Package ‘baseballDBR’

October 12, 2022

Type Package
Title Sabermetrics and Advanced Baseball Statistics
Version 0.1.2
Description A tool for gathering and analyzing data from the Baseball Data-
bank <http://www.baseball-databank.org/>, which includes player performance statistics from major league baseball in the United States beginning in the year 1871.
Depends R (>= 3.3.3)
Imports rvest, xml2, magrittr, dplyr
Suggests testthat, tidyr, rmarkdown, knitr, Lahman
License MIT + file LICENSE
URL https://github.com/keberwein/moneyball
BugReports https://github.com/keberwein/moneyball/issues
LazyData true
RoxygenNote 6.0.1
VignetteBuilder knitr
NeedsCompilation no
Author Kris Eberwein [aut, cre]
Maintainer Kris Eberwein <eberwein@knights.ucf.edu>
Repository CRAN
Date/Publication 2017-06-15 15:19:17 UTC

R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>BA</td>
<td>2</td>
</tr>
<tr>
<td>BABIP</td>
<td>3</td>
</tr>
<tr>
<td>baseballDBR</td>
<td>4</td>
</tr>
<tr>
<td>BBpct</td>
<td>4</td>
</tr>
<tr>
<td>BB_9</td>
<td>5</td>
</tr>
<tr>
<td>Ch</td>
<td>5</td>
</tr>
</tbody>
</table>
### Description

Find batting average for batters with more than zero at bats. Required fields from the Batting table are; "AB", and "H."

### Usage

```r
BA(dat = NULL)
```

### Arguments

- **dat**  
  A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.
BABIP

See Also

Other Batting functions: BABIP, BBpct, CTpct, HRpct, ISO, Kpct, OBP, OPS, PA, RC2002, RCBasic, RCTech, SLG, TBs, XBHpct, XBperH, wOBA, wRAA, wRC

Examples

data("Batting2016")
head(Batting2016)

Batting2016$BABIP <- BABIP(Batting2016)
baseballDBR

baseballDBR: A package for working with data from the Baseball Databank/Lahman Database.

Description

baseballDBR: A package for working with data from the Baseball Databank/Lahman Database.

BBpct

Batting: Calculate base on ball percentage

Description

Find base on ball percentage for batters with more than zero at bats. Required fields from the Batting table are: "AB", "SO", "BB", "HBP", "SF", and "SH." Intentional base on balls (IBB) is added for the years that metric is available.

Usage

BBpct(dat = NULL)

Arguments

dat

A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

See Also

Other Batting functions: BABIP, BA, CTpct, HRpct, ISO, Kpct, OBP, OPS, PA, RC2002, RCbasic, RCtech, SLG, TBs, XBHpct, XBperH, wOBA, wRAA, wRC

Examples

data("Batting2016")
head(Batting2016)

Batting2016$BBpct <- BBpct(Batting2016)
**BB_9**  
*Pitching: Calculate walks per nine innings*

---

**Description**

Find batting average walks per nine innings for pitchers with more one or more inning pitched. Required fields from the Pitching table are; "IPouts", and "BB."

**Usage**

```r
BB_9(dat = NULL)
```

**Arguments**

- **dat**  
  A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

**See Also**

Other Pitching functions: FIP, HR_9, H_9, IP, K_9, LOB_pct, WHIP

**Examples**

```r
data("Pitching2016")
head(Pitching2016)

Pitching2016$BB_9 <- BB_9(Pitching2016)
```

---

**Ch**  
*Fielding: Calculate defensive chances*

---

**Description**

The number of chances a player had to make a defensive play. Required fields from the Fielding table are; "PO", "A", and "E."

**Usage**

```r
Ch(dat = NULL)
```

**Arguments**

- **dat**  
  A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.
See Also

Other Fielding functions: Fld_pct

Examples

```
data("Fielding2016")
head(Fielding2016)
Fielding2016$Ch <- Ch(Fielding2016)
```

---

CTpct  

Battling: Calculate a batter's contact rate

Description

Find the contact rate for batters. Required fields from the batting table are "AB" and "SO."

