Package ‘Lahman’

May 2, 2019

Type Package
Title Sean 'Lahman' Baseball Database
Version 7.0-1
Date 2019-05-01
Author Michael Friendly [aut],
      Chris Dalzell [cre, aut],
      Martin Monkman [aut],
      Dennis Murphy [aut],
      Vanessa Foot [ctb],
      Justeena Zaki-Azat [ctb]
Maintainer Chris Dalzell <cdalzell@gmail.com>
Description Provides the tables from the 'Sean Lahman Baseball Database' as a set of R data.frames. It uses the data on pitching, hitting and fielding performance and other tables from 1871 through 2018, as recorded in the 2019 version of the database. Documentation examples show how many baseball questions can be investigated.
Depends R (>= 2.10)
Suggests lattice, ggplot2, googleVis, data.table, vcd, reshape2,
tidyR, zipcode, knitr, rmarkdown, car
Imports dplyr
License GPL
URL http://lahman.r-forge.r-project.org/
LazyLoad yes
LazyData yes
BugReports https://github.com/cdalzell/Lahman/issues
Repository CRAN
Repository/R-Forge/Project lahan
Repository/R-Forge/Revision 39
Repository/R-Forge/DateTimeStamp 2013-06-01 03:33:30
Date/Publication 2019-05-02 09:30:03 UTC
R topics documented:

NeedsCompilation: no
RoxygenNote: 6.1.1
VignetteBuilder: knitr

R topics documented:

Lahman-package .................................................. 3
AllstarFull ...................................................... 5
Appearances ..................................................... 6
AwardsManagers .................................................. 8
AwardsPlayers ................................................... 10
AwardsShareManagers .......................................... 11
AwardsSharePlayers ............................................ 12
Batting ........................................................... 14
battingLabels ................................................... 18
BattingPost ...................................................... 19
battingStats ..................................................... 21
CollegePlaying ................................................... 22
Fielding ........................................................ 23
FieldingOF ....................................................... 25
FieldingPost ..................................................... 27
HallOfFame ....................................................... 28
Label ............................................................. 32
LahmanData ....................................................... 33
Managers ........................................................ 34
ManagersHalf .................................................... 38
Master .......................................................... 39
Parks .............................................................. 41
People ........................................................... 42
Pitching .......................................................... 44
PitchingPost ..................................................... 47
playerInfo ....................................................... 49
Salaries .......................................................... 50
Schools .......................................................... 53
SeriesPost ........................................................ 54
Teams ............................................................ 56
TeamsFranchises ............................................... 61
TeamsHalf ....................................................... 62

Index 64
Description

This database contains pitching, hitting, and fielding statistics for Major League Baseball from 1871 through 2018. It includes data from the two current leagues (American and National), the four other "major" leagues (American Association, Union Association, Players League, and Federal League), and the National Association of 1871-1875.

This database was created by Sean Lahman, who pioneered the effort to make baseball statistics freely available to the general public. What started as a one man effort in 1994 has grown tremendously, and now a team of researchers have collected their efforts to make this the largest and most accurate source for baseball statistics available anywhere.

This database, in the form of an R package offers a variety of interesting challenges and opportunities for data processing and visualization in R.

In the current version, the examples make extensive use of the dplyr package for data manipulation (tabulation, queries, summaries, merging, etc.), reflecting the original relational database design and ggplot2 for graphics.

Details

Package: Lahman
Type: Package
Version: 7.0-1
Date: 2019-05-01
License: GPL version 2 or newer
LazyLoad: yes
LazyData: yes

The main form of this database is a relational database in Microsoft Access format. The design follows these general principles: Each player is assigned a unique code (playerID). All of the information in different tables relating to that player is tagged with his playerID. The playerIDs are linked to names and birthdates in the People table. Similar links exist among other tables via analogous *ID variables.

The database is composed of the following main tables:

People Player names, dates of birth, death and other biographical info
Batting batting statistics
Pitching pitching statistics
Fielding fielding statistics

A collection of other tables is also provided: Teams:
Teams yearly stats and standings
TeamsHalf split season data for teams
TeamsFranchises franchise information

Post-season play:
BattingPost post-season batting statistics
PitchingPost post-season pitching statistics
FieldingPost post-season fielding data
SeriesPost post-season series information

Awards:
AwardsManagers awards won by managers
AwardsPlayers awards won by players
AwardsShareManagers award voting for manager awards
AwardsSharePlayers award voting for player awards

Hall of Fame: links to People via hofID
HallOfFame Hall of Fame voting data

Other tables:
AllstarFull - All-Star games appearances; Managers - managerial statistics; FieldingOF - outfield position data; ManagersHalf - split season data for managers; Salaries - player salary data; Appearances - data on player appearances; Schools - Information on schools players attended; CollegePlaying - Information on schools players attended, by player and year;

Variable label tables are provided for some of the tables:
batingLabels, pitchingLabels, fieldingLabels

Author(s)
Michael Friendly, Dennis Murphy, Chris Dalzell, Martin Monkman
Maintainer: Chris Dalzell <cdalzell@gmail.com>

Source
**AllstarFull**  

**AllstarFull table**

**Description**

All Star appearances by players

**Usage**

data(AllstarFull)

**Format**

A data frame with 5291 observations on the following 8 variables.

- **playerID**: Player ID code
- **yearID**: Year
- **gameNum**: Game number (for years in which more than one game was played)
- **gameID**: Game ID code
- **teamID**: Team; a factor
- **lgID**: League; a factor with levels AL NL
- **GP**: Game played (zero if player did not appear in game)
- **startingPos**: If the player started, what position he played

**Source**


**Examples**

data(AllstarFull)

# find number of appearances by players in the All Star games  
player_appearances <- with(AllstarFull, rev(sort(table(playerID))))

# How many All-Star players, in total?  
length(player_appearances)

# density plot of the whole distribution  
plot(density(player_appearances), main="Player appearances in All Star Games")  
rug(jitter(player_appearances))

# who has played in more than 10 ASGs?  
player_appearances[player_appearances > 10]  
hist(player_appearances[player_appearances > 10])
# Hank Aaron's All-Star record:
subset(AllstarFull, playerID == "aaronha01")

# Years that Stan Musial played in the ASG:
with(AllstarFull, yearID[playerID == "musiast01"])

# Starting positions he played (NA means did not start)
with(AllstarFull, startPos[playerID == "musiast01"])

# All-Star rosters from the 1966 ASG
subset(AllstarFull, gameID == "NLS196607120")

# All-Stars from the Washington Nationals
subset(AllstarFull, teamID == "WAS")

# Teams with the fewest All-Stars
rare <- names(which(table(AllstarFull$teamID) < 10))

# Records associated with the 'rare' teams:
# (There are a few teamID typos: can you spot them?)
subset(AllstarFull, teamID %in% rare)

---

**Appearances**

**Appearances table**

<table>
<thead>
<tr>
<th>Description</th>
<th>Data on player appearances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usage</td>
<td>data(Appearances)</td>
</tr>
<tr>
<td>Format</td>
<td>A data frame with 105789 observations on the following 21 variables.</td>
</tr>
<tr>
<td></td>
<td>yearID Year</td>
</tr>
<tr>
<td></td>
<td>teamID Team; a factor</td>
</tr>
<tr>
<td></td>
<td>lgID League; a factor with levels AA AL FL NL PL UA</td>
</tr>
<tr>
<td></td>
<td>playerId Player ID code</td>
</tr>
<tr>
<td></td>
<td>G_all Total games played</td>
</tr>
<tr>
<td></td>
<td>GS Games started</td>
</tr>
<tr>
<td></td>
<td>G_batting Games in which player batted</td>
</tr>
<tr>
<td></td>
<td>G_defense Games in which player appeared on defense</td>
</tr>
<tr>
<td></td>
<td>G_p Games as pitcher</td>
</tr>
</tbody>
</table>
Appearances

- G_c: Games as catcher
- G_1b: Games as firstbaseman
- G_2b: Games as secondbaseman
- G_3b: Games as thirdbaseman
- G_ss: Games as shortstop
- G_1f: Games as leftfielder
- G_cf: Games as centerfielder
- G_rf: Games as right fielder
- G_of: Games as outfielder
- G_dh: Games as designated hitter
- G_ph: Games as pinch hitter
- G_pr: Games as pinch runner

Details

The Appearances table in the original version has some incorrect variable names. In particular, the 5th column is `career_year`.

Source


Examples

```r
data(Appearances)
library("dplyr")
library("tidyr")

# Henry Aaron's last two years as a DH in Milwaukee
Appearances %>%
  filter(playerID == "aaronha01" & teamID == "MLA") %>%
  select(yearID,G_batting, G_of,G_ph) # subset variables

# Herb Washington, strictly a pinch runner for Oakland in 1974-5
Appearances %>%
  filter(playerID == "washihe01")

# A true utility player - Jerry Hairston, Jr.
Appearances %>%
  filter(playerID == "hairsje02")

# Appearances for the 1984 Cleveland Indians
Appearances %>%
  filter(teamID == "CLE" & yearID == 1984)

# Pete Rose's primary position each year of his career
Appearances %>%
```
library(dplyr)

# Most pitcher appearances each year since 1950
Appearsances %>%
  filter(yearID >= 1950) %>%
  group_by(yearID) %>%
  summarise(maxPitcher = playerID[which.max(G_p)],
    maxAppear = max(G_p))

# Individuals who have played all 162 games since 1961
all162 <- Appearsances %>%
  filter(yearID > 1960 & G_all == 162) %>%
  arrange(yearID, playerID) %>%
  select(yearID, playerID)

# Number of all-gamers by year (returns a vector)
table(all162$yearID)

# Players with most pinch hitting appearances in a year
Appearsances %>%
  arrange(desc(G_ph)) %>%
  select(playerID, yearID, teamID, lgID, G_all, G_ph) %>%
  head(., 10)

# Players with most pinch hitting appearances, career
Appearsances %>%
  group_by(playerID) %>%
  select(playerID, G_all, G_ph) %>%
  summarise(G = sum(G_all), PH = sum(G_ph)) %>%
  arrange(desc(PH)) %>%
  head(., 10)

# Players with most career appearances at each position
Appearsances %>%
  select(playerID, G_c:G_rf) %>%
  rename(C = G_c, `1B` = G_1b, `2B` = G_2b, SS = G_ss,
    `3B` = G_3b, LF = G_lf, CF = G_cf, RF = G_rf) %>%
  gather(pos, G, C:RF) %>%
  group_by(pos, playerID) %>%
  summarise(G = sum(G)) %>%
  arrange(desc(G)) %>%
  do(head(., 1))
Description

Award information for managers awards

Usage

data(AwardsManagers)

Format

A data frame with 179 observations on the following 6 variables.

playerID  Manager (player) ID code
awardID  Name of award won
yearID  Year
lgID  League; a factor with levels AL NL
tie  Award was a tie (Y or N)
notes  Notes about the award

Source


Examples

# Post-season managerial awards

# Number of recipients of each award by year
with(AwardsManagers, table(yearID, awardID))

# 1996 award winners
subset(AwardsManagers, yearID == 1996)

# AL winners of the BBWAA managerial award
subset(AwardsManagers, awardID == "BBWAA Manager of the year" & lgID == "AL")

# Tony LaRusso’s manager of the year awards
subset(AwardsManagers, playerId == "larusto@1")
**AwardsPlayers**

**AwardsPlayers table**

**Description**

Award information for players awards

**Usage**

```r
data(AwardsPlayers)
```

**Format**

A data frame with 6236 observations on the following 6 variables.

