Package ‘Lahman’

September 15, 2015

Type Package

Title Sean Lahman's Baseball Database

Version 4.0-1

Date 2015-09-04

Author Michael Friendly [aut], Dennis Murphy [ctb], Martin Monkman [ctb], Chris Dalzell [cre, ctb]

Maintainer Chris Dalzell <cdalzell@gmail.com>

Description Provides the tables from Sean Lahman's 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 2014, as recorded in the 2015 version of the database.

Depends R (>= 2.10)

Suggests lattice, ggplot2, googleVis, data.table, vcd, plyr, reshape2, zipcode

License GPL

URL http://lahman.r-forge.r-project.org/

LazyLoad yes

LazyData yes

Repository CRAN

Repository/R-Forge/Project lahman

Repository/R-Forge/Revision 39

Repository/R-Forge/DateTimeStamp 2013-06-01 03:33:30

Date/Publication 2015-09-15 08:35:24

NeedsCompilation no

R topics documented:

`Lahman-package` *Sean Lahman's Baseball Database*

Description

This database contains pitching, hitting, and fielding statistics for Major League Baseball from 1871 through 2012. 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.

Details

Package:	Lahman
Type:	Package
Version:	2.0-3
Date:	2013-05-29
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 `playerID`s are linked to names and birthdates in the `Master` table. Similar links exist among other tables via analogous `*ID` variables.

The database is comprised of the following main tables:

`Master` 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 Master via `hofID`

HallOfFame	Hall of Fame voting data

Others 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:

battingLabels, pitchingLabels, fieldingLabels

Author(s)

Michael Friendly and Dennis Murphy

Maintainer: Michael Friendly <friendly@yorku.ca>

Source

Lahman, S. (2012) Lahman's Baseball Database, 1871-2012, Main page, http://baseball1.com/statistics/

Lahman, S. (2012) Lahman's Baseball Database, 1871-2012, v. 2012, Comma-delimited version, http://seanlahman.com/files/database/lahman2012-csv.zip

Lahman, S. (2012) Lahman's Baseball Database, 1871-2012, MS Access version, http://seanlahman.com/files/database/lahman2012-ms.zip

AllstarFull *AllstarFull table*

Description

All Star appearances by players

Usage

```
data(AllstarFull)
```

Format

A data frame with 4993 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

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, `http://baseball1.com/statistics/`

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, startingPos[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 two teamID typos: can you spot them?)
subset(AllstarFull, teamID %in% rare)
```

Appearances *Appearances table*

Description

Data on player appearances

Usage

```
data(Appearances)
```

Format

A data frame with 99466 observations on the following 21 variables.

`yearID` Year

`teamID` Team; a factor

`lgID` League; a factor with levels `AA` `AL` `FL` `NL` `PL` `UA`

`playerID` Player ID code

`G_all` Total games played

`GS` Games started

`G_batting` Games in which player batted

`G_defense` Games in which player appeared on defense

`G_p` Games as pitcher

`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_lf` 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

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, http://baseball1.com/statistics/

Examples

```
data(Appearances)

# some test cases
# Henry Aaron spent the last two years of his career as DH in Milwaukee
subset(Appearances, playerID == 'aaronha01')
# Herb Washington, strictly a pinch runner for Oakland in 1974-5
subset(Appearances, playerID == 'washihe01')
subset(Appearances, playerID == 'thomeji01')
subset(Appearances, playerID == 'hairsje02')

# Appearances for the 1984 Cleveland Indians
subset(Appearances, teamID == "CLE" & yearID == 1984)


if (require(reshape2) & require(plyr)) {
# Appearances for Pete Rose during his career:
prose <- subset(Appearances, playerID == "rosepe01")


# What was Pete Rose's primary position each year
# of his career?

prose_melt <- melt(prose, id = c("yearID", "teamID"),
                          measure = 9:17)
# Split out the position from variable
prose_melt <- cbind(prose_melt, colsplit(prose_melt$variable,
                                         "_", names = c("G", "pos")))

# Two grouping variables because of an in-season trade in 1984
primary_pos <- ddply(prose_melt, .(yearID, teamID), summarise,
                         top_pos = pos[which.max(value)],
                         games = max(value))
primary_pos

# Most pitcher appearances each year since 1950
ddply(subset(Appearances, yearID >= 1950), .(yearID), summarise,
                              maxPitcher = playerID[which.max(G_p)],
                              maxAppear = max(G_p))

