# Package 'BasketballAnalyzeR'

June 26, 2020

**Type** Package

**Title** Analysis and Visualization of Basketball Data

**Version** 0.5.0

**Author** Marco Sandri, Paola Zuccolotto, Marica Manisera

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

**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.

**License** GPL (>= 2.0)

**Encoding** UTF-8

**LazyData** true

**URL** https://github.com/sndmrc/BasketballAnalyzeR

**Contact** <basketballanalyzer.help@unibs.it>

**Depends** R (>= 3.4), ggplot2 (>= 2.2.0)

**Imports** hexbin (>= 1.27), 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), scales (>= 1.0), MASS (>= 7.3), directlabels (>= 2018.05), corrplot (>= 0.80), ggplotify (>= 0.0.3), network (>= 1.13.0), sna (>= 2.4), dendextend (>= 1.8), circlize (>= 0.4), PBSmapping (>= 2.70), sp (>= 1.3), operators (>= 0.1), stringr (>= 1.3), GGally (>= 1.4), statnet.common (>= 4.2), ggnetwork (>= 0.5), readr (>= 1.3), stats, grDevices, graphics

**RoxygenNote** 7.1.0

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2020-06-26 09:00:11 UTC

# R topics documented:

`assistnet` *Investigates the network of assists-shots in a team*

## Description

Investigates the network of assists-shots in a team

## Usage

```
assistnet(
  data,
  assist = "assist",
  player = "player",
  points = "points",
  event.type = "event_type"
)
```

## 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). |

## 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.

## 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).

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.

## Examples

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

---

# barline *Draws a bar-line plot*

---

## Description

Draws a bar-line plot

## Usage

```
barline(
  data,
  id,
  bars,
  line,
  order.by = id,
  decreasing = TRUE,
  labels.bars = NULL,
  label.line = NULL,
  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). |
| 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
)
```

## Arguments

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

## 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 that 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)
plot(out)
```

---

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

---

## 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

| | |
|---|---|
| data | a data frame whose rows are shots and with the following columns: ShotType, player, points 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 shot density is estimated. Available options: "playlength", "periodTime", "totalTime", "shot_distance". |
| shot.type | character, a string giving the type of shots to be analyzed. Available options: "2P", "3P", "FT", "field". |
| thresholds | numerical vector with two thresholds defining the range boundaries that divide the area under the density curve into three regions. If NULL default values are used. |
| best.scorer | logical; if TRUE, displays the player who scored the highest number of points in the corresponding interval. |
| period.length | numeric, the length of a quarter in minutes (default: 12 minutes as in NBA). |
| bw | 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). |
| title | 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 *Plots expected points of shots as a function of the distance from the basket (default) or another variable*

---

## 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"). |
| col.hline | 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. |
| xlab | character, x-axis label. |
| x.range | numerical vector or character; available options: NULL (x-axis range defined by ggplot2, the default), "auto" (internally defined x-axis range), or a 2-component numerical vector (user-defined x-axis range). |
| title | character, plot title. |
| legend | logical, if TRUE, color legend is displayed (only when players 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 even_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))
plrys <- c("Stephen Curry","Kevin Durant")
expectedpts(data=PbP.GSW, bw=10, players=plrys, col.team='dodgerblue',
        palette=colorRampPalette(c("gray","black")), col.hline="red")
```

---

# fourfactors *Calculates possessions, pace, offensive and defensive rating, and Four Factors*

---

## 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 columms:

- 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 hclust (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` *Reports whether x is a 'corranalysis' object*

## 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` *Reports whether x is a 'inequality' object.*

## 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` *Reports whether x is a 'kclustering' object*

## 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` *Reports whether x is a 'simplereg' object*

## 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
Pl <- Pbox.sel$Player
out <- simplereg(x=X, y=Y, type="lin")
is.simplereg(out)
```

`is.variability` *Reports whether x is a 'variability' object*

## 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` *K-means cluster analysis*

## 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 eterogeineity 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

## 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` *Adapts the standard file supplied by BigDataBall to the format required by BasketballAnalyzeR*

---

## Description

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

## Usage

```
PbPmanipulation(data)
```

## Arguments

| | |
|---|---|
| `data` | a play-by-play data frame supplied by BigDataBall. |

## 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

## 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` *Plots a network from a 'assistnet' object*

---

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

---

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

---

## 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

| | |
|---|---|
| x | an object of class hclustering. |
| title | character or vector of characters (when plotting radial plots of cluster profiles; see Value), plot title(s). |
| profiles | logical; if TRUE, displays radial plots of cluster profiles (active if x$k is not NULL; see Value). |
| ncol.arrange | integer, number of columns when arranging multiple grobs on a page (active when plotting radial plots of cluster profiles; see Value). |
| circlize | logical; if TRUE, plots a circular dendrogram (active when plotting a dendrogram; see Value). |

| | |
|---|---|
| horiz | logical; if TRUE, plots an horizontal dendrogram (active when plotting a non circular dendrogram; see Value). |
| cex.labels | numeric, the magnification to be used for labels (active when plotting a dendrogram; see Value). |
| colored.labels | logical; if TRUE, assigns different colors to labels of different clusters (active when plotting a dendrogram; see Value). |
| colored.branches | logical; if TRUE, assigns different colors to branches of different clusters (active when plotting a dendrogram; see Value). |
| rect | logical; if TRUE, draws rectangles around the branches in order to highlight the corresponding clusters (active when plotting a dendrogram; see Value). |
| lower.rect | numeric, a value of how low should the lower part of the rect be (active when plotting a dendrogram; see option lower_rect of rect.dendrogram). |
| 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). |
| ... | 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)
```

