Analyzing Baseball Data with R - Introduction to R

Tomoka Takegaki

23/11/2021

This project is to learn analyze baseball data with R. The source is from a book "Analyzing Baseball Data with R". This is a section of "Introduction to R".

Setting an environment

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(Lahman)
```

```
## Warning: package 'Lahman' was built under R version 4.1.2
```

Babe Ruth analysis

```
# Create a data frame for Babe Ruth
bruth_pitch_df <- Pitching[Pitching$playerID=="ruthba01",]

# Aggregate ERA data
bruth_pitch_df %>%
    summarize(
        LO = min(ERA),
        QL = quantile(ERA,.25), QU = quantile(ERA,.75),M = median(ERA),
        Hi = max(ERA)
    )
```

```
##     LO    QL     QU     M Hi
## 1 1.75 2.275 4.3525 2.985  9
```

```
# Year of the lowerst ERA of Babe Ruth
bruth_pitch_df %>% filter(ERA==min(ERA)) %>% select(yearID)
```

```
##   yearID
## 1   1916
```

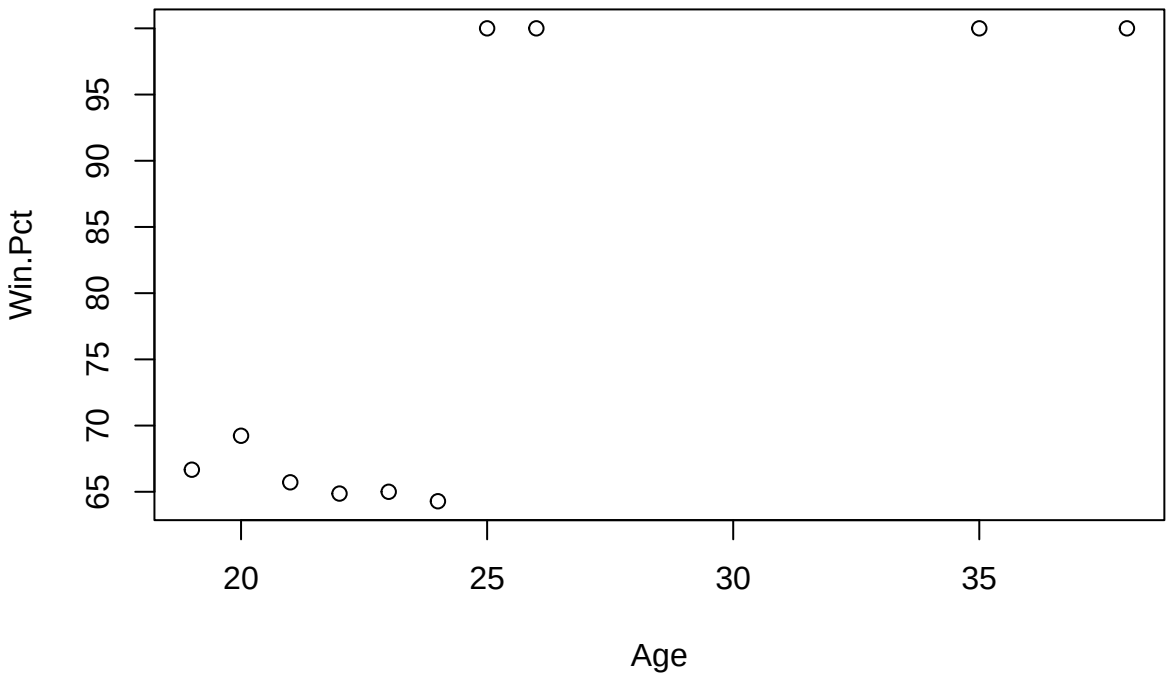
```
# Adding new column "FIP", Fielding independent pitching.
bruth_pitch_df <- bruth_pitch_df %>%
    mutate(FIP = (13 * HR + 3 * BB -2 * SO)/IPouts)

# Sort the data by FIP(ascending)
bruth_pitch_df %>%
    arrange(FIP) %>%
    select(yearID,W,L,ERA,FIP) %>%
    head(10)
```

```
##    yearID  W  L  ERA        FIP
## 1    1930  1  0 3.00 0.00000000
## 2    1916 23 12 1.75 0.01441813
## 3    1917 24 13 2.01 0.09601634
## 4    1915 18  8 2.44 0.10719755
## 5    1918 13  7 2.22 0.16032064
## 6    1933  1  0 5.00 0.33333333
## 7    1919  9  5 2.97 0.35000000
## 8    1914  2  1 3.91 0.40579710
## 9    1920  1  0 4.50 0.50000000
## 10   1921  2  0 9.00 1.33333333
```

```
# Performance for each team Babe ruth played.
bruth_pitch_df %>%
    group_by(teamID) %>%
    summarize(mean_W = mean(W),
              mean_L = mean(L),
              mean_ERA = mean(ERA),
              mean_FIP = mean(FIP))
```

```
## # A tibble: 2 x 5
##   teamID mean_W mean_L mean_ERA mean_FIP
##   <fct>   <dbl>  <dbl>    <dbl>    <dbl>
## 1 BOS     14.8    7.67     2.55    0.189
## 2 NYA      1.25   0        5.38    0.542
```

```
# Vector
Win.Pct <- 100 * bruth_pitch_df$W / (bruth_pitch_df$W + bruth_pitch_df$L)