Usage

```
CTpct(dat = NULL)
```

Arguments

- **dat**: A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

See Also

Other Batting functions: BABIP, BA, BBpct, HRpct, ISO, Kpct, OBP, OPS, PA, RC2002, RCbasic, RCtech, SLG, TBs, XBhpct, XBperH, wOBA, wRAA, wRC

Examples

```
data("Batting2016")
head(Batting2016)
```

```
Batting2016$CTpct <- CTpct(Batting2016)
```
FIP

Pitching: Fielding Independent Pitching (FIP)

Description
Find the FIP for all pitchers with one or strike outs in a particular season. Required fields from the Pitching table are "BB", "HBP", "SO", and "IPouts."

Usage
FIP(dat = NULL, Fangraphs = FALSE, NA_to_zero = TRUE, Sep.Leagues = FALSE)

Arguments
- dat: A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.
- Fangraphs: If TRUE the function will download wOBA values from Fangraphs. If FALSE the function will use the internal formula adapted from Tom Tango’s original wOBA formula. Note, the internal formula is typically identical to Fangraphs and does not require an external download. If not specified, the default is set to FALSE.
- NA_to_zero: If TRUE this will replace NAs with 0 for years that certain stats were not counted. For example, sacrifice hits were not a counted statistic until 1954, therefore we are technically unable to calculate wOBA for any player prior to 1954. The default is set to TRUE. Even though this is bad practice mathematically, many in the sabermetrics community accept the practice. If FALSE, the wOBA calculation will return NaN for years with missing data.
- Sep.Leagues: If TRUE the algorithm will calculate different run environments for the National and American leagues. Grouping the leagues can solve problems introduced by the designated hitter and hitting pitchers. It also serves to further isolate for park factors between the American and National leagues. The default for this argument is FALSE.

See Also
Other Pitching functions: BB_9, HR_9, H_9, IP, K_9, LOB_pct, WHIP

Examples
```r
data("Pitching2016")
head(Pitching2016)
Pitching2016$FIP <- FIP(Pitching2016, Fangraphs=FALSE, NA_to_zero=TRUE, Sep.Leagues=FALSE)
```
fip_values  
*Return FIP constants per season*

**Description**

Get fip constants for each season. By default the function uses a method adapted from Tom Tango and used by Fangraphs. The function returns FIP constants based on ERA FIP_ERA as well as constants based on RA FIP_RA. Both the Tango and Fangraphs formulas use ERA for their FIP constants.

**Usage**

```r
fip_values(dat = NULL, Sep.Leagues = FALSE, Fangraphs = FALSE)
```

**Arguments**

- **dat**  
  A full pitching table from the Lahman package or the Chadwick Bureau GitHub repository. Any subsetting or removal of players will affect your results. All players for each year are recommended.

- **Sep.Leagues**  
  If TRUE, this will split the calculation and return unique FIP constants for the various leagues. This can be helpful in handling Designated Hitters and National League pitchers. It also isolates the park factors to their respective leagues.

- **Fangraphs**  
  If TRUE the function will return the Fangraphs FIP constants. This cannot be used in conjunction with the Sep.Leagues argument because Fangraphs does not separate FIP constants by league.

**Examples**

```r
data("Pitching2016")
head(Pitching2016)

fip_df <- fip_values(Pitching2016, Fangraphs=FALSE)
head(fip_df)
```

**Fld_pct**  
*Fielding: Calculate batting average*

**Description**

Find batting average for batters with more than zero at bats. Required fields from the Fielding table are: "PO", "A", and "E."
get_bbdb

Usage

Fld_pct(dat = NULL)

Arguments

dat A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

See Also

Other Fielding functions: Ch

Examples

data("Fielding2016")
head(Fielding2016)

Fielding2016$Fld_pct <- Fld_pct(Fielding2016)

get_bbdb

Get an up to date copy of the Baseball Databank.

Description

Download the newest version of the Baseball Databank from the Chadwick Bureau GitHub repository. This is the source of Sean Lahman’s baseball database and is always under development. This function will read the .csv files and return them as data frames. There is also an option to download the entire directory.

Usage

get_bbdb(table = NULL, downloadZip = FALSE, AllTables = FALSE)

Arguments

table The tables you would like to download. Uses Lahman table names Ex. "Batting", "Master", "AllstarFull", etc... If this argument is left as NULL, the function will download all twenty-seven tables.
downloadZip If true, this will download a zip file of all twenty-seven tables in .csv format to your working directory.
AllTables If true, this will download all the tables in the database. The default is set to false.
Examples

get_bbdb(table = "Batting")

## Not run:
get_bbdb(table = c("Batting", "Pitching"))

## End(Not run)

## Not run:
get_bbdb(downloadZip = TRUE)

## End(Not run)

HRpct

Batting: Calculate home run percentage

Description

Find home run percentage for batters with more than zero at bats. Required fields from the Batting table are "AB" and "HR."