- `playerID` Player ID code
- `awardID` Name of award won
- `yearID` Year
- `lgID` League; a factor with levels AA AL ML NL
- `tie` Award was a tie (Y or N)
- `notes` Notes about the award

**Source**


**Examples**

```r
data(AwardsPlayers)
# Which awards have been given and how many?
with(AwardsPlayers, table(awardID))
awardtab <- with(AwardsPlayers, table(awardID))

# Plot the awardtab table as a Cleveland dot plot
library("lattice")
dotplot(awardtab)

# Restrict to MVP awards
mvp <- subset(AwardsPlayers, awardID == "Most Valuable Player")
# Who won in 1994?
mvp[mvp$yearID == 1994L, ]

goldglove <- subset(AwardsPlayers, awardID == "Gold Glove")
# which players won most often?
GGCount <- table(goldglove$playerID)
GGCount[GGCount>10]
```
# Triple Crown winners
subset(AwardsPlayers, awardID == "Triple Crown")

# Simultaneous Triple Crown and MVP winners
# (compare merged file to TC)
TC <- subset(AwardsPlayers, awardID == "Triple Crown")
MVP <- subset(AwardsPlayers, awardID == "Most Valuable Player")
keepvars <- c("playerID", "yearID", "lgID.x")
merge(TC, MVP, by = c("playerID", "yearID"))[ ,keepvars]

---

AwardsShareManagers

## Description
Award voting for managers awards

## Usage
data(AwardsShareManagers)

## Format
A data frame with 425 observations on the following 7 variables.

- awardID  name of award votes were received for
- yearID  Year
- lgID  League; a factor with levels AL NL
- playerID  Manager (player) ID code
- pointsWon  Number of points received
- pointsMax  Maximum number of points possible
- votesFirst  Number of first place votes

## Source
Examples

# Voting for the BBWAA Manager of the Year award by year and league
require("dplyr")

# Sort in decreasing order of points by year and league
AwardsShareManagers %>%
group_by(yearID, lgID) %>%
arrange(desc(pointsWon))

# Any unanimous winners?
AwardsShareManagers %>%
filter(pointsWon == pointsMax)

# Manager with highest proportion of possible points
AwardsShareManagers %>%
mutate(propWon = pointsWon/pointsMax) %>%
arrange(desc(propWon)) %>%
head(., 1)

# Bobby Cox's MOY vote tallies
AwardsShareManagers %>%
filter(playerID == "coxbo01")

AwardsSharePlayers table

Description

Award voting for managers awards

Usage

data(AwardsSharePlayers)

Format

A data frame with 6879 observations on the following 7 variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>awardID</td>
<td>name of award votes were received for</td>
</tr>
<tr>
<td>yearID</td>
<td>Year</td>
</tr>
<tr>
<td>lgID</td>
<td>League; a factor with levels AL ML NL</td>
</tr>
<tr>
<td>playerID</td>
<td>Player ID code</td>
</tr>
<tr>
<td>pointsWon</td>
<td>Number of points received</td>
</tr>
<tr>
<td>pointsMax</td>
<td>Maximum number of points possible</td>
</tr>
<tr>
<td>votesFirst</td>
<td>Number of first place votes</td>
</tr>
</tbody>
</table>
AwardsSharePlayers

Source

Examples

# Vote tallies for post-season player awards
require("dplyr")

# Which awards are represented in this data frame?
unique(AwardsSharePlayers$awardID)

# Sort the votes for the Cy Young award in decreasing order.
# Until 1967, the award went to the best pitcher
# in both leagues.
cyvotes <- AwardsSharePlayers %>%
  filter(awardID == "Cy Young") %>%
  group_by(yearID, lgID) %>%
  arrange(desc(pointsWon))

# 2012 votes
subset(cyvotes, yearID == 2012)

# top three votegetters each year by league
cya_top3 <- cyvotes %>%
  group_by(yearID, lgID) %>%
  do(head(.x, 3))
head(cya_top3, 12)

# unanimous Cy Young winners
subset(cyvotes, pointsWon == pointsMax)

## CYA was a major league award until 1967
## Find top five pitchers with most top 3 vote tallies in CYA
## head(with(cya_top3, rev(sort(table(playerID)))), 5)

# Pre-1967
cya_top3 %>%
  filter(yearID <= 1966) %>%
  group_by(playerID) %>%
  summarise(yrs_top3 = n()) %>%
  arrange(desc(yrs_top3)) %>%
  head(., 2)

# 1967+ (both leagues)
cya_top3 %>%
  filter(yearID > 1966) %>%
  group_by(playerID) %>%
  summarise(yrs_top3 = n()) %>%
```r
arrange(desc(yrs_top3)) %>%
head(., 5)

# 1967+ (by league)
cya_top3 %>%
filter(yearID > 1966) %>%
group_by(playerID, lgID) %>%
summarise(yrs_top3 = n()) %>%
arrange(desc(yrs_top3)) %>%
head(., 5)

# Ditto for MVP awards
# Top 3 votegetters for MVP award by year and league
MVP_top3 <- AwardsSharePlayers %>%
  filter(awardID == "MVP") %>%
group_by(yearID, lgID) %>%
  arrange(desc(pointsWon)) %>%
do(head(., 3))
tail(MVP_top3)

## Select players with >= 7 top 3 finishes
MVP_top3 %>%
group_by(playerID) %>%
summarise(n_top3 = n()) %>%
arrange(desc(n_top3)) %>%
filter(n_top3 > 6)
```

---

**Batting**

**Batting table**

**Description**

Batting table - batting statistics

**Usage**

data(Batting)

**Format**

A data frame with 105861 observations on the following 22 variables.

- **playerID**  Player ID code
- **yearID**  Year
- **stint**  player’s stint (order of appearances within a season)
- **teamID**  Team; a factor
- **lgID**  League; a factor with levels AA AL FL NL PL UA
Batting

G  Games: number of games in which a player played
AB  At Bats
R  Runs
H  Hits: times reached base because of a batted, fair ball without error by the defense
X2B  Doubles: hits on which the batter reached second base safely
X3B  Triples: hits on which the batter reached third base safely
HR  Homeruns
RBI  Runs Batted In
SB  Stolen Bases
CS  Caught Stealing
BB  Base on Balls
SO  Strikeouts
IBB  Intentional walks
HBP  Hit by pitch
SH  Sacrifice hits
SF  Sacrifice flies
GIDP  Grounded into double plays

Details

Variables X2B and X3B are named 2B and 3B in the original database

Source


See Also

battingStats for calculating batting average (BA) and other derived statistics
baseball for a similar dataset, but a subset of players who played 15 or more seasons.
Baseball for data on batting in the 1987 season.

Examples

data(Batting)
head(Batting)
require("dplyr")

## Prelude: Extract information from Salaries and People
## to be merged with the batting data.

# Subset of Salaries data
salaries <- Salaries %>%
  select(playerID, yearID, teamID, salary)
# Subset of People table (player metadata)
peopleInfo <- select(people, 
  select(playerID, birthYear, birthMonth, nameLast, nameFirst, bats)
)

# Left join salaries and peopleInfo to batting data,
# create an age variable and sort by playerID, yearID and stint
# Returns an ignorable warning.
batting <- left_join(salaries, 
  by = c("playerID", "yearID", "teamID")) 
left_join(peopleInfo, by = "playerID") 
mutate(age = yearID - birthYear - 
  if (birthMonth >= 10)) 
arrange(playerID, yearID, stint)

## Generate a ggplot similar to the NYT graph in the story about Ted Williams and the last .400 MLB season

# Restrict the pool of eligible players to the years after 1899 and
# players with a minimum of 450 plate appearances (this covers the
# strike year of 1994 when Tony Gwynn hit .394 before play was suspended
# for the season - in a normal year, the minimum number of plate appearances is 502)
eligibleHitters <- batting 
  filter(yearID >= 1900 & PA > 450)

# Find the hitters with the highest BA in MLB each year (there are a
# few ties). Include all players with BA > .400, whether they
# won a batting title or not, and add an indicator variable for
# .400 average in a season.
tophitters <- eligibleHitters 
  group_by(yearID) 
  filter(BA == max(BA) | BA >= .400) 
  mutate(ba400 = BA >= 0.400) 
  select(playerID, yearID, nameLast, nameFirst, BA, ba400)

# Sub-data frame for the .400 hitters plus the outliers after 1950
# (averages above .380) - used to produce labels in the plot below
bignames <- tophitters 
  filter(ba400 | (yearID > 1950 & BA > 0.380)) 
arrange(desc(BA))