# Individuals who have played all 162 games since 1961
all162 <- ddply(subset(Appearances, yearID > 1960), .(yearID), summarise,
                      allGamers = playerID[G_all == 162])
# Number of all-gamers by year
table(all162$yearID)
}
```

AwardsManagers *AwardsManagers table*

Description

Award information for managers awards

Usage

```
data(AwardsManagers)
```

Format

A data frame with 171 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

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, http://baseball1.com/statistics/

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 LaRussa's manager of the year awards
subset(AwardsManagers, playerID == "larusto01")
```

AwardsPlayers *AwardsPlayers table*

Description

Award information for players awards

Usage

```
data(AwardsPlayers)
```

Format

A data frame with 6026 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

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, http://baseball1.com/statistics/

Examples

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

Description

Award voting for managers awards

Usage

```
data(AwardsShareManagers)
```

Format

A data frame with 401 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 numner of points possible

votesFirst Number of first place votes

Source

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, http://baseball1.com/statistics/

Examples

```
# Voting for the BBWAA Manager of the Year award by year and league

require(plyr)

# Sort in decreasing order of points by year and league
MOYsort <- ddply(AwardsShareManagers, .(yearID, lgID), arrange, desc(pointsWon))
```

```
# Any unanimous winners?
subset(AwardsShareManagers, pointsWon == pointsMax)

# OK, how about highest proportion of possible points?
AwardsShareManagers[with(AwardsShareManagers, which.max(pointsWon/pointsMax)), ]

# Bobby Cox's MOY vote tallies
subset(AwardsShareManagers, playerID == "coxbo01")
```

AwardsSharePlayers *AwardsSharePlayers table*

Description

Award voting for managers awards

Usage

```
data(AwardsSharePlayers)
```

Format

A data frame with 6617 observations on the following 7 variables.

awardID name of award votes were received for

yearID Year

lgID League; a factor with levels AL ML NL

playerID Player ID code

pointsWon Number of points received

pointsMax Maximum numner of points possible

votesFirst Number of first place votes

Source

Lahman, S. (2014) Lahman's Baseball Database, 1871-2013, 2014 version, http://baseball1.com/statistics/

Examples

```
# Vote tallies for post-season player awards

require(plyr)

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

```
# Sort the votes for the Cy Young award in decreasing order.
# For the first few years, the award went to the best pitcher
# in both leagues.

cyvotes <- ddply(subset(AwardsSharePlayers, awardID == "Cy Young"),
                 .(yearID, lgID), arrange, desc(pointsWon))

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

# top three votegetters each year by league

cya_top3 <- ddply(cyvotes, .(yearID, lgID), function(d) head(d, 3))

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

# Top five pitchers with most top 3 vote tallies in CYA
head(with(cya_top3, rev(sort(table(playerID)))), 5)

# Ditto for MVP awards

MVP <- subset(AwardsSharePlayers, awardID == "MVP")
MVP_top3 <- ddply(MVP, .(yearID, lgID),
                  function(d) head(arrange(d, desc(pointsWon)), 3))
head(with(MVP_top3, rev(sort(table(playerID)))), 5)
```

`Batting` *Batting table*

Description

Batting table - batting statistics

Usage