Usage

HRpct(dat = NULL)

Arguments

dat A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

See Also

Other Batting functions: BABIP, BA, BBpct, CTpct, ISO, Kpct, OBP, OPS, PA, RC2002, RCBasic, RCtech, SLG, TBs, XBHpct, XBperH, wOBA, wRAA, wRC

Examples

data("Batting2016")
head(Batting2016)

Batting2016$HRpct <- HRpct(Batting2016)
**HR_9**

*Pitching: Calculate Home Runs per Nine innings*

**Description**
Find the number of home runs a pitcher allows per nine innings pitched. Required fields from the Pitching table are; "H" and "IPouts."

**Usage**

```r
HR_9(dat = NULL)
```

**Arguments**

- `dat` A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

**See Also**
Other Pitching functions: **BB_9, FIP, H_9, IP, K_9, LOB_pct, WHIP**

**Examples**

```r
data("Pitching2016")
head(Pitching2016)
Pitching2016$HR_9 <- HR_9(Pitching2016)
```

---

**H_9**

*Pitching: Calculate Hits per Nine innings*

**Description**
Find the number of hits a pitcher throws per nine innings pitched. Required fields from the Pitching table are; "H", "BB", and "IPouts."

**Usage**

```r
H_9(dat = NULL)
```

**Arguments**

- `dat` A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.
See Also

Other Pitching functions: BB_9, FIP, HR_9, IP, K_9, LOB_pct, WHIP

Examples

data("Pitching2016")
head(Pitching2016)

Pitching2016$H_9 <- H_9(Pitching2016)

---

IP

Pitching: Calculate the innings pitched

Description

Find the number of innings a player has pitched for a season. Required fields from the Pitching table are: "IPouts."

Usage

IP(dat = NULL)

Arguments

dat A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

See Also

Other Pitching functions: BB_9, FIP, HR_9, H_9, K_9, LOB_pct, WHIP

Examples

data("Pitching2016")
head(Pitching2016)

Pitching2016$IP <- IP(Pitching2016)
ISO

**Batting: Calculate ISO for batters**

**Description**

Find isolated power (ISO) for batters with more than zero at bats. Required fields from the batting table are "H", "X2B", "X3B", "HR".

**Usage**

ISO(dat = NULL)

**Arguments**

- **dat**
  A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

**See Also**

Other Batting functions: BABIP, BA, BBpct, CTpct, HRPct, Kpct, OBP, OPS, PA, RC2002, RCbasic, RCtech, SLG, TBs, XBHptct, XBperH, wOBA, wRAA, wRC

**Examples**

```r
data("Batting2016")
head(Batting2016)

Batting2016$ISO <- ISO(Batting2016)
```

Kpct

**Batting: Calculate strikeout percentage**

**Description**

Find strikeout percentage for batters with more than zero at bats. Required fields from the Batting table are; "AB", "SO", "BB", "HBP", "SF", and "SH".

**Usage**

Kpct(dat = NULL)
Arguments

dat  A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

See Also

Other Batting functions: BABIP, BA, BBpct, CTpct, HRpct, ISO, OBP, OPS, PA, RC2002, RCbasic, RCtech, SLG, TBs, XBHpct, XBperH, wOBA, wRAA, wRC

Examples

data("Batting2016")
head(Batting2016)

Batting2016$Kpct <- Kpct(Batting2016)


---

K_9  Pitching: Calculate Strikes per Nine innings

Description

Find the number of strikes a pitcher throws per nine innings pitched. Required fields from the Pitching table are; "H", "BB", "IPouts", and "SO."

Usage

K_9(dat = NULL)

Arguments

dat  A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

See Also

Other Pitching functions: BB_9, FIP, HR_9, H_9, IP, LOB_pct, WHIP

Examples

data("Pitching2016")
head(Pitching2016)

Pitching2016$K_9 <- K_9(Pitching2016)
LOB_pct

*Pitching: Calculate the left on base percentage*

**Description**
Find the percentage of base runners that a pitcher leaves on base of the course of a season. Required fields from the Pitching table are; "H", "BB", "HBP", "R", and "HR."