# Variable to provide a vertical offset to certain
# labels in the ggplot below
bignames$yoffset <- c(0, 0, 0, 0, 0.002, 0, 0, 0, 
  0.001, -0.001, 0, -0.002, 0, 0, 0.002, 0, 0)
# Produce the plot

```r
require("ggplot2")
ggplot(tophitters, aes(x = yearID, y = BA)) +
  geom_point(aes(colour = ba400), size = 2.5) +
  geom_hline(yintercept = 0.400, size = 1, colour = "gray70") +
  geom_text(data = bignames, aes(y = BA + yoffset, label = nameLast),
            size = 3, hjust = 1.2) +
  scale_colour_manual(values = c("FALSE" = "black", "TRUE" = "red")) +
  xlim(1899, 2015) +
  scale_y_continuous("Batting average",
                    limits = c(0.330, 0.430),
                    breaks = seq(0.34, 0.42, by = 0.02),
  geom_smooth() +
  theme(legend.position = "none")
```

# after Chris Green,
# http://sabr.org/research/baseball-s-first-power-surge-home-runs-late-19th-century-major-leagues

# Total home runs by year

totalHR <- Batting %>%
  group_by(yearID) %>%
  summarise(Homeruns = sum(as.numeric(HR), na.rm=TRUE),
            Games = sum(as.numeric(G), na.rm=TRUE))

# Plot HR by year, pre-1919 (dead ball era)
totalHR %>% filter(yearID <= 1918) %>%
  ggplot(., aes(x = yearID, y = HomeRuns)) +
  geom_line() +
  geom_point() +
  labs(x = "Year", y = "Home runs hit")

# Take games into account
totalHR %>% filter(yearID <= 1918) %>%
  ggplot(., aes(x = yearID, y = HomeRuns/Games)) +
  geom_line() +
  geom_point() +
  labs(x = "Year", y = "Home runs per game played")

# Widen perspective to all years from 1871
ggplot(totalHR, aes(x = yearID, y = HomeRuns)) +
  geom_point() +
  geom_path() +
  geom_smooth() +
  labs(x = "Year", y = "Home runs hit")

# Similar plot for HR per game played by year -
# shows several eras with spikes in HR hit

ggplot(totalHR, aes(x = yearID, y = HomeRuns/Games)) +
```r
gem_point() +
gem_path() +
gem_smooth(se = FALSE) +
labs(x = "Year", y = "Home runs per game played")
```

<table>
<thead>
<tr>
<th>battingLabels</th>
<th>Variable Labels</th>
</tr>
</thead>
</table>

**Description**

These data frames provide descriptive labels for the variables in the *Batting*, *Pitching* and *Fielding* files (and related *Post* files). They are useful for plots and other output using `label`.

**Usage**

- `data(battingLabels)`
- `data(fieldingLabels)`
- `data(pitchingLabels)`

**Format**

Each is data frame with observations on the following 2 variables.

- `variable` variable name
- `label` variable label

**See Also**

- `Label`

**Examples**

```r
data(battingLabels)
str(battingLabels)

require("dplyr")

# find and plot maximum number of homers per year
batHR <- Batting %>%
  filter(!is.na(HR)) %>%
  group_by(yearID) %>%
  summarise(max=max(HR))

with(batHR, {
  plot(yearID, max,
```
BattingPost

```r
xlab=Label("yearID"), ylab=paste("Maximum", Label("HR")),
cex=0.8)
lines(lowess(yearID, max), col="blue", lwd=2)
abline(lm(max ~ yearID), col="red", lwd=2)
```

<table>
<thead>
<tr>
<th>Description</th>
<th>BattingPost table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Post season batting statistics</td>
</tr>
<tr>
<td>Usage</td>
<td>data(BattingPost)</td>
</tr>
<tr>
<td>Format</td>
<td>A data frame with 14354 observations on the following 22 variables.</td>
</tr>
<tr>
<td>yearID Year</td>
<td></td>
</tr>
<tr>
<td>round Level of playoffs</td>
<td></td>
</tr>
<tr>
<td>playerID Player ID code</td>
<td></td>
</tr>
<tr>
<td>teamID Team</td>
<td></td>
</tr>
<tr>
<td>lgID League; a factor with levels AA AL NL</td>
<td></td>
</tr>
<tr>
<td>G Games</td>
<td></td>
</tr>
<tr>
<td>AB At Bats</td>
<td></td>
</tr>
<tr>
<td>R Runs</td>
<td></td>
</tr>
<tr>
<td>H Hits</td>
<td></td>
</tr>
<tr>
<td>X2B Doubles</td>
<td></td>
</tr>
<tr>
<td>X3B Triples</td>
<td></td>
</tr>
<tr>
<td>HR Homeruns</td>
<td></td>
</tr>
<tr>
<td>RBI Runs Batted In</td>
<td></td>
</tr>
<tr>
<td>SB Stolen Bases</td>
<td></td>
</tr>
<tr>
<td>CS Caught stealing</td>
<td></td>
</tr>
<tr>
<td>BB Base on Balls</td>
<td></td>
</tr>
<tr>
<td>SO Strikeouts</td>
<td></td>
</tr>
<tr>
<td>IBB Intentional walks</td>
<td></td>
</tr>
<tr>
<td>HBP Hit by pitch</td>
<td></td>
</tr>
<tr>
<td>SH Sacrifices</td>
<td></td>
</tr>
<tr>
<td>SF Sacrifice flies</td>
<td></td>
</tr>
<tr>
<td>GIDP Grounded into double plays</td>
<td></td>
</tr>
</tbody>
</table>
Details

Variables X2B and X3B are named 2B and 3B in the original database

Source


Examples

# Post-season batting data
# Requires care since intra-league playoffs have evolved since 1969
# Simplest case: World Series

require("dplyr")

# Create a sub-data frame for modern World Series play
ws <- BattingPost %>%
  filter(round == "WS" & yearID >= 1903) %>%
  mutate(BA = 0 + (AB > 0) * round(H/AB, 3),
         TB = H + X2B + 2 * X3B + 3 * HR,
         SA = 0 + (AB > 0) * round(TB/AB, 3),
         PA = AB + BB + IBB + HBP + SH + SF,
         OB = H + BB + IBB + HBP,
         OBP = 0 + (AB > 0) * round(OB/PA, 3))

# Players with most appearances in the WS:
ws %>% group_by(playerID) %>%
  summarise(appearances = n()) %>%
  arrange(desc(appearances)) %>%
  head(., 10)

# Non-Yankees with most WS appearances
ws %>% filter(teamID != "NYA") %>%
  group_by(playerID) %>%
  summarise(appearances = n()) %>%
  arrange(desc(appearances)) %>%
  head(., 10)

# Top ten single WS batting averages ( >= 10 AB )
ws %>% filter(AB > 10) %>%
  arrange(desc(BA)) %>%
  head(., 10)

# Top ten slugging averages in a single WS
ws %>% filter(AB > 10) %>%
  arrange(desc(SA)) %>%
  head(., 10)

# Hitting stats for the 1946 St. Louis Cardinals, ordered by BA
battingStats

`battingStats` is a function that calculates additional batting statistics for each record in a Batting-like data.frame. The batting data does not contain batting statistics derived from those present in the data.frame. This function calculates batting average (`ba`), plate appearances (`pa`), total bases (`tb`), slugging percentage (`slugpct`), on-base percentage (`obp`), on-base percentage + slugging (`ops`), and batting average on balls in play (`babip`) for each record in a Batting-like data.frame.

**Description**

The `Batting` does not contain batting statistics derived from those present in the data.frame. This function calculates batting average (BA), plate appearances (PA), total bases (TB), slugging percentage (SlugPct), on-base percentage (OBP), on-base percentage + slugging (OPS), and batting average on balls in play (BABIP) for each record in a Batting-like data.frame.

**Usage**

```r
battingStats(data = Lahman::Batting, 
idvars = c("playerID", "yearID", "stint", "teamID", "lgID"), 
cbind = TRUE)
```

**Arguments**

- `data`: input data, typically `Batting`
- `idvars`: ID variables to include in the output data.frame
- `cbind`: If TRUE, the calculated statistics are appended to the input data as additional columns

**Details**

Standard calculations, e.g., `BA <- H/AB` are problematic because of the presence of NAs and zeros. This function tries to deal with those problems.

**Value**

A data.frame with all the observations in `data`. If `cbind==FALSE`, only the `idvars` and the calculated variables are returned.

**Author(s)**

Michael Friendly, Dennis Murphy
See Also

Batting, BattingPost

Examples

```r
bstats <- battingStats()
str(bstats)
bstats <- battingStats(cbind=FALSE)
str(bstats)
```

---

**CollegePlaying**  
*CollegePlaying table*

**Description**

Information on schools players attended, by player

**Usage**

```r
data(CollegePlaying)
```

**Format**

A data frame with 17350 observations on the following 3 variables.

- **playerID**  Player ID code
- **schoolID**  school ID code
- **yearID**  Year player attended school

**Details**

This data set reflects a change in the Lahman schema for the 2015 version. The old SchoolsPlayers table was replaced with this new table called CollegePlaying.

According to the documentation, this change reflects advances in the compilation of this data, largely led by Ted Turocy. The old table reported college attendance for major league players by listing a start date and end date. The new version has a separate record for each year that a player attended. This allows us to better account for players who attended multiple colleges or skipped a season, as well as to identify teammates.

**Source**

Fielding

Examples

```r
data(CollegePlaying)
head(CollegePlaying)

## Q: What are the top universities for producing MLB players?
SPcount <- table(CollegePlaying$schoolID)
SPcount[SPcount>50]

library("lattice")
dotplot(SPcount[SPcount>50])
dotplot(sort(SPcount[SPcount>50]))

## Q: How many schools are represented in this dataset?
length(table(CollegePlaying$schoolID))

# Histogram of the number of players from each school who played in MLB:
with(CollegePlaying,
    hist(table(schoolID), xlab = "Number of players",
    main = ""))
```

Fielding

Fielding table

Description

Fielding table

Usage

data(Fielding)

Format

A data frame with 140921 observations on the following 18 variables.