```
data(Batting)
```

Format

A data frame with 99846 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`

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

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, http://baseball1.com/statistics/

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('plyr')

# calculate batting average and other stats
batting <- battingStats()

# add salary to Batting data; need to match by player, year and team
batting <- merge(batting,
                 Salaries[,c("playerID", "yearID", "teamID", "salary")],
                 by=c(”playerID”, ”yearID”, ”teamID”), all.x=TRUE)

# Add name, age and bat hand information:
masterInfo <- Master[, c('playerID', 'birthYear', 'birthMonth',
                          'nameLast', 'nameFirst', 'bats')]
batting <- merge(batting, masterInfo, all.x = TRUE)
batting$age <- with(batting, yearID - birthYear -
                             ifelse(birthMonth < 10, 0, 1))

batting <- arrange(batting, playerID, yearID, stint)

## Generate a ggplot similar to the NYT graph in the story about Ted
## Williams and the last .400 MLB season
# http://www.nytimes.com/interactive/2011/09/18/sports/baseball/WILLIAMS-GRAPHIC.html

# 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 <- subset(batting, 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
topHitters <- ddply(eligibleHitters, .(yearID), subset, (BA == max(BA))|BA > .400)

# Create a factor variable to distinguish the .400 hitters
topHitters$ba400 <- with(topHitters, BA >= 0.400)

# Sub-data frame for the .400 hitters plus the outliers after 1950
# (averages above .380) - used to produce labels in the plot below
bignames <- rbind(subset(topHitters, ba400),
                  subset(topHitters, yearID > 1950 & BA > 0.380))
# Cut to the relevant set of variables
bignames <- subset(bignames, select = c('playerID', 'yearID', 'nameLast',
                                        'nameFirst', 'BA'))

# Ditto for the original data frame
topHitters <- subset(topHitters, select = c('playerID', 'yearID', 'BA', 'ba400'))

# Positional offsets to spread out certain labels
#                     NL TC JJ TC GS TC RH GS HH RH RH BT TW TW  RC GB TG
bignames$xoffset <- c(0, 0, 0, 0, 0, 0, 0, 0, -8, 0, 3, 3, 0, 0, -2, 0, 0)
bignames$yoffset <- c(0, 0, -0.003, 0, 0, 0, 0, 0, -0.004, 0, 0, 0, 0, 0, -0.003, 0, 0)  +  0.002

require('ggplot2')
ggplot(topHitters, aes(x = yearID, y = BA)) +
    geom_point(aes(colour = ba400), size = 2.5) +
    geom_hline(yintercept = 0.400, size = 1) +
    geom_text(data = bignames, aes(x = yearID + xoffset, y = BA + yoffset,
                                   label = nameLast), size = 3) +
    scale_colour_manual(values = c('FALSE' = 'black', 'TRUE' = 'red')) +
    ylim(0.330, 0.430) +
    xlab('Year') +
    scale_y_continuous('Batting average',
                       breaks = seq(0.34, 0.42, by = 0.02),
                       labels = c('.340', '.360', '.380', '.400', '.420')) +
    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 <- ddply(Batting, .(yearID), summarise,
                       HomeRuns = sum(as.numeric(HR), na.rm=TRUE),
                       Games = sum(as.numeric(G), na.rm=TRUE)
                       )

plot(HomeRuns ~ yearID, data=subset(totalHR, yearID<=1918))
# take games into account?
plot(HomeRuns/Games ~ yearID, data=subset(totalHR, yearID<=1918))

# long term trend?
plot(HomeRuns ~ yearID, data=totalHR)
plot(HomeRuns/Games ~ yearID, data=totalHR)
```

battingLabels *Variable Labels*

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

```
data(battingLabels)
str(battingLabels)

require(plyr)
# find and plot maximum number of homers per year
batHR <- ddply(subset(Batting, !is.na(HR)), .(yearID),
summarise, max=max(HR))

with(batHR, {
  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)
})
```

BattingPost *BattingPost table*

Description

Post season batting statistics

Usage

```
data(BattingPost)
```

Format

A data frame with 11294 observations on the following 22 variables.

`yearID` Year

`round` Level of playoffs

`playerID` Player ID code

`teamID` Team

`lgID` League; a factor with levels `AA` `AL` `NL`

`G` Games

`AB` At Bats

`R` Runs

`H` Hits

X2B Doubles

X3B Triples

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 Sacrifices

SF Sacrifice flies

GIDP Grounded into double plays

Details

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

Source

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, http://baseball1.com/statistics/

Examples

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

require(plyr)

# Create a sub-data frame for modern World Series play
ws <- subset(BattingPost, round == "WS" & yearID >= 1903)

# Add some derived measures
ws <- mutate(ws, BA = ifelse(AB == 0, 0, round(H/AB, 3)),
                 TB = H + X2B + 2 * X3B + 3 * HR,
                 SA = ifelse(AB == 0,  0, round(TB/AB, 3)),
                 PA = AB + BB + IBB + HBP + SH + SF,
                 OB = H + BB + IBB + HBP,
                 OBP = ifelse(AB == 0, 0, round(OB/PA, 3)) )

# Players with most appearances in the WS:
with(subset(BattingPost, round == "WS"), rev(sort(table(playerID))))[1:10]

# OK, how about someone who is *not* a Yankee?
with(subset(BattingPost, round == "WS" & teamID != "NYA"),
       rev(sort(table(playerID))))[1:10]


