uses `ggplot2::geom_bin2d` to map 2D bin counts

Value

a ggplot object of a heatmap on a soccer pitch.

See Also

`soccerPitch` for a background soccer pitch for the purpose of drawing position maps, player trajectories, etc..

Examples

```
# Heatmap of Tromso IL #9's position
data(tromso)
soccerHeatmap(subset(tromso, id == 8), xBins = 10)

# Heatmap of France defensive pressure locations w/ ~5x5m bins (pitchLength / 5 = 21, pitchWidth / 5 = 13.6)
statsbomb %>%
  filter(type.name == "Pressure" & team.name == "France") %>%
  soccerHeatmap(x = "location.x", y = "location.y", xBins = 21, yBins = 14,
                title = "France (vs Argentina, 30th June 2016)",
                subtitle = "Defensive pressure heatmap")
```

---

soccerPath *Draw a path of player trajectory on a soccer pitch.*

Description

Draws a path connecting consecutive x,y-coordinates of a player on a soccer pitch.

Usage

```
soccerPath(dat, lengthPitch = 105, widthPitch = 68, col = "black",
  arrow = c("none", "r", "l"), fillPitch = "white", colPitch = "grey60",
  grass = FALSE, lwd = 1, title = NULL, subtitle = NULL,
  legend = TRUE, x = "x", y = "y", id = NULL, plot = NULL)
```

Arguments

`dat`	dataframe containing x,y-coordinates of player position
`lengthPitch, widthPitch`	length and width of pitch in metres
`col`	colour of path if no `'id'` is provided. If an `'id'` is present, colours from ColorBrewer's 'Paired' palette are used
`arrow`	optional, adds arrow showing team attack direction as right (`'r'`) or left (`'l'`)
`fillPitch, colPitch`	pitch background and line colour
`grass`	if `TRUE`, uses a more realistic pitch
`lwd`	player path thickness
`title, subtitle`	optional, adds title and subtitle to plot
`legend`	boolean, include legend
`x, y`	= name of variables containing x,y-coordinates
`id`	character, the name of the column containing player identity (only required if `'dat'` contains multiple players)
`plot`	plot to add path to, if desired

Value

a ggplot object

Examples

```
data(tromso)
# draw path of Tromso #8 over first 3 minutes (1800 frames)
subset(tromso, id == 8)[1:1800,] %>%
  soccerPath(col = "red", grass = TRUE, arrow = "r")

# draw path of all Tromso players over first minute (600 frames)
tromso %>%
  dplyr::group_by(id) %>%
  dplyr::slice(1:1200) %>%
  soccerPath(id = "id")
```

`soccerPitch` *Plot a soccer pitch ggplot object*

Description

Draws a soccer pitch as a ggplot object for the purpose of adding layers such as player positions, player trajectories, etc..

Usage

```
soccerPitch(lengthPitch = 105, widthPitch = 68, fillPitch = "white",
  colPitch = "grey60", grass = FALSE, arrow = c("none", "r", "l"),
  arrow_col = "default", lwd = 0.5, title = NULL, subtitle = NULL)
```

Arguments

`lengthPitch, widthPitch`	length and width of pitch in metres
`fillPitch, colPitch`	pitch fill and line colour
`grass`	if `TRUE`, uses a more realistic pitch
`arrow`	optional, adds arrow showing team attack direction as right (`'r'`) or left (`'l'`)
`arrow_col`	colour of attack direction arrow
`lwd`	numeric, pitch line width
`title, subtitle`	optional, adds title and subtitle to plot

Value

a ggplot object

See Also

`soccerPitchFG` for adding soccer pitch lines to an existing ggplot object

Examples

```
# get x,y-coords of player #8 during first 10 minutes
data(tromso)
dd <- subset(tromso, id == 9)[1:1200,]
# draw player path on pitch
soccerPitch(grass = TRUE) +
  geom_path(data = dd, aes(x, y))
```

`soccerPitchFG` *Helper function to add soccer pitch outlines to an existing ggplot object*

Description

Adds soccer pitch outlines (with transparent fill) to an existing ggplot object (e.g. heatmaps, passing maps, etc..)

Usage

```
soccerPitchFG(plot, lengthPitch = 105, widthPitch = 68,
  colPitch = "black", arrow = c("none", "r", "l"), arrow_col = "black",
  lwd = 0.5, title = NULL, subtitle = NULL)
```

Arguments

`plot`	an existing ggplot object to add pitch lines to
`lengthPitch, widthPitch`	length and width of pitch in metres
`colPitch`	pitch fill and line colour
`arrow`	optional, adds arrow showing team attack direction as right (`'r'`) or left (`'l'`)
`arrow_col`	colour of attack direction arrow
`lwd`	numeric, pitch line width
`title, subtitle`	optional, adds title and subtitle to plot

Value

a ggplot object

See Also

`soccerPitch` for plotting a soccer pitch for the purpose of drawing over event data, average position, player trajectories, etc..

soccerPositionMap *Plot average player position*

Description

Draws the average x,y-positions of each player from one or both teams on a soccer pitch.