- `playerID` Player ID code
- `yearID` Year
- `stint` player’s stint (order of appearances within a season)
- `teamID` Team; a factor
- `lgID` League; a factor with levels AA AL FL NL PL UA
- `POS` Position
- `G` Games
- `GS` Games Started
- `InnOuts` Time played in the field expressed as outs
- `PO` Putouts
Fielding

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Assists</td>
</tr>
<tr>
<td>E</td>
<td>Errors</td>
</tr>
<tr>
<td>DP</td>
<td>Double Plays</td>
</tr>
<tr>
<td>PB</td>
<td>Passed Balls (by catchers)</td>
</tr>
<tr>
<td>WP</td>
<td>Wild Pitches (by catchers)</td>
</tr>
<tr>
<td>SB</td>
<td>Opponent Stolen Bases (by catchers)</td>
</tr>
<tr>
<td>CS</td>
<td>Opponents Caught Stealing (by catchers)</td>
</tr>
<tr>
<td>ZR</td>
<td>Zone Rating</td>
</tr>
</tbody>
</table>

Source


Examples

data(fielding)
# Basic fielding data

require("dplyr")

# Roberto Clemente’s fielding profile
# pitching and catching related data removed
# subset( Fielding, playerID == "clemero01")[, 1:13]
Fielding %>%
  filter(playerID == "clemero01") %>%
  select(1:13)

# Yadier Molina’s fielding profile
# PB, WP, SP and CS apply to catchers
Fielding %>%
  subset(playerID == "molinya01") %>%
  select(-WP, -ZR)

# Pedro Martinez’s fielding profile
Fielding %>% subset(playerID == "martipe02")

# Table of games played by Pete Rose at different positions
with(subset( Fielding, playerID == "rosepe01"), xtabs(G ~ POS))

# Career total G/PO/A/E/DP for Luis Aparicio
Fielding %>%
  filter(playerID == "aparilu01") %>%
  select(G, PO, A, E, DP) %>%
  summarise_each(funs(sum))

# Top ten 2B/SS in turning DPs
Fielding %>%
subset(PO5 %in% c("2B", "SS")) %>%
group_by(playerID) %>%
summarise(TDP = sum(DP, na.rm = TRUE)) %>%
arrange(desc(TDP)) %>%
head(.., 10)

# League average fielding statistics, 1961-present
Fielding %>%
  filter(yearID >= 1961 & POS != "DH") %>%
  select(yearID, lgID, POS, InnOuts, PO, A, E) %>%
  group_by(yearID, lgID) %>%
  summarise_at(vars(InnOuts, PO, A, E), funs(sum), na.rm = TRUE) %>%
  mutate(fpct = round((PO + A)/(PO + A + E), 3),
         OPE = round(InnOuts/E, 3))

---

### FieldingOF table

**Description**

Outfield position data: information about positions played in the outfield

**Usage**

```
data(FieldingOF)
```

**Format**

A data frame with 12028 observations on the following 6 variables.

- `playerID` Player ID code
- `yearID` Year
- `stint` player’s stint (order of appearances within a season)
- `GLf` Games played in left field
- `Gcf` Games played in center field
- `Grf` Games played in right field

**Source**

Examples

```r
require("dplyr")
require("tidyr")

## Data set only goes through 1955
## Can get a more complete record from the Fielding data frame
## or from the Appearances data (see below)

## Output directly from the FieldingOF data

## Barry Bonds (no records: post-1955 player)
FieldingOF %>%
  filter(playerID == "bondsba01")

## Willie Mays (first few years)
FieldingOF %>%
  filter(playerID == "mayswi01")

## Ty Cobb (complete)
FieldingOF %>%
  filter(playerID == "cobbty01")

## One way to get OF game information from the Fielding data
## Note: OF games != sum(LF, CF, RF) because players can switch
## OF positions within a game. Players can also switch from
## other positions to outfield during a game. OF represents
## the number of games a player started in the outfield.
Fielding %>%
  select(playerID, yearID, stint, POS, G) %>%
  filter(POS %in% c("LF", "CF", "RF", "OF")) %>%
  tidyr::spread(POS, G, fill = 0) %>%
  filter(playerID == "trumbma01")

## Another way is through the Appearances data (no stint).
## Provides a somewhat nicer table than the above.

## Mark Trumbo (active player)
Appearances %>%
  select(playerID, yearID, G_Lf, G_cf, G_rf, G_of) %>%
  filter(playerID == "trumbma01")

## A slightly better format, perhaps
Appearances %>%
  select(playerID, yearID, G_Lf, G_cf, G_rf, G_of) %>%
  rename(LF = G_Lf, CF = G_cf, RF = G_rf, OF = G_of) %>%
  filter(playerID == "trumbma01")

## Willie Mays (1951-1973)
Appearances %>%
  select(playerID, yearID, G_Lf, G_cf, G_rf, G_of) %>%
  filter(playerID == "mayswi01")
```
## FieldingPost data

### Description

Post season fielding data

### Usage

```r
data(FieldingPost)
```

### Format

A data frame with 13534 observations on the following 17 variables.

- `playerID` Player ID code
- `yearID` Year
- `teamID` Team; a factor
- `lgID` League; a factor with levels AL NL
- `round` Level of playoffs
- `pos` Position
- `G` Games
- `GS` Games Started
- `innouts` Time played in the field expressed as outs
- `PO` Putouts
- `A` Assists
- `E` Errors
- `DP` Double Plays
- `TP` Triple Plays
- `PB` Passed Balls
- `SB` Stolen Bases allowed (by catcher)
- `CS` Caught Stealing (by catcher)

### Source

Examples

```r
require("dplyr")

## World Series fielding record for Yogi Berra
FieldingPost %>%
  filter(playerID == "berrayo01" & round == "WS")

## Yogi's career efficiency in throwing out base stealers
## in his WS appearances and CS as a percentage of his
## overall assists
FieldingPost %>%
  filter(playerID == "berrayo01" & round == "WS" & POS == "C") %>%
  summarise(cs_pct = round(100 * sum(CS)/sum(SB + CS), 2),
             cs_assists = round(100 * sum(CS)/sum(A), 2))

## Innings per error for several selected shortstops in the WS
FieldingPost %>%
  filter(playerID %in% c("belanma01", "jeterde01", "campabe01",
                         "conceda01", "bowala01"), round == "WS") %>%
  group_by(playerID) %>%
  summarise(G = sum(G),
             InnOuts = sum(InnOuts),
             Eper9 = round(27 * sum(E)/sum(InnOuts), 3))

## Top 10 center fielders in innings played in the WS
FieldingPost %>%
  filter(POS == "CF" & round == "WS") %>%
  group_by(playerID) %>%
  summarise(inn_total = sum(InnOuts)) %>%
  arrange(desc(inn_total)) %>%
  head(., 10)

## Most total chances by position
FieldingPost %>%
  filter(round == "WS" & !(POS %in% c("DH", "OF", "P"))) %>%
  group_by(POS, playerID) %>%
  summarise(TC = sum(P0 + A + E)) %>%
  arrange(desc(TC)) %>%
  do(head(., 1))  # provides top player by position
```

---

**HallOfFame**

**Hall of Fame Voting Data**

**Description**

Hall of Fame table. This is composed of the voting results for all candidates nominated for the Baseball Hall of Fame.
Usage
data(HallOfFame)

Format
A data frame with 4191 observations on the following 9 variables.

playerID  Player ID code
yearID   Year of ballot
votedBy  Method by which player was voted upon. See Details
ballots  Total ballots cast in that year
needed   Number of votes needed for selection in that year
votes    Total votes received
inducted Whether player was inducted by that vote or not (Y or N)
category Category of candidate; a factor with levels Manager Pioneer/Executive Player Umpire
needed_note Explanation of qualifiers for special elections

details
This table links to the People table via the playerID.

votedBy: Most Hall of Fame inductees have been elected by the Baseball Writers Association of America (BBWAA). Rules for election are described in [http://en.wikipedia.org/wiki/National_Baseball_Hall_of_Fame_and_Museum#Selection_process](http://en.wikipedia.org/wiki/National_Baseball_Hall_of_Fame_and_Museum#Selection_process).

Source

Examples
## Some examples for Hall of Fame induction data

```R
require("dplyr")
require("ggplot2")

# Some simple queries

# What are the different types of HOF voters?
table(HallOfFame$votedBy)

# What was the first year of Hall of Fame elections?
sort(unique(HallOfFame$yearID))[1]

# Who comprised the original class?
subset(HallOfFame, yearID == 1936 & inducted == "Y")

# Result of a player's last year on the BBWAA ballot
```
# Restrict to players voted by BBWAA:
HOFplayers <- subset(HallOfFame, votedBy == "BBWAA" & category == "Player")

# Number of years as HOF candidate, last pct vote, etc.
# for a given player
playerOutcomes <- HallOfFame %>%
  filter(votedBy == "BBWAA" & category == "Player") %>%
  group_by(playerID) %>%
  mutate(nyears = length(ballots)) %>%
  arrange(yearID) %>%
  do(tail(., 1)) %>%
  mutate(lastPct = 100 * round(votes/ballots, 3)) %>%
  select(playerID, nyears, inducted, lastPct, yearID) %>%
  rename(lastYear = yearID)

# How many voting years until election?
inducted <- subset(playerOutcomes, inducted == "Y")
table(inducted$nyears)

# Bar chart of years to induction for inductees
barplot(table(inducted$nyears),
        main="Number of voting years until election",
        ylab="Number of players", xlab="Years")
box()

# What is the form of this distribution?
require("vcd")
goodfit(inducted$nyears)
plot(goodfit(inducted$nyears), xlab="Number of years",
     main="Poissonness plot of number of years voting until election")
ord_plot(table(inducted$nyears), xlab="Number of years")

# First ballot inductees sorted by vote percentage:
playerOutcomes %>%
  filter(nyears == 1L & inducted == "Y") %>%
  arrange(desc(lastPct))

# Who took at least ten years on the ballot before induction?
playerOutcomes %>%
  filter(nyears >= 10L & inducted == "Y")

# Plots of voting percentages over time for the borderline HOF candidates, according to the BBWAA:

# Identify players on the BBWAA ballot for at least 10 years
# Returns a character vector of playerIDs
longTimers <- as.character(unlist(subset(playerOutcomes, nyears >= 10, select = "playerID")))

# Extract their information from the HallOfFame data
HOF1t <- HallOfFame %>%
  filter(playerID %in% longTimers & votedBy == "BBWAA") %>%
  group_by(playerID) %>%
  mutate(elected = ifelse(any(inducted == "Y"), "Elected", "Not elected"),
          pct = 100 * round(votes/ballots, 3))

# Plot the voting profiles:
HOFplot <- ggplot(HOF1t, aes(x = yearID, y = pct, group = playerID)) +
  ggtitle("Profiles of BBWAA voting percentage, long-time HOF candidates") +
  geom_line() +
  geom_hline(yintercept = 75, colour = "red") +
  labs(x = "Year", y = "Percentage of votes") +
  facet_wrap(~ elected, ncol = 1)

## Eventually inductees tend to have increasing support over time.
## Fit simple linear regression models to each player’s voting
## percentage profile and extract the slopes. Then compare the
## distributions of the slopes in each group.