**Usage**
LOB_pct(dat = NULL)

**Arguments**
- **dat**
  A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

**See Also**
Other Pitching functions: BB_9, FIP, HR_9, H_9, IP, K_9, WHIP

**Examples**
```r
data("Pitching2016")
head(Pitching2016)

Pitching2016$LOB_pct <- LOB_pct(Pitching2016)
```

OBP

*Batting: Calculate on base percentage (OBP)*

**Description**
Find the OBP for batters with more than zero hits. Required fields from the batting table are "H", "X2B", "X3B", "HR."

**Usage**
OBP(dat = NULL)

**Arguments**
- **dat**
  A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.
See Also

Other Batting functions: BABIP, BA, BBpct, CTpct, HRpct, ISO, Kpct, OPS, PA, RC2002, RCBasic, RCtech, SLG, TBs, XBhpct, XBperH, wOBA, wRAA, wRC

Examples

```r
data("Batting2016")
head(Batting2016)

Batting2016$OPS <- OPS(Batting2016)
```

---

**OPS**

*Batting: Calculate on base percentage plus slugging (OPS)*

Description

Find the OPS for batters with more than zero hits. Required fields from the batting table are "H", "X2B", "X3B", "HR", "BB", "HBP", "AB" and "SF."

Usage

```r
OPS(dat = NULL)
```

Arguments

dat

A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

See Also

Other Batting functions: BABIP, BA, BBpct, CTpct, HRpct, ISO, Kpct, OBP, PA, RC2002, RCBasic, RCtech, SLG, TBs, XBhpct, XBperH, wOBA, wRAA, wRC

Examples

```r
data("Batting2016")
head(Batting2016)

Batting2016$OPS <- OPS(Batting2016)
```
**PA**  
*Batting: Calculate plate appearances for batters*

**Description**

Find the plate appearances (PA) for batters. Required fields from the batting table are "AB", "BB", "HBP", "SH", and "SF."

**Usage**

```r
PA(dat = NULL)
```

**Arguments**

- `dat`: A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

**See Also**

Other Batting functions: BABIP, BA, BBpct, CTpct, HRPct, ISO, Kpct, OBP, OPS, RC2002, RCbasic, RCtech, SLG, TBs, XBPct, XBperH, wOBA, wRAA, wRC

**Examples**

```r
data("Batting2016")
head(Batting2016)

Batting2016$PA <- PA(Batting2016)
```

---

**RC2002**  
*Batting: Calculate Runs Created using the updated 2002 formula.*

**Description**

The "2002 Version" is an updated version of the "Technical Version" by Bill James. The 2002 RC uses the same counting stats as the Technical Version but applies weights to many of the raw stats. Required fields from the batting table are "AB", "H", "BB", "X2B", "X3B", "HR", "GIDP", "HBP", "SB", "CS", "SF" and "SH," "SO", and "IBB."

**Usage**

```r
RC2002(dat = NULL)
```
Arguments

dat  A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

See Also

Other Batting functions: BABIP, BA, BBpct, CTpct, HRpct, ISO, Kpct, OBP, OPS, PA, RCbasic, RCtech, SLG, TBs, XBHpct, XBperH, wOBA, wRAA, wRC

Examples

data("Batting2016")
head(Batting2016)

Batting2016$RC2002 <- RC2002(Batting2016)

RCbasic  Batting: Calculate Runs Created using the basic formula.

Description

Find the runs created using the basic formula presented by Bill James in the late 1970s. Required fields from the batting table are "AB", "H", "BB", "X2B", "X3B", and "HR."

Usage

RCbasic(dat = NULL)

Arguments

dat  A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

See Also

Other Batting functions: BABIP, BA, BBpct, CTpct, HRpct, ISO, Kpct, OBP, OPS, PA, RC2002, RCtech, SLG, TBs, XBHpct, XBperH, wOBA, wRAA, wRC

Examples

data("Batting2016")
head(Batting2016)

Batting2016$RCbasic <- RCbasic(Batting2016)
**Batting: Calculate Runs Created using the technical formula.**

**Description**

The "Technical Version" is the most well-known formula for RC. It adds several factors to the basic formula such as sacrifice hits, stolen bases and intentional base on balls. Required fields from the batting table are "AB", "H", "BB", "X2B", "X3B", "HR", "GIDP", "HBP", "SB", "CS", "SF" and "SH," and "IBB."