Age <- bruth_pitch_df$yearID - 1895
plot(Age,Win.Pct)
```




```
summary(Win.Pct)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   64.29   65.18   67.95   79.58  100.00  100.00
```

WS analysis

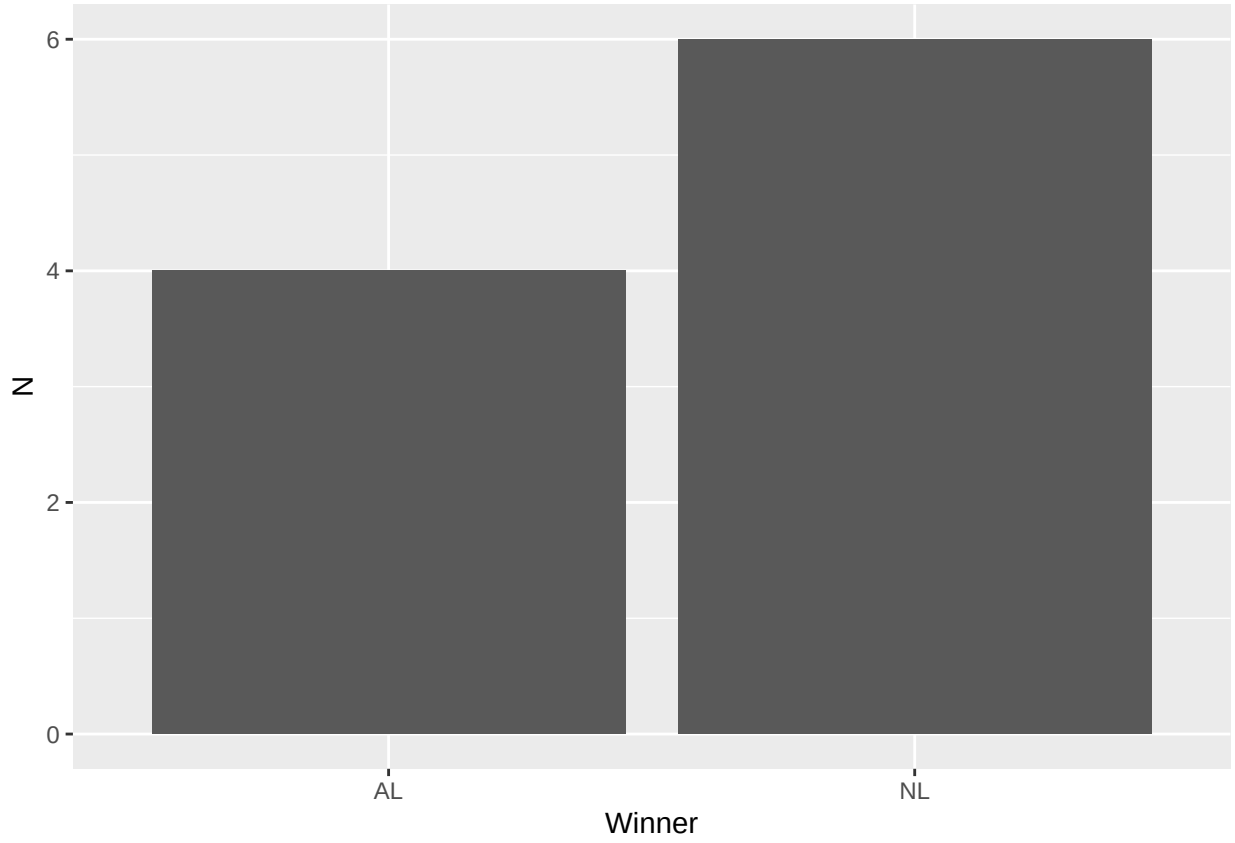
```
# Character data and data frame
Year <- 2008:2017
NL <- c("PHI","PHI","SFN","SLN","SFN",
        "SLN","SFN","NYN","CHN","LAN")
AL <- c("TBA","NYA","TEX","TEX","DET",
        "BOS","KCA","KCA","CLE","HOU")
Winner <- c("NL","AL","NL","NL","NL",
            "AL","NL","AL","NL","AL")
N_Games <- c(5,6,5,7,4,7,7,5,7,7)

# Create a data frame
WS_results <- tibble(
    Year = Year, NL_Team =  NL, AL_Team = AL,
    N_Games = N_Games, Winner = Winner)

# Find patterns
grep("NY",c(AL,NL),value=TRUE)
```

```
## [1] "NYA" "NYN"
```

```
WS <- WS_results %>%
    group_by(Winner) %>%
    summarize(N=n())
WS
```

```
## # A tibble: 2 x 2
##   Winner     N
##   <chr>  <int>
## 1 AL         4
## 2 NL         6
```

```
# plot bar chart
ggplot(WS,mapping = aes(x=Winner,y=N)) +
    geom_col()
```




```
# Factors
WS_results %>%
    group_by(NL_Team) %>%
    summarize(N=n())
```

```
## # A tibble: 6 x 2
##   NL_Team     N
```

```
##   <chr>   <int>
## 1 CHN         1
## 2 LAN         1
## 3 NYN         1
## 4 PHI         2
## 5 SFN         3
## 6 SLN         2
```

```
WS_results <- WS_results %>%
    mutate(NL_Team = factor(NL_Team, levels = c("NYN","PHI","CHN",
                                                "SLN","LAN","SFN")))
str(WS_results$NL_Team)
```

