Analysis of Sports Big Data HW 2

2021311175 Jae-Hyun Lee

Chapter 3

1. Problem # 1 in Section 2.13 (Top Base Stealers in the Hall of Fame)

> The following table gives the number of stolen bases (SB), the number of times caught stealing (CS), and the number of games played (G) for nine players currently inducted in the Hall of Fame.

Player	SB	CS	G
Rickey Henderson	1406	335	3081
Lou Brock	938	307	2616
Ty Cobb	897	212	3034
Eddie Collins	741	195	2826
Max Carey	738	109	2476
Joe Morgan	689	162	2649
Luis Aparicio	506	136	2599
Paul Molitor	504	131	2683
Roberto Alomar	474	114	2379

(a) In `R`, place the stolen base, caught stealing, and game counts in the vectors `SB`, `CS`, and `G`.

```
> Player <- c("RH", "LB", "TC", "EC", "MC", "JM", "LA", "PM", "RA")
> 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)
```

(b) For all players, compute the number of stolen base attempts `SB` + `CS` and store in the vector `SB.Attempt`.

```
> SB.Attempt <- SB + CS
> SB.Attempt
[1] 1741 1245 1109  936  847  851  642  635  588
```

(c) For all players, compute the success rate `Success.Rate` = `SB` / `SB.Attempt`.

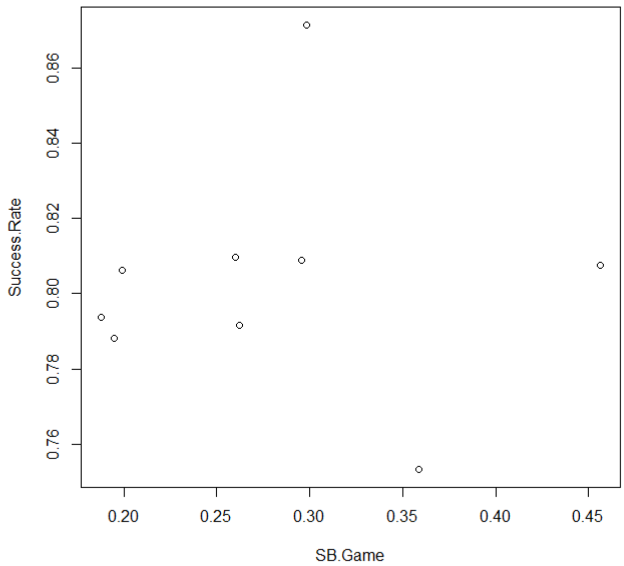
```
> Success.Rate <- SB / SB.Attempt
> Success.Rate
[1] 0.8075818 0.7534137 0.8088368 0.7916667 0.8713105 0.8096357 0.7881620
0.7937008
[9] 0.8061224
```

(d) Compute the number of stolen bases per game `SB.Game` = `SB` / `G`.

```
> SB.Game <- SB / G
> SB.Game
[1] 0.4563453 0.3585627 0.2956493 0.2622081 0.2980614 0.2600982 0.1946903
0.1878494
[9] 0.1992434
```

(e) Construct a scatter plot of the stolen bases per game against the success rate. Are there particular players with unusually high or low stolen base success rates? Which player had the greatest number of stolen bases per game?

```
> plot(SB.Game, Success.Rate)
> Player[rank(Success.Rate) == 1] # Worst success rate
[1] "LB"
> Player[rank(Success.Rate) == 9] # Best success rate
[1] "MC"
> Player[rank(SB.Game) == 9] # Best number of stolen bases per game
[1] "RH"
```



There are one player with unusually low stolen base success rate, and one player with unusually high success rate. The names are **Lou Brock** and **Max Carey** respectively. The player with best number of stolen bases per game is **Rickey Henderson**.

2. Problem # 5 in Section 2.13 (Pitcher Strikeout / Walk Ratios)

(a) Read the `Lahman Pitching` data into `R`.

```
> library(Lahman)
> head(Pitching, 1)
   playerID yearID stint teamID lgID W L G GS CG SHO SV IPouts  H ER HR BB SO
BAOpp  ERA
1 bechtge01   1871     1    PH1   NA 1 2 3  3  2   0  0     78 43 23  0 11  1
 NA 7.96
  IBB WP HBP BK BFP GF  R SH SF GIDP
1  NA  7  NA  0 146  0 42 NA NA   NA
```

(b) The following script computes the cumulative strikeouts, cumulative walks, mid career year, and the total innings pitched (measured in terms of outs) for all pitchers in the data file. This new data frame is named `career.pitching`. Run this code and use the `inner_join()` function to merge the `Pitching` and `career.pitching` data frames.

