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".

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

```
setwd("C:/Users/ttake/Documents/My_Data_Analysis_Projects/Analyzing_Baseball_Data_R//")
```

```
# 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
```

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

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

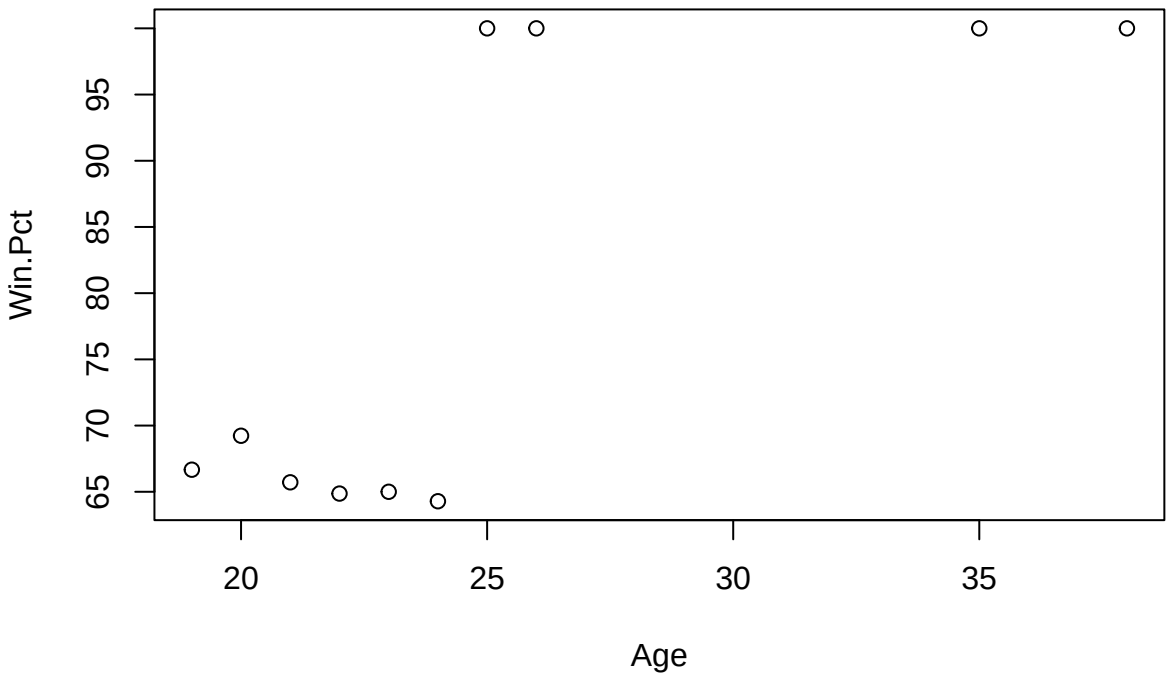
```r
# Adding new column "FIP", Fielding independent pitching.
bruth_pitch_df <- bruth_pitch_df %>%
    mutate(FIP = (13 * HR + 3 * BB -2 * SO)/IPouts)

# 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
```

```r
# 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
```

```r
# 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
```