```
##  Factor w/ 6 levels "NYN","PHI","CHN",..: 2 2 6 4 6 4 6 1 3 5
```

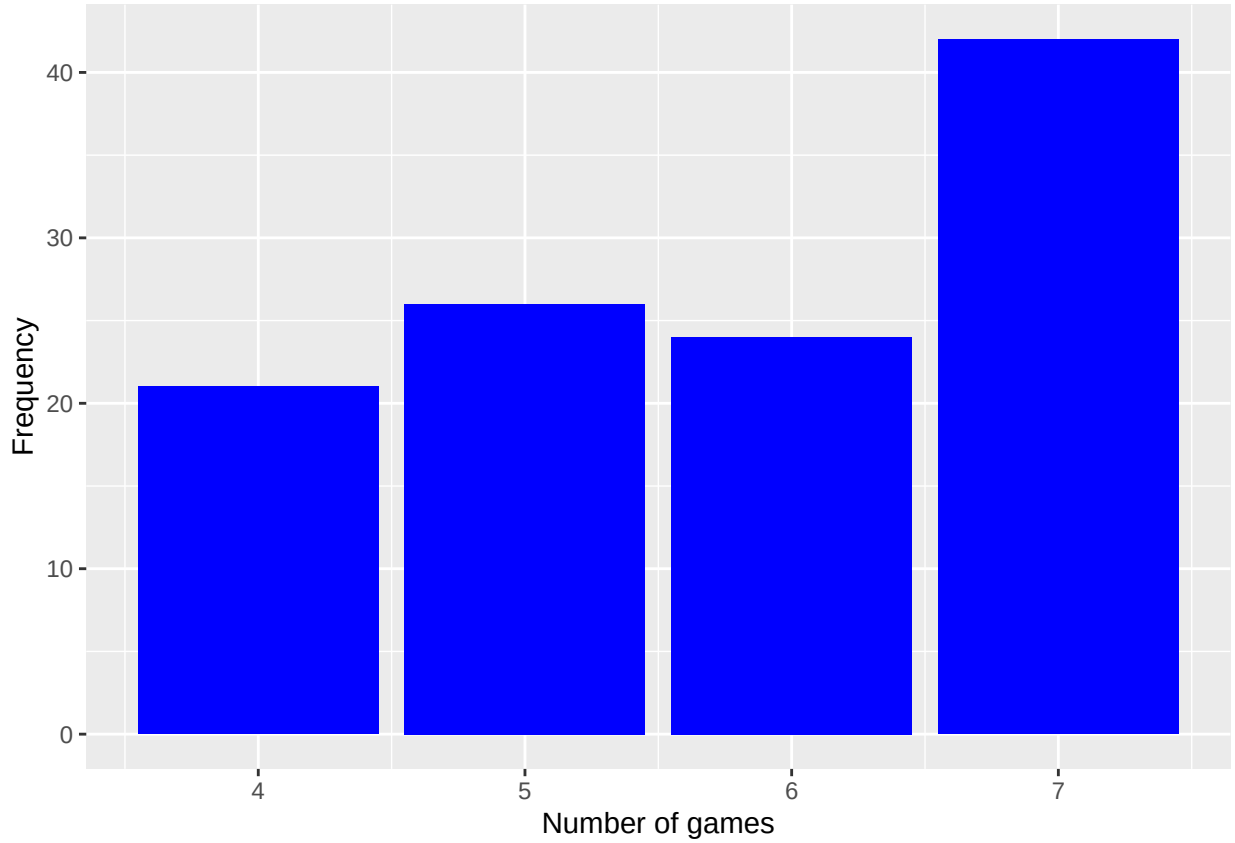
```
WS_results %>%
    group_by(NL_Team) %>%
    summarize(N=n())
```

```
## # A tibble: 6 x 2
##   NL_Team     N
##   <fct>   <int>
## 1 NYN         1
## 2 PHI         2
## 3 CHN         1
## 4 SLN         2
## 5 LAN         1
## 6 SFN         3
```

