

Package ‘BasketballAnalyzeR’

July 21, 2025

Type Package

Title Analysis and Visualization of Basketball Data

Version 0.8.0

Date 2025-04-17

Description Contains data and code to accompany the book
P. Zuccolotto and M. Manisera (2020) Basketball Data Science. Applications with R. CRC Press. ISBN 9781138600799.
For more details, see the page bdsports.unibs.it/basketballanalyzer/.

License GPL (>= 2.0)

Encoding UTF-8

LazyData true

URL <https://github.com/sndmrc/BasketballAnalyzeR/>

BugReports <https://github.com/sndmrc/BasketballAnalyzeR/issues>

Contact <basketballanalyzer.help@unibs.it>

Depends R (>= 3.4), ggplot2 (>= 3.4.0)

Imports plyr (>= 1.8.4), dplyr (>= 0.7.6), tidyr (>= 0.8.1), rlang (>= 0.4.3), magrittr (>= 1.5), ggrepel (>= 0.8), gridExtra (>= 2.3), MASS (>= 7.3), directlabels (>= 2018.05), corrplot (>= 0.80), PBSmapping (>= 2.70), sp (>= 1.3), operators (>= 0.1), stringr (>= 1.3), GGally (>= 1.4), statnet.common (>= 4.2), readr (>= 1.3), utils (>= 4.2.3), gtools (>= 3.9.4), data.table (>= 1.14), mathjaxr (>= 1.6), stats, grDevices, graphics

Suggests dendextend (>= 1.8), ggnetwork (>= 0.5), ggplotify (>= 0.0.3), network (>= 1.13.0)

RdMacros mathjaxr

RoxygenNote 7.3.2

NeedsCompilation no

Author Marco Sandri [aut, cre] (ORCID:

<<https://orcid.org/0000-0002-1422-5695>>),

Paola Zuccolotto [aut] (ORCID: <<https://orcid.org/0000-0003-4399-7018>>),

Marica Manisera [aut] (ORCID: <<https://orcid.org/0000-0002-2982-0243>>)

Maintainer Marco Sandri <basketballanalyzer.help@unibs.it>

Repository CRAN

Date/Publication 2025-04-18 03:40:02 UTC

Contents

assistnet	3
barline	5
bubbleplot	6
corranalysis	8
CreateRadialPlot	9
densityplot	11
drawNBACourt	13
expectedpts	14
fourfactors	16
hclustering	18
inequality	19
is.assistnet	20
is.corranalysis	21
is.fourfactors	22
is.hclustering	23
is.inequality	24
is.kclustering	25
is.MDSmap	26
is.simplereg	27
is.variability	28
kclustering	29
MDSmap	30
Obox	32
Pbox	33
PbP.BDB	34
PbPmanipulation	36
plot.assistnet	37
plot.corranalysis	38
plot.fourfactors	39
plot.hclustering	40
plot.inequality	42
plot.kclustering	43
plot.MDSmap	44
plot.simplereg	46
plot.variability	48
radialprofile	50
scatterplot	51
scoredifference	53
scoringprob	54
shotchart	56
simplereg	58

Tadd	60
Tbox	60
TOPboxes	62
variability	63

Index	65
--------------	-----------

assistnet	<i>Investigates the network of assists-shots in a team</i>
-----------	--

Description

The assistnet command provides a comprehensive analysis of a team's assist-shot network, revealing crucial insights into player interactions and on-court dynamics.

Usage

```
assistnet(
  data,
  assist = "assist",
  player = "player",
  points = "points",
  event.type = "event_type",
  normalize = FALSE,
  period.length = 12,
  time.thr = 0
)
```

Arguments

data	a data frame whose rows are field shots and columns are variables to be specified in assist, player, points, event.type (see Details).
assist	character, indicating the name of the variable with players who made the assists, if any.
player	character, indicating the name of the variable with players who made the shot.
points	character, indicating the name of the variable with points.
event.type	character, indicating the name of the variable with type of event (mandatory categories are "miss" for missed field shots and "shot" for field goals).
normalize	logical, if TRUE the number of assists is normalized (default normalize=FALSE, see Details).
period.length	numerical, the length of a quarter in minutes (default: 12 minutes as in NBA)
time.thr	numerical, Minimum number of minutes played together by a pair of players required for computing their normalized assist count. Pairs below time.thr are excluded to avoid inflation due to small denominators (default: time.thr = 0).

Details

The data data frame could also be a play-by-play dataset provided that rows corresponding to events different from field shots are not coded as "shot" in the event.type variable. (To be completed)

Normalization:

$$4 \cdot (\text{period.length}) \cdot \frac{(\text{number of assists})}{(\text{minutes played in attack by each couple of players})}$$

Value

A list with 3 elements, assistTable (a table), nodeStats (a data frame), and assistNet (a network object). See Details.

assistTable, the cross-table of assists made and received by the players.

nodeStats, a data frame with the following variables:

- FGM (fields goals made),
- FGM_AST (field goals made thanks to a teammate's assist),
- FGM_ASTp (percentage of FGM_AST over FGM),
- FGPTS (points scored with field goals),
- FGPTS_AST (points scored thanks to a teammate's assist),
- FGPTS_ASTp (percentage of FGPTS_AST over FGPTS),
- AST (assists made),
- ASTPTS (point scored by assist's teammates).

minTable, a square matrix with the total number of minutes played in attack by each pair of players; the elements on the principal diagonal are set to zero.

assistminTable, a matrix showing the assist frequency between player pairs, adjusted for minutes played together in attack and expressed per $4 \cdot \text{period.length}$ minutes.

assistNet, an object of class network that can be used for further network analysis with specific R packages (see [network](#))

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

P. Zuccolotto, M. Manisera and M. Sandri (2026) Advanced Basketball Data Science: With Applications in R. CRC Press.

Examples

```
PbP <- PbPmanipulation(PbP.BDB)
PbP.GSW <- subset(PbP, team=="GSW")
out <- assistnet(PbP.GSW)
plot(out)
## Not run:
  out <- assistnet(PbP.GSW, normalize=TRUE, time.thr=50)
  plot(out, edge.thr=5)

## End(Not run)
```

barline	<i>Draws a bar-line plot</i>
---------	------------------------------

Description

Draws a bar-line plot

Usage

```
barline(
  data,
  id,
  bars,
  line,
  order.by = id,
  decreasing = TRUE,
  labels.bars = NULL,
  label.line = NULL,
  position.bars = "stack",
  title = NULL
)
```

Arguments

data	a data frame.
id	character, name of the ID variable.
bars	character vector, names of the bar variables.
line	character, name of the line variable.
order.by	character, name of the variable used to order bars (on the x-axis).
decreasing	logical; if TRUE, decreasing order.
labels.bars	character vector, labels for the bar variables.
label.line	character, label for the line variable on the second y-axis (on the right).
position.bars	character, used to adjust the positioning of the bars in the plot; there are four main options: stack, fill, dodge, and identity.
title	character, plot title.

Value

A ggplot2 object

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (< basketballanalyzer.help@unibs.it >)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

Examples

```
dts <- subset(Pbox, Team=="Houston Rockets" & MIN>=500)
barline(data=dts, id="Player", bars=c("P2p", "P3p", "FTp"),
        line="MIN", order.by="Player",
        labels.bars=c("2P", "3P", "FT"), title="Houston Rockets")
```

bubbleplot

Draws a bubble plot

Description

Draws a bubble plot

Usage

```
bubbleplot(
  data,
  id,
  x,
  y,
  col,
  size,
  text.col = NULL,
  text.size = 2.5,
  scale.size = TRUE,
  labels = NULL,
  mx = NULL,
  my = NULL,
  mcol = NULL,
  title = NULL,
  repel = TRUE,
  text.legend = TRUE,
  hline = TRUE,
  vline = TRUE
)
```

Arguments

<code>data</code>	a data frame.
<code>id</code>	character, name of the ID variable.
<code>x</code>	character, name of the x-axis variable.
<code>y</code>	character, name of the y-axis variable.
<code>col</code>	character, name of variable on the color axis.
<code>size</code>	character, name of variable on the size axis.
<code>text.col</code>	character, name of variable for text colors.
<code>text.size</code>	numeric, text font size (default 2.5).
<code>scale.size</code>	logical; if TRUE, size variable is rescaled between 0 and 100.
<code>labels</code>	character vector, variable labels (on legend and axis).
<code>mx</code>	numeric, x-coordinate of the vertical axis; default is the mean value of x variable.
<code>my</code>	numeric, y-coordinate of the horizontal axis; default is the mean value of y variable.
<code>mcol</code>	numeric, midpoint of the diverging scale (see scale_colour_gradient2); default is the mean value of col variable.
<code>title</code>	character, plot title.
<code>repel</code>	logical; if TRUE, activate text repelling.
<code>text.legend</code>	logical; if TRUE, show the legend for text color.
<code>hline</code>	logical; if TRUE, a horizontal line is drawn with y intercept at the mean value of the variable on the y axis.
<code>vline</code>	logical; if TRUE, a vertical line is drawn with x intercept at the mean value of the variable on the x axis.

Value

A ggplot2 object

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

Examples

```
X <- with(Tbox, data.frame(T=Team, P2p=P2p, P3p=P3p, FTp=FTp, AS=P2A+P3A+FTA))
labs <- c("2-point shots (% made)", "3-point shots (% made)",
         "free throws (% made)", "Total shots attempted")
bubbleplot(X, id="T", x="P2p", y="P3p", col="FTp",
           size="AS", labels=labs)
```

corranalysis

Correlation analysis

Description

Correlation analysis

Usage

```
corranalysis(data, threshold = 0, sig.level = 0.95)
```

Arguments

data	a numeric matrix or data frame (see cor).
threshold	numeric, correlation cutoff (default 0); correlations in absolute value below threshold are set to 0.
sig.level	numeric, significance level (default 0.95); correlations with p-values greater than 1-sig.level are set to 0.

Value

A list with the following elements:

- `corr.mtx` (the complete correlation matrix)
- `corr.mtx.trunc` (the truncated correlation matrix)
- `cor.mtest` (the output of the significance test on correlations; see [cor.mtest](#))
- `threshold` correlation cutoff
- `sig.level` significance level

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) *Basketball Data Science: With Applications in R*. CRC Press.