Usage

```
soccerPositionMap(df, lengthPitch = 105, widthPitch = 68, fill1 = "red",
  col1 = "white", fill2 = "blue", col2 = "white", node_size = 6,
  label = TRUE, label_size = 3, label_col = "black", repel = FALSE,
  fillPitch = "white", colPitch = "grey60", lwd = 0.5, grass = FALSE,
  arrow = c("none", "r", "l"), title = NULL, subtitle = NULL, x = "x",
  y = "y", id = "id", team = NULL)
```

Arguments

df	dataframe containing x,y-coordinates of player position
lengthPitch, widthPitch	numeric, length and width of pitch in metres
fill1, fill2	character, fill colour of position points (team 1, team 2 (if present))
col1, col2	character, border colour of position points and labels (team 1, team 2 (if present))
node_size	numeric, size of position points
label	boolean, draw labels
label_size	numeric, size of labels
label_col	colour of labels
fillPitch	pitch fill colour
colPitch	pitch line colour
lwd	pitch line width
grass	if TRUE, uses a more realistic pitch
arrow	optional, adds arrow showing team attack direction as right ('r') or left ('l')
title, subtitle	optional, adds title and subtitle to plot
x, y	= name of variables containing x,y-coordinates
id	character, the name of the column containing player identity. Defaults to 'id'
team	character, the name of the column containing team identity. Optional, defaults to 'NULL'

See Also

soccerPitch for plotting a soccer pitch for the purpose of drawing over event data, average position, player trajectories, etc..

Examples

```
# average player position; one team w/ labels as player numbers
data(tromso)
soccerPositionMap(tromso, grass = TRUE)

# average pass position; one team w/ labels as shortened, non-overlapping player names
data(statsbomb)
statsbomb$name <- soccerShortenName(statsbomb$player.name)
statsbomb %>%
filter(type.name == "Pass" & team.name == "France" & minute < 43) %>%
soccerPositionMap(id = "name", x = "location.x", y = "location.y",
                  fill1 = "blue", label_col = "black",
                  arrow = "r", repel = T,
                  title = "France (vs Argentina, 30th June 2018)",
                  subtitle = "Average pass position (1' - 42')")

# average pass position; two teams w/ labels as shortened, non-overlapping player names (requires flipping one t
lengthPitch <- 105
widthPitch <- 68
statsbomb %>%
  filter(type.name == "Pass" & minute < 43) %>%
 mutate(location.x = if_else(team.name == "Argentina", lengthPitch - location.x, location.x),
     location.y = if_else(team.name == "Argentina", widthPitch - location.y, location.y)) %>%
 soccerPositionMap(team = "team.name", id = "name", x = "location.x", y = "location.y",
                   fill1 = "blue", fill2 = "lightblue", label_col = "black",
                   repel = T,
                   title = "France vs Argentina, 30th June 2018",
                   subtitle = "Average pass position (1' - 42')")
```

`soccerResample` *Resample the frequency of x,y,t- time series with linear interpolation of x,y-coordinates.*

Description

Downsample or upsample dataframe containing x,y-coordinates and a time variable 't' with linear interpolation of x,y-coordinates and constant interpolation of all other variables.

Usage

```
soccerResample(dat, r = 10, x = "x", y = "y", z = "z", t = "t")
```

Arguments

`dat`	= dataframe containing x,y-coordinates with time variable
`r`	resampling rate in frames per second
`x, y, z`	= name of variables containing x,y(,z)-coordinates
`t`	= name of variable containing time data

Value

a dataframe

Examples

```
# resample tromso dataset from ~21 fps to 10 fps
soccerResample(tromso)
```

---

soccerShortenName *Extract shortened player names*

Description

Helper function to extract last name from full player names

Usage

```
soccerShortenName(names)
```

Arguments

names	vector of strings containing full player names

Examples

```
attach(statsbomb)
statsbomb$name <- soccerShortenName(statsbomb$player.name)
```

---

soccerShotmap *Draw a shotmap on a half pitch from StatsBomb data*

Description

Draw a shotmap on a half pitch from StatsBomb data. Compatability with other (non-StatsBomb) shot data will be added soon.