```
# Lists
world_series <- list(Winner=Winner, Number.Games=N_Games, Seasons="2008 to 2017")
world_series
```

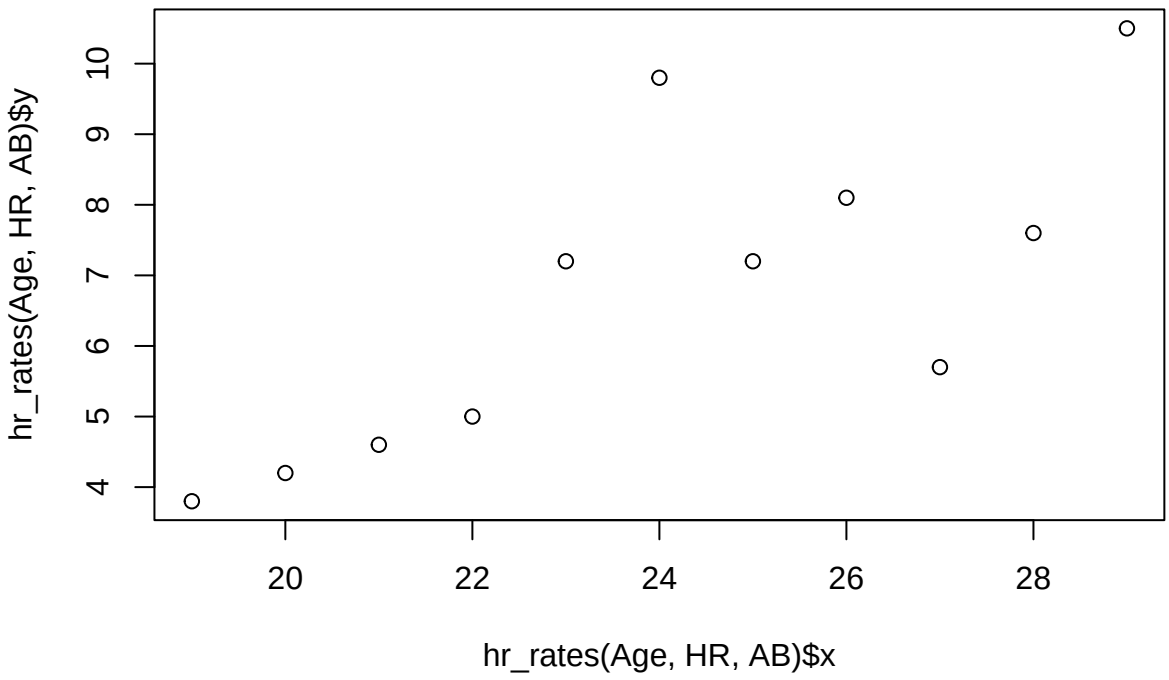
```
## $Winner
##  [1] "NL" "AL" "NL" "NL" "NL" "AL" "NL" "AL" "NL" "AL"
##
## $Number.Games
##  [1] 5 6 5 7 4 7 7 5 7 7
##
## $Seasons
## [1] "2008 to 2017"
```

```
# Frequency of number of games (less than 8) in 1903.
ws <- filter(SeriesPost, yearID >= 1903,
       round == "WS", wins+losses < 8)

ggplot(ws,mapping = aes(x=wins+losses)) +
    geom_bar(fill="blue") +
    labs(x="Number of games", y="Frequency")
```



Home run rate

```
# Calculate Home run rate (Micky mantle)
hr_rates <- function(age,hr,ab){
    rates <- round(100 * hr / ab, 1)
    list(x=age, y=rates)
}

HR <- c(13,23,21,27,37,52,34,42,31,40,54)
AB <- c(341,549,461,543,517,533,474,519,541,527,514)
Age <- c(19:29)

# Scatter plot
plot(hr_rates(Age,HR,AB))
```




```
hr_rates <- hr_rates(Age,HR,AB)

# Writing csv file
Mantle <- data.frame(Age, HR,AB,Rates=hr_rates$y)
#write.csv(Mantle,"csv_files/mantle.csv")
Mantle
```

```
##    Age HR  AB Rates
## 1   19 13 341   3.8
## 2   20 23 549   4.2
## 3   21 21 461   4.6
## 4   22 27 543   5.0
## 5   23 37 517   7.2
## 6   24 52 533   9.8
## 7   25 34 474   7.2
## 8   26 42 519   8.1
## 9   27 31 541   5.7
## 10  28 40 527   7.6
## 11  29 54 514  10.5
```

Splitting, Applying, and Combining data

```r
# Batting data between 1960 and 1969.
Batting %>%
    filter(yearID>=1960, yearID <=1969) -> Batting_60

# Total number of homeruns for each player
Batting_60 %>%
    group_by(playerID) %>%
    summarize(Total_HR = sum(HR)) -> hr_60


# Sort the hr_60 data in desc order
hr_60 %>%
    arrange(desc(Total_HR))->hr_60

head(hr_60)
```

```
## # A tibble: 6 x 2
##   playerID  Total_HR
##   <chr>        <int>
## 1 killeha01      393
## 2 aaronha01      375
## 3 mayswi01       350
## 4 robinfr02      316
## 5 mccovwi01      300
## 6 howarfr01      288
```

```r
# Iterating using map()
hr_leader <- function(data){
    data %>%
        group_by(playerID) %>%
        summarize(Total_HR = sum(HR)) %>%
        arrange(desc(Total_HR)) %>%
        head(1)
}

# Home run leader for each decade.
Batting %>%
    mutate(decade = 10 * floor(yearID/10)) %>%
    split(pull(.,decade)) %>%
    map_df(hr_leader, .id="decade") -> hr_by_decade