See Also

[plot.corranalysis](#).

Examples

```

data <- data.frame(Pbox$PTS,Pbox$P3M,Pbox$P2M,
                  Pbox$OREB + Pbox$DREB,Pbox$AST,
                  Pbox$TOV,Pbox$STL,Pbox$BLK)/Pbox$MIN
names(data) <- c("PTS","P3M","P2M","REB","AST","TOV","STL","BLK")
data <- subset(data, Pbox$MIN >= 500)
out <- corranalysis(data, threshold = 0.5)

```

CreateRadialPlot *R function CreateRadialPlot by William D. Vickers, freely downloadable from the web*

Description

R function CreateRadialPlot by William D. Vickers, freely downloadable from the web

Usage

```

CreateRadialPlot(
  plot.data,
  axis.labels = colnames(plot.data)[-1],
  grid.min = -0.5,
  grid.mid = 0,
  grid.max = 0.5,
  centre.y = grid.min - ((1/9) * (grid.max - grid.min)),
  plot.extent.x.sf = 1.2,
  plot.extent.y.sf = 1.2,
  x.centre.range = 0.02 * (grid.max - centre.y),
  label.centre.y = FALSE,
  grid.line.width = 0.5,
  gridline.min.linetype = "longdash",
  gridline.mid.linetype = "longdash",
  gridline.max.linetype = "longdash",
  gridline.min.colour = "grey",
  gridline.mid.colour = "blue",
  gridline.max.colour = "grey",
  grid.label.size = 4,
  gridline.label.offset = -0.02 * (grid.max - centre.y),
  label.gridline.min = TRUE,
  axis.label.offset = 1.15,
  axis.label.size = 2.5,
  axis.line.colour = "grey",
  group.line.width = 1,
  group.point.size = 4,
  background.circle.colour = "yellow",
  background.circle.transparency = 0.2,
  plot.legend = if (nrow(plot.data) > 1) TRUE else FALSE,

```

```

    legend.title = "Player",
    legend.text.size = grid.label.size,
    titolo = FALSE
)

```

Arguments

```

plot.data      plot.data
axis.labels    axis.labels
grid.min       grid.min
grid.mid       grid.mid
grid.max       grid.max
centre.y       centre.y
plot.extent.x.sf
                plot.extent.x.sf
plot.extent.y.sf
                plot.extent.y.sf
x.centre.range x.centre.range
label.centre.y label.centre.y
grid.line.width
                grid.line.width
gridline.min.linetype
                gridline.min.linetype
gridline.mid.linetype
                gridline.mid.linetype
gridline.max.linetype
                gridline.max.linetype
gridline.min.colour
                gridline.min.colour
gridline.mid.colour
                gridline.mid.colour
gridline.max.colour
                gridline.max.colour
grid.label.size
                grid.label.size
gridline.label.offset
                gridline.label.offset
label.gridline.min
                label.gridline.min
axis.label.offset
                axis.label.offset
axis.label.size
                axis.label.size

```

```

axis.line.colour      axis.line.colour
group.line.width      group.line.width
group.point.size      group.point.size
background.circle.colour  background.circle.colour
background.circle.transparency  background.circle.transparency
plot.legend           plot.legend
legend.title          legend.title
legend.text.size      legend.text.size
titolo                plot title

```

Details

A description of the function can be found at the following link: http://rstudio-pubs-static.s3.amazonaws.com/5795_e6e6411731bb4f1b9cc7eb49499c2082.html

References

Vickers D.W. (2006) Multi-Level Integrated Classifications Based on the 2001 Census, PhD Thesis, School of Geography, The University of Leeds

densityplot	<i>Computes and plots kernel density estimation of shots with respect to a concurrent variable</i>
-------------	--

Description

Computes and plots kernel density estimation of shots with respect to a concurrent variable

Usage

```

densityplot(
  data,
  var,
  shot.type = "field",
  thresholds = NULL,
  best.scorer = FALSE,
  period.length = 12,
  bw = NULL,
  title = NULL
)

```

Arguments

<code>data</code>	a data frame whose rows are shots and with the following columns: <code>ShotType</code> , <code>player</code> , <code>points</code> and at least one of <code>playlength</code> , <code>periodTime</code> , <code>totalTime</code> , <code>shot_distance</code> (the column specified in <code>var</code> , see Details).
<code>var</code>	character, a string giving the name of the numerical variable according to which the shot density is estimated. Available options: <code>"playlength"</code> , <code>"periodTime"</code> , <code>"totalTime"</code> , <code>"shot_distance"</code> .
<code>shot.type</code>	character, a string giving the type of shots to be analyzed. Available options: <code>"2P"</code> , <code>"3P"</code> , <code>"FT"</code> , <code>"field"</code> .
<code>thresholds</code>	numerical vector with two thresholds defining the range boundaries that divide the area under the density curve into three regions. If <code>NULL</code> default values are used.
<code>best.scorer</code>	logical; if <code>TRUE</code> , displays the player who scored the highest number of points in the corresponding interval.
<code>period.length</code>	numeric, the length of a quarter in minutes (default: 12 minutes as in NBA).
<code>bw</code>	numeric, the value for the smoothing bandwidth of the kernel density estimator or a character string giving a rule to choose the bandwidth (see density).
<code>title</code>	character, plot title.

Details

The data data frame could also be a play-by-play dataset provided that rows corresponding to events different from shots have `NA` in the `ShotType` variable.

Required columns:

- `ShotType`, a factor with the following levels: `"2P"`, `"3P"`, `"FT"` (and `NA` for events different from shots)
- `player`, a factor with the name of the player who made the shot
- `points`, a numeric variable (integer) with the points scored by made shots and `0` for missed shots
- `playlength`, a numeric variable with time between the shot and the immediately preceding event
- `periodTime`, a numeric variable with seconds played in the quarter when the shot is attempted
- `totalTime`, a numeric variable with seconds played in the whole match when the shot is attempted
- `shot_distance`, a numeric variable with the distance of the shooting player from the basket (in feet)

Value

A `ggplot2` plot

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

Examples

```
PbP <- PbPmanipulation(PbP.BDB)
data.team <- subset(PbP, team=="GSW" & result!="")
densityplot(data=data.team, shot.type="2P", var="playlength", best.scorer=TRUE)
data.opp <- subset(PbP, team!="GSW" & result!="")
densityplot(data=data.opp, shot.type="2P", var="shot_distance", best.scorer=TRUE)
```

drawNBAcourt

Add lines of NBA court to an existing ggplot2 plot

Description

Add lines of NBA court to an existing ggplot2 plot

Usage

```
drawNBAcourt(p, size = 1.5, col = "black", full = FALSE)
```

Arguments

p	a ggplot2 object.
size	numeric, line size.
col	line color.
full	logical; if TRUE draws a complete NBA court; if FALSE draws a half court.

Value

A ggplot2 object

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

Examples

```
library(ggplot2)
p <- ggplot(data.frame(x=0, y=0), aes(x,y)) + coord_fixed()
drawNBAcourt(p)
```

expectedpts	<i>Plots expected points of shots as a function of the distance from the basket (default) or another variable</i>
-------------	---

Description

Plots expected points of shots as a function of the distance from the basket (default) or another variable

Usage

```
expectedpts(
  data,
  var = "shot_distance",
  players = NULL,
  bw = 10,
  period.length = 12,
  palette = gg_color_hue,
  team = TRUE,
  col.team = "gray",
  col.hline = "black",
  xlab = NULL,
  x.range = "auto",
  title = NULL,
  legend = TRUE
)
```

Arguments

data	a data frame whose rows are field shots and with the following columns: points, event_type, player (only if the players argument is not NULL) and at least one of playlength, periodTime, totalTime, shot_distance (the column specified in var, see Details).
var	character, a string giving the name of the numerical variable according to which the expected points are estimated; available options "playlength", "periodTime", "totalTime", "shot_distance" (default).
players	subset of players to be displayed (optional; it can be used only if the player column is present in data).
bw	numeric, smoothing bandwidth of the kernel density estimator (see ksmooth).
period.length	numeric, the length of a quarter in minutes (default: 12 minutes as in NBA).
palette	color palette.
team	logical; if TRUE, draws the expected points for all the shots in data.
col.team	character, color of the expected points line for all the shots in data (default "gray").

<code>col.hline</code>	character, color of the dashed horizontal line (default "black") denoting the expected points for all the shots in data, not conditional to the variable in the x-axis.
<code>xlab</code>	character, x-axis label.
<code>x.range</code>	numerical vector or character; available options: NULL (x-axis range defined by <code>ggplot2</code> , the default), "auto" (internally defined x-axis range), or a 2-component numerical vector (user-defined x-axis range).
<code>title</code>	character, plot title.
<code>legend</code>	logical, if TRUE, color legend is displayed (only when <code>players</code> is not NULL).

Details

The data data frame could also be a play-by-play dataset provided that rows corresponding to events different from field shots have values different from "shot" or "miss" in the `event_type` variable.

Required columns:

- `event_type`, a factor with the following levels: "shot" for made field shots and "miss" for missed field shots
- `player`, a factor with the name of the player who made the shot
- `points`, a numeric variable (integer) with the points scored by made shots and 0 for missed shots
- `playlength`, a numeric variable with time between the shot and the immediately preceding event
- `periodTime`, a numeric variable with seconds played in the quarter when the shot is attempted
- `totalTime`, a numeric variable with seconds played in the whole match when the shot is attempted
- `shot_distance`, a numeric variable with the distance of the shooting player from the basket (in feet)

Value

A `ggplot2` plot

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) *Basketball Data Science: With Applications in R*. CRC Press.

Examples

```
PbP <- PbPmanipulation(PbP.BDB)
PbP.GSW <- subset(PbP, team=="GSW" & !is.na(shot_distance))
plyrs <- c("Stephen Curry", "Kevin Durant")
expectedpts(data=PbP.GSW, bw=10, players=plyrs, col.team='dodgerblue',
            palette=colorRampPalette(c("gray", "black")), col.hline="red")
```

fourfactors	<i>Calculates possessions, pace, offensive and defensive rating, and Four Factors</i>
-------------	---

Description

Calculates possessions, pace, offensive and defensive rating, and Four Factors

Usage

```
fourfactors(TEAM, OPP)
```

Arguments

TEAM	a data frame whose rows are the analyzed teams and with columns referred to the team achievements in the considered games (a box score); required variables: Team, P2A, P2M, P3A, P3M, FTA, FTM, OREB, DREB, TOV, MIN (see Details).
OPP	a data frame whose rows are the analyzed teams and with columns referred to the achievements of the opponents of each team in the considered game; required variables: Team, P2A, P2M, P3A, P3M, FTA, FTM, OREB, DREB, TOV, MIN (see Details).