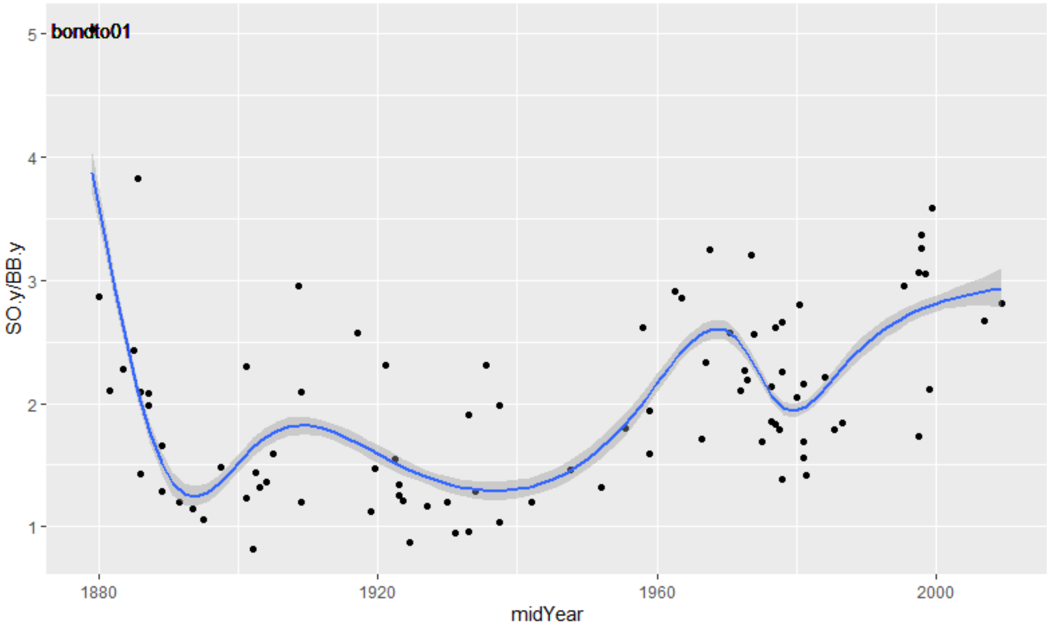
```
> career.pitching <- Pitching %>%
+   group_by(playerID) %>%
+   summarize(SO = sum(SO, na.rm = TRUE),
+             BB = sum(BB, na.rm = TRUE),
+             IPouts = sum(IPouts, na.rm = TRUE),
+             midYear = median(yearID, na.rm = TRUE))
> head(career.pitching)
# A tibble: 6 x 5
  playerID     SO    BB IPouts midYear
  <chr>     <int> <int>  <int>   <dbl>
1 aardsda01   340   183   1011    2009
2 aasedo01    641   457   3328    1984
3 abadfe01    280   116    992    2014.
4 abbeybe01   161   192   1704    1894.
5 abbeych01     0     0      6    1896
6 abbotda01     1     8     39    1890
>
> Pitching <- inner_join(Pitching, career.pitching, by = "playerID")
> head(Pitching, 1) #old: .x , new: .y
   playerID yearID stint teamID lgID W L G GS CG SHO SV IPouts.x  H ER HR BB.x
SO.x BAOpp
1 bechtge01   1871     1    PH1   NA 1 2 3  3  2   0  0       78 43 23  0   11
 1    NA
   ERA IBB WP HBP BK BFP GF  R SH SF GIDP SO.y BB.y IPouts.y midYear
1 7.96  NA  7  NA  0 146  0 42 NA NA   NA   10   22      729    1874
```

(c) Use the `filter()` function to construct a new data frame `career.10000` consisting of data for only those pitchers with at least 10, 000 career `IPouts`.

```
> Pitching %>%
+   filter(IPouts.y >= 10000) -> career.10000
```

(d) For the pitchers with at least 10, 000 career `IPouts`, construct a scatter plot of mid career year and ratio of strikeouts to walks. Comment on the general pattern in this scatter plot.

```
> ggplot(career.10000, aes(midYear, SO.y/BB.y)) +
+   geom_point() +
+   geom_smooth() +
+   geom_text(data = filter(career.10000, SO.y/BB.y > 5),
+                 aes(midYear, SO.y/BB.y, label = playerID))
```