hr_by_decade
```

```
## # A tibble: 16 x 3
##    decade playerID  Total_HR
##    <chr>  <chr>        <int>
##  1 1870   pikeli01        21
##  2 1880   stoveha01       89
##  3 1890   duffyhu01       83
##  4 1900   davisha01       67
##  5 1910   cravaga01      116
##  6 1920   ruthba01       467
```

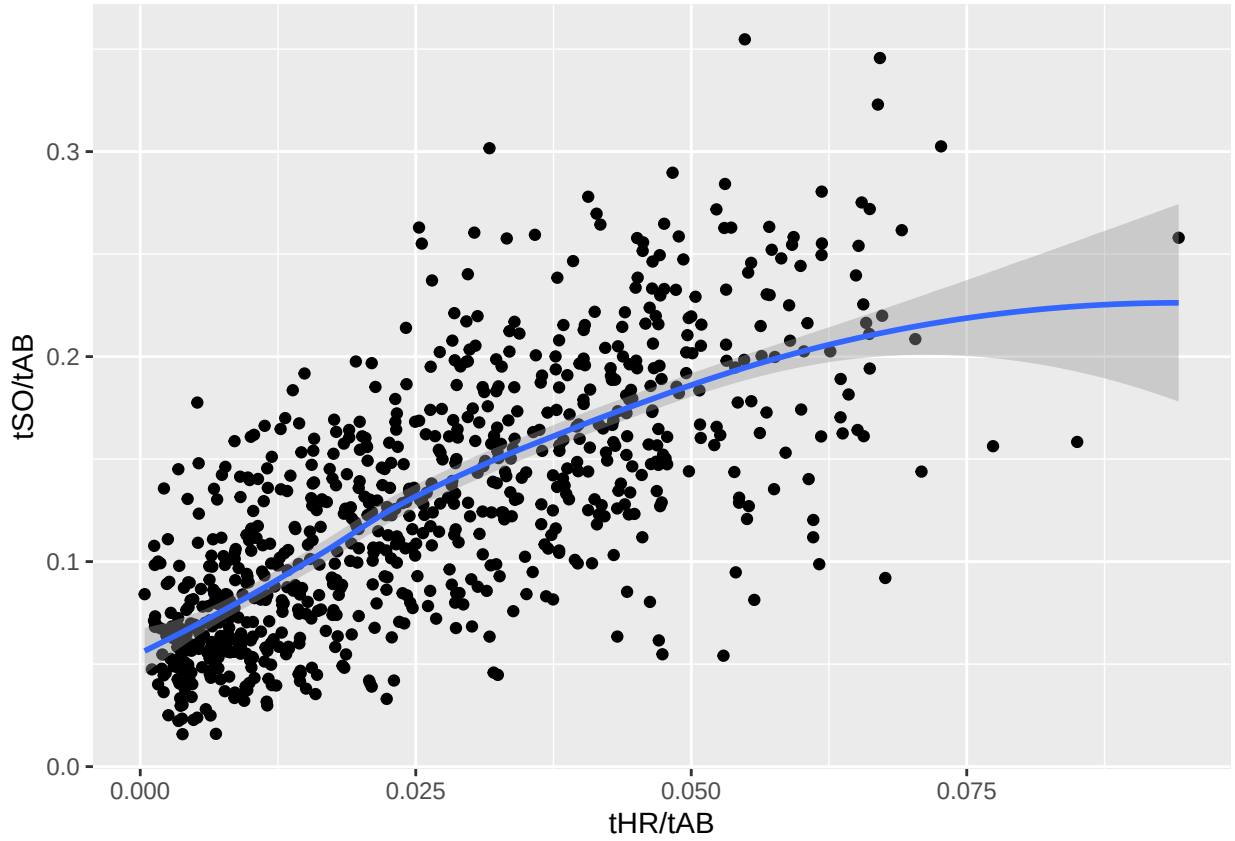
```
##  7 1930   foxxji01       415
##  8 1940   willite01      234
##  9 1950   snidedu01      326
## 10 1960   killeha01      393
## 11 1970   stargwi01      296
## 12 1980   schmimi01      313
## 13 1990   mcgwima01      405
## 14 2000   rodrial01      435
## 15 2010   cruzne02       346
## 16 2020   voitlu01        22
```

```
# Collect the career batting statistics
Batting %>%
    group_by(playerID) %>%
    summarize(tAB = sum(AB,na.rm = TRUE),
              tHR = sum(HR,na.rm = TRUE),
              tSO = sum(SO,na.rm = TRUE)) -> long_careers

# filter tAB >= 5000 players
Batting_5000 <- filter(long_careers, tAB >= 5000)
head(Batting_5000)
```

```
## # A tibble: 6 x 4
##   playerID    tAB   tHR   tSO
##   <chr>     <int> <int> <int>
## 1 aaronha01 12364   755  1383
## 2 abreubo01  8480   288  1840
## 3 adamssp01  5557     9   223
## 4 adcocjo01  6606   336  1059
## 5 alfoned01  5385   146   617
## 6 allendi01  6332   351  1556
```

```
# Correlation between HR rates & SO rates
ggplot(Batting_5000, mapping = aes(x=tHR/tAB, y=tSO/tAB))+
    geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



We can see clearly that batters with higher home run rates tend to have higher strikeout rates.

Exercises

1. Top Base Stealers in the Hall of Fame

```
# (a) Create a data frame
players <- c("Rickey Henderson","Lou Brock","Ty Cobb","Eddie Collins","Max Carey","Joe Morgan","Luis Ap
SB <- c(1406,938,897,741,738,689,506,504,474)
CS <- c(335,307,212,195,109,162,136,131,114)
G <- c(3081,2616,3034,2826,2476,2649,2599,2683,2379)
sb_df <- data.frame(players,SB,CS,G)
sb_df
```

```
##            players   SB  CS    G
## 1 Rickey Henderson 1406 335 3081
## 2        Lou Brock  938 307 2616
## 3          Ty Cobb  897 212 3034
## 4    Eddie Collins  741 195 2826
## 5        Max Carey  738 109 2476
## 6       Joe Morgan  689 162 2649
## 7     Luis Aparico  506 136 2599
## 8     Paul Molitor  504 131 2683
## 9   Roberto Alomar  474 114 2379
```