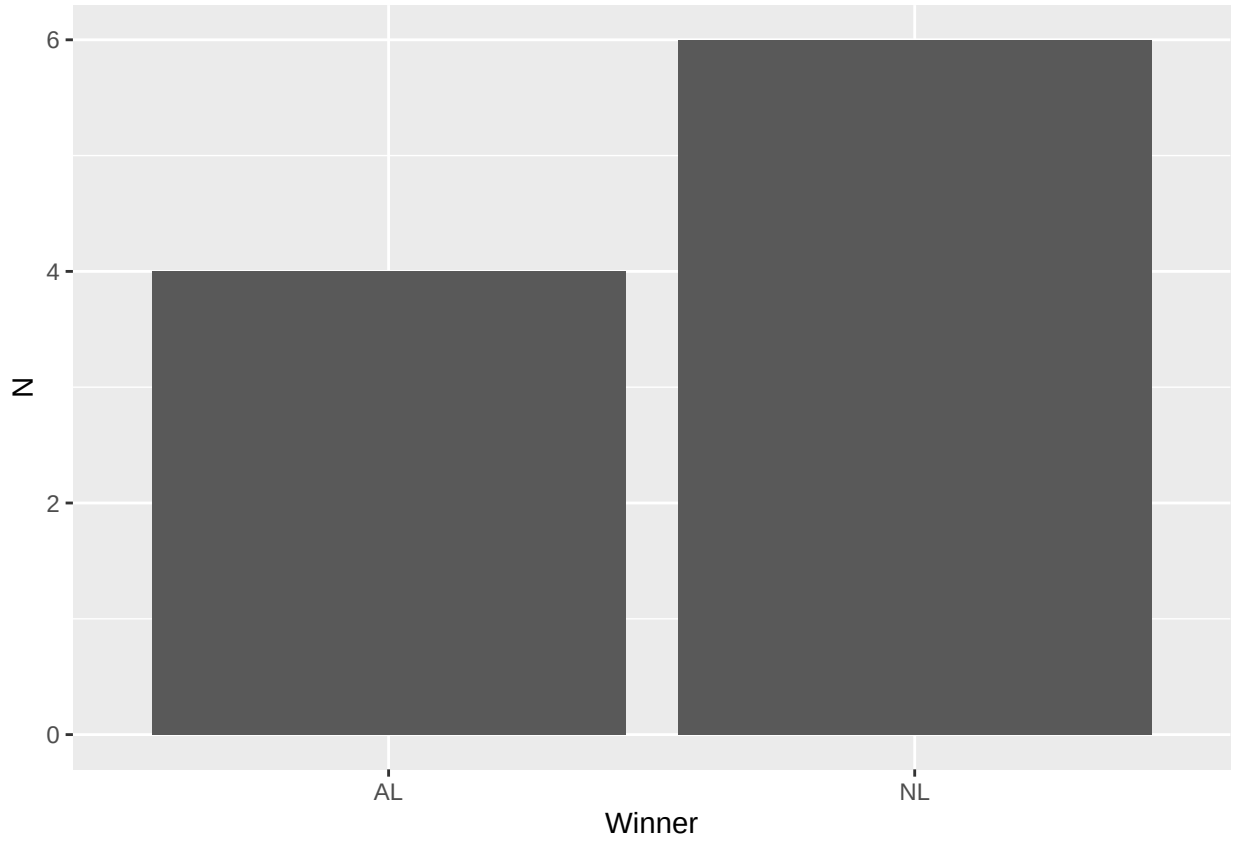
```
# Character data and data frame
Year <- 2008:2017
NL <- c("PHI","PHI","SFN","SLN","SFN",
        "SLN","SFN","NYN","CHN","LAN")
AL <- c("TBA","NYA","TEX","TEX","DET",
        "BOS","KCA","KCA","CLE","HOU")
Winner <- c("NL","AL","NL","NL","NL",
            "AL","NL","AL","NL","AL")
N_Games <- c(5,6,5,7,4,7,7,5,7,7)

# 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
```