**Usage**

```r
RCtech(dat = NULL)
```

**Arguments**

- `dat`: A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

**See Also**

Other Batting functions: BABIP, BA, BBpct, CTpct, HRpct, ISO, Kpct, OBP, OPS, PA, RC2002, RCBasic, SLG, TBs, XBPct, XBperH, wOBA, wRAA, wRC

**Examples**

```r
data("Batting2016")
head(Batting2016)

Batting2016$RCtech <- RCtech(Batting2016)
```

---

**Batting: Calculate slugging percentage (SLG)**

**Description**

Find the SLG for batters with more than zero hits. Required fields from the batting table are "H", "X2B", "X3B", "HR".

**Usage**

```r
SLG(dat = NULL)
```
Arguments

dat  A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

See Also

Other Batting functions: BABIP, BA, BBpct, CTPct, HRPct, ISO, Kpct, OBP, OPS, PA, RC2002, RCBasic, RCtech, TBs, XBHpct, XBperH, wOBA, wRAA, wRC

Examples

data("Batting2016")
head(Batting2016)

Batting2016$SLG <- SLG(Batting2016)

TBs

Batting: Calculate a batter's total bases

Description

Find total bases. Required fields from the batting table are "AB","H", "X2B", "X3B" and "HR."

Usage

TBs(dat = NULL)

Arguments

dat  A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

See Also

Other Batting functions: BABIP, BA, BBpct, CTPct, HRPct, ISO, Kpct, OBP, OPS, PA, RC2002, RCBasic, RCtech, SLG, XBHpct, XBperH, wOBA, wRAA, wRC

Examples

data("Batting2016")
head(Batting2016)

Batting2016$TBs <- TBs(Batting2016)
**urlExists**

Description

A utility function to run a tryCatch on a URL.

Usage

`urlExists(target)`

Arguments

target  url

---

**WHIP**

*Pitching: Calculate Walks plus Hits per Innings Pitched*

Description

Find the number of walks plus hits a pitcher allows per inning pitched. Required fields from the Pitching table are; "H", "BB", and "IPouts."

Usage

`WHIP(dat = NULL)`

Arguments

dat  A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

See Also

Other Pitching functions: `BB_9`, `FIP`, `HR_9`, `H_9`, `IP`, `K_9`, `LOB_pct`

Examples

```r

data("Pitching2016")
head(Pitching2016)

Pitching2016$WHIP <- WHIP(Pitching2016)
```
Description

Find the wOBA for all players with one or more hits for a particular season. Required fields from the batting table are "AB", "H", "BB", "X2B", "X3B", "HR", "HBP", "SF", "IBB."

Usage

wOBA(BattingTable = NULL, PitchingTable = NULL, FieldingTable = NULL, Fangraphs = FALSE, NA_to_zero = TRUE, Sep.Leagues = FALSE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BattingTable</td>
<td>A full batting table from the Lahman package or the Chadwick Bureau GitHub repository. Any subsetting or removal of players will affect your results. All players for each year are recommended.</td>
</tr>
<tr>
<td>PitchingTable</td>
<td>A full pitching table from the Lahman package or the Chadwick Bureau GitHub repository. Any subsetting or removal of players will affect your results. All players for each year are recommended.</td>
</tr>
<tr>
<td>FieldingTable</td>
<td>A full batting table from the Lahman package or the Chadwick Bureau GitHub repository. Any subsetting or removal of players will affect your results. All players for each year are recommended.</td>
</tr>
<tr>
<td>Fangraphs</td>
<td>If TRUE the function will download wOBA values from Fangraphs. If FALSE the function will use the internal formula adapted from Tom Tango's original wOBA formula. Note, the internal formula is typically identical to Fangraphs and does not require an external download. If not specified, the default is set to FALSE.</td>
</tr>
<tr>
<td>NA_to_zero</td>
<td>If TRUE this will replace NAs with 0 for years that certain stats were not counted. For example, sacrifice hits were not a counted statistic until 1954, therefore we are technically unable to calculate wOBA for any player prior to 1954. The default is set to TRUE. Even though this is bad practice mathematically, many in the sabermetrics community accept the practice. If FALSE, the wOBA calculation will return NaN for years with missing data.</td>
</tr>
<tr>
<td>Sep.Leagues</td>
<td>If TRUE the algorithm will calculate different run environments for the National and American leagues. Grouping the leagues can solve problems introduced by the designated hitter and hitting pitchers. It also serves to further isolate for park factors between the American and National leagues. The default for this argument is FALSE.</td>
</tr>
</tbody>
</table>