Details

The rows of the TEAM and the OPP data frames must be referred to the same teams in the same order.
Required columns:

- Team, a factor with the name of the analyzed team
- P2A, a numeric variable (integer) with the number of 2-points shots attempted
- P2M, a numeric variable (integer) with the number of 2-points shots made
- P3A, a numeric variable (integer) with the number of 3-points shots attempted
- P3M, a numeric variable (integer) with the number of 3-points shots made
- FTA, a numeric variable (integer) with the number of free throws attempted
- FTM, a numeric variable (integer) with the number of free throws made

- OREB, a numeric variable (integer) with the number of offensive rebounds
- DREB, a numeric variable (integer) with the number of defensive rebounds
- TOV, a numeric variable (integer) with the number of turnovers
- MIN, a numeric variable (integer) with the number of minutes played

Value

An object of class `fourfactors`, i.e. a data frame with the following columns:

- Team, a factor with the name of the analyzed team
- POSS.Off, a numeric variable with the number of possessions of each team calculated with the formula $POSS = (P2A + P3A) + 0.44 * FTA - OREB + TOV$
- POSS.Def, a numeric variable with the number of possessions of the opponents of each team calculated with the formula $POSS = (P2A + P3A) + 0.44 * FTA - OREB + TOV$
- PACE.Off, a numeric variable with the pace of each team (number of possessions per minute played)
- PACE.Def, a numeric variable with the pace of the opponents of each team (number of possessions per minute played)
- ORtg, a numeric variable with the offensive rating (the points scored by each team per 100 possessions)
- DRtg, a numeric variable with the defensive rating (the points scored by the opponents of each team per 100 possessions)
- F1.Off, a numeric variable with the offensive first factor (effective field goal percentage)
- F2.Off, a numeric variable with the offensive second factor (turnovers per possession)
- F3.Off, a numeric variable with the offensive third factor (rebouding percentage)
- F4.Off, a numeric variable with the offensive fourth factor (free throw rate)
- F1.Def, a numeric variable with the defensive first factor (effective field goal percentage)
- F2.Def, a numeric variable with the defensive second factor (turnovers per possession)
- F3.Def, a numeric variable with the defensive third factor (rebouding percentage)
- F4.Def, a numeric variable with the defensive fourth factor (free throw rate)

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[plot.fourfactors](#)

Examples

```
selTeams <- c(2,6,10,11)
FF <- fourfactors(Tbox[selTeams,], Obox[selTeams,])
plot(FF)
```

hclustering

Agglomerative hierarchical clustering

Description

Agglomerative hierarchical clustering

Usage

```
hclustering(data, k = NULL, nclumax = 10, labels = NULL, linkage = "ward.D")
```

Arguments

data	numeric data frame.
k	integer, number of clusters.
nclumax	integer, maximum number of clusters (when k=NULL).
labels	character, row labels.
linkage	character, the agglomeration method to be used in <code>hclust</code> (see method in hclust).

Details

The `hclustering` function performs a preliminary standardization of columns in data.

Value

A `hclustering` object.

If `k` is `NULL`, the `hclustering` object is a list of 3 elements:

- `k` `NULL`
- `clusterRange` integer vector, values of `k` (from 1 to `nclumax`) at which the *variance between* of the clusterization is evaluated
- `VarianceBetween` numeric vector, values of the *variance between* evaluated for `k` in `clusterRange`

If `k` is not `NULL`, the `hclustering` object is a list of 5 elements:

- `k` integer, number of clusters
- Subjects data frame, subjects' cluster identifiers
- `ClusterList` list, clusters' composition
- Profiles data frame, clusters' profiles, i.e. the average of the variables within clusters and the cluster eterogeineity index (CHI)
- `Hclust` an object of class `hclust`, see [hclust](#)

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[plot.hclustering](#), [hclust](#)

Examples

```
data <- with(Pbox, data.frame(PTS, P3M, REB=OREB+DREB, AST, TOV, STL, BLK, PF))
data <- subset(data, Pbox$MIN >= 1500)
ID <- Pbox$Player[Pbox$MIN >= 1500]
hclu1 <- hclustering(data)
plot(hclu1)
hclu2 <- hclustering(data, labels=ID, k=7)
plot(hclu2)
```

inequality

Inequality analysis

Description

Inequality analysis

Usage

```
inequality(data, nplayers)
```

Arguments

data	numeric vector containing the achievements (e.g. scored points) of the players whose inequality has to be analyzed.
nplayers	integer, number of players to include in the analysis (ranked in nondecreasing order according to the values in data).

Value

A list with the following elements: Lorenz (cumulative distributions used to plot the Lorenz curve) and Gini (Gini coefficient).

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[plot.inequality](#)

Examples

```
Pbox.BN <- subset(Pbox, Team=="Brooklyn Nets")
out <- inequality(Pbox.BN$PTS, nplayers=8)
print(out)
plot(out)
```

is.assistnet

Reports whether x is a 'networkdata' object

Description

Reports whether x is a 'networkdata' object

Usage

```
is.assistnet(x)
```

Arguments

x an object to test.

Value

Returns TRUE if its argument is of class networkdata and FALSE otherwise.

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[assistnet](#)

Examples

```
PbP <- PbPmanipulation(PbP.BDB)
PbP.GSW <- subset(PbP, team=="GSW" & player!="")
out <- assistnet(PbP.GSW)
is.assistnet(out)
```

is.corranalysis	<i>Reports whether x is a 'corranalysis' object</i>
-----------------	---

Description

Reports whether x is a 'corranalysis' object

Usage

```
is.corranalysis(x)
```

Arguments

x an object to test.

Value

Returns TRUE if its argument is of class corranalysis and FALSE otherwise.

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketball.analyzer.help@gmail.com>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[corranalysis](#)

Examples

```
data <- data.frame(Pbox$PTS,Pbox$P3M,Pbox$P2M,
                  Pbox$OREB + Pbox$DREB,Pbox$AST,
                  Pbox$TOV,Pbox$STL,Pbox$BLK)/Pbox$MIN
names(data) <- c("PTS","P3M","P2M","REB","AST","TOV","STL","BLK")
data <- subset(data, Pbox$MIN >= 500)
out <- corranalysis(data)
is.corranalysis(out)
```

is.fourfactors *Reports whether x is a 'fourfactors' object*

Description

Reports whether x is a 'fourfactors' object

Usage

```
is.fourfactors(x)
```

Arguments

x an object to test.

Value

Returns TRUE if its argument is of class fourfactors and FALSE otherwise.

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketball.analyzer.help@gmail.com>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[fourfactors](#)

Examples

```
selTeams <- c(2,6,10,11)
out <- fourfactors(Tbox[selTeams,], Obox[selTeams,])
is.fourfactors(out)
```

is.hclustering *Reports whether x is a 'hclustering' object*

Description

Reports whether x is a 'hclustering' object

Usage

```
is.hclustering(x)
```

Arguments

x an object to test.

Value

Returns TRUE if its argument is of class hclustering and FALSE otherwise.

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketball.analyzer.help@gmail.com>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[hclustering](#)

Examples

```
data <- data.frame(Pbox$PTS, Pbox$P3M,
                  Pbox$OREB + Pbox$DREB, Pbox$AST,
                  Pbox$TOV, Pbox$STL, Pbox$BLK, Pbox$PF)
names(data) <- c("PTS", "P3M", "REB", "AST", "TOV", "STL", "BLK", "PF")
data <- subset(data, Pbox$MIN >= 1500)
ID <- Pbox$Player[Pbox$MIN >= 1500]
hclu <- hclustering(data, labels=ID, k=7)
is.hclustering(hclu)
```

is.inequality	<i>Reports whether x is a 'inequality' object.</i>
---------------	--

Description

Reports whether x is a 'inequality' object.

Usage

```
is.inequality(x)
```

Arguments

x an object to test.

Value

Returns TRUE if its argument is of class inequality and FALSE otherwise.

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketball.analyzer.help@gmail.com>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[inequality](#)

Examples

```
Pbox.BN <- subset(Pbox, Team=="Brooklyn Nets")
out <- inequality(Pbox.BN$PTS, npl=8)
is.inequality(out)
```

is.kclustering	<i>Reports whether x is a 'kclustering' object</i>
----------------	--

Description

Reports whether x is a 'kclustering' object

Usage

```
is.kclustering(x)
```

Arguments

x an object to test.

Value

Returns TRUE if its argument is of class kclustering and FALSE otherwise.

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketball.analyzer.help@gmail.com>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[kclustering](#)

Examples

```
FF <- fourfactors(Tbox,Obox)
X <- with(FF, data.frame(OD.Rtg=ORtg/DRtg,
                        F1.r=F1.Def/F1.Off, F2.r=F2.Off/F2.Def,
                        F3.O=F3.Def, F3.D=F3.Off))
X$P3M <- Tbox$P3M
X$STL.r <- Tbox$STL/Obox$STL
kclu <- kclustering(X)
is.kclustering(kclu)
```

is.MDSmap *Reports whether x is a 'MDSmap' object*

Description

Reports whether x is a 'MDSmap' object

Usage

```
is.MDSmap(x)
```

Arguments

x an object to test.

Value

Returns TRUE if its argument is of class MDSmap and FALSE otherwise.

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[MDSmap](#)

Examples

```
data <- subset(Pbox, MIN >= 1500)
data <- data.frame(data$PTS, data$P3M, data$P2M, data$OREB + data$DREB, data$AST,
                  data$TOV, data$STL, data$BLK)
mds <- MDSmap(data)
is.MDSmap(mds)
```

is.simplereg	<i>Reports whether x is a 'simplereg' object</i>
--------------	--

Description

Reports whether x is a 'simplereg' object

Usage

```
is.simplereg(x)
```

Arguments

x an object to test.

Value

Returns TRUE if its argument is of class simplereg and FALSE otherwise.