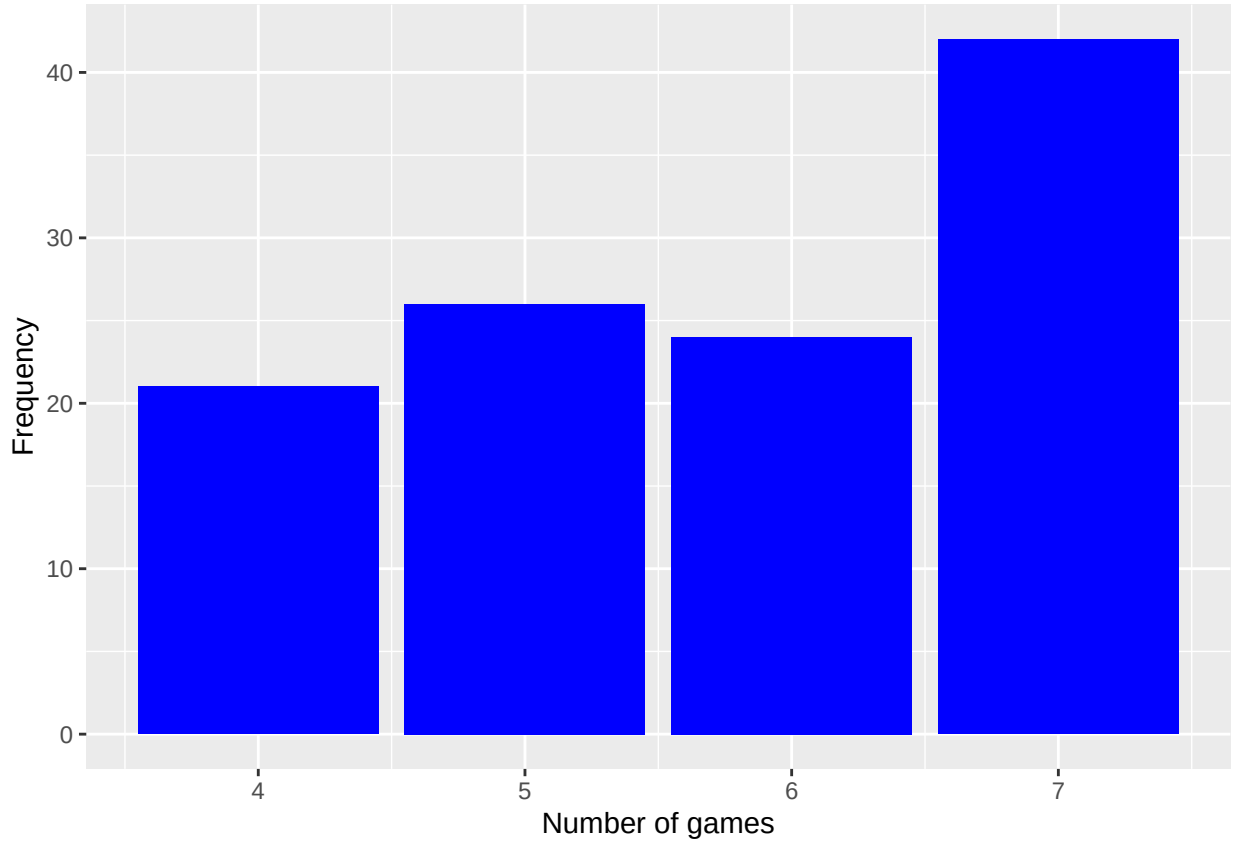
```
WS_results %>%
    group_by(NL_Team) %>%
    summarize(N=n())
```

```
## # 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
```

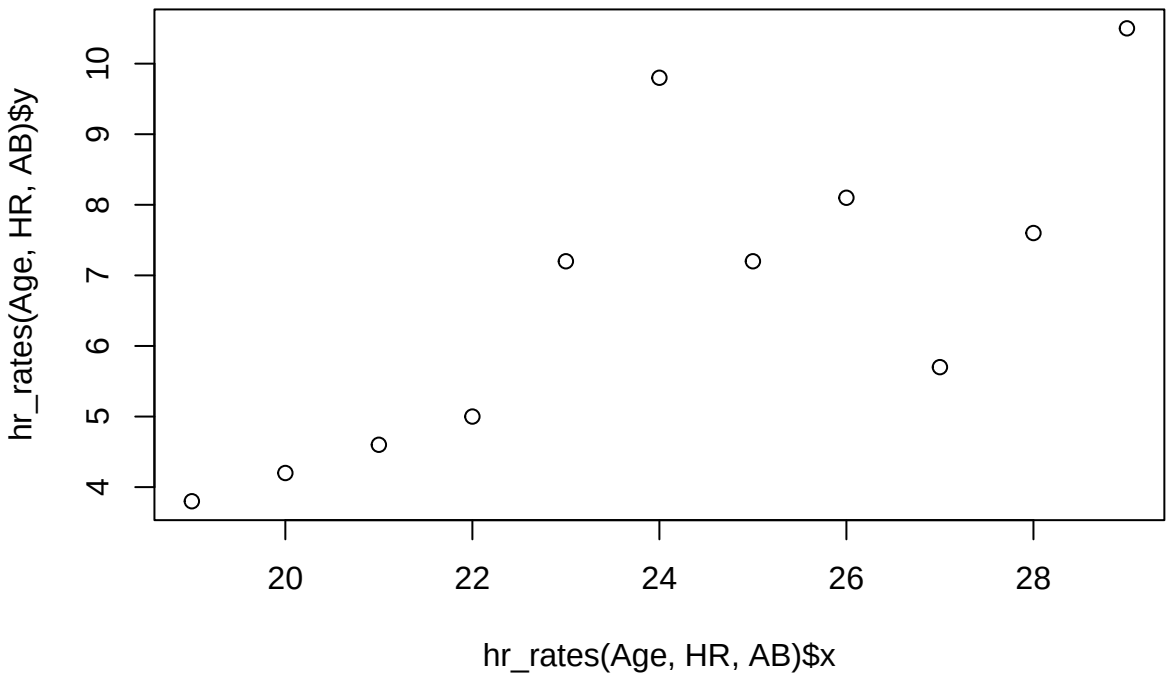
```
# Lists
world_series <- list(Winner=Winner, Number.Games=N_Games, Seasons="2008 to 2017")
world_series
```

```
## $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")
```




```
# 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")
```

```
# Splitting, Applying, and Combining data

# 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
```

```
# 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
```