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

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

# Hitting stats for the 1946 St. Louis Cardinals, ordered by BA
arrange(subset(ws, teamID == "SLN" & yearID == 1946), desc(BA))

# Babe Ruth's WS profile
subset(ws, playerID == "ruthba01")
```

---

battingStats *Calculcate additional batting statistics*

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

```
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

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

`CollegePlaying` *CollegePlaying table*

Description

Information on schools players attended, by player

Usage

```
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

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, `http://baseball1.com/statistics/`

Examples

```
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 167938 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

`A` Assists

E Errors

DP Double Plays

PB Passed Balls (by catchers)

WP Wild Pitches (by catchers)

SB Opponent Stolen Bases (by catchers)

CS Opponents Caught Stealing (by catchers)

ZR Zone Rating

Source

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, http://baseball1.com/statistics/

Examples

```
data(Fielding)
# Basic fielding data

require(plyr)


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

# Yadier Molina's fielding profile
# PB, WP, SP and CS apply to catchers
subset(Fielding, playerID == "molinya01")

# Pedro Martinez's fielding profile
# Notice what pitchers get away with in this data frame :)
subset(Fielding, 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
luis <- subset(Fielding, playerID == "aparilu01",
                  select = c("G", "PO", "A", "E", "DP"))
colwise(sum)(luis)


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

# League average fielding statistics, 1961-present

fldg <- subset(Fielding, yearID >= 1961 & POS != "DH",
                  select = c("yearID", "lgID", "POS", "InnOuts",
                             "PO", "A", "E"))
lgTotalsF <- ddply(fldg, .(yearID, lgID), numcolwise(sum, na.rm = TRUE))
(lgTotalsF <- mutate(lgTotalsF,
                      fpct = round( (PO + A)/(PO + A + E), 3),
                      OPE = round(InnOuts/E, 3) ))
```

FieldingOF *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

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, http://baseball1.com/statistics/

`FieldingPost` *FieldingPost data*

Description

Post season fielding data

Usage

```
data(FieldingPost)
```

Format

A data frame with 11924 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

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, http://baseball1.com/statistics/

HallOfFame *Hall of Fame Voting Data*

Description

Hall of Fame table. This is comprised of the voting results for all candidates nominated for the Baseball Hall of Fame.

Usage

```
data(HallOfFame)
```

Format

A data frame with 4088 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 `Master` 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`.

Source

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, `http://baseball1.com/statistics/`

Examples

```
## Some examples for  Hall of Fame induction data

data('HallOfFame')
require('plyr')         ## extensive use of plyr for data manipulation
require('ggplot2')
```

```
############################################################
## Some simple queries

# What are the different types of votedBy?
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')


# Function to calculate number of years as HOF candidate, last pct vote, etc.
# for a given player
HOFun <- function(d) {
    nyears <- nrow(d)
    fy <- d[nyears, ]
    lastPct <- with(fy, 100 * round(votes/ballots, 3))
    data.frame(playerID = fy$playerID, nyears, induct = fy$inducted,
               lastPct, lastYear = fy$yearID)
}

playerOutcomesHOF <- ddply(HOFplayers, .(playerID), HOFun)


############################################################
# How many voting years until election?
inducted <- subset(playerOutcomesHOF,induct == 'Y')
table(inducted$nyears)
barplot(table(inducted$nyears), main="Number of voting years until election",
ylab="Number of players", xlab="Years")

# 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:
subset(playerOutcomesHOF, nyears == 1L & induct == 'Y')

# Who took at least ten years on the ballot before induction?
# (Doesn't include Bert Blyleven, who was inducted in 2011.)
subset(playerOutcomesHOF, nyears >= 10L & induct == 'Y')

############################################################
```

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

# (1) Set up the data:
longTimers <- as.character(unlist(subset(playerOutcomesHOF,
                                         nyears >= 10, select = 'playerID')))
HOFlt <- subset(HallOfFame, playerID %in% longTimers & votedBy == 'BBWAA')
HOFlt <- ddply(HOFlt, .(playerID), mutate,
                  elected = ifelse(any(inducted == 'Y'),"Elected", "Not elected"),
                  pct = 100 * round(votes/ballots, 3))

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

# Note: All but one of the players whose maximum voting percentage
# was over 60% and was not elected by the BBWAA has eventually been inducted
# into the HOF. Red Ruffing was elected in a 1967 runoff election while
# the others have been voted in by the Veterans Committee. The lone
# exception is Gil Hodges; his profile is the one that flatlines around 60%
# for several years in the late 70s and early 80s.
```

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

```
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

```
require(plyr)
# find and plot maximum number of homers per year
batHR <- ddply(subset(Batting, !is.na(HR)), .(yearID),
summarise, max=max(HR))

with(batHR, {
  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)
})
```

LahmanData *Lahman Datasets*

Description

This dataset gives a consise description of the data files in the Lahman package. It may be useful for computing on the various files.