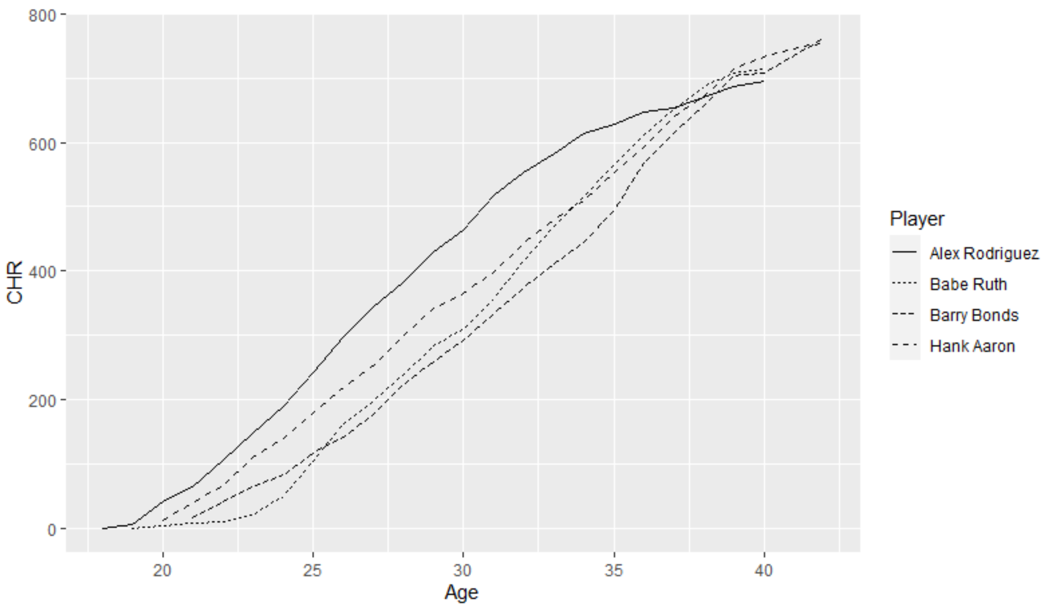


After the drastic decrease of the strikeouts-to-walks ratio in the late 19th century, there seems to be a slightly increasing trend until today. In the late 19th century there was a player whose strikeouts-to-walks ratio is over five; **Tommy Bonds**. Except for him, no player even reached over four in the ratio, which makes Bonds more legendary.

3. Section 3.7 of the Textbook: Reproduce Figure 3.15

```
> library(Lahman)
> get_birthyear <- function(Name) {
    Names <- unlist(strsplit(Name, " "))
    People %>%
      filter(nameFirst == Names[1],
             nameLast == Names[2]) %>%
      mutate(birthyear = ifelse(birthMonth >= 7,
                                birthYear + 1, birthYear),
             Player = paste(nameFirst, nameLast)) %>%
      select(playerID, Player, birthyear)
  }
> PlayerInfo <- bind_rows(get_birthyear("Babe Ruth"),
                          get_birthyear("Hank Aaron"),
                          get_birthyear("Barry Bonds"),
                          get_birthyear("Alex Rodriguez")
  )
> Batting %>%
    inner_join(PlayerInfo, by = "playerID") %>%
    mutate(Age = yearID - birthyear) %>%
```

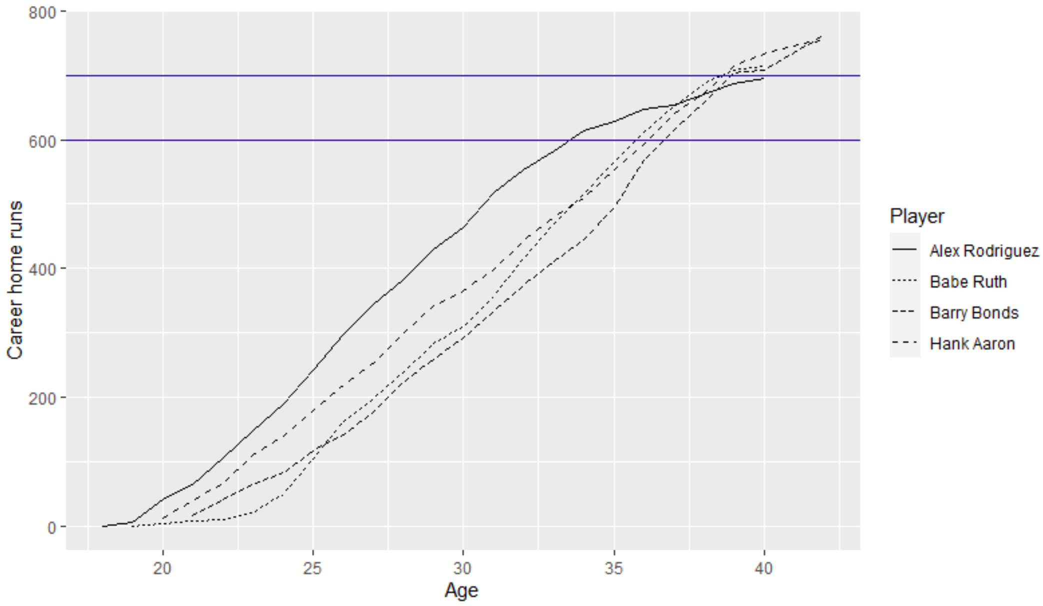
```
  select(Player, Age, HR) %>%
  group_by(Player) %>%
  mutate(CHR = cumsum(HR)) -> HRdata
> ggplot(HRdata, aes(x = Age, y = CHR, linetype = Player)) +
  geom_line()
```



