Package: unitedR (via r-universe)

August 24, 2024

Title Assessment and Evaluation of Formations in United

Version 0.4

Description United is a software tool which can be downloaded at the following website <http://www.schroepl.net/pbm/software/united/>. In general, it is a virtual manager game for football teams. This package contains helpful functions for determining an optimal formation for a virtual match in United. E.g. knowing that the opponent has a strong defensive it is advisable to beat him in the midfield. Furthermore, this package contains functions for computing the optimal usage of hardness in a game.

Depends R (>= 3.1.2), methods, plyr

License GPL (>= 2)

LazyData true

Collate 'simRedCard.R' 'getLineup.R' 'formation.R' 'overtime.R' 'penaltyGoalsProb.R' 'summary.R' 'unitedRPackage.R' 'unitedRoverview.R' 'unitedSimClass.R' 'unitedSimResults.R' 'unitedSimOne.R' 'unitedSim.R'

Suggests testthat, knitr

VignetteBuilder knitr

RoxygenNote 7.1.0

NeedsCompilation no

Author David Schindler [aut, cre]

Maintainer David Schindler <dv.schindler@gmail.com>

Repository CRAN

Date/Publication 2020-06-23 15:40:02 UTC

Contents

`unitedR-package` *Assessment and Evaluation of United Formations*

Description

Assessment and Evaluation of United Formations

Details

Package:	unitedR
Type:	Package
Version:	0.4
Date:	2020-06-27
License:	GPL (>= 2)
LazyLoad:	yes

This package provides functionality for the assessment of lineups and formations in United. The rules for United in detail can be found under: United-rules.

Author(s)

David Schindler <dv.schindler@gmail.com>

References

omido, United Software, United-Forum

formation *Representing a formation*

Description

Represents a valid united formation.

Usage

```
formation(
  GK,
  SW,
  DF,
  MF,
  ST,
  hardness = c(0, 0, 0, 0, 0),
  homeAdv = c(0, 0, 0, 0, 0)
)
```

Arguments

GK	integer for the strength goalkeeper
SW	vector for the strength of the sweeper, can be NA or a numeric
DF	numeric vector for the strengths of the players in the defense
MF	numeric vector for the strengths of the players in the midfield
ST	numeric vector of integers for the strenghts of the strikers
hardness	numeric vector of length five with integers for the used hardness
homeAdv	numeric vector of length five with integers for the used hardness

Value

S4 object of the class formation.

getLineup *Lineup of a united formation*

Description

Generates a numeric vector which specifies the used united lineup

Usage

```
getLineup(obj)

## S4 method for signature 'formation'
getLineup(obj)
```

Arguments

obj	object of the class formation.

Value

vector of the used lineup

---

overtime *Computing overtime results*

Description

Computes the final overtime outcome.

Usage

```
overtime(chancesHome, chancesAway, probGoalHome, probGoalAway)
```

Arguments

chancesHome	goalscoring chances of home team
chancesAway	goalscoring chances of away team
probGoalHome	probability of scoring a goal for home team
probGoalAway	probability of scoring a goal for away team

Value

list with probabilities of final outcome.

---

overview *Overview over the parameters used in the* unitedR *package*

Description

This list of parameters yields a comprehensive overview of the parameters used in the unitedR package.

Arguments

away	away team (an object of the S4class formation)
chancesAway	goalscoring chances of away team
chancesHome	goalscoring chances of home team
DF	numeric vector for the strengths of the players in the defense
formation	object of the S4class formation
GK	integer for the strength goalkeeper
hardness	numeric vector of length five with integers for the used hardness
hardnessMatrix	matrix matrix with eleven columns which contain the probability for yellow cards dependent on the used hardness
home	home team (an object of the S4class formation)
homeAdv	numeric vector of length five with integers for the used hardness
L	list with elements of class formation
MF	numeric vector for the strengths of the players in the midfield
overtime	logical, if True overtime win probabilites are calculated. Only available if total hardness is zero or one.
penaltyGoalProb	probability of a goal by a singular penalty
penaltyProb	occurrence probability of a penalty
posPenalties	number of possible penalties in a game
preventGoalGK	factor multiplicied with the strength of the GK for computing the probability of preventing a goal by the goalkeeper
preventGoalSW	factor multiplicied with the strength of the SW for computing the probability of preventing a goal by the sweeper
probGoalAway	probability of scoring a goal for away team
probGoalHome	probability of scoring a goal for home team
probPenaltySaveAway	probability of saving a penalty for away team
probPenaltySaveHome	probability of saving a penalty for home team
r	number of replications for the simulation of hardness and penalties, can be missing (exact results will be computed)
ST	numeric vector of integers for the strenghts of the strikers
SW	vector for the strength of the sweeper, can be NA or a numeric
x	a variable x.

penaltyGoalsProb *Computing goals by united*

Description

Computes the distribution function of possible goals by penalties.

Usage

```
penaltyGoalsProb(posPenalties, penaltyGoalProb, penaltyProb = 0.1)
```

Arguments

posPenalties	number of possible penalties in a game
penaltyGoalProb	probability of a goal by a singular penalty
penaltyProb	occurrence probability of a penalty

Value

A data.frame with two columns: the possible goals and the probability for achieving this number of goals.

penaltyShootout *Computing outcome of penalty shootout*

Description

Computes outcome of a penalty shootout.