Usage

```
data(LahmanData)
```

Format

A data frame with 24 observations on the following 5 variables.

`file` name of dataset

`class` class of dataset

`nobs` number of observations

`nvar` number of variables

`title` dataset title

Details

This dataset is generated using `vcdExtra::datasets(package="Lahman")` with some post-processing.

Examples

```
data(LahmanData)

# find ID variables in the datasets
IDvars <- lapply(LahmanData[,"file"], function(x) grep('.*ID$', colnames(get(x)), value=TRUE))
names(IDvars) <- LahmanData[,"file"]
str(IDvars)
# vector of unique ID variables
unique(unlist(IDvars))

# which datasets have playerID?
names(which(sapply(IDvars, function(x) "playerID" %in% x)))

################################################
# Visualize relations among datasets via an MDS
################################################
# jaccard distance between two sets; assure positivity
jaccard <- function(A, B) {
max(1 - length(intersect(A,B)) / length(union(A,B)), .00001)
}

distmat <- function(vars, FUN=jaccard) {
nv <- length(vars)
d <- matrix(0, nv, nv, dimnames=list(names(vars), names(vars)))
for(i in 1:nv) {
for (j in 1:nv) {
if (i != j) d[i,j] <- FUN(vars[[i]], vars[[j]])
}
}
d
}

# do an MDS on distances
distID <- distmat(IDvars)
config <- cmdscale(distID)

pos=rep(1:4, length=nrow(config))
plot(config[,1], config[,2], xlab = "", ylab = "", asp = 1, axes=FALSE,
main="MDS of ID variable distances of Lahman tables")
abline(h=0, v=0, col="gray80")
text(config[,1], config[,2], rownames(config), cex = 0.75, pos=pos, xpd=NA)
```

Managers *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 3370 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

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

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, http://baseball1.com/statistics/

Examples

```
####################################
# Basic career summaries by manager
####################################

library('plyr')
mgrsumm <- function(d) {
    df <- data.frame(with(d,
              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)))
    df
}

mgrTotals <- ddply(Managers, .(playerID), summarise,
                  nyear = length(unique(yearID)),
                  yearBegin = min(yearID),
                  yearEnd = max(yearID),
                  nTeams = length(unique(teamID)),
                  nfirst = sum(rank == 1L),
                  games = sum(W + L),
                  W = sum(W),
                  L = sum(L),
                  WinPct = round(sum(W)/sum(W + L), 3))
mgrTotals <- merge(mgrTotals,
                   subset(Master, !is.na(playerID),
                          select = c('playerID', 'nameLast', 'nameFirst')),
                   by = 'playerID')

##########################
# Some basic queries
##########################

# Top 20 managers in terms of years of service:
head(arrange(mgrTotals, -nyear), 20)

# Top 20 winningest managers (500 games minimum)
head(arrange(subset(mgrTotals, games >= 500), -WinPct), 20)

# Hmm. Most of these are 19th century managers.
# How about the modern era?
head(arrange(subset(mgrTotals, yearBegin >= 1900 & games >= 500), -WinPct), 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
head(arrange(subset(mgrTotals, yearBegin >= 1900 & games >= 500),
                -round(nfirst/nyear, 3)), 10)

# How about pre-1969?
head(arrange(subset(mgrTotals,
                    yearBegin >= 1900 & yearEnd <= 1969 & games >= 500),
                    -round(nfirst/nyear, 3)), 10)

##############################################
# 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')

# Who managed more than 4000 games?
subset(mgrTotals, games >= 4000)
# Connie Mack had an advantage: he owned the Philadelphia A's :)

# Table of Tony LaRussa's team finishes:
with(subset(Managers, playerID == 'larusto01'), table(rank))

# To include zero frequencies, one alternative is the tabulate() function:
```

```
with(subset(Managers, playerID == 'larusto01'), tabulate(rank, 7))


##############################################
# 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 with at least 8 years
# of experience in the divisional era (>= 1969):

divMgr <- subset(mgrTotals, yearBegin >= 1969 & nyear >= 8)

# Response is the number of titles
ggplot(divMgr, aes(x = nyear, y = nfirst)) +
    geom_point(position = position_jitter(w = 0.2)) +
    labs(x = 'Number of years', y = 'Number of divisional titles') +
    geom_smooth()

# Response is the proportion of titles
ggplot(divMgr, aes(x = nyear, y = round(nfirst/nyear, 3))) +
    geom_point(position = position_jitter(w = 0.2)) +
    labs(x = 'Number of years', y = 'Proportion of divisional titles') +
    geom_smooth()
```