See Also

Other Batting functions: BABIP, BA, BBpct, CTpct, HRpct, ISO, Kpct, OBP, OPS, PA, RC2002, RCbasic, RCtech, SLG, TBs, XBHpct, X8perH, wRAA, wRC
wOBA_values

Examples

```r
data("Batting2016")
head(Batting2016)
data("Pitching2016")
head(Pitching2016)
data("Fielding2016")
head(Fielding2016)

Batting2016$wOBA <- wOBA(Batting2016, Pitching2016, Fielding2016, Fangraphs=FALSE,
                           NA_to_zero=TRUE, Sep.Leagues=FALSE)
```

wOBA_values  Return wOBA values per season

Description

Get wOBA values for each year in your database. This calculation requires all fields of the Pitching, Fielding and Batting tables from the Lahman package, or a comparable data set. The function uses a version of Tom Tango’s wOBA formula by default, but can also return Fangraphs wOBA values.

Usage

```r
wOBA_values(BattingTable, PitchingTable, FieldingTable, Sep.Leagues = FALSE,
             Fangraphs = FALSE)
```

Arguments

- **BattingTable**: A full batting table from the Lahman package or the Chadwick Bureau GitHub repository. Any subsetting or removal of players will affect your results. All players for each year are recommended.
- **PitchingTable**: A full pitching table from the Lahman package or the Chadwick Bureau GitHub repository. Any subsetting or removal of players will affect your results. All players for each year are recommended.
- **FieldingTable**: A full batting table from the Lahman package or the Chadwick Bureau GitHub repository. Any subsetting or removal of players will affect your results. All players for each year are recommended.
- **Sep.Leagues**: If TRUE, this will split the calculation and return unique wOBA values for the various leagues. This can be helpful in handling Designated Hitters and National League pitchers. It also isolates the park factors to their respective leagues.
- **Fangraphs**: if TRUE the function will return the Fangraphs wOBA values. By default the function uses a method adapted from Tom Tango. These values are often very close to Fangraphs, but are not the same due to Fangraphs using a different algorithm. This can not be used in conjunction with the Sep.Leagues argument because Fangraphs does not separate FIP constants by league.
Examples

```r
data("Batting2016")
head(Batting2016)
data("Pitching2016")
head(Pitching2016)
data("Fielding2016")
head(Fielding2016)

woba_df <- wOBA_values(Batting2016, Pitching2016, Fielding2016, Sep.Leagues=FALSE, Fangraphs=FALSE)
```

---

**wRAA**

*Batting: Calculate Weighted Runs Above Average (wRAA)*

---

**Description**

Find the wRAA for all players with one or more hits for a particular season. Required fields from the batting table are "AB", "H", "BB", "X2B", "X3B", "HR", "HBP", "SF", "IBB."

**Usage**

```r
wRAA(BattingTable = NULL, PitchingTable = NULL, FieldingTable = NULL, Fangraphs = FALSE, NA_to_zero = TRUE, Sep.Leagues = FALSE)
```

**Arguments**

- **BattingTable**
  A full batting table from the Lahman package or the Chadwick Bureau GitHub repository. Any subsetting or removal of players will affect your results. All players for each year are recommended.

- **PitchingTable**
  A full pitching table from the Lahman package or the Chadwick Bureau GitHub repository. Any subsetting or removal of players will affect your results. All players for each year are recommended.

- **FieldingTable**
  A full batting table from the Lahman package or the Chadwick Bureau GitHub repository. Any subsetting or removal of players will affect your results. All players for each year are recommended.

- **Fangraphs**
  If TRUE the function will download wOBA values from Fangraphs. Both wOBA scale and league wOBA are used in the wRAA calculation. If FALSE the function will use the internal wOBA algorithm, which is adapted from Tom Tango’s original wOBA formula. This algorithm produces a slightly different wOBA scale than the Fangraphs wOBA scale, so variations in wRAA should be expected. The default internal method does not require an external download from Fangraphs. If not specified, the default is set to FALSE.