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[simplereg](#)

Examples

```
Pbox.sel <- subset(Pbox, MIN >= 500)
X <- Pbox.sel$AST/Pbox.sel$MIN
Y <- Pbox.sel$TOV/Pbox.sel$MIN
P1 <- Pbox.sel$Player
out <- simplereg(x=X, y=Y, type="lin")
is.simplereg(out)
```

is.variability	<i>Reports whether x is a 'variability' object</i>
----------------	--

Description

Reports whether x is a 'variability' object

Usage

```
is.variability(x)
```

Arguments

x an object to test.

Value

Returns TRUE if its argument is of class `variability` and FALSE otherwise.

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[variability](#)

Examples

```
Pbox.BC <- subset(Pbox, Team=="Oklahoma City Thunder" & MIN >= 500,
                 select=c("P2p", "P3p", "FTp", "P2A", "P3A", "FTA"))
out <- variability(data=Pbox.BC, data.var=c("P2p", "P3p", "FTp"),
                 size.var=c("P2A", "P3A", "FTA"), weight=TRUE)
is.variability(out)
```

kclustering	<i>K-means cluster analysis</i>
-------------	---------------------------------

Description

K-means cluster analysis

Usage

```
kclustering(  
  data,  
  k = NULL,  
  labels = NULL,  
  nclumax = 10,  
  nruns = 10,  
  iter.max = 50,  
  algorithm = "Hartigan-Wong"  
)
```

Arguments

data	numeric data frame.
k	integer, number of clusters.
labels	character, row labels.
nclumax	integer, maximum number of clusters (when k=NULL) used for calculating the explained variance as function of the number of clusters.
nruns	integer, run the k-means algorithm nruns times and chooses the best solution according to a maximum explained variance criterion.
iter.max	integer, maximum number of iterations allowed in k-means clustering (see kmeans).
algorithm	character, the algorithm used in k-means clustering (see kmeans).

Details

The `kclustering` function performs a preliminary standardization of columns in data.

Value

A `kclustering` object.

If `k` is `NULL`, the `kclustering` object is a list of 3 elements:

- `k` `NULL`
- `clusterRange` integer vector, values of `k` (from 1 to `nclumax`) at which the *variance between* of the clusterization is evaluated
- `VarianceBetween` numeric vector, values of the *variance between* evaluated for `k` in `clusterRange`

If `k` is not `NULL`, the `kclustering` object is a list of 4 elements:

- `k` integer, number of clusters
- Subjects data frame, subjects' cluster identifiers
- `ClusterList` list, clusters' composition
- Profiles data frame, clusters' profiles, i.e. the average of the variables within clusters and the cluster heterogeneity index (CHI)

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) *Basketball Data Science: With Applications in R*. CRC Press.

See Also

[plot.kclustering](#), [kmeans](#)

Examples

```
FF <- fourfactors(Tbox,Obox)
X <- with(FF, data.frame(OD.Rtg=ORtg/DRtg,
                        F1.r=F1.Def/F1.Off, F2.r=F2.Off/F2.Def,
                        F3.O=F3.Def, F3.D=F3.Off))
X$P3M <- Tbox$P3M
X$STL.r <- Tbox$STL/Obox$STL
kclu1 <- kclustering(X)
plot(kclu1)
kclu2 <- kclustering(X, k=9)
plot(kclu2)
```

MDSmap

Multidimensional scaling (MDS) in 2 dimensions

Description

Multidimensional scaling (MDS) in 2 dimensions

Usage

```
MDSmap(data, std = TRUE)
```

Arguments

- `data` a numeric matrix, data frame or "dist" object (see [dist](#)).
- `std` logical; if TRUE, data columns are standardized (centered and scaled).

Details

If data is an object of class "dist", std is not active and data is directly inputted into MASS::isoMDS.

Value

An object of class MDSmap, i.e. a list with 4 objects:

- `points`, a 2-column vector of the fitted configuration (see [isoMDS](#));
- `stress`, the final stress achieved in percent (see [isoMDS](#));
- `data`, the input data frame;
- `std`, the logical std input.

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[isoMDS](#), [plot.MDSmap](#).

Examples

```
data <- with(Pbox, data.frame(PTS, P3M, P2M, REB=OREB+DREB, AST, TOV, STL, BLK))
selp <- which(Pbox$MIN >= 1500)
data <- data[selp, ]
id <- Pbox$Player[selp]
mds <- MDSmap(data)
plot(mds, labels=id, z.var="P2M", level.plot=FALSE, palette=rainbow)
```

Obox

Opponents box scores dataset - NBA 2017-2018

Description

In this data frame cases (rows) are teams and variables (columns) are referred to achievements of the opponents in the NBA 2017-2018 Championship

Usage

Obox

Format

A data frame with 30 rows and 23 variables:

Team Analyzed team, character
GP Games Played, numeric
MIN Minutes Played, numeric
PTS Points Made, numeric
W Games won, numeric
L Games lost, numeric
P2M 2-Point Field Goals (Made), numeric
P2A 2-Point Field Goals (Attempted), numeric
P2p 2-Point Field Goals (Percentage), numeric
P3M 3-Point Field Goals (Made), numeric
P3A 3-Point Field Goals (Attempted), numeric
P3p 3-Point Field Goals (Percentage), numeric
FTM Free Throws (Made), numeric
FTA Free Throws (Attempted), numeric
FTp Free Throws (Percentage), numeric
OREB Offensive Rebounds, numeric
DREB Defensive Rebounds, numeric
AST Assists, numeric
TOV Turnovers, numeric
STL Steals, numeric
BLK Blocks, numeric
PF Personal Fouls, numeric
PM Plus/Minus, numeric

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

Pbox

Players box scores dataset - NBA 2017-2018

Description

In this data frame, cases (rows) are players and variables (columns) are referred to the individual achievements in the NBA 2017-2018 Championship

Usage

Pbox

Format

A data.frame with 605 rows and 22 variables:

Team Analyzed team, character

Player Analyzed player, character

GP Games Played, numeric

MIN Minutes Played, numeric

PTS Points Made, numeric

P2M 2-Point Field Goals (Made), numeric

P2A 2-Point Field Goals (Attempted), numeric

P2p 2-Point Field Goals (Percentage), numeric

P3M 3-Point Field Goals (Made), numeric

P3A 3-Point Field Goals (Attempted), numeric

P3p 3-Point Field Goals (Percentage), numeric

FTM Free Throws (Made), numeric

FTA Free Throws (Attempted), numeric

FTp Free Throws (Percentage), numeric

OREB Offensive Rebounds, numeric

DREB Defensive Rebounds, numeric

AST Assists, numeric

TOV Turnovers, numeric

STL Steals, numeric

BLK Blocks, numeric

PF Personal Fouls, numeric

PM Plus/Minus, numeric

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

PbP.BDB

Play-by-play dataset - NBA 2017-2018

Description

In this play-by-play data frame (NBA 2017-2018 Championship), the cases (rows) are the events occurred during the analyzed games and the variables (columns) are descriptions of the events in terms of type, time, players involved, score, area of the court.

Usage

PbP.BDB

Format

A data.frame with 37430 rows and 48 variables:

game_id Identification code for the game
data_set Season: years and type (Regular or Playoffs)
date Date of the game
a1 ... a5; h1 ... h5 Five players on the court (away team; home team)
period Quarter (>= 5: over-time)
away_score; home_score Score of the away/home team
remaining_time Time left in the quarter (h:mm:ss)
elapsed Time played in the quarter (h:mm:ss)
play_length Time since the immediately preceding event (h:mm:ss)
play_id Identification code for the play
team Team responsible for the event
event_type Type of event
assist Player who made the assist
away; home Players for the jump ball
block Player who blocked the shot
entered; left Player who entered/left the court
num Sequence number of the free throw
opponent Player who made the foul

outof Number of free throws accorded

player Player responsible for the event

points Scored points

possession Player who the jump ball is tipped to

reason Reason of the turnover

result Result of the shot (made or missed)

steal Player who stole the ball

type Type of play

shot_distance Field shots: distance from the basket

original_x ; original_y ; converted_x ; converted_y Coordinates of the shooting player. original: tracking coordinate system half court, (0,0) center of the basket; converted: coordinates in feet full court, (0,0) bottom-left corner

description Textual description of the event

Details

This data set has been kindly made available by BigDataBall (www.bigdataball.com), a data provider which leverages computer-vision technologies to enrich and extend sports datasets with lots of unique metrics. Since its establishment, BigDataBall has also supported many academic studies and is referred as a reliable source of validated and verified stats for NBA, MLB, NFL and WNBA.

The functions of BasketballAnalyzeR requiring play-by-play data as input need a data frame with some additional variables with respect to PbP.BDB. It can be obtained by means of the function [PbPmanipulation](#).

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

Source

<https://github.com/sndmrc/BasketballAnalyzeR>

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

PbPmanipulation	<i>Adapts the standard file supplied by BigDataBall to the format required by BasketballAnalyzeR</i>
-----------------	--

Description

Adapts the standard file supplied by BigDataBall to the format required by BasketballAnalyzeR

Usage

```
PbPmanipulation(data, period.length = 12, overtime.length = 5)
```

Arguments

data	a play-by-play data frame supplied by BigDataBall (www.bigdataball.com).
period.length	numeric, the length of a quarter in minutes (default: 12 minutes as in NBA)
overtime.length	numeric, the length of an overtime period in minutes (default: 5 minutes as in NBA)

Value

A play-by-play data frame.

The data frame generated by `PbPmanipulation` has the same variables of `PbP.BDB` (when necessary, coerced from one data type to another, e.g from factor to numeric) plus the following five additional variables:

- `periodTime`, time played in the quarter (in seconds)
- `totalTime`, time played in the match (in seconds)
- `playlength`, time since the immediately preceding event (in seconds)
- `ShotType`, type of shot (FT, 2P, 3P)
- `oppTeam`, name of the opponent team
- `hometeam`, name of the home team (generated conditionally on the presence of the variable `home_score`)

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) *Basketball Data Science: With Applications in R*. CRC Press.