- **Horizontal lines at 600 and 700**
- **Change the label for the vertical axis as "Career home runs".**

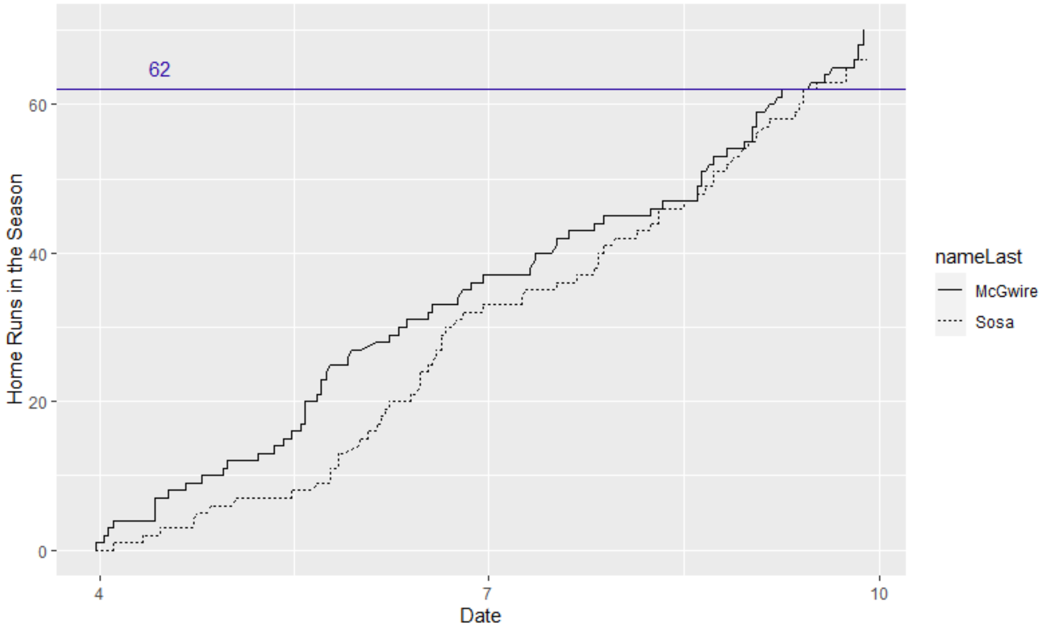
```
> fields <- read_csv(file.choose()) # Click the fields.csv wherever it is in
> data1998 <- read_csv(file.choose(),
                       col_names = pull(fields, Header))
>
> sosa_id <- People %>%
  filter(nameFirst == "Sammy", nameLast == "Sosa") %>%
  pull(retroID)
> mac_id <- People %>%
  filter(nameFirst == "Mark", nameLast == "McGwire") %>%
  pull(retroID)
> hr_race <- data1998 %>%
  filter(BAT_ID %in% c(sosa_id, mac_id))
> library(lubridate)
> cum_hr <- function(d) {
  d %>%
    mutate(Date = ymd(str_sub(GAME_ID, 4, 11))) %>%
    arrange(Date) %>%
    mutate(HR = ifelse(EVENT_CD == 23, 1, 0),
           cumHR = cumsum(HR)) %>%
    select(Date, cumHR)
 }
> hr_ytd <- hr_race %>%
  split(pull(., BAT_ID)) %>%
  map_df(cum_hr, .id = "BAT_ID") %>%
  inner_join(People, by = c("BAT_ID" = "retroID"))
> ggplot(hr_ytd, aes(Date, cumHR, linetype = nameLast)) +
  geom_line() +
  geom_hline(yintercept = 62, color = crcblue) +
  annotate("text", ymd("1998-04-15"), 65,
           label = "62", color = crcblue) +
```

```
    ylab("Home Runs in the Season")
```