- **NA_to_zero**
  If TRUE this will replace NAs with 0 for years that certain stats were not counted. For example, sacrifice hits were not a counted statistic until 1954, therefore we are technically unable to calculate wRAA for any player prior to
The default is set to TRUE. Even though this is bad practice mathematically, many in the sabermetrics community accept the practice. If FALSE, the wRAA calculation will return NaN for years with missing data.

Sep.Leagues
If TRUE the algorithm will calculate different run environments for the National and American leagues. Grouping the leagues can solve problems introduced by the designated hitter and hitting pitchers. It also serves to further isolate for park factors between the American and National leagues. The default for this argument is FALSE.

See Also
Other Batting functions: BABIP, BA, BBpct, CTpct, HRpct, ISO, Kpct, OBP, OPS, PA, RC2002, RCbasic, RCtech, SLG, TBs, XBHpct, XBperH, wOBA, wRC

Examples

```r
data("Batting2016")
head(Batting2016)
data("Pitching2016")
head(Pitching2016)
data("Fielding2016")
head(Fielding2016)

Batting2016$wRAA <- wRAA(Batting2016, Pitching2016, Fielding2016, Fangraphs=FALSE,
                           NA_to_zero=TRUE, Sep.Leagues=FALSE)
```

---

**wRC**

**Batting: Calculate Weighted Runs Created (wRC)**

**Description**

Find the wRC for all players with one or more hits for a particular season. Required fields from the batting table are "AB", "H", "BB", "X2B", "X3B", "HR", "HBP", "SF", "IBB."

**Usage**

```r
wRC(BattingTable = NULL, PitchingTable = NULL, FieldingTable = NULL,
   Fangraphs = FALSE, NA_to_zero = TRUE, Sep.Leagues = FALSE)
```

**Arguments**

- **BattingTable**: A full batting table from the Lahman package or the Chadwick Bureau GitHub repository. Any subsetting or removal of players will affect your results. All players for each year are recommended.
- **PitchingTable**: A full pitching table from the Lahman package or the Chadwick Bureau GitHub repository. Any subsetting or removal of players will affect your results. All players for each year are recommended.
FieldingTable

A full fielding table from the Lahman package or the Chadwick Bureau GitHub repository. Any subsetting or removal of players will affect your results. All players for each year are recommended.

Fangraphs

If TRUE the function will download wOBA values from Fangraphs. Both wOBA scale and league wOBA are used in the wRC calculation. If FALSE the function will use the internal wOBA algorithm, which is adapted from Tom Tango’s original wOBA formula. This algorithm produces a slightly different wOBA scale than the Fangraphs wOBA scale, so variations in wRC should be expected. The default internal method does not require an external download from Fangraphs. If not specified, the default is set to FALSE.

NA_to_zero

If TRUE this will replace NAs with 0 for years that certain stats were not counted. For example, sacrifice hits were not a counted statistic until 1954, therefore we are technically unable to calculate wRC for any player prior to 1954. The default is set to TRUE. Even though this is bad practice mathematically, many in the sabermetrics community accept the practice. If FALSE, the wRC calculation will return NaN for years with missing data.

Sep.Leagues

If TRUE the algorithm will calculate different run environments for the National and American leagues. Grouping the leagues can solve problems introduced by the designated hitter and hitting pitchers. It also serves to further isolate for park factors between the American and National leagues. The default for this argument is FALSE.

See Also

Other Batting functions: BABIP, BA, BBpct, CTpct, HRpct, ISO, Kpct, OBP, OPS, PA, RC2002, RCbasic, RCTech, SLG, TBS, XBPct, XBperH, wOBA, wRAA

Examples

data("Batting2016")
head(Batting2016)
data("Pitching2016")
head(Pitching2016)
data("Fielding2016")
head(Fielding2016)


XBHpct

*Batting: Calculate extra base percentage*

Description

Find extra base percentage for batters with more than zero at bats. Required fields from the batting table are "AB", "BB", "HBP", "SF", "SH", "X2B", "X3B", "HR".
**Usage**

```
XBHpct(dat = NULL)
```

**Arguments**

- `dat`: A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

**See Also**

Other Batting functions: `BABIP`, `BA`, `BBpct`, `CTpct`, `HRpct`, `ISO`, `Kpct`, `OBP`, `OPS`, `PA`, `RC2002`, `RCbasic`, `RCtech`, `SLG`, `TBs`, `XBperH`, `wOBA`, `wRAA`, `wRC`

**Examples**

```r
data("Batting2016")
head(Batting2016)

Batting2016$XBHpct <- XBHpct(Batting2016)
```

---

**Description**

Find the average extra bases per hit for batters with more than zero hits. Required fields from the batting table are "H", "X2B", "X3B", "HR".