See Also[PbP.BDB](#)**Examples**

```
PbP <- PbPmanipulation(PbP.BDB)
```

plot.assistnet	<i>Plots a network from a 'assistnet' object</i>
----------------	--

Description

Plots a network from a 'assistnet' object

Usage

```
## S3 method for class 'assistnet'
plot(
  x,
  layout = "kamadakawai",
  layout.par = list(),
  edge.thr = 0,
  edge.col.lim = NULL,
  edge.col.lab = NULL,
  node.size = NULL,
  node.size.lab = NULL,
  node.col = NULL,
  node.col.lim = NULL,
  node.col.lab = NULL,
  node.pal = colorRampPalette(c("white", "blue", "red")),
  edge.pal = colorRampPalette(c("white", "blue", "red")),
  ...
)
```

Arguments

x	an object of class assistnet.
layout	character, network vertex layout algorithm (see gplot.layout) such as "kamadakawai" (the default).
layout.par	a list of parameters for the network vertex layout algorithm (see gplot.layout).
edge.thr	numeric, threshold for edge values; values below the threshold are set to 0.
edge.col.lim	numeric vector of length two providing limits of the scale for edge color.
edge.col.lab	character, label for edge color legend.
node.size	character, indicating the name of the variable for node size (one of the columns of the nodeStats data frame in the x object, see assistnet).

node.size.lab	character, label for node size legend.
node.col	character, indicating the name of the variable for node color (one of the columns of the nodeStats data frame in the x object, see assistnet).
node.col.lim	numeric vector of length two providing limits of the scale for node color.
node.col.lab	character, label for node color legend.
node.pal	color palette for node colors.
edge.pal	color palette for edge colors.
...	other graphical parameters.

Value

A ggplot2 object

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[assistnet](#)

Examples

```
PbP <- PbPmanipulation(PbP.BDB)
PbP.GSW <- subset(PbP, team=="GSW" & player!="")
out <- assistnet(PbP.GSW)
plot(out, layout="circle", edge.thr=30, node.col="FGM_ASTp", node.size="ASTPTS")
```

plot.corranalysis	<i>Plots the correlation matrix and the correlation network from a 'corranalysis' object</i>
-------------------	--

Description

Plots the correlation matrix and the correlation network from a 'corranalysis' object

Usage

```
## S3 method for class 'corranalysis'
plot(x, horizontal = TRUE, title = NULL, ...)
```

Arguments

x an object of class `corranalysis`.
horizontal logical; if TRUE, the two plots are arranged horizontally.
title character, plot title.
... other graphical parameters

Value

A `ggplot2` object

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) *Basketball Data Science: With Applications in R*. CRC Press.

See Also

[corranalysis](#)

Examples

```

data <- data.frame(Pbox$PTS, Pbox$P3M, Pbox$P2M,
                  Pbox$OREB + Pbox$DREB, Pbox$AST,
                  Pbox$TOV, Pbox$STL, Pbox$BLK) / Pbox$MIN
names(data) <- c("PTS", "P3M", "P2M", "REB", "AST", "TOV", "STL", "BLK")
data <- subset(data, Pbox$MIN >= 500)
out <- corranalysis(data, threshold=0.5)
plot(out)

```

plot.fourfactors	<i>Plot possessions, pace, offensive and defensive rating, and Four Factors from a 'fourfactors' object</i>
------------------	---

Description

Plot possessions, pace, offensive and defensive rating, and Four Factors from a 'fourfactors' object

Usage

```

## S3 method for class 'fourfactors'
plot(x, title = NULL, ...)

```

Arguments

x an object of class `fourfactors`.
title character, plot title.
... other graphical parameters.

Details

The height of the bars in the two four factor plots are given by the difference between the team value and the average on the analyzed teams.

Value

A list of four `ggplot2` plots.

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) *Basketball Data Science: With Applications in R*. CRC Press.

See Also

[fourfactors](#)

Examples

```
selTeams <- c(2,6,10,11)
FF <- fourfactors(Tbox[selTeams,], Obox[selTeams,])
plot(FF)
```

plot.hclustering

Plots hierarchical clustering from a 'hclustering' object

Description

Plots hierarchical clustering from a 'hclustering' object

Usage

```
## S3 method for class 'hclustering'
plot(
  x,
  title = NULL,
  profiles = FALSE,
  ncol.arrange = NULL,
  circlize = FALSE,
  horiz = TRUE,
  cex.labels = 0.7,
  colored.labels = TRUE,
  colored.branches = FALSE,
  rect = FALSE,
  lower.rect = NULL,
  min.mid.max = NULL,
  ...
)
```

Arguments

<code>x</code>	an object of class <code>hclustering</code> .
<code>title</code>	character or vector of characters (when plotting radial plots of cluster profiles; see Value), plot title(s).
<code>profiles</code>	logical; if TRUE, displays radial plots of cluster profiles (active if <code>x\$k</code> is not NULL; see Value).
<code>ncol.arrange</code>	integer, number of columns when arranging multiple grobs on a page (active when plotting radial plots of cluster profiles; see Value).
<code>circlize</code>	logical; if TRUE, plots a circular dendrogram (active when plotting a dendrogram; see Value).
<code>horiz</code>	logical; if TRUE, plots an horizontal dendrogram (active when plotting a non circular dendrogram; see Value).
<code>cex.labels</code>	numeric, the magnification to be used for labels (active when plotting a dendrogram; see Value).
<code>colored.labels</code>	logical; if TRUE, assigns different colors to labels of different clusters (active when plotting a dendrogram; see Value).
<code>colored.branches</code>	logical; if TRUE, assigns different colors to branches of different clusters (active when plotting a dendrogram; see Value).
<code>rect</code>	logical; if TRUE, draws rectangles around the branches in order to highlight the corresponding clusters (active when plotting a dendrogram; see Value).
<code>lower.rect</code>	numeric, a value of how low should the lower part of the rect be (active when plotting a dendrogram; see option <code>lower_rect</code> of rect.dendrogram).
<code>min.mid.max</code>	numeric vector with 3 elements: lower bound, middle dashed line, upper bound for radial axis (active when plotting radial plots of cluster profiles; see Value).
<code>...</code>	other graphical parameters.

Value

If `x$k` is NULL, `plot.hclustering` returns a single `ggplot2` object, displaying the pattern of the explained variance vs the number of clusters.

If `x$k` is not NULL and `profiles=FALSE`, `plot.hclustering` returns a single `ggplot2` object, displaying the dendrogram.

If `x$k` is not NULL and `profiles=TRUE`, `plot.hclustering` returns a list of `ggplot2` objects, displaying the radial plots of the cluster profiles.

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[hclustering](#), [radialprofile](#).

Examples

```
data <- with(Pbox, data.frame(PTS, P3M, REB=OREB+DREB, AST, TOV, STL, BLK, PF))
data <- subset(data, Pbox$MIN >= 1500)
ID <- Pbox$Player[Pbox$MIN >= 1500]
hclu1 <- hclustering(data)
plot(hclu1)
hclu2 <- hclustering(data, labels=ID, k=7)
plot(hclu2)
```

<code>plot.inequality</code>	<i>Plot Lorenz curve from a 'inequality' object</i>
------------------------------	---

Description

Plot Lorenz curve from a 'inequality' object

Usage

```
## S3 method for class 'inequality'
plot(x, title = NULL, ...)
```

Arguments

<code>x</code>	an object of class <code>inequality</code> .
<code>title</code>	character, plot title.
<code>...</code>	other graphical parameters.

Value

A ggplot2 object.

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[inequality](#)

Examples

```
Pbox.BN <- subset(Pbox, Team=="Brooklyn Nets")
out <- inequality(Pbox.BN$PTS, nplayers=8)
print(out)
plot(out)
```

plot.kclustering

Plot k-means clustering from a 'kclustering' object

Description

Plot k-means clustering from a 'kclustering' object

Usage

```
## S3 method for class 'kclustering'
plot(
  x,
  title = NULL,
  ncol.arrange = NULL,
  min.mid.max = NULL,
  label.size = 2.5,
  ...
)
```

Arguments

x an object of class kclustering.

title character or vector of characters (when plotting radial plots of cluster profiles; see Value), plot title(s).

ncol.arrange	integer, number of columns when arranging multiple grobs on a page (active when plotting radial plots of cluster profiles; see Value).
min.mid.max	numeric vector with 3 elements: lower bound, middle dashed line, upper bound for radial axis (active when plotting radial plots of cluster profiles; see Value).
label.size	numeric; label font size (default 2.5).
...	other graphical parameters.

Value

If `x$k` is NULL, `plot.kclustering` returns a single `ggplot2` object, displaying the pattern of the explained variance vs the number of clusters.

If `x$k` is not NULL, `plot.kclustering` returns a list of `ggplot2` objects, displaying the radial plots of the cluster profiles.

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[kclustering](#), [radialprofile](#)

Examples

```
FF <- fourfactors(Tbox,Obox)
X <- with(FF, data.frame(OD.Rtg=ORtg/DRtg,
                        F1.r=F1.Def/F1.Off, F2.r=F2.Off/F2.Def,
                        F3.O=F3.Def, F3.D=F3.Off))
X$P3M <- Tbox$P3M
X$STL.r <- Tbox$STL/Obox$STL
kclu1 <- kclustering(X)
plot(kclu1)
kclu2 <- kclustering(X, k=9)
plot(kclu2)
```

plot.MDSmap	<i>Draws two-dimensional plots for multidimensional scaling (MDS) from a 'MDSmap' object</i>
-------------	--

Description

Draws two-dimensional plots for multidimensional scaling (MDS) from a 'MDSmap' object

Usage

```
## S3 method for class 'MDSmap'
plot(
  x,
  z.var = NULL,
  level.plot = TRUE,
  title = NULL,
  labels = NULL,
  repel_labels = FALSE,
  text_label = TRUE,
  label_size = 3,
  subset = NULL,
  col.subset = "gray50",
  zoom = NULL,
  palette = NULL,
  contour = FALSE,
  ncol.arrange = NULL,
  ...
)
```

Arguments

<code>x</code>	an object of class <code>MDSmap</code> .
<code>z.var</code>	character vector; defines the set of variables (available in the data data frame of <code>MDSmap</code>) used to color-coding the points in the map (for scatter plots) or, alternatively, overlap to the map a colored level plot.
<code>level.plot</code>	logical; if <code>TRUE</code> , draws a level plot, otherwise draws a scatter plot (not active if <code>zvar=NULL</code>).
<code>title</code>	character, plot title.
<code>labels</code>	character vector, labels for (x, y) points (only for single scatter plot).
<code>repel_labels</code>	logical; if <code>TRUE</code> , draw text labels using repelling (not for highlighted points) (see geom_text_repel).
<code>text_label</code>	logical; if <code>TRUE</code> , draw a rectangle behind the text labels (not active if <code>subset=NULL</code>).
<code>label_size</code>	numeric; label font size (default <code>label_size=3</code> , for scatter plots).
<code>subset</code>	logical vector, to select a subset of points to be highlighted.
<code>col.subset</code>	character, color for the subset of points.
<code>zoom</code>	numeric vector with 4 elements; <code>c(xmin, xmax, ymin, ymax)</code> for the x- and y-axis limits of the plot.
<code>palette</code>	color palette.
<code>contour</code>	logical; if <code>TRUE</code> , contour lines are plotted (not active if <code>level.plot=FALSE</code>).
<code>ncol.arrange</code>	integer, number of columns when arranging multiple grobs on a page.
<code>...</code>	other graphical parameters.