`plot.inequality` *Plot Lorenz curve from a 'inequality' object*

## Description

Plot Lorenz curve from a 'inequality' object

## Usage

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

## Arguments

| | |
|---|---|
| `x` | an object of class `inequality`. |
| `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

`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, ...)
```

## 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). |
| `...` | 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` *Draws two-dimensional plots for multidimensional scaling (MDS) from a 'MDSmap' object*

---

## 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,
  subset = NULL,
  col.subset = "gray50",
  zoom = NULL,
  palette = NULL,
  contour = FALSE,
  ncol.arrange = NULL,
  ...
)
```

## Arguments

| | |
|---|---|
| `x` | an object of class `MDSmap`. |
| `z.var` | character vector; defines the set of variables (available in the `data` data frame of `MDSmap`) used to color-coding the points in the map (for scatter plots) or, alternatively, overlap to the map a colored level plot. |

| | |
|---|---|
| level.plot | logical; if TRUE, draws a level plot, otherwise draws a scatter plot (not active if zvar=NULL). |
| title | character, plot title. |
| labels | character vector, labels for (x, y) points (only for single scatter plot). |
| repel_labels | logical; if TRUE, draw text labels using repelling (not for highlighted points) (see geom_text_repel). |
| text_label | logical; if TRUE, draw a rectangle behind the text labels (not active if subset=NULL). |
| subset | logical vector, to select a subset of points to be highlighted. |
| col.subset | character, color for the subset of points. |
| zoom | numeric vector with 4 elements; c(xmin,xmax,ymin,ymax) for the x- and y-axis limits of the plot. |
| palette | color palette. |
| contour | logical; if TRUE, contour lines are plotted (not active if level.plot=FALSE). |
| ncol.arrange | integer, number of columns when arranging multiple grobs on a page. |
| ... | 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 simplereg. |
| labels | character, labels for subjects. |
| subset | an optional vector specifying a subset of observations to be highlighted in the graph or subset='quant' to highligh observations with coordinates above and below the upper and lower quantiles of the variables on the x- and y-axis (Lx, Ux, Ly, Uy). |
| Lx | numeric; if subset='quant', lower quantile for the variable on the x-axis (default = 0.01). |
| Ux | numeric; if subset='quant', upper quantile for the variable on the x-axis (default = 0.99). |
| Ly | numeric; if subset='quant', lower quantile for the variable on the y-axis (default = 0.01). |
| Uy | numeric; if subset='quant', upper quantile for the variable on the y-axis (default = 0.99). |
| title | character, plot title. |
| xtitle | character, x-axis label. |
| ytitle | character, y-axis label. |
| repel | logical, if TRUE (the default) text labels repel 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 *Plots a variability diagram from a 'variability' object*

---

### 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 c(x,y) numeric vector (x and y are coordinates of the legend box; their values should be between 0 and 1; c(0,0) corresponds to the bottom-left and c(1,1) 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
)
```

## 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. |

## 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,
  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 data.frame. |
| 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). |
| 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")
```

---

# 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

| | |
|---|---|
| data | a data frame whose rows are shots and with the following columns: result, ShotType, 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, the string giving the name of the numerical variable according to which the scoring probability is estimated. Available options: "playlength", "periodTime", "totalTime", "shot_distance". |
| shot.type | character, the type of shots to be analyzed; available options: "2P", "3P", "FT", "field". |
| players | subset of players to be displayed (optional; it can be used only if the player column is present in data). |
| bw | numeric, the 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). |
| xlab | character, x-axis label. |
| x.range | numerical vector or character; available options: NULL (x-axis range defined by ggplot2, the default), "auto" (internally defined x-axis range), or a 2-component numerical vector (user-defined x-axis range). |
| title | character, plot title. |
| palette | color palette. |
| team | character; if TRUE draws the scoring probability for all the shots in data. |
| col.team | character, color of the scoring probability line for all the shots in data. |
| legend | character; if TRUE, color legend is displayed (only when players is not NULL). |

## 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

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

---

# simplereg *Simple linear and nonparametric regression*

---

## 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
Pl <- Pbox.sel$Player
mod <- simplereg(x=X, y=Y, type="lin")
```

## Tadd *Tadd dataset - NBA 2017-2018*

### 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 *Teams box scores dataset - NBA 2017-2018*

### 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.

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 variability. |
| size.var | a vector of variable names or of column numbers defining variables for weights (active only if weight=TRUE). |
| VC | logical; if TRUE, calculates variation coefficients of variables in data.var. |
| weight | logical; if TRUE, 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