Week 5 statistical computing

MS 276

September, 2017

Project

Today's class: Data analysis

1. Importing data
2. Combining data
3. Missing values

Data import

- CSV formats

```
data.states <- read_csv("https://raw.githubusercontent.com/statsbylopez/DataViz/master/Homeworks/Wiki.S
head(data.states)
```

```
## # A tibble: 6 x 8
##    Rank        State Population House Elect Pop.House Pop.elect Pop.Senate
##   <int>        <chr>      <int> <int> <int>     <int>     <int>      <int>
## 1     1   California   38802500    53    55    717763    691662   19401250
## 2     2        Texas   26956958    36    38    723867    685769   13478479
## 3     3      Florida   19893297    27    29    715465    666123    9946649
## 4     4     New York   19746227    27    29    724824    674837    9873114
## 5     5     Illinois   12880580    18    20    715292    643763    6440290
## 6     6 Pennsylvania   12787209    18    20    709085    638177    6393605
```

- How to adjust for data from hard drive?
- What's in a csv?

Understanding your data

Today we'll work with the `Lahman` package.

```
#install.packages("Lahman")
library(Lahman)
```

From the R description:

This database contains pitching, hitting, and fielding statistics for Major League Baseball from 1871 through 2015. 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.

This package comes with a few data sets. These include `Master`, `Batting`, `Pitching`, and `Fielding`, which each contain relevant information for baseball players over time.

```
head(Master)
```

```
##    playerID birthYear birthMonth birthDay birthCountry birthState
## 1 aardsda01      1981         12       27          USA         CO
## 2 aaronha01      1934          2        5          USA         AL
## 3 aaronto01      1939          8        5          USA         AL
## 4  aasedo01      1954          9        8          USA         CA
## 5  abadan01      1972          8       25          USA         FL
## 6  abadfe01      1985         12       17         D.R.  La Romana
##    birthCity deathYear deathMonth deathDay deathCountry deathState
## 1     Denver        NA         NA       NA         <NA>       <NA>
## 2     Mobile        NA         NA       NA         <NA>       <NA>
## 3     Mobile      1984          8       16          USA         GA
## 4     Orange        NA         NA       NA         <NA>       <NA>
## 5 Palm Beach        NA         NA       NA         <NA>       <NA>
## 6  La Romana        NA         NA       NA         <NA>       <NA>
##   deathCity nameFirst nameLast        nameGiven weight height bats throws
## 1      <NA>     David  Aardsma      David Allan    220     75    R      R
## 2      <NA>      Hank    Aaron      Henry Louis    180     72    R      R
## 3   Atlanta    Tommie    Aaron       Tommie Lee    190     75    R      R
## 4      <NA>       Don     Aase   Donald William    190     75    R      R
## 5      <NA>      Andy     Abad    Fausto Andres    184     73    L      L
## 6      <NA>  Fernando     Abad Fernando Antonio    220     73    L      L
##        debut  finalGame  retroID   bbrefID  deathDate  birthDate
## 1 2004-04-06 2015-08-23 aardd001 aardsda01       <NA> 1981-12-27
## 2 1954-04-13 1976-10-03 aaroh101 aaronha01       <NA> 1934-02-05
## 3 1962-04-10 1971-09-26 aarot101 aaronto01 1984-08-16 1939-08-05
## 4 1977-07-26 1990-10-03 aased001  aasedo01       <NA> 1954-09-08
## 5 2001-09-10 2006-04-13 abada001  abadan01       <NA> 1972-08-25
## 6 2010-07-28 2015-10-03 abadf001  abadfe01       <NA> 1985-12-17
```

```
head(Batting)
```

```
##    playerID yearID stint teamID lgID  G  AB  R  H X2B X3B HR RBI SB CS BB
## 1 abercda01   1871     1    TRO   NA  1   4  0  0   0   0  0   0  0  0  0
## 2  addybo01   1871     1    RC1   NA 25 118 30 32   6   0  0  13  8  1  4
## 3 allisar01   1871     1    CL1   NA 29 137 28 40   4   5  0  19  3  1  2
## 4 allisdo01   1871     1    WS3   NA 27 133 28 44  10   2  2  27  1  1  0
## 5 ansonca01   1871     1    RC1   NA 25 120 29 39  11   3  0  16  6  2  2
## 6 armstbo01   1871     1    FW1   NA 12  49  9 11   2   1  0   5  0  1  0
##   SO IBB HBP SH SF GIDP
## 1  0  NA  NA NA NA   NA
## 2  0  NA  NA NA NA   NA
## 3  5  NA  NA NA NA   NA
## 4  2  NA  NA NA NA   NA
## 5  1  NA  NA NA NA   NA
## 6  1  NA  NA NA NA   NA
```

Combining data sets

Let's say we were interested in the `Batting` data set, but noticed that player names are not available. How could we combine this information?