# data frame for playerID and induction status among
# long term candidates
HOFstatus <- HOF1t %>%
  group_by(playerID) %>%
  select(playerID, elected, inducted) %>%
  do(tail(., 1))

# data frame of regression slopes, which represent average
# increase in percentage support by BBWAA members over a
# player’s candidacy.
HOFslope <- HOF1t %>%
  group_by(playerID) %>%
  do(mod = lm(pct ~ yearID, data = .)) %>%
  do(data.frame(slope = coef(mod)[2]))

## Boxplots of regression slopes by induction group
HOFplot(data.frame(HOFstatus, HOFslope),
        aes(x = elected, y = slope)) +
  geom_boxplot(width = 0.5) +
  geom_point(position = position_jitter(width = 0.2))

# Note 1: Only two players whose maximum voting percentage
# was over 66% were not eventually inducted
# into the HOF: Gil Hodges and Jack Morris.
# Red Ruffing was elected in a 1967 runoff election while
# the others have been voted in by the Veterans Committee.

# Note 2: Of the players whose slope was >= 2.5 among
Label

Extract the Label for a Variable

Description

Extracts the label for a variable from one or more of the *Labels files. This is useful for plots and other displays because the variable names are often cryptically short.

Usage

```r
Label(var, labels = rbind(Lahman::battingLabels,
                           Lahman::pitchingLabels,
                           Lahman::fieldingLabels))
```

Arguments

- `var`: name of a variable
- `labels`: label table(s) to search, a 2-column dataframe containing variable names and labels.

Value

Returns the variable label, or `var` if no label is found

Author(s)

Michael Friendly

See Also

`battingLabels`, `pitchingLabels`, `fieldingLabels`

Examples

```r
require("dplyr")
# find and plot maximum number of home runs per year
bathR <- Batting %>%
  filter(!is.na(HR)) %>%
  group_by(yearID) %>%
  summarise(max = max(HR))

with(bathR, {
```
**LahmanData**

```
plot(yearID, max,
  xlab=Label("yearID"), ylab=paste("Maximum", Label("HR")),
  cex=0.8)
lines(lowess(yearID, max), col="blue", lwd=2)
abline(lm(max ~ yearID), col="red", lwd=2)
```
Managers table

Description
Managers table: information about individual team managers, teams they managed and some basic statistics for those teams in each year.

Usage
data(Managers)

Format
A data frame with 3504 observations on the following 10 variables.

playerID Manager (player) ID code
yearID Year
teamID Team; a factor
lgID League; a factor with levels AA AL FL NL PL UA
Managers

Managers

inseason  Managerial order. Zero if the individual managed the team the entire year. Otherwise denotes where the manager appeared in the managerial order (1 for first manager, 2 for second, etc.)

G  Games managed

W  Wins

L  Losses

rank  Team’s final position in standings that year

plyrMgr  Player Manager (denoted by ‘Y’); a factor with levels N Y

Source


Examples

# Basic career summaries by manager
library(dplyr)
mgrSumm <- Managers %>%
  group_by(playerID) %>
  summarise(nyear = length(unique(yearID)),
    yearBegin = min(yearID),
    yearEnd = max(yearID),
    nTeams = length(unique(teamID)),
    nFirst = sum(rank == 1L),
    W = sum(W),
    L = sum(L),
    WinPct = round(W/(W + L), 3))

MgrInfo <- People %>%
  filter(!is.na(playerID)) %>
  select(playerID, nameLast, nameFirst)

# Merge names into the table
mgrTotals <- right_join(MgrInfo, mgrSumm, by = "playerID")

# add total games managed
mgrTotals <- mgTotals %>%
  mutate(games = W + L)

# Some basic queries

# Top 20 managers in terms of years of service:
mgrTotals %>%
  arrange(desc(nyear)) %>%
  select(playerID, nameLast, nameFirst, nyear)
Managers

head(., 20)

# Top 20 winningest managers (500 games minimum)
mgrTotals %>%
  filter((W + L) >= 500) %>%
  arrange(desc(WinPct)) %>%
  head(., 20)

# Most of these are 19th century managers.
# How about the modern era?
mgrTotals %>%
  filter(yearBegin >= 1901 & (W + L) >= 500) %>%
  arrange(desc(WinPct)) %>%
  head(., 20)

# Top 10 managers in terms of percentage of titles
# (league or divisional) - should bias toward managers
# post-1970 since more first place finishes are available
mgrTotals %>%
  filter(yearBegin >= 1901 & (W + L) >= 500) %>%
  arrange(desc(round(nfirst/nyear, 3))) %>%
  head(., 10)

# How about pre-1969?
mgrTotals %>%
  filter(yearBegin >= 1901 & yearEnd <= 1969 &
         (W + L) >= 500) %>%
  arrange(desc(round(nfirst/nyear, 3))) %>%
  head(., 10)

## Tony LaRussa's managerial record by team
Managers %>%
  filter(playerID == "larusto01") %>%
  group_by(teamID) %>%
  summarise(nyear = length(unique(yearID)),
            yearBegin = min(yearID),
            yearEnd = max(yearID),
            games = sum(G),
            nfirst = sum(rank == 1L),
            W = sum(W),
            L = sum(L),
            WinPct = round(W/(W + L), 3))

# Density plot of the number of games managed:

library("ggplot2")

ggplot(mgrTotals, aes(x = games)) +
  geom_density(fill = "red", alpha = 0.3) +
  labs(x = "Number of games managed")
Managers

# Who managed more than 4000 games?
mgrTotals %>%
  filter(W + L >= 4000) %>%
  arrange(desc(W + L))
# Connie Mack’s advantage: he owned the Philadelphia A’s :) 

# Table of Tony LaRussa’s team finishes (rank order):
managers %>%
  filter(playerID == "larusto01") %>%
  count(rank)

# Scatterplot of winning percentage vs. number of games managed (min 100)

ggplot(subset(mgrTotals, yearBegin >= 1900 & games >= 100),
  aes(x = games, y = WinPct)) +
  geom_point() + geom_smooth() +
  labs(x = "Number of games managed")

# Division titles

# Plot of number of first place finishes by managers who
# started in the divisional era (>= 1969) with
# at least 8 years of experience

mgrTotals %>%
  filter(yearBegin >= 1969 & nyear >= 8) %>%
  ggplot(. , aes(x = nyear, y = nfirst)) +
  geom_point(position = position_jitter(width = 0.2)) +
  labs(x = "Number of years",
      y = "Number of divisional titles") +
  geom_smooth()

# Change response to proportion of titles relative
# to years managed

mgrTotals %>%
  filter(yearBegin >= 1969 & nyear >= 8) %>%
  ggplot(. , aes(x = nyear, y = round(nfirst/nyear, 3))) +
  geom_point(position = position_jitter(width = 0.2)) +
  labs(x = "Number of years",
      y = "Proportion of divisional titles") +
  geom_smooth()
ManagersHalf

ManagersHalf table

**Description**

Split season data for managers

**Usage**

```r
data(ManagersHalf)
```

**Format**

A data frame with 93 observations on the following 10 variables.

- `playerID`: Manager (player) ID code
- `yearID`: Year
- `teamID`: Team; a factor
- `lgID`: League; a factor with levels AL NL
- `inseason`: Managerial order. One if the individual managed the team the entire year. Otherwise denotes where the manager appeared in the managerial order (1 for first manager, 2 for second, etc.). A factor with levels 1 2 3 4 5
- `half`: First or second half of season
- `G`: Games managed
- `W`: Wins
- `L`: Losses
- `rank`: Team's position in standings for the half

**Source**


**Examples**

```r
library("dplyr")
library("reshape2")

# Only have data for 1892 and 1893
ManagersHalf %>%
  filter(yearID >= 1891) %>%
  group_by(teamID, yearID) %>%
  filter(all(playerID == playerID[1])) %>% # same manager in both halves
Master table

Description

**Deprecation Notice**: The Master table is now the People table in the Lahman dataset. Master is now a copy of People and is being retained for backward compatibility. Please change your code to use the People table.

Master table - Player names, DOB, and biographical info. This file is to be used to get details about players listed in the Batting, Pitching, and other files where players are identified only by playerId.

Usage

data(Master)

Format

A data frame with 19617 observations on the following 26 variables.

- **playerID**: A unique code assigned to each player. The `playerID` links the data in this file with records on players in the other files.
- **birthYear**: Year player was born
- **birthMonth**: Month player was born
- **birthDay**: Day player was born
- **birthCountry**: Country where player was born
- **birthState**: State where player was born
- **birthCity**: City where player was born
- **deathYear**: Year player died
- **deathMonth**: Month player died
- **deathDay**: Day player died
- **deathCountry**: Country where player died
- **deathState**: State where player died
- **deathCity**: City where player died
- **nameFirst**: Player’s first name
- **nameLast**: Player’s last name
- **nameGiven**: Player’s given name (typically first and middle)
- **weight**: Player’s weight in pounds
height  Player's height in inches
bats  a factor: Player's batting hand (left (L), right (R), or both (B))
throws  a factor: Player's throwing hand (left(L) or right(R))
debut  Date that player made first major league appearance
finalGame  Date that player made first major league appearance (blank if still active)
retroID  ID used by retrosheet, http://www.retrosheet.org/
birthDate  Player's birthdate, in as.Date format
deathDate  Player's deathdate, in as.Date format

Details

debut, finalGame were converted from character strings with as.Date.

Source


Examples

data(Master); data(Batting)

## add player's name to Batting data
Master$name <- paste(Master$nameFirst, Master$nameLast, sep=" ")
batting <- merge(Batting,
    Master[,c("playerID","name")],
    by="playerID", all.x=TRUE)

## batting and throwing
# right-handed batters are much less ambidexterous in throwing than left-handed batters
# (should only include batters)

BT <- with(Master, table(bats, throws))
require(vcd)
structable(BT)
mosaic(BT, shade=TRUE)

## Who is Shoeless Joe Jackson?
subset(Master, nameLast="Jackson" & nameFirst="Joe")
subset(Master, nameLast="Jackson" & nameFirst="Shoeless Joe")

joeID <- c(subset(Master, nameLast="Jackson" & nameFirst="Shoeless Joe")$"playerID")

subset(Batting, playerID==joeID)
subset(Fielding, playerID==joeID)
**Description**

Name and location data for baseball stadiums.

**Usage**