Usage

```
soccerShotmap(dat, lengthPitch = 105, widthPitch = 68,
  colGoal = "skyblue", colMiss = "grey60", alpha = 0.8, legend = FALSE,
  theme = c("light", "dark", "grass"), lwd = 0.5)
```

Arguments

lengthPitch, widthPitch	length and width of pitch in metres
colGoal, colMiss	colour of points representing scored and missed shots
alpha	transparency of points
legend	boolean, include legend or not
lwd	numeric, pitch line width
fillPitch, colPitch	pitch background and line colour
grass	if TRUE, uses a more realistic pitch

Value

a ggplot object

Examples

```
data(statsbomb)

# shot map of France vs. Argentina (2018-06-30)
statsbomb %>%
  filter(team.name == "France") %>%
  soccerShotmap()
```

`soccerSpokes` *Visualise movement direction on a soccer pitch.*

Description

Draws spokes showing the direction of x,y-movements made in each sector of the pitch.

Usage

```
soccerSpokes(plot, df, lengthPitch = 105, widthPitch = 68, xBins,
  yBins = NULL, angleBins = 16, lwd = 0.5, minLength = 0.6,
  minAlpha = 0.4, legend = TRUE, x = "x", y = "y", angle = "angle")
```

Arguments

`plot`	plot of soccer pitch returned by `soccerPitch` to add spokes to
`df`	dataframe containing x,y-coordinates of player position in columns named `x` and `y` and angular information (in radians, ranging between `-pi` and `pi`) in a column specified by argument `angle`
`lengthPitch, widthPitch`	length and width of pitch in metres
`xBins, yBins`	integer, the number of horizontal (length-wise) and vertical (width-wise) bins the soccer pitch is to be divided up into. If no value for `yBins` is provided, it will take the value of `xBins`
`angleBins`	integer, the number of angle bins movement directions are divided up into. For example, a value of 4 clusters has direction vectors north, east, south and west
`lwd`	thickness of arrow lines
`minLength`	numeric, ratio between size of shortest arrow and longest arrow depending on number of events.
`legend`	if `TRUE`, adds legend for arrow thickness and transparency
`x, y, angle`	= name of variables containing x,y-coordinates and angular data

Value

a ggplot object

See Also

soccerPitch for drawing a heatmap of player position, soccerHeatmap for drawing a heatmap of player position

Examples

```
data(tromso_extra)
# resample movement dataset to plot 100 movement directions
# (in absence of pass / shot event data as yet)
id8 <- tromso_extra %>%
  dplyr::filter(id == 8) %>%
  dplyr::sample_n(100)
# 5x5 x,y-bins, 16 angle-bins, blank pitch
soccerPitch(pitchLength, pitchWidth) %>%
  soccerSpokes(id8, xBins = 5, angleBins = 16, minLength = 0.4)
# 10x10 x,y-bins, 8 angle-bins, grass pitch
soccerPitch(pitchLength, pitchWidth, grass = T) %>%
  soccerSpokes(id8, xBins = 10, angleBins = 8, minLength = 0.2, lwd = 1)
# draw spokes over player heatmap w/ 5x5 x,y-bins, 8 angle-bins
soccerHeatmap(id8, xBins = 5) %>%
  soccerSpokes(id8, xBins = 5, angleBins = 8, lwd = 1)
```

---

soccerTransform *Normalises x,y-coordinates to metres units for use with soccermatics functions*

Description

Normalises x,y-coordinates from between any arbitrary bounds to metre units bounded by [0 < x < pitchLength, 0 < y < pitchWidth]

Usage

```
soccerTransform(dat, xMin, xMax, yMin, yMax, pitchLength = 105,
  pitchWidth = 68, method = c("manual", "statsbomb"))
```

Arguments

dat	dataframe containing unnormalised x,y-coordinates named ‘x‘ and ‘y‘
xMin, xMax, yMin, yMax	range of x,y-coordinates possible in the raw dataset
pitchLength, pitchWidth	length, width of pitch in metres