The `join` commands are quite useful:

- `left_join(df1, df2)`

- `inner_join(df1, df2)`

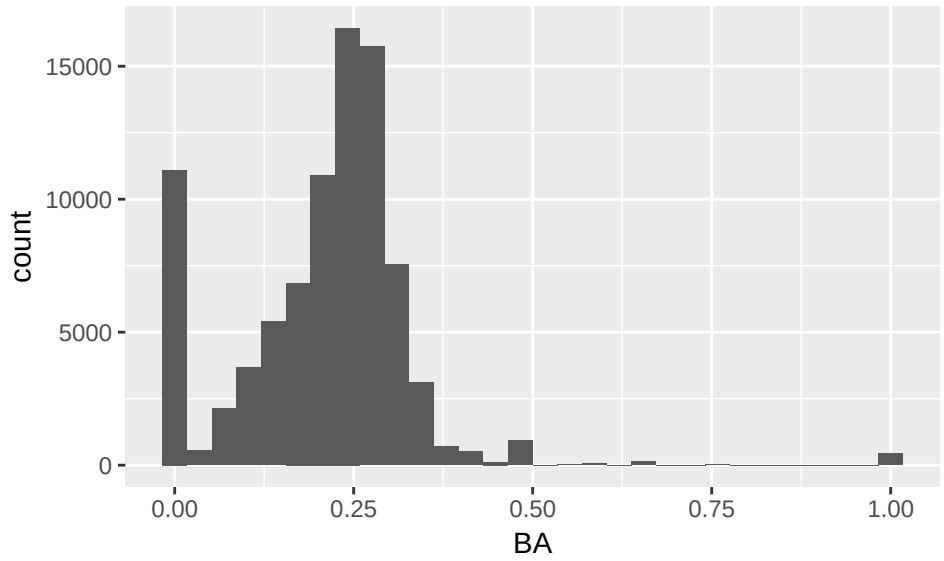
- `right_join(df1, df2)`

```
Master.sum <- Master %>% select(playerID, birthYear, nameFirst, nameLast)
Batting1 <- left_join(Batting, Master.sum)
head(Batting1)
```

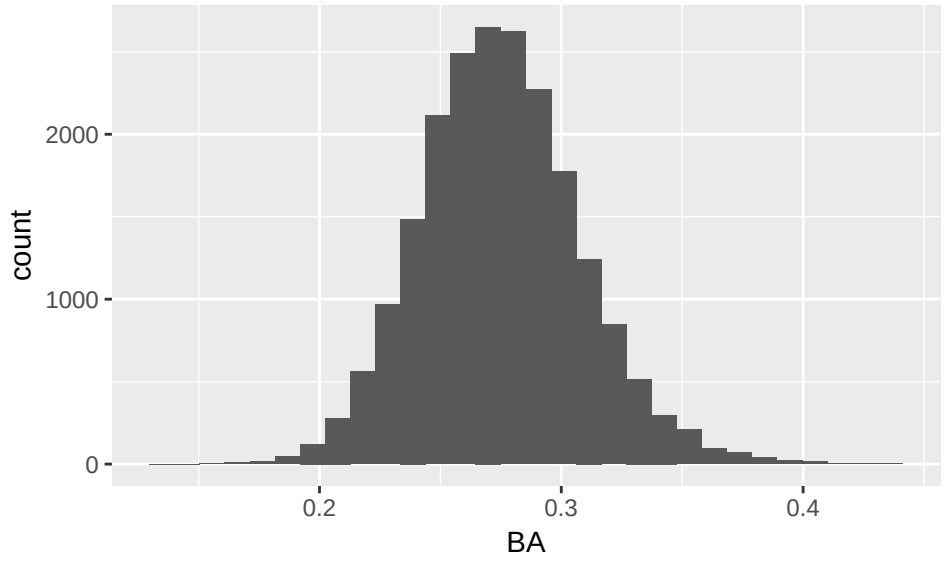
```
##    playerID yearID stint teamID lgID  G  AB  R  H X2B X3B HR RBI SB CS BB
## 1 abercda01   1871     1    TRO   NA  1   4  0  0   0   0  0   0  0  0  0
## 2  addybo01   1871     1    RC1   NA 25 118 30 32   6   0  0  13  8  1  4
## 3 allisar01   1871     1    CL1   NA 29 137 28 40   4   5  0  19  3  1  2
## 4 allisdo01   1871     1    WS3   NA 27 133 28 44  10   2  2  27  1  1  0
## 5 ansonca01   1871     1    RC1   NA 25 120 29 39  11   3  0  16  6  2  2
## 6 armstbo01   1871     1    FW1   NA 12  49  9 11   2   1  0   5  0  1  0
##   SO IBB HBP SH SF GIDP birthYear nameFirst    nameLast
## 1  0  NA  NA NA NA   NA      1850     Frank Abercrombie
## 2  0  NA  NA NA NA   NA      1842       Bob        Addy
## 3  5  NA  NA NA NA   NA      1849       Art     Allison
## 4  2  NA  NA NA NA   NA      1846      Doug     Allison
## 5  1  NA  NA NA NA   NA      1852       Cap       Anson
## 6  1  NA  NA NA NA   NA      1850    Robert   Armstrong
```

Unusual or missing values

```
Batting1 <- Batting1 %>%
  mutate(BA = H/AB)
ggplot(Batting1, aes(BA)) + geom_histogram()
```