**Usage**

```
XBperH(dat = NULL)
```

**Arguments**

- `dat`: A data frame you would wish to calculate. The data frame must have the same column names found in The Lahman package or the Chadwick Bureau GitHub repository.

**See Also**

Other Batting functions: `BABIP`, `BA`, `BBpct`, `CTpct`, `HRpct`, `ISO`, `Kpct`, `OBP`, `OPS`, `PA`, `RC2002`, `RCbasic`, `RCtech`, `SLG`, `TBs`, `XBperH`, `wOBA`, `wRAA`, `wRC`
Examples

data("Batting2016")
head(Batting2016)

Batting2016$XBperH <- XBperH(Batting2016)
Index

* Above
  wRAA, 24
  wRC, 25
* Average
  wOBA, 22
  wRAA, 24
  wRC, 25
* BABIP
  BABIP, 3
* BA
  BA, 2
* BB9
  BB_9, 5
* BB_9
  BB_9, 5
* BBpct
  BBpct, 4
* BB
  BB_9, 5
* CTpct
  CTpct, 6
* Chances
  Ch, 5
* Ch
  Ch, 5
* Defensive
  Ch, 5
* FIP
  FIP, 7
* Fld_pct
  Fld_pct, 8
* HRpct
  HRpct, 10
* Hits
  WHIP, 21
* ISO
  ISO, 13
* Innings
  WHIP, 21
* Kpct
  Kpct, 13
* LOB_pct
  LOB_pct, 15
* LOB
  LOB_pct, 15
* OBP
  OBP, 15
* OPS
  OPS, 16
* On-Base
  wOBA, 22
* PA
  PA, 17
* Pitched
  WHIP, 21
* RC2002
  RC2002, 17
* RChasic
  RChasic, 18
* RCtech
  RCtech, 19
* Runs
  wRAA, 24
  wRC, 25
* SLG
  SLG, 19
* TBs
  TBs, 20
* WHIP
  WHIP, 21
* Walks
  WHIP, 21
* Weighted
  wOBA, 22
  wRAA, 24
  wRC, 25
* XBHpct
  XBHpct, 26
INDEX

* XBperH
  XBperH, 27
* average,
  fip_values, 8
  wOBA_values, 23
* ball
  BA, 2
  BABIP, 3
  BBpct, 4
* bases
  TBs, 20
* base
  BA, 2
  BABIP, 3
  BBpct, 4
  fip_values, 8
  OBP, 15
  OPS, 16
  PA, 17
  RC2002, 17
  RCBasic, 18
  RCtech, 19
  SLG, 19
  wOBA_values, 23
  XBHpct, 26
  XBperH, 27
* bb/9
  BB_9, 5
* bb
  BA, 2
  BABIP, 3
  BBpct, 4
* contact
  CTpct, 6
* database,
  get_bbdb, 9
* data
  get_bbdb, 9
* extra
  RC2002, 17
  RCBasic, 18
  RCtech, 19
  XBHpct, 26
  XBperH, 27
* fangraphs
  fip_values, 8
  wOBA_values, 23
* fielding
  FIP, 7
  Fld_pct, 8
* frame
  get_bbdb, 9
* hits
  H_9, 11
  HR_9, 11
* hit
  RC2002, 17
  RCBasic, 18
  RCtech, 19
  XBperH, 27
* home
  HRpct, 10
* independent
  FIP, 7
* innings
  H_9, 11
  HR_9, 11
  IP, 12
  K_9, 14
* isolated
  ISO, 13
* nine
  H_9, 11
  HR_9, 11
  K_9, 14
* on
  BA, 2
  BABIP, 3
  BBpct, 4
  fip_values, 8
  OBP, 15
  OPS, 16
  PA, 17
  SLG, 19
  wOBA_values, 23
* percentage
  BA, 2
  BABIP, 3
  BBpct, 4
  Fld_pct, 8
  HRpct, 10
  Kpct, 13
  LOB_pct, 15
  OBP, 15
  OPS, 16
  PA, 17