Value

A single ggplot2 plot or a list of ggplot2 plots

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[MDSmap](#)

Examples

```
data <- data.frame(Pbox$PTS, Pbox$P3M, Pbox$P2M, Pbox$OREB + Pbox$DREB, Pbox$AST,
Pbox$TOV, Pbox$STL, Pbox$BLK)
names(data) <- c('PTS', 'P3M', 'P2M', 'REB', 'AST', 'TOV', 'STL', 'BLK')
selp <- which(Pbox$MIN >= 1500)
data <- data[selp,]
id <- Pbox$Player[selp]
mds <- MDSmap(data)
plot(mds, labels=id, z.var="P2M", level.plot=FALSE, palette=rainbow)
```

plot.simplereg

Plot simple regression from a 'simplereg' object

Description

Plot simple regression from a 'simplereg' object

Usage

```
## S3 method for class 'simplereg'
plot(
  x,
  labels = NULL,
  subset = NULL,
  Lx = 0.01,
  Ux = 0.99,
  Ly = 0.01,
  Uy = 0.99,
  title = "Simple regression",
  xtitle = NULL,
  ytitle = NULL,
  repel = TRUE,
```

```
    ...  
  )
```

Arguments

x	an object of class <code>simplereg</code> .
labels	character, labels for subjects.
subset	an optional vector specifying a subset of observations to be highlighted in the graph or <code>subset='quant'</code> to highlight observations with coordinates above and below the upper and lower quantiles of the variables on the x- and y-axis (<code>Lx</code> , <code>Ux</code> , <code>Ly</code> , <code>Uy</code>).
Lx	numeric; if <code>subset='quant'</code> , lower quantile for the variable on the x-axis (default = 0.01).
Ux	numeric; if <code>subset='quant'</code> , upper quantile for the variable on the x-axis (default = 0.99).
Ly	numeric; if <code>subset='quant'</code> , lower quantile for the variable on the y-axis (default = 0.01).
Uy	numeric; if <code>subset='quant'</code> , upper quantile for the variable on the y-axis (default = 0.99).
title	character, plot title.
xtitle	character, x-axis label.
yttitle	character, y-axis label.
repel	logical, if TRUE (the default) text labels <code>repel</code> away from each other.
...	other graphical parameters.

Value

A `ggplot2` object

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) *Basketball Data Science: With Applications in R*. CRC Press.

See Also

[simplereg](#)

Examples

```
Pbox.sel <- subset(Pbox, MIN >= 500)
X <- Pbox.sel$AST/Pbox.sel$MIN
Y <- Pbox.sel$TOV/Pbox.sel$MIN
Pl <- Pbox.sel$Player
mod <- simplereg(x=X, y=Y, type="lin")
plot(mod)
```

plot.variability	<i>Plots a variability diagram from a 'variability' object</i>
------------------	--

Description

Plots a variability diagram from a 'variability' object

Usage

```
## S3 method for class 'variability'
plot(
  x,
  title = "Variability diagram",
  ylim = NULL,
  ylab = NULL,
  size.lim = NULL,
  max.circle = 25,
  n.circle = 4,
  leg.brk = NULL,
  leg.pos = "right",
  leg.just = "left",
  leg.nrow = NULL,
  leg.title = NULL,
  leg.title.pos = "top",
  ...
)
```

Arguments

x	an aobject of class variability.
title	character, plot title.
ylim	numeric vector of length two, y-axis limits.
ylab	character, y-axis label.
size.lim	numeric vector of length two, set limits of the bubbles' size scale (see limits of scale_size).
max.circle	numeric, maximum size of the size plotting symbol (see range of scale_size).
n.circle	integer; if leg.brk=NULL, set a sequence of about n.circle+1 equally spaced 'round' values which cover the range of the values used to set the bubbles' size.

leg.brk	numeric vector, breaks for bubbles' size legend (see breaks of scale_size).
leg.pos	character or numeric vector of length two, legend position; available options "none", "left", "right" (default), "bottom", "top", or a <code>c(x,y)</code> numeric vector (x and y are coordinates of the legend box; their values should be between 0 and 1; <code>c(0,0)</code> corresponds to the bottom-left and <code>c(1,1)</code> corresponds to the top-right position).
leg.just	character or numeric vector of length two; anchor point for positioning legend inside plot ("left" (default), "center", "right" or two-element numeric vector) or the justification according to the plot area when positioned outside the plot.
leg.nrow	integer, number of rows of the bubbles' size legend.
leg.title	character, title of the bubbles' size legend.
leg.title.pos	character, position of the legend title; available options: "top" (default for a vertical legend), "bottom", "left" (default for a horizontal legend), or "right".
...	other graphical parameters.

Value

A `ggplot2` object

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) *Basketball Data Science: With Applications in R*. CRC Press.

See Also

[variability](#)

Examples

```
Pbox.BC <- subset(Pbox, Team=="Oklahoma City Thunder" & MIN >= 500,
  select=c("P2p", "P3p", "FTp", "P2A", "P3A", "FTA"))
out <- variability(data=Pbox.BC, data.var=c("P2p", "P3p", "FTp"),
  size.var=c("P2A", "P3A", "FTA"), weight=TRUE)
plot(out, leg.brk=c(10,25,50,100,500,1000), max.circle=30)
```

radialprofile *Draws radial plots for player profiles*

Description

Draws radial plots for player profiles

Usage

```
radialprofile(  
  data,  
  perc = FALSE,  
  std = TRUE,  
  title = NULL,  
  ncol.arrange = NULL,  
  min.mid.max = NULL,  
  label.size = 2.5  
)
```

Arguments

data	a data frame.
perc	logical; if perc=TRUE, std=FALSE and min.mid.max=NULL, set axes range between 0 and 100 and set the middle dashed line at 50.
std	logical; if std=TRUE, variables are preliminarily standardized.
title	character vector, titles for radial plots.
ncol.arrange	integer, number of columns in the grid of arranged plots.
min.mid.max	numeric vector with 3 elements: lower bound, middle dashed line, upper bound for radial axis.
label.size	numeric; label font size (default 2.5).

Value

A list of ggplot2 radial plots or, if ncol.arrange=NULL, a single ggplot2 plot of arranged radial plots

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketball.analyzer.help@gmail.com>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[plot.kclustering](#)

Examples

```

data("Pbox")
Pbox.PG <- Pbox[1:6,]
X <- data.frame(Pbox.PG$P2M, Pbox.PG$P3M, Pbox.PG$OREB+Pbox.PG$DREB,
               Pbox.PG$AST, Pbox.PG$TO)/Pbox.PG$MIN
names(X) <- c("P2M", "P3M", "REB", "AST", "TO")
radialprofile(data=X, ncol.arrange=3, title=Pbox.PG$Player)

```

scatterplot

Draws a scatter plot or a matrix of scatter plots

Description

Draws a scatter plot or a matrix of scatter plots

Usage

```

scatterplot(
  data,
  data.var,
  z.var = NULL,
  palette = NULL,
  labels = NULL,
  repel_labels = FALSE,
  text_label = TRUE,
  label_size = 3,
  subset = NULL,
  col.subset = "gray50",
  zoom = NULL,
  title = NULL,
  legend = TRUE,
  upper = list(continuous = "cor", combo = "box_no_facet", discrete = "facetbar", na =
    "na"),
  lower = list(continuous = "points", combo = "facethist", discrete = "facetbar", na =
    "na"),
  diag = list(continuous = "densityDiag", discrete = "barDiag", na = "naDiag")
)

```

Arguments

data	an object of class <code>data.frame</code> .
data.var	character or numeric vector, name or column number of variables (in data object) used on the axes of scatter plot(s).
z.var	character or number, name or column number of variable (in data object) used to assign colors to points (see Details).
palette	color palette (active when plotting a single scatter plot; see Value).

labels	character vector, labels for points (active when plotting a single scatter plot, see Value).
repel_labels	logical; if TRUE, draws text labels of not highlighted points using repelling (active when plotting a single scatter plot; see Value).
text_label	logical; if TRUE, draws a rectangle behind the labels of highlighted points (active when plotting a single scatter plot; see Value).
label_size	numeric; label font size (default label_size=3).
subset	logical or numeric vector, to select a subset of points to be highlighted (active when plotting a single scatter plot; see Value).
col_subset	character, color for the labels and rectangles of highlighted points (active when plotting a single scatter plot; see Value).
zoom	numeric vector with 4 elements; c(xmin, xmax, ymin, ymax) for the x- and y-axis limits of the plot (active when plotting a single scatter plot; see Value).
title	character, plot title.
legend	logical, if legend=FALSE legend is removed (active when plotting a single scatter plot with z.var not NULL; see Value).
upper	list, may contain the variables continuous, combo, discrete, and na (active when plotting a matrix of scatter plot; see Value and upper in ggpairs)
lower	list, may contain the variables continuous, combo, discrete, and na (active when plotting a matrix of scatter plot; see Value and lower in ggpairs)
diag	list, may contain the variables continuous, discrete, and na (active when plotting a matrix of scatter plot; see Value and diag in ggpairs)

Details

If `length(data.var)=2`, the variable specified in `z.var` can be numeric or factor; if `length(data.var)>2`, the variable specified in `z.var` must be a factor.