`ManagersHalf` *ManagersHalf table*

Description

Split season data for managers

Usage

```
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

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, `http://baseball1.com/statistics/`

`Master` *Master table*

Description

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 18589 observations on the following 26 variables.

`playerID` A unique code asssigned 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/

bbrefID ID used by Baseball Reference website, http://www.baseball-reference.com/

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

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, http://baseball1.com/statistics/

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)
```

Pitching *Pitching table*

Description

Pitching table

Usage

```
data(Pitching)
```

Format

A data frame with 43330 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

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

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, http://baseball1.com/statistics/

Examples

```
# Pitching data

require(plyr)

###################################
# 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 <- subset(Pitching, yearID >= 1901 & lgID %in% c("AL", "NL"))[, -(28:30)]

# Approximate missing BAOpp values (most common remaining missing value)
pitching$BAOpp <- with(pitching, round(H/(BFP - BB - HBP), 3))
# Compute WHIP (hits + walks per inning pitched -- lower is better)
pitching <- mutate(pitching,
                   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 <- subset(pitching, yearID == 1993 & teamID == "TOR")
arrange(tor93, ERA)

# Career pitching statistics, Greg Maddux
subset(pitching, playerID == "maddugr01")

# Best ERAs for starting pitchers post WWII
postwar <- subset(pitching, yearID >= 1946 & IPouts >= 600)
head(arrange(postwar, ERA), 10)

# Best K/BB ratios post-1955 among starters (excludes intentional walks)
post55 <- subset(pitching, yearID >= 1955 & IPouts >= 600)
post55 <- mutate(post55, KperBB = SO/(BB - IBB))
head(arrange(post55, desc(KperBB)), 10)

# Best K/BB ratios among relievers post-1950 (min. 20 saves)
head(arrange(subset(pitching, yearID >= 1950 & SV >= 20), desc(KperBB)), 10)

################################################
# Winningest pitchers in each league each year:
################################################

# Add name & throws information:
masterInfo <- Master[, c('playerID',
                         'nameLast', 'nameFirst', 'throws')]
pitching <- merge(pitching, masterInfo, all.x=TRUE)

wp <- ddply(pitching, .(yearID, lgID), subset, W == max(W),
         select = c("playerID", "teamID", "W", "throws"))

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

# an eye-catching, but naive, specious graph

require('ggplot2')
# compare loess smooth with quadratic fit
ggplot(wp, aes(x = yearID, y = W)) +
    geom_point(aes(colour = throws, shape=lgID), size = 2) +
    geom_smooth(method="loess", size=1.5, color="blue") +
    geom_smooth(method = "lm", se=FALSE, color="black", formula = y ~ poly(x,2)) +
    ylab("Maximum Wins") + xlab("Year") +
    ggtitle("Why can't pitchers win 30+ games any more?")
```

PitchingPost *PitchingPost table*

Description

Post season pitching statistics

Usage

```
data(PitchingPost)
```

Format

A data frame with 4945 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

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, `http://baseball1.com/statistics/`

playerInfo *Lookup Information for Players and Teams*

Description

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

Usage

```
playerInfo(playerID, nameFirst, nameLast, data = Lahman::Master, 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)
```

Format

A data frame with 23956 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

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, http://baseball1.com/statistics/