4. Section 3.8 of the Textbook: Reproduce Figure 3.16

```
> fields <- read_csv(file.choose()) # Click the fields.csv whereever it is in
> data1998 <- read_csv(file.choose(),
                       col_names = pull(fields, Header))
>
> sosa_id <- People %>%
    filter(nameFirst == "Sammy", nameLast == "Sosa") %>%
    pull(retroID)
> mac_id <- People %>%
    filter(nameFirst == "Mark", nameLast == "McGwire") %>%
    pull(retroID)
> hr_race <- data1998 %>%
    filter(BAT_ID %in% c(sosa_id, mac_id))
> library(lubridate)
> cum_hr <- function(d) {
    d %>%
      mutate(Date = ymd(str_sub(GAME_ID, 4, 11))) %>%
      arrange(Date) %>%
      mutate(HR = ifelse(EVENT_CD == 23, 1, 0),
             cumHR = cumsum(HR)) %>%
      select(Date, cumHR)
  }
> hr_ytd <- hr_race %>%
    split(pull(., BAT_ID)) %>%
    map_df(cum_hr, .id = "BAT_ID") %>%
    inner_join(People, by = c("BAT_ID" = "retroID"))
> ggplot(hr_ytd, aes(Date, cumHR, linetype = nameLast)) +
    geom_line() +
    geom_hline(yintercept = 62, color = crcblue) +
    annotate("text", ymd("1998-04-15"), 65,
             label = "62", color = crcblue) +
    ylab("Home Runs in the Season")
```



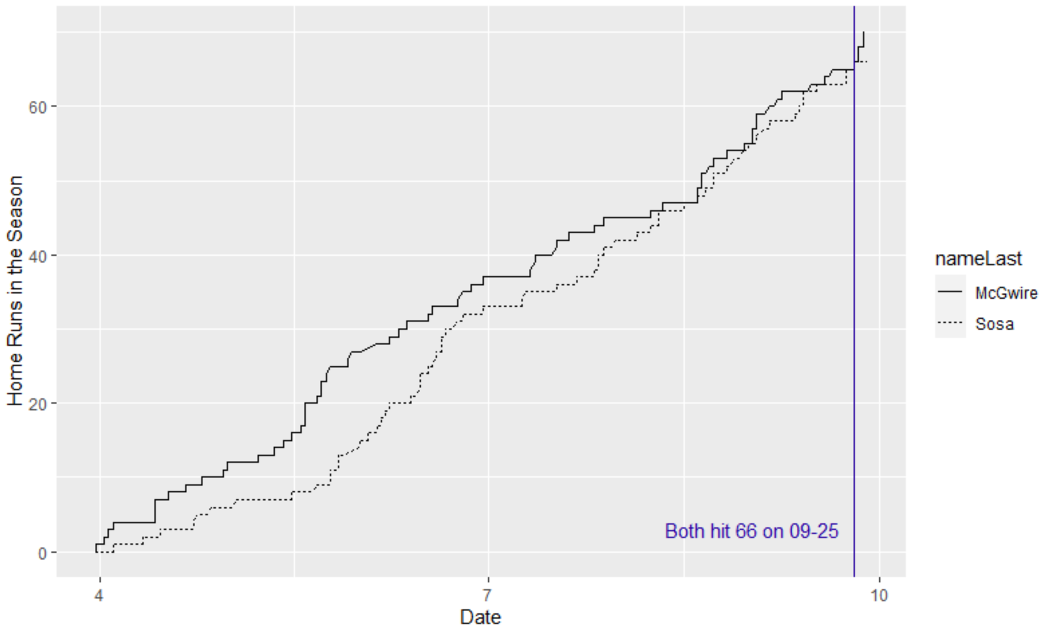
- **Identify when they were tied at 66 and add a vertical line on that day with a label saying “66”**

```
> hr_ytd %>%
    filter(cumHR == 66) %>%
    select(BAT_ID, cumHR, Date)
# A tibble: 20 x 3
   BAT_ID   cumHR Date
   <chr>    <dbl> <date>
 1 mcgwm001    66 1998-09-25
 2 mcgwm001    66 1998-09-25
 3 mcgwm001    66 1998-09-25
 4 mcgwm001    66 1998-09-26
 5 sosas001    66 1998-09-25
 6 sosas001    66 1998-09-25
 7 sosas001    66 1998-09-25
 8 sosas001    66 1998-09-26
 9 sosas001    66 1998-09-26
10 sosas001    66 1998-09-26
11 sosas001    66 1998-09-26
12 sosas001    66 1998-09-27
13 sosas001    66 1998-09-27
14 sosas001    66 1998-09-27
15 sosas001    66 1998-09-27
16 sosas001    66 1998-09-27
17 sosas001    66 1998-09-28
18 sosas001    66 1998-09-28
19 sosas001    66 1998-09-28
20 sosas001    66 1998-09-28
>
> hr_ytd %>%
    filter(Date == "1998-09-25" | Date == "1998-09-26") %>%
    select(BAT_ID, cumHR, Date) %>%
    arrange(Date)
# A tibble: 18 x 3
   BAT_ID   cumHR Date
   <chr>    <dbl> <date>
 1 mcgwm001    65 1998-09-25
 2 mcgwm001    65 1998-09-25
```

```
 3 mcgwm001    66 1998-09-25
 4 mcgwm001    66 1998-09-25
 5 mcgwm001    66 1998-09-25
 6 sosas001    65 1998-09-25
 7 sosas001    66 1998-09-25
 8 sosas001    66 1998-09-25
 9 sosas001    66 1998-09-25
10 mcgwm001    66 1998-09-26
11 mcgwm001    67 1998-09-26
12 mcgwm001    67 1998-09-26
13 mcgwm001    68 1998-09-26
14 mcgwm001    68 1998-09-26
15 sosas001    66 1998-09-26
16 sosas001    66 1998-09-26
17 sosas001    66 1998-09-26
18 sosas001    66 1998-09-26
```