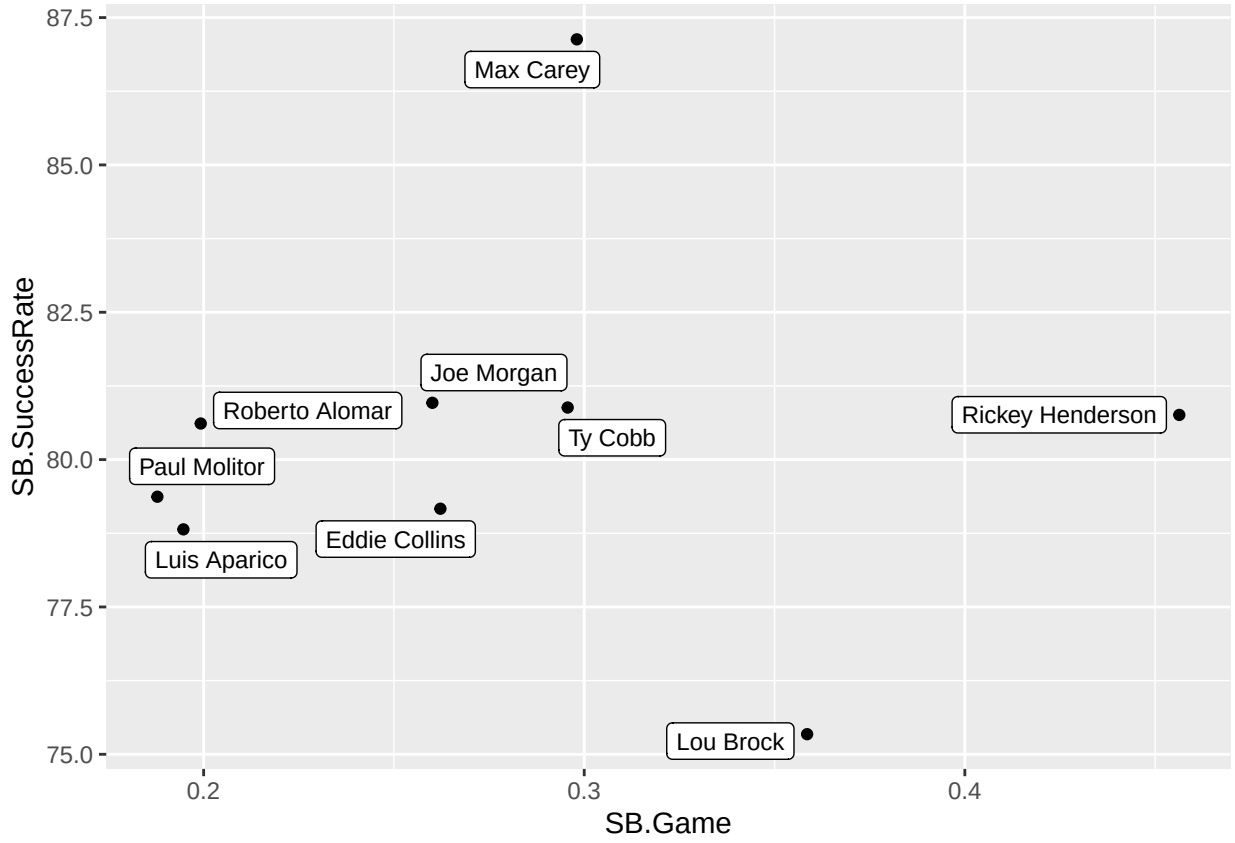
```r
# (b) Create New column "SB.Attempt" (SB+CS)
sb_df <- sb_df %>%
    mutate(SB.Attempt = SB + CS)

# (c) Create New column "SB.Game" (SB/G) Stolen bases per game
sb_df <- sb_df %>%
    mutate(SB.Game = SB / G)

sb_df <- sb_df %>%
    mutate(SB.SuccessRate = 100 * SB / SB.Attempt)

sb_df
```

```
##            players   SB  CS    G SB.Attempt   SB.Game SB.SuccessRate
## 1 Rickey Henderson 1406 335 3081       1741 0.4563453       80.75818
## 2        Lou Brock  938 307 2616       1245 0.3585627       75.34137
## 3          Ty Cobb  897 212 3034       1109 0.2956493       80.88368
## 4    Eddie Collins  741 195 2826        936 0.2622081       79.16667
## 5        Max Carey  738 109 2476        847 0.2980614       87.13105
## 6       Joe Morgan  689 162 2649        851 0.2600982       80.96357
## 7     Luis Aparico  506 136 2599        642 0.1946903       78.81620
## 8     Paul Molitor  504 131 2683        635 0.1878494       79.37008
## 9   Roberto Alomar  474 114 2379        588 0.1992434       80.61224
```

```r
#install.packages("ggrepel")
library(ggrepel)
```

```
## Warning: package 'ggrepel' was built under R version 4.1.2
```

```r
ggplot(sb_df, mapping = aes(x=SB.Game,y=SB.SuccessRate))+
    geom_point() + geom_label_repel(aes(label = players), size = 3)
```



1. Are there are particular players with unusually high or low stolen base success rates?

- Max Carey had the highest stolen base success rate with 87.1%.
- Lou Brock had the lowest stolen base success rate with 75.3%.

2. Which player had the greatest number of stolen bases per game?

- Rickey Henderson had the greatest number of stolen bases per game : 0.46 / game.