Value

a dataframe

Examples

```
# Three examples with true pitch dimesions (in metres):
lengthPitch <- 101
widthPitch <- 68

# Example 1. Opta-style ------------------------------------------------------
# limits = [0 < x < 100, 0 < y < 100]
# centre of pitch = [50,50]

df <- data.frame(t = 1:12,
                 x = c(50,55,61,66,62,58,51,44,45,42,41,32),
                 y = c(50,48,47,40,42,45,49,51,59,75,88,100))

df <- soccerTransform(df, 0, 100, 0, 100, lengthPitch, widthPitch)

soccerPath(df, lengthPitch = lengthPitch, widthPitch = widthPitch)


# Example 2. StrataBet-style -------------------------------------------------
# limits = [0 < x < 420, -136 < y < 136]
# centre of pitch = [210,0]

df <- data.frame(t = 1:12,
                 x = c(210,222,201,192,178,170,143,122,104,91,75,60),
                 y = c(0,-5,-20,-12,-8,-2,4,8,13,20,30,45))

df <- soccerTransform(df, 0, 420, -136, 136, lengthPitch, widthPitch)

soccerPath(df, lengthPitch = lengthPitch, widthPitch = widthPitch)


# Example 3. Other -----------------------------------------------------------
# limits = [-5250 < x < 5250, -3400 < y < 3400]
# centre of pitch = [0,0]

xMin <- -5250
xMax <- 5250
yMin <- -3400
yMax <- 3400

df <- data.frame(x = c(0,-452,-982,-1099,-1586,-2088,-2422,-2999,-3200,-3857),
                 y = c(0,150,300,550,820,915,750,620,400,264))

df <- soccerTransform(df, -5250, 5250, -3400, 3400, lengthPitch, widthPitch)

soccerPath(df, lengthPitch = lengthPitch, widthPitch = widthPitch)
```

`soccerVelocity` *Compute instantaneous distance, speed and direction from x,y-coordinates*

Description

Compute instantaneous distance moved (in metres), speed (in metres per second), and direction (in radians) between subsequent frames in a dataframe of x,y-coordinates.

Usage

```
soccerVelocity(dat)
```

Arguments

`dat`	= dataframe containing unnormalised x,y-coordinates 'x' and 'y', time variable 't', and identifier 'id'

Value

a dataframe

Examples

```
# calculate distance, speed, and direction for tromso dataset
soccerVelocity(tromso)
```

`statsbomb` *Sample StatsBomb event data containing the x,y-locations and identity of players involved in pass events, shot events, defensive actions, and more.*

Description

Sample StatsBomb event data from the France vs. Argentina World Cup 2018 game on the 30th June 2018, made publicly available by StatsBomb here. Data contains 145 variables in total, including x,y-coordinates (`location.x`, `location.y`). StatsBomb pitch dimensions are 120m long and 80m wide, meaning `lengthPitch` should be specified as `120` and `widthPitch` as `80`. Event data for all World Cup games (and other competitions) are accessible via the StatsBombR package available here.

Usage

```
data(statsbomb)
```

Format

A dataframe containing 12000 frames of x,y-coordinates and timestamps from 11 players.

Source

ZXY Sport Tracking

References

StatsBomb Open Data

Examples

```
data(statsbomb)

# heatmap of France defensive pressure vs. Argentina (2018-06-30)
statsbomb %>%
  filter(type.name == "Pressure" & team.name == "France") %>%
  soccerHeatmap(x = "location.x", y = "location.y")
```

`tromso` *x,y-coordinates of 11 soccer players over 12000 frames each*

Description

x,y-coordinates of 11 soccer players over 10 minutes (Tromsø IL vs. Anzhi, 2013-11-07), captured at 20 Hz using the ZXY Sport Tracking system and made available in the publication ZXY Sport Tracking.

Usage

```
data(tromso)
```

Format

A dataframe containing 12000 frames of x,y-coordinates and timestamps from 11 players.

Source

ZXY Sport Tracking

References

Pettersen et al. (2014) Proceedings of the International Conference on Multimedia Systems (MMSys)

Examples

```
data(tromso)
# draw path of player #8 on a soccer pitch
soccerPitchBG(lengthPitch = 105, widthPitch = 68, grass = TRUE) +
  geom_path(data = subset(tromso, id == 8), aes(x, y), lwd = 2)
```

`tromso_extra` *x,y-coordinates and additional positional information on 11 soccer players over 12000 frames each*

Description

x,y-coordinates of 11 soccer players over 10 minutes (Tromsø IL vs. Anzhi, 2013-11-07), plus additional information on player heading, direction, energy, speed, and total distance. Data captured at 20 Hz using the ZXY Sport Tracking system and made available in the publication ZXY Sport Tracking.

Usage

```
data(tromso_extra)
```

Format

A dataframe containing 12000 frames of x,y-coordinates and timestamps from 11 players.

Source

ZXY Sport Tracking

References

Pettersen et al. (2014) Proceedings of the International Conference on Multimedia Systems (MMSys) (pdf)

Examples

```
data(tromso_extra)
# draw flow field showing mean direction of player #8's movement
soccerFlow(subset(tromso_extra, id == 8), bins = 5, grass = TRUE)
```

Index