```r
data(Parks)
```

**Format**

A data frame with 252 observations on the following 6 variables.

- `park.key` unique identifier for each ballpark
- `park.name` the name of the ballpark
- `park.alias` a semicolon delimited list of other names for the ballpark if they exist
- `city` city where the ballpark is located
- `state` state where the ballpark is located
- `country` country where the ballpark is located

**Details**

This dataset apparently includes all ballparks that were ever used in baseball. There is no indication of the years they were used, nor the teams that played there.

The ballparks can be associated with teams through the `park` variable in the `Teams` table.

**Source**


**See Also**

`Teams`

**Examples**

```r
data(Parks)
library(dplyr)
# how many parks in each country?
table(Parks$country)

# how many parks in each US state?
Parks %>%
  filter(country=="US") %>%
```
count(state, sort=TRUE)

# ballparks in NYC
Parks %>%
  filter(state=="NY") %>%
  filter(city %in% c("New York", "Brooklyn", "Queens"))

# ballparks in Canada
Parks %>%
  filter(country=="CA") %>%
  count(state, sort=TRUE)

# what are the Canadian parks?
Parks %>%
dplyr::filter(country=="CA")

---

### People table

**Description**

People table - Player names, DOB, and biographical info. This file is to be used to get details about players listed in the Batting, Pitching, and other files where players are identified only by playerID.

**Usage**

data(People)

**Format**

A data frame with 19617 observations on the following 26 variables.

playerID A unique code assigned to each player. The playerID links the data in this file with records on players in the other files.
birthYear Year player was born
birthMonth Month player was born
birthDay Day player was born
birthCountry Country where player was born
birthState State where player was born
birthCity City where player was born
deathYear Year player died
deathMonth Month player died
deathDay Day player died
deathCountry Country where player died
People

deathState  State where player died
deathCity  City where player died
nameFirst  Player’s first name
nameLast  Player’s last name
nameGiven  Player’s given name (typically first and middle)
weight  Player’s weight in pounds
height  Player’s height in inches
bats  a factor: Player’s batting hand (left (L), right (R), or both (B))
throws  a factor: Player’s throwing hand (left(L) or right(R))
debut  Date that player made first major league appearance
finalGame  Date that player made first major league appearance (blank if still active)
retroID  ID used by retrosheet, http://www.retrosheet.org/
birthDate  Player’s birthdate, in as.Date format
deathDate  Player’s deathdate, in as.Date format

Details

debut, finalGame were converted from character strings with as.Date.

Source


Examples

data(People); data(Batting)

## add player’s name to Batting data
People$name <- paste(People$nameFirst, People$nameLast, sep= " ")
batting <- merge(Batting, 
  People[,c("playerID","name")],
  by="playerID", all.x=TRUE)

## batting and throwing
# right-handed batters are much less ambidexterous in throwing than left-handed batters
# (should only include batters)

BT <- with(People, table(bats, throws))
require(vcd)
structable(BT)
mosaic(BT, shade=TRUE)

## Who is Shoeless Joe Jackson?
subset(People, nameLast=="Jackson" & nameFirst=="Joe")
subset(People, nameLast=="Jackson" & nameFirst=="Shoeless Joe")

joeID <- c(subset(People, nameLast=="Jackson" & nameFirst=="Shoeless Joe")["playerID"])

subset(Batting, playerID==joeID)
subset(Fielding, playerID==joeID)

<table>
<thead>
<tr>
<th>Pitching</th>
<th>Pitching table</th>
</tr>
</thead>
</table>

**Description**

Pitching table

**Usage**

data(Pitching)

**Format**

A data frame with 46699 observations on the following 30 variables.

- **playerID**  Player ID code
- **yearID**  Year
- **stint**  player's stint (order of appearances within a season)
- **teamID**  Team; a factor
- **lgID**  League; a factor with levels AA AL FL NL PL UA
- **W**  Wins
- **L**  Losses
- **G**  Games
- **GS**  Games Started
- **CG**  Complete Games
- **SHO**  Shutouts
- **SV**  Saves
- **IPouts**  Outs Pitched (innings pitched x 3)
- **H**  Hits
- **ER**  Earned Runs
- **HR**  Homeruns
- **BB**  Walks
- **SO**  Strikeouts
- **BAOpp**  Opponent's Batting Average
- **ERA**  Earned Run Average
Pitching

IBB  Intentional Walks
WP   Wild Pitches
HBP  Batters Hit By Pitch
BK   Balks
BFP  Batters faced by Pitcher
GF   Games Finished
R    Runs Allowed
SH   Sacrifices by opposing batters
SF   Sacrifice flies by opposing batters
GIDP Grounded into double plays by opposing batter

Source

Examples

# Pitching data
require("dplyr")

####################################################
# cleanup, and add some other stats
####################################################

# Restrict to AL and NL data, 1901+
# All data re SH, SF and GIDP are missing, so remove
# Intentional walks (IBB) not recorded until 1955
pitching <- Pitching %>%
  filter(yearID >= 1901 & lgID %in% c("AL", "NL")) %>%
  select(-c(28:30)) %>%
  # remove SH, SF, GIDP
  mutate(BAopp = round(H/(H + IPouts), 3),
          # loose def'n
          WHIP = round((H + BB) / 3/IPouts, 2),
          KperBB = round(ifelse(yearID >= 1955,
                              SO/(BB - IBB), SO/BB), 2))

####################################################
# some simple queries
####################################################

# Team pitching statistics, Toronto Blue Jays, 1993
tor93 <- pitching %>%
  filter(yearID == 1993 & teamID == "TOR") %>%
  arrange(ERA)

# Career pitching statistics, Greg Maddux
subset(pitching, playerId == "maddug01")
# Best ERAs for starting pitchers post WWII

```r
testing <- function(yearID, IPouts) {
  group_by(lgID) %>%
  arrange(ERA) %>%
  head(10)
}
```

# Best K/BB ratios post-1955 among starters (excludes intentional walks)

```r
kbb <- function(yearID, IPouts) {
  mutate(KperBB = SO/(BB - IBB)) %>%
  arrange(desc(KperBB)) %>%
  head(10)
}
```

# Best K/BB ratios among relievers post-1950 (min. 20 saves)

```r
kbb relief <- function(yearID, SV) {
  arrange(desc(KperBB)) %>%
  head(10)
}
```

# Winningest pitchers in each league each year

```r
winningest <- function(playerID, nameLast, nameFirst, throws) {
  select(playerID, nameLast, nameFirst, throws) %>%
  right_join(pitching, by = "playerID") %>%
  group_by(yearID, lgID) %>%
  filter(W == max(W)) %>%
  select(nameLast, nameFirst, teamID, W, throws)
}
```

# A simple ANCOVA model of wins vs. year, league and hand (L/R)

```r
anova(lm(formula = W ~ yearID + I(yearID^2) + lgID + throws, data = winningest))
```

# Nature of managing pitching staffs has altered importance of wins over time

```r
require("ggplot2")
```

# compare loess smooth with quadratic fit

```r
ggplot(winningest, aes(x = yearID, y = W)) +
ggplot2::geom_point(aes(colour = throws, shape=lgID, size = 2)) +
ggplot2::geom_smooth(method="loess", size=1.5, color="blue") +
ggplot2::geom_smooth(method = "lm", se=FALSE, color="black",
```
```r

formula = y - poly(x,2)) + 
  ylab("League maximum Wins") + xlab("Year") + 
  ggtitle("Maximum pitcher wins by year")

## To reinforce this, plot the mean IPouts by year and league, 
## which gives some idea of pitcher usage. Restrict pitcher 
## pool to those who pitched at least 100 innings in a year.

pitching %>% filter(IPouts >= 300) %>% # # >= 100 IP 

  ggplot(. , aes(x = yearID, y = IPouts, color = lgID)) + 
  geom_smooth(method="loess") + 
  labs(x = "Year", y = "IPouts")

## Another indicator: total number of complete games pitched 
## (Mirrors the trend from the preceding plot.) 

pitching %>%
  group_by(yearID, lgID) %>%
  summarise(totalCG = sum(CG, na.rm = TRUE)) %>%
  ggplot(. , aes(x = yearID, y = totalCG, color = lgID)) + 
  geom_point() +
  geom_path() +
  labs(x = "Year", y = "Number of complete games")