Rather than working on the magical function that just automatically finds the date of the home run and draws a vertical line, I followed a primitive approach. First, I looked up the dates when both players hit 66 home runs, and noticed from McGwire's data that the date is around 09-25 ~ 09-26. Then I investigated what happened on those two days and concluded that on **09-25**, after the game, **both were tied at 66** home runs. The day after, which was 09-26, McGwire was able to lead the race, adding another two home runs to his career. I added the exact date in `geom_vline`, with a label containing more information than suggested.

```
ggplot(hr_ytd, aes(Date, cumHR, linetype = nameLast)) +
  geom_line() +
  geom_vline(xintercept = ymd("1998-09-25"), color = crcblue) +
  annotate("text", ymd("1998-09-01"), 3,
           label = "Both hit 66 on 09-25", color = crcblue) +
  ylab("Home Runs in the Season")
```



5. Problem # 7 in Section 3.10 (Working with the Retrosheet Play-by-Play Dataset)

> In Section 3.8, we used the Retrosheet play-by-play data to explore the home run race between Mark McGwire and Sammy Sosa in the 1998 season. Another way to compare the patterns of home run hitting of the two players is to compute the spacings, the number of plate appearances between home runs.

(a) Following the work in Section 3.8, create the two data frames `mac.data` and `sosa.data` containing the batting data for the two players.

```
> sosa_id <- People %>%
    filter(nameFirst == "Sammy", nameLast == "Sosa") %>%
    pull(retroID)
> mac_id <- People %>%
    filter(nameFirst == "Mark", nameLast == "McGwire") %>%
    pull(retroID)
>
> data1998 %>%
    filter(BAT_ID == sosa_id) -> sosa.data
>
> data1998 %>%
    filter(BAT_ID == mac_id) -> mac.data
```

(b) Use the following R commands to restrict the two data frames to the plays where a batting event occurred.

```
> mac.data <- filter(mac.data, BAT_EVENT_FL == TRUE)
> sosa.data <- filter(sosa.data, BAT_EVENT_FL == TRUE)
```

(c) For each data frame, create a new variable `PA` that numbers the plate appearance 1, 2, . . .

```
> mac.data <- mutate(mac.data, PA = 1:nrow(mac.data))
> sosa.data <- mutate(sosa.data, PA = 1:nrow(sosa.data))
```

(d) The following commands will return the numbers of the plate appearances when the players hit home runs.