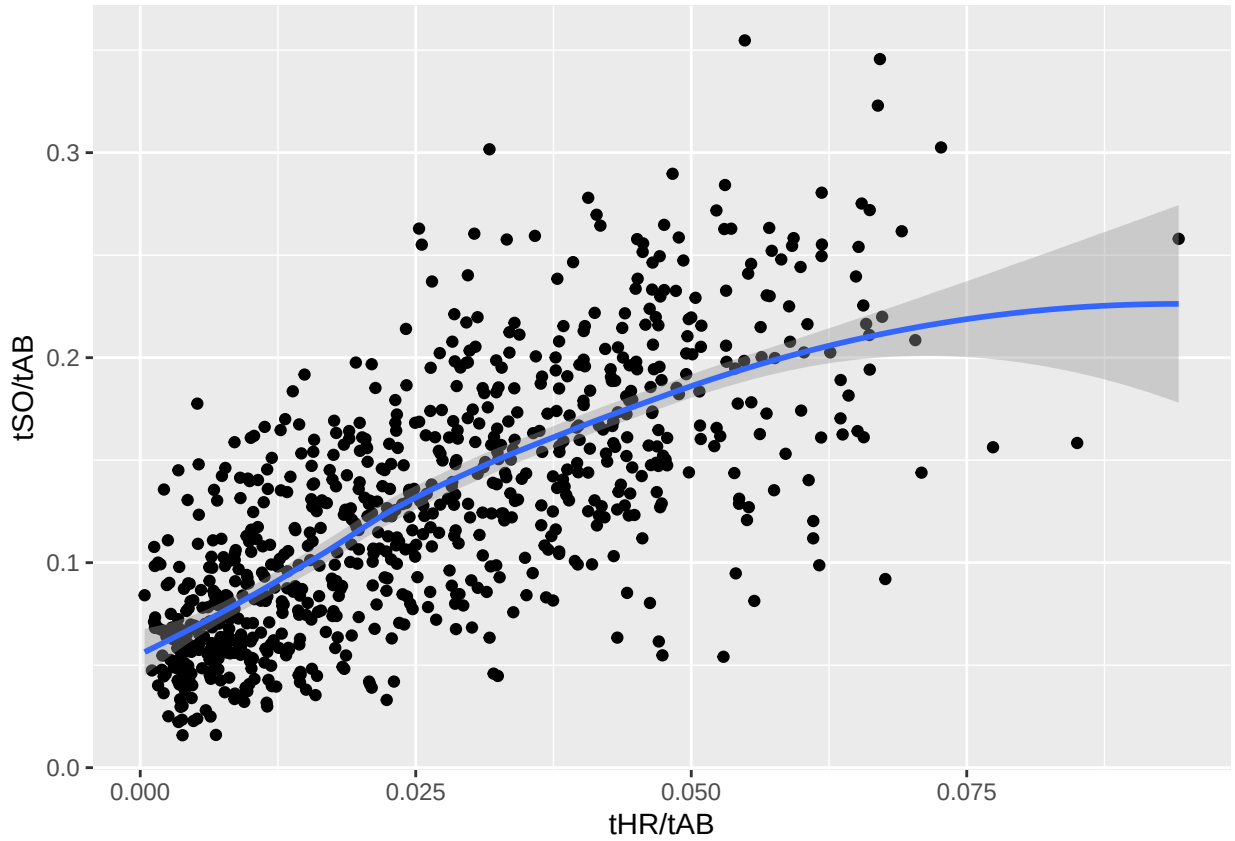
```
# Collect the career batting statistics
Batting %>%
    group_by(playerID) %>%
    summarize(tAB = sum(AB,na.rm = TRUE),
              tHR = sum(HR,na.rm = TRUE),
              tSO = sum(SO,na.rm = TRUE)) -> long_careers
```

```r
# 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
```

```r
# 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

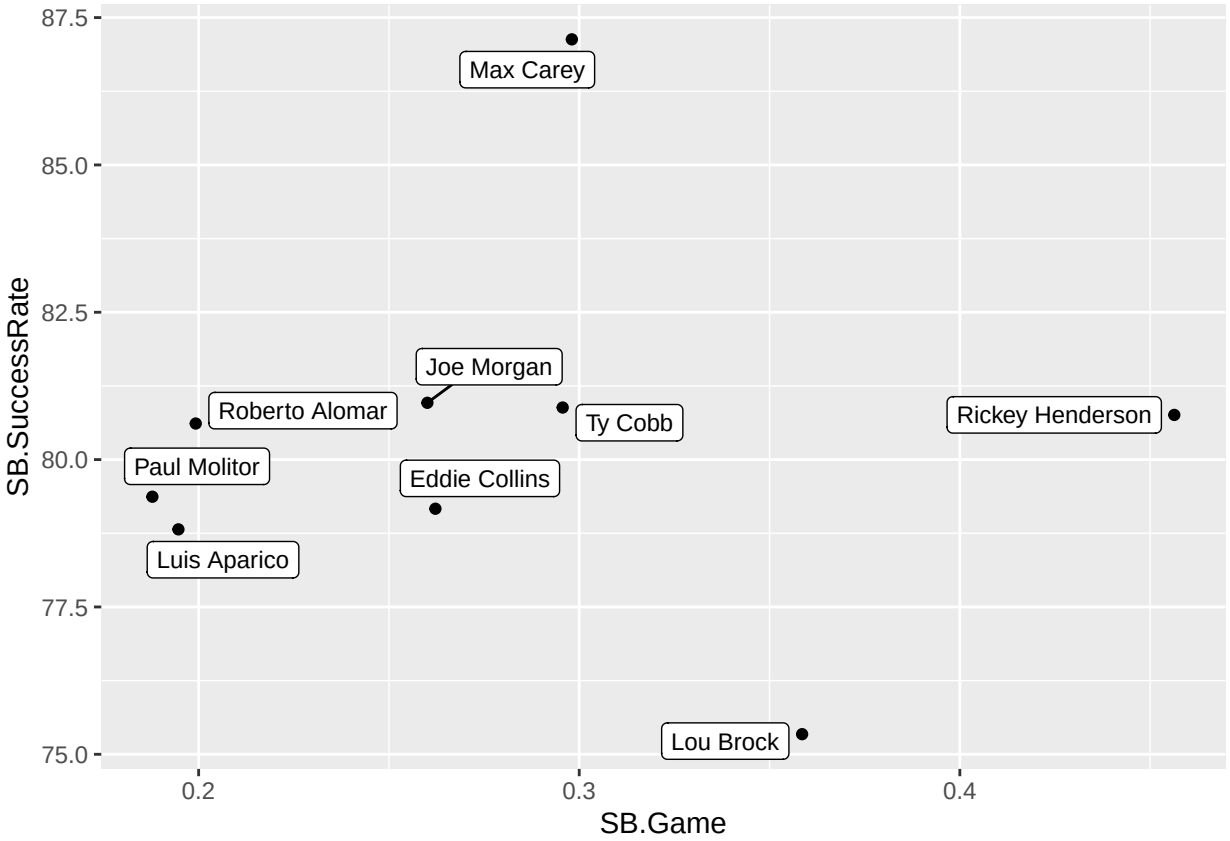
```
# (a) Create a data frame
players <- c("Rickey Henderson","Lou Brock","Ty Cobb","Eddie Collins","Max Carey","Joe Morgan","Luis Ap
SB <- c(1406,938,897,741,738,689,506,504,474)
CS <- c(335,307,212,195,109,162,136,131,114)
G <- c(3081,2616,3034,2826,2476,2649,2599,2683,2379)
sb_df <- data.frame(players,SB,CS,G)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

```
# (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)

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

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

```
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.