## End(Not run)
```

---

### Description

Post season pitching statistics

### Usage

```r
data(PitchingPost)
```

### Format

A data frame with 5624 observations on the following 30 variables.

- **playerID**: Player ID code
- **yearID**: Year
- **round**: Level of playoffs
- **teamID**: Team; a factor
- **lgID**: League; a factor with levels AA AL NL
- **w**: Wins
L Losses
G Games
GS Games Started
CG Complete Games
SHO Shutouts
SV Saves
IPouts Outs Pitched (innings pitched x 3)
H Hits
ER Earned Runs
HR Homeruns
BB Walks
SO Strikeouts
BAOpp Opponents’ batting average
ERA Earned Run Average
IBB Intentional Walks
WP Wild Pitches
HBP Batters Hit By Pitch
BK Balks
BFP Batters faced by Pitcher
GF Games Finished
R Runs Allowed
SH Sacrifice Hits allowed
SF Sacrifice Flies allowed
GIDP Grounded into Double Plays

Source

Examples
library("dplyr")
library(ggplot2)

# Restrict data to World Series in modern era
ws <- PitchingPost %>%
  filter(yearID >= 1903 & round == "WS")
# Pitchers with ERA 0.00 in WS play (> 10 IP)
ws %>%
  filter(IPouts > 30 & ERA == 0.00) %>%
  arrange(desc(IPouts)) %>%
  select(playerID, yearID, teamID, lgID, IPouts, W, L, G
playerInfo

Lookup Information for Players and Teams

Description

These functions use grep to lookup information about players (from the People file) and teams (from the Teams file).
Salaries

Usage

 playerInfo(playerID, nameFirst, nameLast, data = Lahman::People, extra = NULL, ...)

 teamInfo(teamID, name, data = Lahman::Teams, extra = NULL, ...)

Arguments

 playerID pattern for playerID
 nameFirst pattern for first name
 nameLast pattern for last name
 data The name of the dataset to search
 extra A character vector of other fields to include in the result
 ... other arguments passed to grep
 teamID pattern for teamID
 name pattern for team name

Value

 Returns a data frame for unique matching rows from data

Author(s)

 Michael Friendly

See Also

grep.

Examples

 playerInfo("aaron")

 teamInfo("CH", extra="park")

Salaries  Salaries table

Description

 Player salary data.

Usage

 data(Salaries)
Salaries

Format

A data frame with 26428 observations on the following 5 variables.

- **yearID** Year
- **teamID** Team; a factor
- **lgID** League; a factor
- **playerID** Player ID code
- **salary** Salary

Details

There is no real coverage of player's salaries until 1985.

Source


Examples

```r
# what years are included?
summary(Salaries$yearID)

# how many players included each year?
table(Salaries$yearID)

# Team salary data
require("dplyr")
require("ggplot2")

# Total team salaries by league, team and year
teamSalaries <- Salaries %>%
group_by(lgID, teamID, yearID) %>%
summarise(Salary = sum(as.numeric(salary))) %>%
group_by(yearID, lgID) %>%
arrange(desc(Salary))

# Highest paid players each year:
maxSal <- Salaries %>%
group_by(yearID) %>%
filter(salary == max(salary))
maxPlayers <- bind_rows(lapply(maxSal$playerID, playerInfo)) %>%
select(-playerID)
maxSal <- bind_cols(maxPlayers, maxSal)

# Plot maximum MLB salary by year (1985-present)
ggplot(maxSal, aes(x = yearID, y = salary/1e6)) +
geom_point() +
```

geom_smooth(se = FALSE) +
labs(x = "Year", y = "Salary (millions)"")

# Plot salary distributions by year for all players
ggplot(Salaries, aes(x = factor(yearID), y = salary/1e5)) +
geom_boxplot(fill = "lightblue", outlier.size = 1) +
labs(x = "Year", y = "Salary ($100,000)") +
coord_flip()

# Plot median MLB salary per year
Salaries %>%
  group_by(yearID) %>%
  summarise(medsal = median(salary)) %>%
  ggplot(., aes(x = yearID, y = medsal/1e6)) +
geom_point() +
geom_smooth() +
labs(x = "Year", y = "Median MLB salary (millions)")

# add salary to batting data
batting <- batting %>%
  filter(yearID >= 1985) %>%
  left_join(select(Salaries, playerID, yearID, teamID, salary),
            by = c("playerID", "yearID", "teamID"))

str(batting)

# Average salaries by teams, over years
# Some franchises are multiply named, so add a new variable
# 'franchise' to the Salaries data as a lookup table
franchise <- c("ANA" = "LAA", "ARI" = "ARI", "ATL" = "ATL",
               "BAL" = "BAL", "BOS" = "BOS", "CAL" = "LAA",
               "CHA" = "CHA", "CHN" = "CHN", "CIN" = "CIN",
               "CLE" = "CLE", "COL" = "COL", "DET" = "DET",
               "FLO" = "MIA", "HOU" = "HOU", "KCA" = "KCA",
               "LAA" = "LAA", "LAN" = "LAN", "MIA" = "MIA",
               "ML1" = "ML1", "MIN" = "MIN", "ML4" = "ML1",
               "MON" = "WAS", "NYA" = "NYA", "NYM" = "NYN",
               "NYN" = "NYN", "OAK" = "OAK", "PHI" = "PHI",
               "PIT" = "PIT", "SD1" = "SD1", "SEA" = "SEA",
               "SFG" = "SFN", "SFN" = "SFN", "SLN" = "SLN",
               "TBA" = "TBA", "TEX" = "TEX", "TOR" = "TOR",
               "WAS" = "WAS")

Salaries$franchise <- unname(franchise[Salaries$teamID])

# Average salaries annual salaries by team, in millions USD
avg_team_salaries <- Salaries %>%
  group_by(yearID, franchise, lgID) %>%
  summarise(salary = mean(salary)/1e6) %>%
  filter(!is.na(salary))
```r
# Spaghetti plot of team salary over time by team
# Yankees have largest average team salary since 2003
ggplot(avg_team_salaries,
       aes(x = yearID, y = salary, group = factor(franchise))) +
       geom_path() +
       labs(x = "Year", y = "Average team salary (millions USD)")
```

## Schools table

### Description

Information on schools players attended, by school

### Usage

```r
data(Schools)
```

### Format

A data frame with 1207 observations on the following 5 variables.

- `schoolID` school ID code
- `name_full` school name
- `city` city where school is located
- `state` state where school’s city is located
- `country` country where school is located

### Source


### Examples

```r
require("dplyr")

# How many different schools are listed in each state?
table(Schools$state)

# How many different schools are listed in each country?
table(Schools$country)

# Top 20 schools
schoolInfo <- Schools %>% select(-country)
```
schoolCount <- CollegePlaying %>%
group_by(schoolID) %>%
  summarise(players = length(schoolID)) %>%
  left_join(schoolInfo, by = "schoolID") %>%
  arrange(desc(players))
head(schoolCount, 20)

# sum counts by state
schoolStates <- schoolCount %>%
  group_by(state) %>%
  summarise(players = sum(players),
            schools = length(state))
str(schoolStates)
summary(schoolStates)

## Not run:
if(require(zipcode)) {
  # in lieu of more precise geocoding via schoolName,
  # find lat/long of Schools from zipcode file
  zips <- zipcode %>%
    group_by(city, state) %>%
    summarise(latitude=mean(latitude),
              longitude=mean(longitude))
names(zips)[1:2] <- c("city", "state")
str(zips)

  # merge lat/long from zips
  schoolsXY <- merge(Schools, zips, by=c("city", "state"), all.x=TRUE)
  str(schoolsXY)

  # plot school locations
  with(subset(schoolsXY, schoolState != 'HI'),
    plot(jitter(longitude), jitter(latitude))
  )
}
## End(Not run)

---

### SeriesPost table

#### Description
Post season series information

#### Usage
```r
data(SeriesPost)
```
Format

A data frame with 334 observations on the following 9 variables.

- `yearID`  Year
- `round`  Level of playoffs
- `teamIDwinner`  Team ID of the team that won the series; a factor
- `lgIDwinner`  League ID of the team that won the series; a factor with levels AL NL
- `teamIDloser`  Team ID of the team that lost the series; a factor
- `lgIDloser`  League ID of the team that lost the series; a factor with levels AL NL
- `wins`  Wins by team that won the series
- `losses`  Losses by team that won the series
- `ties`  Tie games

Source


Examples

data(SeriesPost)

# How many times has each team won the World Series?

# Notes:
# - the SeriesPost table includes an identifier for the
#   team (teamID), but not the franchise (e.g. the Brooklyn Dodgers
#   [BRO] and Los Angeles Dodgers [LAN] are counted separately)
# - the World Series was first played in 1903, but the
#   Lahman data tables have the final round of the earlier
#   playoffs labelled "WS", so it is necessary to
#   filter the SeriesPost table to exclude years prior to 1903.

# using the dplyr data manipulation package
library("dplyr")
library("tidyr")
library("ggplot2")

## WS winners, arranged in descending order of titles won
ws_winner_table <- SeriesPost %>%
  filter(yearID > "1902", round == "WS") %>%
  group_by(teamIDwinner) %>%
  summarise(wincount = n()) %>%
  arrange(desc(wincount))
ws_winner_table

## Expanded form of World Series team data in modern era
Teams table

```
ws <- SeriesPost %>%
  filter(yearID >= 1903 & round == "WS") %>%
  select(-ties, -round) %>%
  mutate(lgIDloser = droplevels(lgIDloser),
         lgIDwinner = droplevels(lgIDwinner))

# Bar chart of length of series (# games played)
# 1903, 1919 and 1921 had eight games
ggplot(ws, aes(x = wins + losses)) +
  geom_bar(fill = "dodgerblue") +
  labs(x = "Number of games", y = "Frequency")

# Last year the Cubs appeared in the WS
ws %>%
  filter(teamIDwinner == "CHN" | teamIDloser == "CHN") %>%
  summarise(max(yearID))

# Dot chart of number of WS appearances by teamID
ws %>%
  gather(wl, team, teamIDwinner, teamIDloser) %>%
  count(team) %>%
  arrange(desc(n)) %>%
  ggplot(. , aes(x = reorder(team, n), y = n)) +
  theme_bw() +
  geom_point(size = 3, color = "dodgerblue") +
  geom_segment(aes(xend = reorder(team, n), yend = 0),
               linetype = "dotted", color = "dodgerblue",
               size = 1) +
  labs(x = NULL, y = "Number of WS appearances") +
  scale_y_continuous(expand = c(0, 0), limits = c(0, 42)) +
  coord_flip() +
  theme(axis.text.y = element_text(size = rel(0.8)),
        axis.ticks.y = element_blank())

# Initial year of each round of championship series in modern era
SeriesPost %>%
  filter(yearID >= 1903) %>%  # modern WS started in 1903
  group_by(round) %>%
  summarise(first_year = min(yearID)) %>%
  arrange(first_year)

