Package ‘unitedR’

January 5, 2018

Title Assessment and Evaluation of Formations in United

Version 0.3.1

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'
  'penaltyGoalsProb.R' 'summary.R' 'unitedRPackage.R'
  'unitedRoverview.R' 'unitedSimClass.R' 'unitedSimResults.R'
  'unitedSimOne.R' 'unitedSim.R'

Suggests testthat, knitr

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Author David Schindler [aut, cre]

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

Repository CRAN

Date/Publication 2018-01-05 12:39:45 UTC

R topics documented:

  unitedR-package .................................................. 2
  formation .......................................................... 2
  getLineup ........................................................ 3
  overview .......................................................... 4
### unitedR-package

**Assessment and Evaluation of United Formations**

**Description**

Assessment and Evaluation of United Formations

**Details**

- **Package:** unitedR
- **Type:** Package
- **Version:** 0.3.1
- **Date:** 2018-01-03
- **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))
getLineup

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 strengths 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.

Description

Generates a numeric vector which specifies the used united lineup

Usage

gLineup(obj)

## S4 method for signature 'formation'
gLineup(obj)

Arguments

- **obj**: object of the class formation.

Value

vector of the used lineup
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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>away</td>
<td>away team (an object of the S4class formation)</td>
</tr>
<tr>
<td>DF</td>
<td>numeric vector for the strengths of the players in the defense</td>
</tr>
<tr>
<td>formation</td>
<td>object of the S4class formation</td>
</tr>
<tr>
<td>GK</td>
<td>integer for the strength goalkeeper</td>
</tr>
<tr>
<td>hardness</td>
<td>numeric vector of length five with integers for the used hardness</td>
</tr>
<tr>
<td>hardnessMatrix</td>
<td>matrix matrix with eleven columns which contain the probability for yellow cards dependent on the used hardness</td>
</tr>
<tr>
<td>home</td>
<td>home team (an object of the S4class formation)</td>
</tr>
<tr>
<td>homeAdv</td>
<td>numeric vector of length five with integers for the used hardness</td>
</tr>
<tr>
<td>MF</td>
<td>numeric vector for the strengths of the players in the midfield</td>
</tr>
<tr>
<td>penaltyGoalProb</td>
<td>probability of a goal by a singular penalty</td>
</tr>
<tr>
<td>penaltyProb</td>
<td>occurrence probability of a penalty</td>
</tr>
<tr>
<td>posPenalties</td>
<td>number of possible penalties in a game</td>
</tr>
<tr>
<td>preventGoalGK</td>
<td>factor multiplicied with the strength of the GK for computing the probability of preventing a goal by the goalkeeper</td>
</tr>
<tr>
<td>preventGoalSW</td>
<td>factor multiplicied with the strength of the SW for computing the probability of preventing a goal by the sweeper</td>
</tr>
<tr>
<td>r</td>
<td>number of replications for the simulation of hardness and penalties, can be missing (exact results will be computed)</td>
</tr>
<tr>
<td>ST</td>
<td>numeric vector of integers for the strengths of the strikers</td>
</tr>
<tr>
<td>SW</td>
<td>vector for the strength of the sweeper, can be NA or a numeric</td>
</tr>
<tr>
<td>x</td>
<td>a variable x.</td>
</tr>
</tbody>
</table>
### penaltyGoalsProb

**Description**

Computes the distribution function of possible goals by penalties.

**Usage**

```r
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.

### simRedCard

**Simulate red card(s)**

**Description**

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

**Usage**

```r
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 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.

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)
Arguments

- `home`: home team (an object of the S4 class `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 multiplied with the strength of the GK for computing the probability of preventing a goal by the goalkeeper
- `preventGoalSW`: factor multiplied 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

Value

Creates an object of the `unitedSim` class.

See Also

- `unitedSimOne`

Examples

```r
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 simulated
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 specified 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(90,10,0,0,0,0,0,0,0,50,40,10,
  0,0,0,0,30,50,20,0,0,0,0,20,40,30,10,0,0,
  0,0,10,30,40,20,0,0,0,0,20,40,30,10,0,0,0,0,
  10,30,40,20,0,0,0,0,0,20,40,30,10,0,0,0,10,20,
  0,20,10,0,0,0,0,10,40,20,20,10), 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)

Arguments

- home: home team (an object of the `formation` class)
- away: away team (an object of the `formation` class)
- 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 multiplied with the strength of the GK for computing the probability of preventing a goal by the goalkeeper
- preventGoalSW: factor multiplied 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

Value

Creates an object of the unitedSim class.

See Also

unitedSim

Examples

```r
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)
# you can even simulated the game
unitedSimOne(home, away, r = 100)
```
Index

formation, 2
getLineup, 3
getLineup, formation-method (getLineup), 3

overview, 4

penaltyGoalsProb, 5

simRedCard, 5
simRedCard, formation, numeric, matrix-method (simRedCard), 5

summary, 6
summary, unitedSim-method (summary), 6
summary, unitedSimResults-method (summary), 6

unitedR (unitedR-package), 2
unitedR-package, 2
unitedSim, 6, 8
unitedSimOne, 7, 8