Value

A `ggplot2` object with a single scatter plot if `length(data.var)=2` or a matrix of scatter plots if `length(data.var)>2`.

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketball.analyzer.help@gmail.com>)

References

P. Zuccolotto and M. Manisera (2020) *Basketball Data Science: With Applications in R*. CRC Press.

See Also

[ggpairs](#)

Examples

```
# Single scatter plot
Pbox.sel <- subset(Pbox, MIN>= 500)
X <- data.frame(AST=Pbox.sel$AST/Pbox.sel$MIN, TOV=Pbox.sel$TOV/Pbox.sel$MIN)
X$PTSpm <- Pbox.sel$PTS/Pbox.sel$MIN
mypal <- colorRampPalette(c("blue", "yellow", "red"))
scatterplot(X, data.var=c("AST", "TOV"), z.var="PTSpm", labels=1:nrow(X), palette=mypal)
# Matrix of scatter plots
data <- Pbox[1:50, c("PTS", "P3M", "P2M", "OREB", "Team")]
scatterplot(data, data.var=1:4, z.var="Team")
```

scoredifference

*Computes the score difference between the two teams in the match***Description**

Computes the score difference between the two teams in the match

Usage

```
scoredifference(PbP_data, team_name, player_data, team_data)
```

Arguments

PbP_data	a play-by-play data frame, previously handled by PbPmanipulation
team_name	name of the team we are interested in. The name can be either shortened (e.g. CLE) or extended (e.g. Cleveland Cavaliers)
player_data	dataframe containing the boxscore data of all players of a particular season. We need it to know the players who have played at least one match for a team during the season. This dataframe might be substituted by a dataframe which has a column Player containing in each row the name of the players and a second column Team containing the extended name (e.g. Golden State Warriors) of the team in which the player has played at least one match. If a player has played at least one match for more than one team during the same season, he/she will have a row for each franchise where he has played
team_data	dataframe, contains several data regarding the teams in the NBA. Inside this function it is used only to check if team_name corresponds to a team in the NBA. If the teams in the play-by-play data studied are the same as in the 2017-18 season, Tadd (the dataframe contained in the BasketballAnalyzeR package, regarding the 2017-18 season) can be used

Details

The score difference computed by the function can be different from the simple difference between the score of the home team and the one of the away team, as we have to take account of the points scored during an action. Indeed, the value of `score.diff` indicates the difference in the score while the action was played

Value

the initial play-by-play dataframe, with two additional columns:

- `score.diff`: difference between the score of `team_name` and the score of the opposite team (see details for more informations)

`*isHome`: boolean which indicates if `team_name` is the home team in that play-by-play row

Author(s)

Andrea Fox

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

P. Zuccolotto, M. Manisera and M. Sandri (2018) Big data analytics for modeling scoring probability in basketball: The effect of shooting under high pressure conditions. International Journal of Sports Science & Coaching.

Examples

```
## Not run:
PbP <- PbPmanipulation(PbP.BDB)
PbP <- scoredifference(PbP, team_name="GSW", player_data=Pbox, team_data=Tadd)

## End(Not run)
```

scoringprob

Plots scoring probability of shots as a function of a given variable

Description

Plots scoring probability of shots as a function of a given variable

Usage

```
scoringprob(
  data,
  var,
  shot.type,
  players = NULL,
  bw = 20,
  period.length = 12,
  xlab = NULL,
  x.range = "auto",
  title = NULL,
  palette = gg_color_hue,
```

```

    team = TRUE,
    col.team = "dodgerblue",
    legend = TRUE
  )

```

Arguments

<code>data</code>	a data frame whose rows are shots and with the following columns: <code>result</code> , <code>ShotType</code> , <code>player</code> (only if the <code>players</code> argument is not <code>NULL</code>) and at least one of <code>playlength</code> , <code>periodTime</code> , <code>totalTime</code> , <code>shot_distance</code> (the column specified in <code>var</code> , see <code>Details</code>).
<code>var</code>	character, the string giving the name of the numerical variable according to which the scoring probability is estimated. Available options: <code>"playlength"</code> , <code>"periodTime"</code> , <code>"totalTime"</code> , <code>"shot_distance"</code> .
<code>shot.type</code>	character, the type of shots to be analyzed; available options: <code>"2P"</code> , <code>"3P"</code> , <code>"FT"</code> , <code>"field"</code> .
<code>players</code>	subset of players to be displayed (optional; it can be used only if the <code>player</code> column is present in <code>data</code>).
<code>bw</code>	numeric, the smoothing bandwidth of the kernel density estimator (see ksmooth).
<code>period.length</code>	numeric, the length of a quarter in minutes (default: 12 minutes as in NBA).
<code>xlab</code>	character, x-axis label.
<code>x.range</code>	numerical vector or character; available options: <code>NULL</code> (x-axis range defined by <code>ggplot2</code> , the default), <code>"auto"</code> (internally defined x-axis range), or a 2-component numerical vector (user-defined x-axis range).
<code>title</code>	character, plot title.
<code>palette</code>	color palette.
<code>team</code>	character; if <code>TRUE</code> draws the scoring probability for all the shots in data.
<code>col.team</code>	character, color of the scoring probability line for all the shots in data.
<code>legend</code>	character; if <code>TRUE</code> , color legend is displayed (only when <code>players</code> is not <code>NULL</code>).

Details

The `data` data frame could also be a play-by-play dataset provided that rows corresponding to events different from shots have `NA` in the `ShotType` variable.

Required columns:

- `result`, a factor with the following levels: `"made"` for made shots, `"miss"` for missed shots, and `" "` for events different from shots
- `ShotType`, a factor with the following levels: `"2P"`, `"3P"`, `"FT"` (and `NA` for events different from shots)
- `player`, a factor with the name of the player who made the shot
- `playlength`, a numeric variable with time between the shot and the immediately preceding event

- `periodTime`, a numeric variable with seconds played in the quarter when the shot is attempted
- `totalTime`, a numeric variable with seconds played in the whole match when the shot is attempted
- `shot_distance`, a numeric variable with the distance of the shooting player from the basket (in feet)

Value

A `ggplot2` plot

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) *Basketball Data Science: With Applications in R*. CRC Press.

Examples

```
PbP <- PbPmanipulation(PbP.BDB)
PbP.GSW <- subset(PbP, team=="GSW" & result!="")
players <- c("Kevin Durant", "Draymond Green", "Klay Thompson")
scoringprob(data=PbP.GSW, shot.type="2P", players=players,
             var="shot_distance", col.team="gray")
```

shotchart

Plots different kinds of charts based on shot coordinates

Description

Plots different kinds of charts based on shot coordinates

Usage

```
shotchart(
  data,
  x,
  y,
  z = NULL,
  z.fun = median,
  result = NULL,
  type = NULL,
  scatter = FALSE,
  num.sect = 7,
  n = 1000,
  col.limits = c(NA, NA),
```



```

    courtline.col = "black",
    bg.col = "white",
    sectline.col = "white",
    text.col = "white",
    legend = FALSE,
    drop.levels = TRUE,
    pt.col = "black",
    pt.alpha = 0.5,
    nbins = 25,
    palette = "mixed"
)

```

Arguments

<code>data</code>	A data frame whose rows are field shots and columns are half-court shot coordinates <code>x</code> and <code>y</code> , and optionally additional variables to be specified in <code>z</code> and/or <code>result</code> (see Details).
<code>x</code>	character, indicating the variable name of the <code>x</code> coordinate.
<code>y</code>	character, indicating the variable name of the <code>y</code> coordinate.
<code>z</code>	character, indicating the name of the variable used to color the points (if <code>type=NULL</code>) or the sectors (if <code>type="sectors"</code> , in this case <code>z</code> must be a numeric variable).
<code>z.fun</code>	function (active when <code>type="sectors"</code>), used to summarize the values of <code>z</code> variable within each sector (recommended: <code>mean</code> , <code>median</code>).
<code>result</code>	character (active when <code>type="sectors"</code> and <code>scatter=FALSE</code>), indicating the name of the factor with the shot result (allowed categories <code>made</code> and <code>missed</code>).
<code>type</code>	character, indicating the plot type; available options are <code>NULL</code> , <code>"sectors"</code> , <code>"density-polygons"</code> , <code>"density-raster"</code> , <code>"density-hexbin"</code> .
<code>scatter</code>	logical, if <code>TRUE</code> a scatter plot of the shots is added to the plot.
<code>num.sect</code>	integer (active when <code>type="sectors"</code>), number of sectors.
<code>n</code>	integer (active when <code>type="sectors"</code>), number of points used to draw arcs (must be <code>> 500</code>).
<code>col.limits</code>	numeric vector, (active when <code>z</code> is a numeric variable), limits <code>c(min, max)</code> for the gradient color scale of <code>z</code> variable.
<code>courtline.col</code>	color of court lines.
<code>bg.col</code>	background color.
<code>sectline.col</code>	color of sector lines (active when <code>type="sectors"</code>).
<code>text.col</code>	color of text annotation within sectors (active when <code>type="sectors"</code>).
<code>legend</code>	logical, if <code>TRUE</code> a legend for <code>z</code> is plotted.
<code>drop.levels</code>	logical, if <code>TRUE</code> unused levels of the <code>z</code> variable are dropped.
<code>pt.col</code>	color of points in the scatter plot.
<code>pt.alpha</code>	numeric, transparency of points in the scatter plot.
<code>nbins</code>	integer (active when <code>type="density-hexbin"</code>), number of bins.
<code>palette</code>	color palette; available options <code>"main"</code> , <code>"cool"</code> , <code>"hot"</code> , <code>"mixed"</code> , <code>"grey"</code> , <code>"bwr"</code> (blue, white, red).

Details

The data dataframe could also be a play-by-play dataset provided that rows corresponding to events different from field shots have missing x and y coordinates.

x and y coordinates must be expressed in feet; the origin of the axes is positioned at the center of the field.

Value

A ggplot2 object.