# Ditto, but with more information about each series played
SeriesPost %>%
  filter(yearID >= 1903) %>%
  group_by(round) %>%
  arrange(yearID) %>%
  do(head(., 1)) %>%
  select(-lgIDwinner, -lgIDloser) %>%
  arrange(yearID, round)
```
Teams

Description
Yearly statistics and standings for teams

Usage
data(Teams)

Format
A data frame with 2895 observations on the following 48 variables.

  yearID Year
  lgID League; a factor with levels AA AL FL NL PL UA
  teamID Team; a factor
  franchID Franchise (links to TeamsFranchises table)
  divID Team's division; a factor with levels C E W
  Rank Position in final standings
  G Games played
  Ghome Games played at home
  W Wins
  L Losses
  DivWin Division Winner (Y or N)
  WCWin Wild Card Winner (Y or N)
  LgWin League Champion(Y or N)
  WSWin World Series Winner (Y or N)
  R Runs scored
  AB At bats
  H Hits by batters
  X2B Doubles
  X3B Triples
  HR Homeruns by batters
  BB Walks by batters
  SO Strikeouts by batters
  SB Stolen bases
  CS Caught stealing
  HBP Batters hit by pitch
  SF Sacrifice flies
  RA Opponents runs scored
  ER Earned runs allowed
  ERA Earned run average
CG Complete games
SHO Shutouts
SV Saves
IPouts Outs Pitched (innings pitched x 3)
HA Hits allowed
HRA Homeruns allowed
BBA Walks allowed
SOA Strikeouts by pitchers
E Errors
DP Double Plays
FP Fielding percentage
name Team’s full name
park Name of team’s home ballpark
attendance Home attendance total
BPF Three-year park factor for batters
PPF Three-year park factor for pitchers
teamIDBR Team ID used by Baseball Reference website
teamIDlahman45 Team ID used in Lahman database version 4.5
teamIDretro Team ID used by Retrosheet

Details
Variables X2B and X3B are named 2B and 3B in the original database

Source

Examples

data(Teams)
library("dplyr")
library("tidyr")

# Add some selected measures to the Teams data frame
# Restrict to AL and NL in modern era
teams <- Teams %>%
  filter(yearID >= 1901 & lgID %in% c("AL", "NL")) %>%
  group_by(yearID, teamID) %>%
  mutate(TB = H + X2B + 2 * X3B + 3 * HR,
         WinPct = W/G,
         rpg = R/G,
         hrpg = HR/G,
         tbpg = TB/G,
# Function to create a ggplot by year for selected team stats
# Both arguments are character strings
yrPlot <- function(year, label) {
  require("ggplot2")
  ggplot(teams, aes_string(x = "yearID", y = year)) +
  geom_point(size = 0.5) +
  geom_smooth(method="loess") +
  labs(x = "Year", y = paste(label, "per game"))
}

## Run scoring in the modern era by year
yrPlot("rpg", "Runs")

## Home runs per game by year
yrPlot("hrpg", "Home runs")

## Total bases per game by year
yrPlot("tbpg", "Total bases")

## Strikeouts per game by year
yrPlot("kpg", "Strikeouts")

## Plot win percentage vs. run differential (R - RA)
ggplot(teams, aes(x = R - RA, y = WinPct)) +
  geom_point(size = 0.5) +
  geom_smooth(method="loess") +
  geom_hline(yintercept = 0.5, color = "orange") +
  geom_vline(xintercept = 0, color = "orange") +
  labs(x = "Run differential", y = "Win percentage")

## Plot attendance vs. win percentage by league, post-1980
teams %>% filter(yearID >= 1980) %>%
ggplot(., aes(x = WinPct, y = attendance/1000)) +
  geom_point(size = 0.5) +
  geom_smooth(method="loess", se = FALSE) +
  facet_wrap(~ lgID) +
  labs(x = "Win percentage", y = "Attendance (1000s)")

## Teams with over 4 million attendance in a season
teams %>%
  filter(attendance >= 4e6) %>%
  select(yearID, lgID, teamID, rank, attendance) %>%
  arrange(desc(attendance))

## Average season HRs by park, post-1980
teams %>%
  filter(yearID >= 1980) %>%
  group_by(park) %>%

kpg = S0/G,
k2bb = S0/BB,
whip = 3 * (H + BB/IPouts)
summarise(meanHRpg = mean((HR + HRA)/Ghome), nyears = n()) %>%
filter(nyears >= 10) %>%
arrange(desc(meanHRpg)) %>%
head(., 10)

## Home runs per game at Fenway Park and Wrigley Field,
## the two oldest MLB parks, by year. Fenway opened in 1912.
teams %>%
filter(yearID >= 1912 & teamID %in% c("BOS", "CHN")) %>%
mutate(hrpg = (HR + HRA)/Ghome) %>%
ggplot(. , aes(x = yearID, y = hrpg, color = teamID)) +
  geom_line(size = 1) +
  labs(x = "Year", y = "Home runs per game", color = "Team") +
  scale_color_manual(values = c("red", "blue"))

## Ditto for total strikeouts per game
teams %>%
filter(yearID >= 1912 & teamID %in% c("BOS", "CHN")) %>%
mutate(kpg = (SO + SOA)/Ghome) %>%
ggplot(. , aes(x = yearID, y = kpg, color = teamID)) +
  geom_line(size = 1) +
  geom_point() +
  labs(x = "Year", y = "Strikeouts per game", color = "Team") +
  scale_color_manual(values = c("red", "blue"))

## Not run:
if(require(googleVis)) {
motion1 <- gvisMotionChart(as.data.frame(teams),
  idvar="teamID", timevar="yearID", chartid="gvisTeams",
  options=list(width=700, height=600))
plot(motion1)
# print(motion1, file="gvisteamssal.html")

# Merge with avg salary for years where salary is available

teamssal <- Salaries %>%
group_by(yearID, teamID) %>%
summarise(Salary = sum(salary, na.rm = TRUE)) %>%
select(yearID, teamID, Salary)

teamsSal <- teams %>%
filter(yearID >= 1985) %>%
left_join(teamssal, by = c("yearID", "teamID")) %>%
select(yearID, teamID, attendance, Salary, WinPct) %>%
as.data.frame();

motion2 <- gvisMotionChart(teamsSal, idvar="teamID", timevar="yearID",
  xvar="attendance", yvar="salary", sizevar="WinPct",
  chartid="gvisteamssal", options=list(width=700, height=600))
plot(motion2)
# print(motion2, file="gvisteamssal.html")
TeamFranchises

Description

Information about team franchises

Usage

data(TeamFranchises)

Format

A data frame with 120 observations on the following 4 variables.

franchid  Franchise ID; a factor
franchName  Franchise name
active  Whether team is currently active (Y or N)
NAassoc  ID of National Association team franchise played as

Source


Examples

data(TeamFranchises)

# Which of the active Major League Baseball teams had a National Association predecessor?

# Notes:
# - the National Association was founded in 1871, and continued through the
# 1875 season. In 1876, six clubs from the National Association and two other
# independent clubs formed the National League, which exists to this day.
# - the 'active' field has "NA" for the National Association franchises
# - where appropriate, the 'NAassoc' field has the 'franchid' of the successor National League team

# using the dplyr data manipulation package
library("dplyr")

NatAssoc_active_table <- TeamFranchises %>%
  filter(active == "Y") %>%
  filter(!is.na(NAassoc))
TeamsHalf

### Description

Split season data for teams

### Usage

```r
data(TeamsHalf)
```

### Format

A data frame with 52 observations on the following 10 variables.

- **yearID** Year
- **lgID** League; a factor with levels AL NL
- **teamID** Team; a factor
- **Half** First or second half of season
- **divID** Division
- **DivWin** Won Division (Y or N)
- **Rank** Team’s position in standings for the half
- **G** Games played
- **W** Wins
- **L** Losses

### Source

Examples

# 1981 season team data split into half seasons
data(TeamsHalf)
library("dplyr")

# List standings with winning percentages by
# season half, league and division
TeamsHalf %>%
group_by(Half, lgID, divID) %>%
mutate(WinPct = round(W/G, 3)) %>%
arrange(Half, lgID, divID, Rank) %>%
select(Half, lgID, divID, Rank, teamID, WinPct)
Index

*Topic datasets
   AllstarFull, 5
   Appearances, 6
   AwardsManagers, 8
   AwardsPlayers, 10
   AwardsShareManagers, 11
   AwardsSharePlayers, 12
   Batting, 14
   battingLabels, 18
   BattingPost, 19
   CollegePlaying, 22
   Fielding, 23
   FieldingOF, 25
   FieldingPost, 27
   HallOfFame, 28
   LahmanData, 33
   Managers, 34
   ManagersHalf, 38
   Master, 39
   Parks, 41
   People, 42
   Pitching, 44
   PitchingPost, 47
   Salaries, 50
   Schools, 53
   SeriesPost, 54
   Teams, 57
   TeamsFranchises, 61
   TeamHalf, 62

*Topic manip
   battingStats, 21
   Label, 32
   playerInfo, 49

AllstarFull, 4, 5
Appearances, 4, 6
AwardsManagers, 4, 8
AwardsPlayers, 4, 10
AwardsShareManagers, 4, 11
AwardsSharePlayers, 4, 12

Baseball, 15
baseball, 15
Batting, 3, 14, 18, 21, 22, 39, 42
battingLabels, 4, 18, 32
BattingPost, 4, 19, 22
battingStats, 15, 21

CollegePlaying, 4, 22

Fielding, 3, 18, 23
fieldingLabels, 4, 32
fieldingLabels (battingLabels), 18
FieldingOF, 4, 25
FieldingPost, 4, 27

grep, 50

HallOfFame, 4, 28

Label, 18, 32
Lahman (Lahman-package), 3
Lahman-package, 3
LahmanData, 33

Managers, 4, 34
ManagersHalf, 4, 38
Master, 39

Parks, 41
People, 3, 29, 42, 49
Pitching, 3, 18, 39, 42, 44
pitchingLabels, 4, 32
pitchingLabels (battingLabels), 18
PitchingPost, 4, 47
playerInfo, 49

Salaries, 4, 50
Schools, 4, 53
SeriesPost, 4, 54

teamInfo (playerInfo), 49
Teams, 4, 41, 49, 56
TeamsFranchises, 4, 57, 61
TeamsHalf, 4, 62