2. Character, Factor, and Logical Variables in R

Suppose one records the outcomes of a batter in ten plate appearance.

```
outcomes <- c("Single","Out","Out","Single","Out","Double","Out","Walk","Out","Single")

# (a) Construct a frequency table
table(outcomes)
```

```
## outcomes
## Double    Out Single   Walk
##      1      5      3      1
```

```
# (b) Ordered from least_successful to most-successful
f.outcomes <- factor(outcomes,levels = c("Out","Walk","Single","Double"))
table(f.outcomes)
```

```
## f.outcomes
##    Out   Walk Single Double
##      5      1      3      1
```

table a and b appeared in different order. Factor enables us to reorder the table.

(d) Suppose you only want to focus on the walks in the plate appearances. Describe what is done in each of the following statements.

```
outcomes == "Walk"
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
```

```
sum(outcomes == "Walk")
```

```
## [1] 1
```

In the first line, it checks the all the value in the outcome vector if the value is equal to "Walk". It returns True or False. In the second line, it counts how many "Walk" is in the outcome vector. In this case, it returns 1.

3. Pitches in the 350-Wins Club

```
#(a) Create vectors
p_players <- c("Pete Alexander","Roger Clements","Pud Galvin","Walter Johnson","Greg Maddux","Christy Ma
W <- c(373,354,364,417,355,373,361,363,511)
L <- c(208,184,310,279,227,188,208,245,316)
SO <- c(2198,4672,1806,3509,3371,2502,1868,2583,2803)
BB <- c(951,1580,745,1363,999,844,1268,1434,1217)
# (b) Calculate Winning percentage
Win.PCT = (100 * W/W+L)

# (c) Create a data frame
Wins.350 <- data.frame(p_players,W,L,Win.PCT)

# (d) Sort the data by Win.PCT
Wins.350 <- Wins.350 %>% arrange(Win.PCT)
Wins.350
```

```
##           p_players   W   L Win.PCT
## 1    Roger Clements 354 184     284
## 2 Christy Mathewson 373 188     288
## 3    Pete Alexander 373 208     308
## 4       Kid Nichols 361 208     308
## 5       Greg Maddux 355 227     327
```

```
## 6      Warren Spahn 363 245     345
## 7    Walter Johnson 417 279     379
## 8        Pud Galvin 364 310     410
## 9          Cy Young 511 316     416
```

1. Who had the largest winning percentage?

- Roger Clements

2. Who had the smallest winning percentage?

- Cy Young

4. Pitchers in the 350-Wins Club, Continued

```
# (b)Create a vecor strikeout-walk ratio
SO.BB.Ratio <- SO/BB

# (c)Create a data frame
SO.BB <- data.frame(p_players,SO,BB,SO.BB.Ratio)

# (d) filter the data who had strikeout-ratio more than 2.8.
SO.BB<- SO.BB %>% filter(SO.BB.Ratio > 2.8)

# (e) Sort by Walk
SO.BB <- SO.BB %>% arrange(desc(BB))

SO.BB
```

```
##           p_players   SO   BB SO.BB.Ratio
## 1    Roger Clements 4672 1580    2.956962
## 2       Greg Maddux 3371  999    3.374374
## 3 Christy Mathewson 2502  844    2.964455
```

1. Did the pitcher with the highest walks have a high or low strikeout-walk ratio?

- Roger Clements has the highest walk but lowest SO.BB.Ratio.