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketball.analyzer.help@gmail.com>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[drawNBAcourt](#), [geom_density_2d](#), [geom_hex](#)

Examples

```
PbP <- PbPmanipulation(PbP.BDB)
subdata <- subset(PbP, player=="Kevin Durant")
subdata$xx <- subdata$original_x/10
subdata$yy <- subdata$original_y/10-41.75
shotchart(data=subdata, x="xx", y="yy", scatter=TRUE)
shotchart(data=subdata, x="xx", y="yy", scatter=TRUE, z="result")
shotchart(data=subdata, x="xx", y="yy", scatter=TRUE, z="result",
          bg.col="black", courtline.col="white", palette="hot")
shotchart(data=subdata, x="xx", y="yy", result="result",
          type="sectors", sectline.col="gray", text.col="red")
shotchart(data=subdata, x="xx", y="yy", z="playlength", result="result",
          type="sectors", num.sect=5)
shotchart(data=subdata, x="xx", y="yy", type="density-polygons", palette="bwr")
shotchart(data=subdata, x="xx", y="yy", type="density-raster",
          scatter=TRUE, pt.col="tomato", pt.alpha=0.1)
shotchart(data=subdata, x="xx", y="yy", type="density-hexbin", nbins=30)
```

Description

Simple linear and nonparametric regression

Usage

```
simplereg(x, y, type = "lin", sp = NULL)
```

Arguments

x	numerical vector, input x values.
y	numerical vector, input y values.
type	character, type of regression; available options are: lin (linear regression, the default), pol (local polynomial regression of degree 2), ks (nonparametric kernel smoothing).
sp	numeric, parameter to control the degree of smoothing; span for local polynomial regression and bandwidth for ksmooth.

Value

An object of class `simplereg`, i.e. a list with the following objects:

- `Model`, the output model (linear regression, local polynomial regression, or kernel smoothing)
- `R2`, (in-sample) coefficient of determination
- `x`, input x values
- `y`, input y values
- `type`, type of regression

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketball.analyzer.help@gmail.com>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

See Also

[loess](#), [ksmooth](#)

Examples

```
Pbox.sel <- subset(Pbox, MIN >= 500)
X <- Pbox.sel$AST/Pbox.sel$MIN
Y <- Pbox.sel$TOV/Pbox.sel$MIN
P1 <- Pbox.sel$Player
mod <- simplereg(x=X, y=Y, type="lin")
```

Tadd	<i>Tadd dataset - NBA 2017-2018</i>
------	-------------------------------------

Description

In this data frame, the cases (rows) are the analyzed teams and the variables (columns) are qualitative information such as Conference, Division, final rank, qualification for Playoffs for the NBA 2017-2018 Championship.

Usage

Tadd

Format

A data frame with 30 rows and 6 variables:

Team Analyzed team (long name), factor

team Analyzed team (short name), factor

Conference Conference, factor

Division Division, factor

Rank Rank (end season), numeric

Playoff Playoff qualification (Yes or No), factor

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

Tbox	<i>Teams box scores dataset - NBA 2017-2018</i>
------	---

Description

In this data frame, cases (rows) are teams and variables (columns) are referred to team achievements in the different games in the NBA 2017-2018 Championship.

Usage

Tbox

Format

A data frame with 30 rows and 23 variables:

Team Analyzed team, character

GP Games Played, numeric

MIN Minutes Played, numeric

PTS Points Made, numeric

W Games won, numeric

L Games lost, numeric

P2M 2-Point Field Goals (Made), numeric

P2A 2-Point Field Goals (Attempted), numeric

P2p 2-Point Field Goals (Percentage), numeric

P3M 3-Point Field Goals (Made), numeric

P3A 3-Point Field Goals (Attempted), numeric

P3p 3-Point Field Goals (Percentage), numeric

FTM Free Throws (Made), numeric

FTA Free Throws (Attempted), numeric

FTp Free Throws (Percentage), numeric

OREB Offensive Rebounds, numeric

DREB Defensive Rebounds, numeric

AST Assists, numeric

TOV Turnovers, numeric

STL Steals, numeric

BLK Blocks, numeric

PF Personal Fouls, numeric

PM Plus/Minus, numeric

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto and M. Manisera (2020) Basketball Data Science: With Applications in R. CRC Press.

TOPboxes	<i>Calculate Team, Opponents and Players box scores (Tbox, Obox and Pbox)</i>
----------	---

Description

Calculate Team, Opponents and Players box scores (Tbox, Obox and Pbox)

Usage

```
TOPboxes(data, team)
```

Arguments

data	a data frame containing play-by-play data (see Details)
team	character, indicating the name of the team

Details

To compute Tbox and Obox, the function needs the following variables: `game_id`, `playlength`, `ShotType`, `points`, `result`, `team`, `oreb`, `dreb`, `PF`, `turnover`, `assist`, `block` and `steal`. If any of these variables is missing, an error message is displayed.

To compute Pbox, also the variables `player`, `a1 . . . a5`, `h1 . . . h5` and `hometeam` are needed. If any is omitted, only Tbox and Obox are given in output.

Note that the variables `assist`, `block` and `steal` can contain the logical indicator of whether the corresponding event has occurred (TRUE/FALSE or numerical 0/1) or the name of the involved player (character). In the former case, Tbox and Obox are fully computed, while the variables `AST`, `BLK` and `STL` are missing in the Pbox data frame. In the latter case, all the data frames Tbox, Obox and Pbox are fully computed.

TOPboxes omits the computation of the variables `W` (Games won) and `L` (Games lost). In fact, since we aim at computing box scores starting from whatever portion of play-by-play data (e.g., only a part of a game), in some cases, calculating the number of won and lost games does not make sense.

Value

A list with the following elements

- Tbox, the data frame of team box scores
- Obox, the data frame of opponents box scores
- Pbox, the data frame of player box scores

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

References

P. Zuccolotto, M. Manisera and M. Sandri (2026) *Advanced Basketball Data Science: With Applications in R*. CRC Press.

P. Zuccolotto and M. Manisera (2020) *Basketball Data Science: With Applications in R*. CRC Press.

See Also

[PbPmanipulation](#)

Examples

```
library(operators)
library(dplyr)
PbP <- PbPmanipulation(PbP.BDB)
PbP <- PbP %>%
  mutate(oreb = type %~% "rebound offensive",
         dreb = type %~% "rebound defensive",
         turnover = event_type=="turnover",
         PF = (event_type == "foul") & !(type %~% "technical") ) %>%
  mutate(across(c(player, assist, steal, block, h1:h5, a1:a5), as.character)) %>%
  as.data.frame()
out <- TOPboxes(PbP, team="GSW")
```

 variability

Variability analysis

Description

Variability analysis

Usage

```
variability(data, data.var, size.var, VC = TRUE, weight = FALSE)
```

Arguments

data	a data frame.
data.var	a vector of variable names or of column numbers defining (numeric) variables whose variability will be analyzed by <code>variability</code> .
size.var	a vector of variable names or of column numbers defining variables for weights (active only if <code>weight=TRUE</code>).
VC	logical; if <code>TRUE</code> , calculates variation coefficients of variables in <code>data.var</code> .
weight	logical; if <code>TRUE</code> , calculates weighted variation coefficients and standard deviations.

Value

A list with the following elements: ranges, standard deviations, variation coefficients, and two dataframes (data, size).

Author(s)

Marco Sandri, Paola Zuccolotto, Marica Manisera (<basketballanalyzer.help@unibs.it>)

Examples

```
Pbox.BC <- subset(Pbox, Team=="Oklahoma City Thunder" & MIN >= 500,
                 select=c("P2p", "P3p", "FTp", "P2A", "P3A", "FTA"))
list_variability <- variability(data=Pbox.BC, data.var=c("P2p", "P3p", "FTp"),
                               size.var=c("P2A", "P3A", "FTA"), weight=TRUE)
print(list_variability)
plot(list_variability, leg.brk=c(10,25,50,100,500,1000), max.circle=30)
```


Index

- * **datasets**
 - Obox, [32](#)
 - Pbox, [33](#)
 - PbP.BDB, [34](#)
 - Tadd, [60](#)
 - Tbox, [60](#)
- assistnet, [3](#), [21](#), [37](#), [38](#)
- barline, [5](#)
- bubbleplot, [6](#)
- cor, [8](#)
- cor.mtest, [8](#)
- corranalysis, [8](#), [21](#), [39](#)
- CreateRadialPlot, [9](#)
- density, [12](#)
- densityplot, [11](#)
- dist, [31](#)
- drawNBACourt, [13](#), [58](#)
- expectedpts, [14](#)
- fourfactors, [16](#), [22](#), [40](#)
- geom_density_2d, [58](#)
- geom_hex, [58](#)
- geom_text_repel, [45](#)
- ggpairs, [52](#)
- gplot.layout, [37](#)
- hclust, [18](#), [19](#)
- hclustering, [18](#), [23](#), [42](#)
- inequality, [19](#), [24](#), [43](#)
- is.assistnet, [20](#)
- is.corranalysis, [21](#)
- is.fourfactors, [22](#)
- is.hclustering, [23](#)
- is.inequality, [24](#)
- is.kclustering, [25](#)
- is.MDSmap, [26](#)
- is.simplereg, [27](#)
- is.variability, [28](#)
- isoMDS, [31](#)
- kclustering, [25](#), [29](#), [44](#)
- kmeans, [29](#), [30](#)
- ksmooth, [14](#), [55](#), [59](#)
- loess, [59](#)
- MDSmap, [26](#), [30](#), [45](#), [46](#)
- network, [4](#)
- Obox, [32](#)
- Pbox, [33](#)
- PbP.BDB, [34](#), [37](#)
- PbPmanipulation, [35](#), [36](#), [63](#)
- plot.assistnet, [37](#)
- plot.corranalysis, [8](#), [38](#)
- plot.fourfactors, [18](#), [39](#)
- plot.hclustering, [19](#), [40](#)
- plot.inequality, [20](#), [42](#)
- plot.kclustering, [30](#), [43](#), [50](#)
- plot.MDSmap, [31](#), [44](#)
- plot.simplereg, [46](#)
- plot.variability, [48](#)
- radialprofile, [42](#), [44](#), [50](#)
- rect.dendrogram, [41](#)
- scale_colour_gradient2, [7](#)
- scale_size, [48](#), [49](#)
- scatterplot, [51](#)
- scoredifference, [53](#)
- scoringprob, [54](#)
- shotchart, [56](#)
- simplereg, [27](#), [47](#), [58](#)

Tadd, [60](#)

Tbox, [60](#)

TOPboxes, [62](#)

variability, [28](#), [49](#), [63](#)