Usage

```
penaltyShootout(probPenaltySaveHome, probPenaltySaveAway, initial = 5)
```

Arguments

probPenaltySaveHome	probability of saving a penalty for home team
probPenaltySaveAway	probability of saving a penalty for away team
initial	number of initial penalties (default 5)

Value

list with probabilities of final outcome (winProbabilityHome, winProbabilityAway).

simRedCard *Simulate red card(s)*

Description

Simulates red card(s) in the united and returns the adjusted lineup.

Usage

```
simRedCard(obj, lineup, Hard)

## S4 method for signature 'formation,numeric,matrix'
simRedCard(obj, lineup, Hard)
```

Arguments

obj	object of the class formation
lineup	lineup of the corresponding object obj
Hard	matrix of hardness to be used

Value

list with two elements:

- vector adjusted lineup for the red card(s)
- numeric number of red cards

summary *Summary of assessments of a randomization procedure*

Description

Summary of assessments of a randomization procedure

Usage

```
summary(object, ...)

## S4 method for signature 'unitedSim'
summary(object)

## S4 method for signature 'unitedSimResults'
summary(object)
```

Arguments

object	object of class unitedSimResults
...	additional arguments affecting the summary that will be produced.

Value

data.frame with a summary of the assessed object.

---

unitedSim *Simulating a formation*

Description

Simulates a formation against another formations (several formations of away are possible).

Usage

```
unitedSim(
  home,
  ...,
  r,
  penaltyProb = 0.1,
  preventGoalGK = 1/14,
  preventGoalSW = 1/15,
  hardnessMatrix,
  L,
  overtime = FALSE
)
```

Arguments

home	home team (an object of the S4class formation)
...	several objects of the class formation
r	number of replications for the simulation of hardness and penalties, can be missing (exact results will be computed)
penaltyProb	occurrence probability of a penalty
preventGoalGK	factor multiplicied with the strength of the GK for computing the probability of preventing a goal by the goalkeeper
preventGoalSW	factor multiplicied with the strength of the SW for computing the probability of preventing a goal by the sweeper
hardnessMatrix	matrix matrix with eleven columns which contain the probability for yellow cards dependent on the used hardness
L	list with elements of class formation
overtime	logical, if True overtime win probabilites are calculated. Only available if total hardness is zero or one.

Value

Creates an object of the `unitedSim` class.

See Also

`unitedSimOne`

Examples

```
home <- formation(10, NA, c(7,5,3), c(8,8), c(10,10,8))
away <- formation(5, 8, c(8,8), c(10,10), c(10,10,10),
 hardness = c(0,0,0,0,1))
set.seed(123)
unitedSim(home, away)
# can also be simualated
unitedSim(home, away, r = 100)
# several away lineups
unitedSim(home, away, away)
# several away lineups simulated
unitedSim(home, away, away, r = 100)
# used hardness matrix (default)
# shows the probability of receiving a specifed number of yellow cards
# dependent on the used points of hardness
dimNams <-  list(paste(0:7, "cards"), paste(0:10, "hardness points"))
(hardnessMatrix <- matrix(c(90,10,0,0,0,0,0,0,
70,30,0,0,0,0,0,0,50,40,10,
0,0,0,0,0,30,50,20,0,0,0,0,0,20,40,30,10,0,0,
0,0,10,30,40,20,0,0,0,0,0,20,40,30,10,0,0,0,0,
10,30,40,20,0,0,0,0,0,20,40,30,10,0,0,0,0,10,20,
40,20,10,0,0,0,0,10,40,20,20,10), nrow = 8,
dimnames = dimNams))
```

`unitedSimOne` *Simulating a formation*

Description

Simulates a formation against another formation.

Usage

```
unitedSimOne(
  home,
  away,
  r,
  penaltyProb = 0.1,
  preventGoalGK = 1/14,
```

```
  preventGoalSW = 1/15,
  hardnessMatrix,
  overtime = FALSE
)
```

Arguments

home	home team (an object of the S4class formation)
away	away team (an object of the S4class formation)
r	number of replications for the simulation of hardness and penalties, can be missing (exact results will be computed)
penaltyProb	occurrence probability of a penalty
preventGoalGK	factor multiplicied with the strength of the GK for computing the probability of preventing a goal by the goalkeeper
preventGoalSW	factor multiplicied with the strength of the SW for computing the probability of preventing a goal by the sweeper
hardnessMatrix	matrix matrix with eleven columns which contain the probability for yellow cards dependent on the used hardness
overtime	logical, if True overtime win probabilites are calculated. Only available if total hardness is zero or one.

Value

Creates an object of the unitedSim class.

See Also

unitedSim

Examples

```
home <- formation(10, NA, c(7,5,3), c(8,8), c(10,10,8))
away <- formation(5, 8, c(8,8), c(10,10), c(10,10,10),
 hardness = c(0,0,0,0,1))
set.seed(123)
unitedSimOne(home, away)
# results with overtime
# Note: Only key statistics are adjusted for overtime
unitedSimOne(home, away, overtime = TRUE)
# simulating the game
unitedSimOne(home, away, r = 100)
```

Index