Examples

```
# what years are included?
summary(Salaries$yearID)
# how many players included each year?
table(Salaries$yearID)

# Team salary data

require(plyr)

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

# Arrange in decreasing order within year and league:
teamSalaries <- ddply(teamSalaries, .(yearID, lgID), arrange, desc(Salary))
```

```
#######################################
# Highest paid players each year:
maxSal <- ddply(Salaries, .(yearID), subset, salary == max(salary))
names <- apply(t(sapply(maxSal$playerID, playerInfo))[,2:3], 2, paste)
maxSal <- cbind(maxSal, names)
maxSal
plot(salary/100000 ~ yearID, data=maxSal, type='b', ylab='Salary (100,000$)')
# see the whole distribution
boxplot(salary/100000 ~ yearID, data=Salaries, col="lightblue")

# add salary to Batting data
batting <- merge(Batting,
                 Salaries[,c("playerID", "yearID", "teamID", "salary")],
                 by=c("playerID", "yearID", "teamID"), all.x=TRUE)
str(batting)

#######################################
# Average salaries by teams, over years
#######################################

require(plyr)
avesal <- ddply(Salaries, .(yearID, teamID, lgID), summarise,
salary= mean(salary)/100000)

# remove infrequent teams
tcount <- table(avesal$teamID)
avesal <- subset(avesal, avesal$teamID %in% names(tcount)[tcount>=15], drop=TRUE)
avesal$teamID <- factor(avesal$teamID, levels=names(tcount)[tcount>=15])

require(lattice)
xyplot(salary ~ yearID | teamID, data=avesal, ylab="Salary (100,000$)")
```

Schools *Schools table*

Description

Information on schools players attended, by school

Usage

```
data(Schools)
```

Format

A data frame with 749 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

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, http://baseball1.com/statistics/

Examples

```
require(plyr)

# 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[, c("schoolID", "name_full", "city", "state")]

schoolCount <- ddply(CollegePlaying, .(schoolID), summarise,
                       players = length(schoolID))
schoolCount <- merge(schoolCount, schoolInfo, by="schoolID", all.x=TRUE)

# Arrange in decreasing order:
schoolCount <- arrange(schoolCount, desc(players))
head(schoolCount, 20)

# sum counts by state
schoolStates <- ddply(schoolCount, .(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 <- ddply(zipcode, .(city, state), summarize,
                latitude=mean(latitude), longitude=mean(longitude))
  colnames(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 *SeriesPost table*

Description

Post season series information

Usage

```
data(SeriesPost)
```

Format

A data frame with 298 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

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, http://baseball1.com/statistics/

`Teams` *Teams table*

Description

Yearly statistics and standings for teams

Usage

```
data(Teams)
```

Format

A data frame with 2775 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

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, http://baseball1.com/statistics/

Examples

```
data(Teams)

# subset on a few variables
teams <- subset(Teams, lgID %in% c("AL", "NL"))
teams <- subset(teams, yearID>1900)
# drop some variables
teams <- subset(teams, select=-c(Ghome,divID,DivWin:WSWin,name,park,teamIDBR:teamIDretro))
teams <- subset(teams, select=-c(HBP,CS,BPF,PPF))
```

```
# subset to remove infrequent teams
tcount <- table(teams$teamID)
teams <- subset(teams, teams$teamID %in% names(tcount)[tcount>15], drop=TRUE)
teams$teamID <- factor(teams$teamID, levels=names(tcount)[tcount>15])

# relevel lgID
teams$lgID <- factor(teams$lgID, levels= c("AL", "NL"))
# create new variables

teams <- within(teams, {
   WinPct = W / G    ## Winning percentage
   })

library(lattice)
xyplot(attendance/1000 ~ WinPct|yearID, groups=lgID, data=subset(teams, yearID>1980),
type=c("p", "r"), col=c("red","blue"))

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

#### merge with ave salary, for those years where salary is available

avesal <- aggregate(salary ~ yearID + teamID, data=Salaries, FUN=mean)

# salary data just starts after 1980
teamsSal <- subset(teams, yearID>=1980)

# add salary to team data
teamsSal <- merge(teamsSal,
                 avesal[,c("yearID", "teamID", "salary")],
                 by=c("yearID", "teamID"), all.x=TRUE)

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")

}

## End(Not run)
```

TeamsFranchises *TeamFranchises table*

Description

Information about team franchises

Usage

```
data(TeamsFranchises)
```

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

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, http://baseball1.com/statistics/

`TeamsHalf` *TeamsHalf table*

Description

Split season data for teams

Usage

```
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

Lahman, S. (2015) Lahman's Baseball Database, 1871-2014, 2015 version, `http://baseball1.com/statistics/`

Index