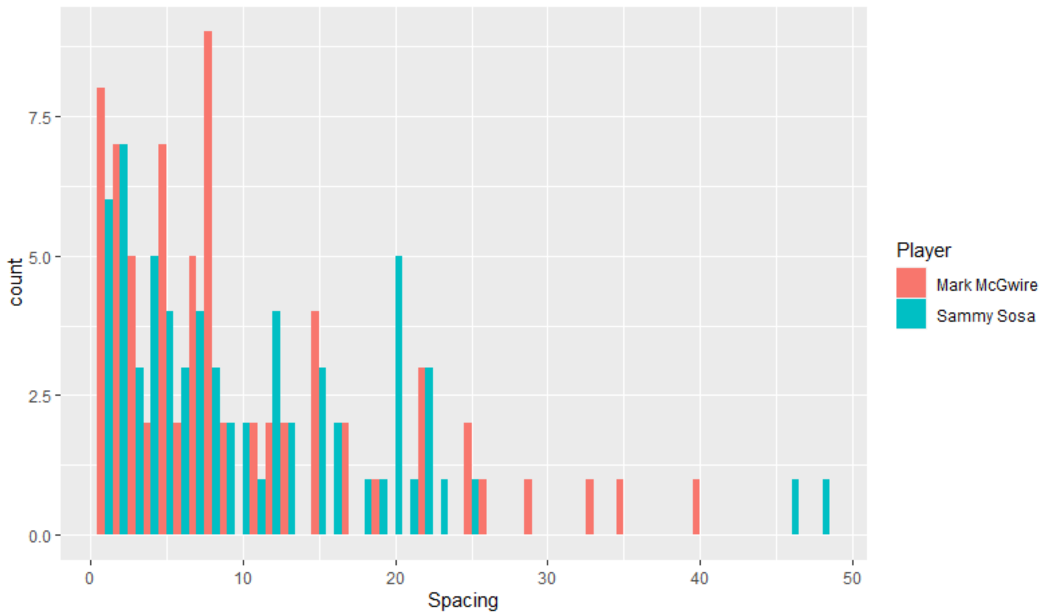
```
> mac.HR.PA <- mac.data %>%
    filter(EVENT_CD == 23) %>%
    pull(PA)
> sosa.HR.PA <- sosa.data %>%
    filter(EVENT_CD == 23) %>%
    pull(PA)
```

(e) Using the `R` function `diff()`, the following commands compute the spacings between the occurrences of home runs. Create a new data frame `HR_Spacing` with two variables, `Player`, the player name, and `Spacing`, the value of the spacing.

```
> mac.spacings <- diff(c(0, mac.HR.PA))
> sosa.spacings <- diff(c(0, sosa.HR.PA))
>
> HR_Spacing <- rbind(cbind("Mark McGwire", mac.spacings),
                    cbind("Sammy Sosa", sosa.spacings))
> colnames(HR_Spacing) <- c("Player", "Spacing")
> HR_Spacing <- data.frame(HR_Spacing)
> HR_Spacing <- transform(HR_Spacing, Spacing=as.numeric(Spacing))
```

(f) By use of the `summarize()` and `geom_histogram()` functions on the data frame HR Spacing, compare the home run spacings of the two players.

```
> ggplot(HR_Spacing, aes(x=Spacing, group=Player, fill=Player)) +
  geom_histogram(binwidth = 1, position = "dodge")
```



- Overall **Mark McGwire seems to hit his next home run earlier** than Sammy Sosa. McGwire's distribution seems more left-shifted.
- "Spacing = 1" means that as a player hits a home run, he hits another at his next PA.
- There were **two big slumps for Sosa**, being not able to hit a home run for more than 45 PA.
- McGwire had a few slumps too, but the difference was that they weren't as long as Sosa's

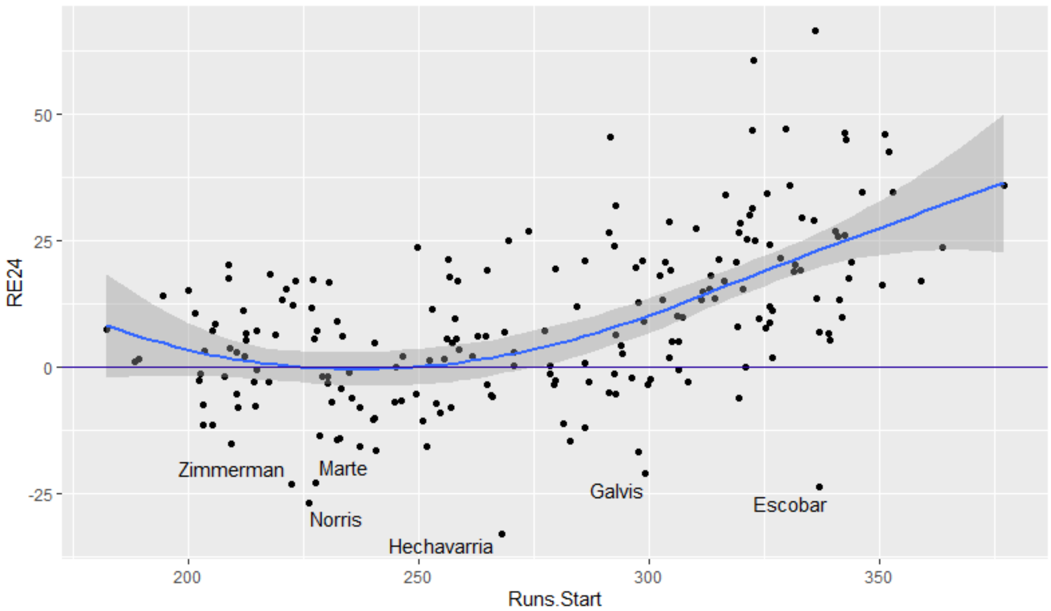
Chapter 5

1. `RE24` for Batters with 400 PAs or More

(a) Identify the batters whose `RE24` values are smaller than -20. Who are they?