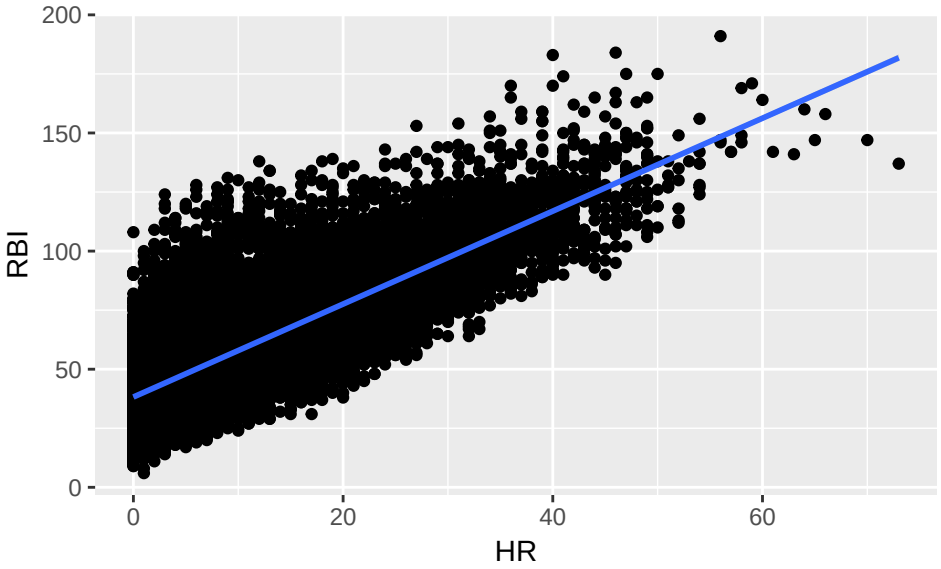



```
Batting2 <- Batting1 %>%
  filter(AB > 300)
ggplot(Batting2, aes(BA)) + geom_histogram()
```



Modeling

```
Batting2 <- Batting2 %>% filter(yearID >= 1900)
ggplot(Batting2, aes(HR, RBI)) + geom_point() + geom_smooth(method = "lm")
```



- Model types

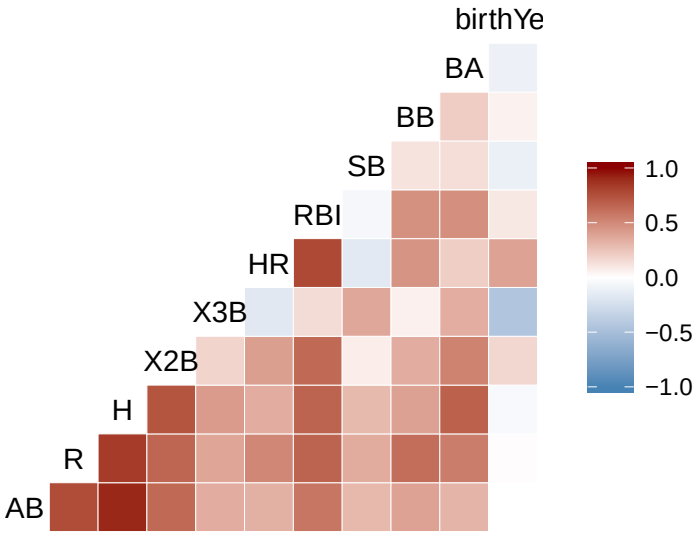
- Linear models

```
mod <- lm(RBI ~ HR, data = Batting2)
summary(mod)
```

```
##
## Call:
## lm(formula = RBI ~ HR, data = Batting2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.836 -11.540  -2.228   8.903  79.870
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 38.22798    0.17726   215.7   <2e-16 ***
## HR           1.96724    0.01163   169.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.08 on 18785 degrees of freedom
## Multiple R-squared:  0.6038, Adjusted R-squared:  0.6038
## F-statistic: 2.863e+04 on 1 and 18785 DF,  p-value: < 2.2e-16
```

- Correlations

```
library(GGally)
Batting2.short <- Batting2 %>%
  select(AB, R, H, X2B, X3B, HR, RBI, SB, BB, BA, birthYear)
ggcorr(Batting2.short, low = "steelblue", mid = "white", high = "darkred")
```



Lab goals

1. Download the Lahman package
2. Merge the Pitching data frame (`Pitching`) with the master list of players, as in today's notes.
3. Identify the 10 individual players who have recorded seasons with the most strikeouts (`SO`). It is okay to have a few of the same players more than once.
4. Identify the 10 individual seasons with the most strikeouts, when taking into account *all* pitchers.
5. What has happened to the number of total shutouts (`SHO`) by year over time? Make a visualization to explore this.
6. ERA stands for earned run average. Make two histograms of ERA. First, use all players. Second, use all players who have recorded at least 300 outs (`IPouts >=300`).
7. Identify the regression line of earned runs (`ER`) as a function of hits (`H`). What does this suggest?
8. Identify which subset of variables in the `Pitching` data frame have the strongest (positive or negative) correlations.
9. Poke around the suggested project topics!