5. Pitcher Strikeout/Walk Ratios

```
# (a) Read Pitching file
head(Pitching)
```

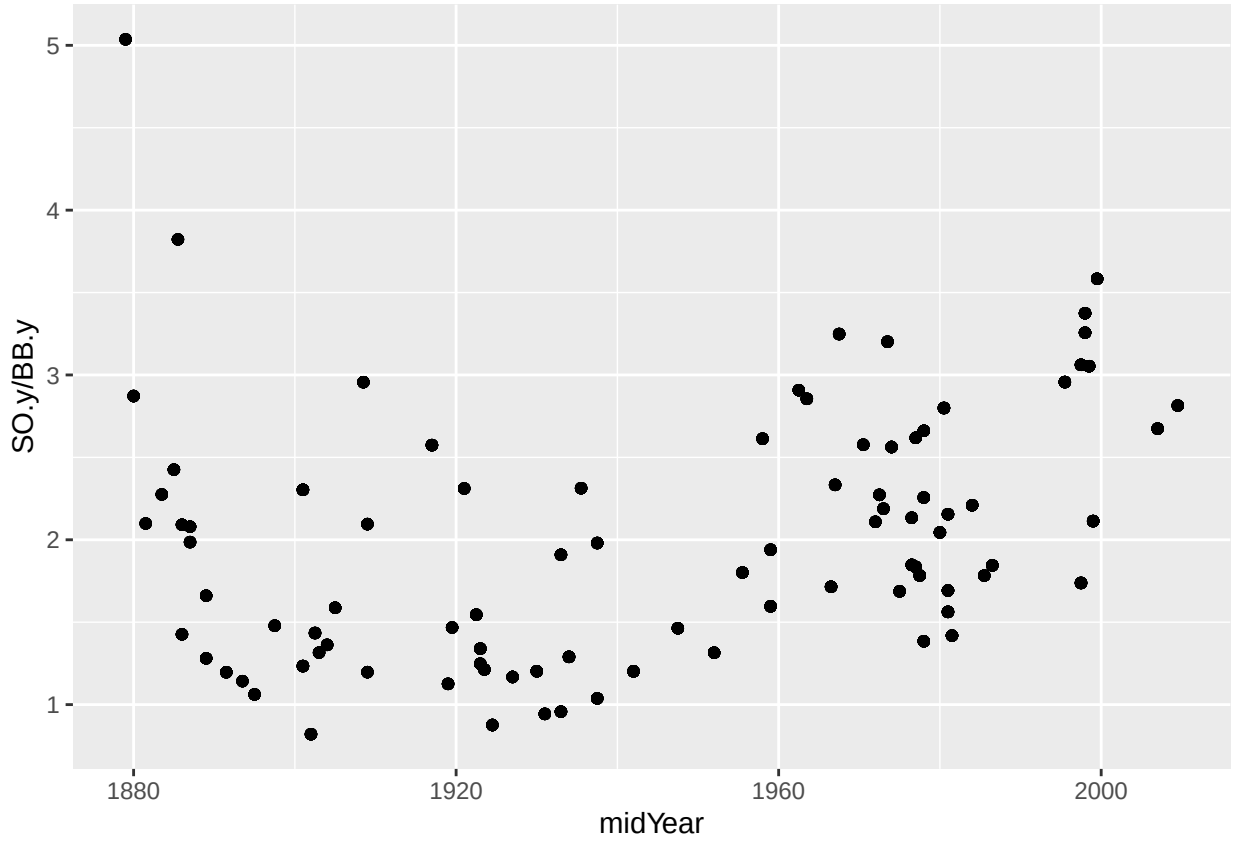
```
##    playerID yearID stint teamID lgID  W  L  G GS CG SHO SV IPouts   H  ER HR BB
## 1 bechtge01   1871     1    PH1   NA  1  2  3  3  2   0  0     78  43  23  0 11
## 2 brainas01   1871     1    WS3   NA 12 15 30 30 30   0  0    792 361 132  4 37
## 3 fergubo01   1871     1    NY2   NA  0  0  1  0  0   0  0      3   8   3  0  0
## 4 fishech01   1871     1    RC1   NA  4 16 24 24 22   1  0    639 295 103  3 31
```

```
## 5 fleetfr01   1871     1    NY2   NA  0  1  1  1  1   0  0     27  20  10  0  3
## 6 flowedi01   1871     1    TRO   NA  0  0  1  0  0   0  0      3   1   0  0  0
##   SO BAOpp   ERA IBB WP HBP BK  BFP GF   R SH SF GIDP
## 1  1    NA  7.96  NA  7  NA  0  146  0  42 NA NA   NA
## 2 13    NA  4.50  NA  7  NA  0 1291  0 292 NA NA   NA
## 3  0    NA 27.00  NA  2  NA  0   14  0   9 NA NA   NA
## 4 15    NA  4.35  NA 20  NA  0 1080  1 257 NA NA   NA
## 5  0    NA 10.00  NA  0  NA  0   57  0  21 NA NA   NA
## 6  0    NA  0.00  NA  0  NA  0    3  1   0 NA NA   NA
```

```
# (b) Compute the cumulative strikeouts, cumulative walks,mid career year,
# and the total innings pitched for all pitchers on the data file.

career_pitching <- Pitching %>%
    group_by(playerID) %>%
    summarize(SO = sum(SO, na.rm = TRUE),
              BB = sum(BB, na.rm = TRUE),
              IPouts = sum(IPouts, na.rm = TRUE),
              midYear = median(yearID, na.rm = TRUE))

# Merge data sets
career_pitching <- inner_join(Pitching,career_pitching,by="playerID")

# (c) filter data (IPouts >= 10000)
career_pitching %>%
    filter(IPouts.y >10000) -> career.10000

head(career.10000)
```

```
##    playerID yearID stint teamID lgID  W  L  G GS CG SHO SV IPouts.x   H  ER HR
## 1 mathebo01   1871     1    FW1   NA  6 11 19 19 19   1  0      507 261  97  5
## 2 mathebo01   1872     1    BL1   NA 25 18 49 47 39   0  0     1218 480 144  3
## 3 mathebo01   1873     1    NY2   NA 29 23 52 52 47   2  0     1329 489 127  5
## 4  bondto01   1874     1    BR2   NA 22 32 55 55 55   1  0     1491 606 112 15
## 5 mathebo01   1874     1    NY2   NA 42 22 65 65 62   4  0     1734 652 122  3
## 6  bondto01   1875     1    HR1   NA 19 16 40 39 37   6  0     1056 302  55  3
##   BB.x SO.x BAOpp  ERA IBB WP HBP BK  BFP GF   R SH SF GIDP SO.y BB.y IPouts.y
## 1   21   17    NA 5.17  NA 15  NA  2  876  0 243 NA NA   NA 1528  532    14868
## 2   52   57    NA 3.19  NA 25  NA  0 1922  4 356 NA NA   NA 1528  532    14868
## 3   62   79    NA 2.58  NA 23  NA  0 2008  0 348 NA NA   NA 1528  532    14868
## 4    8   42    NA 2.03  NA  6  NA  0 2288  0 440 NA NA   NA  972  193    10886
## 5   41  101    NA 1.90  NA 32  NA  0 2543  0 371 NA NA   NA 1528  532    14868
## 6    7   70    NA 1.41  NA 17  NA  0 1408  2 152 NA NA   NA  972  193    10886
##   midYear
## 1    1880
## 2    1880
## 3    1880
## 4    1879
## 5    1880
## 6    1879
```

```
# (d) Scatter plot
ggplot(career.10000, mapping = aes(x=midYear, y=SO.y/BB.y))+
    geom_point()
```



This scatter plot shows correlation between mid career year and ratio of strikeouts to walks.