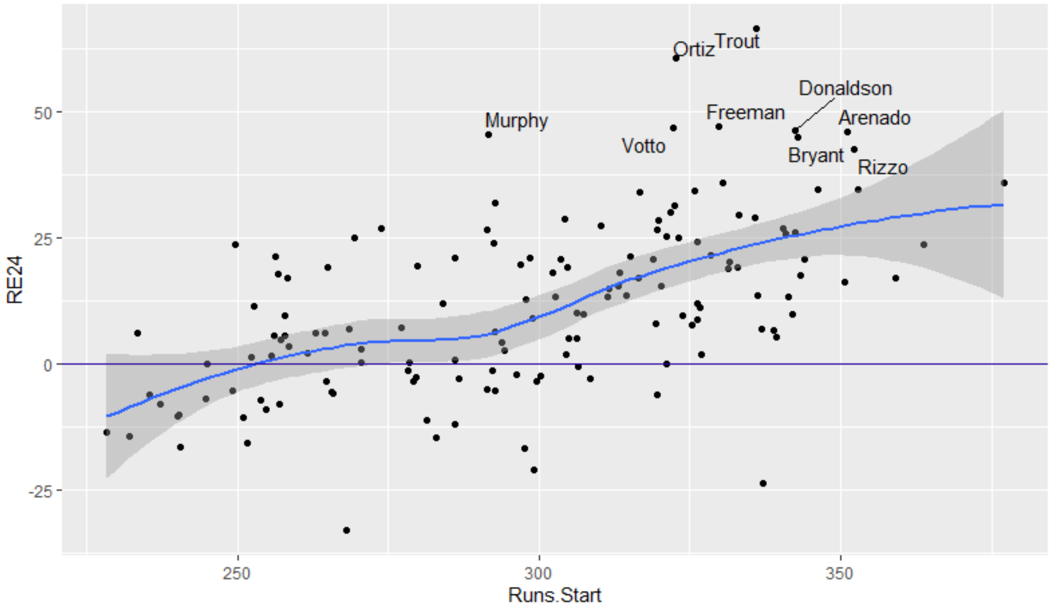
The codes are too long, thus only the relevant lines are introduced. Please check `Sports_HW2_LeeJH.R` for full codes including the lines from the lecture note.

```
> ggplot(runs400, aes(Runs.Start, RE24)) +
    geom_point() +
    geom_smooth() +
    geom_hline(yintercept = 0, color = crcblue) +
    geom_text_repel(data = filter(runs400, RE24 < -20),
                    aes(label = nameLast))
```



(b) Change this condition to 502 PAs or more and reconstruct Figure 5.2. Are there any changes in the overall pattern?

```
> runs %>%
    filter(PA >= 502) -> runs502 # restricting players
>
> runs502 %>%
    inner_join(People, by = c("BAT_ID" = "retroID")) -> runs502
>
> ggplot(runs502, aes(Runs.Start, RE24)) +
    geom_point() +
    geom_smooth() +
    geom_hline(yintercept = 0, color = crcblue) +
    geom_text_repel(data = filter(runs502, RE24 >= 40),
                    aes(label = nameLast))
```



- The overall increasing trend seems to be robust. **The more the opportunity, the higher the `RE24` value**.
- The x-axis `Runs.Start` can be interpreted as number of chances for batters. Since the lower bound of PA has increased, the **lower bound of the number of chances also increased**. Thus the decreasing trend around "`Runs.Start` ≈ 200" is trimmed away.
- Players such that "`RE24` > 40": Murphy, Votto, Bryant, Freeman, Ortiz, Trout, Donaldson, Arenado and Rizzo (from lecture note Figure 5.2) all appeared on more than 502 plates.

2. Run Values for Doubles and Triples

Doubles

```
> data2016 %>% filter(EVENT_CD == 21) -> doubles
>
> double_STATE <- cbind(
  matrix(table(doubles$STATE),8,3,byrow=T),
  matrix(round(prop.table(table(doubles$STATE)),2),8,3,byrow=T))
> dimnames(double_STATE)[[2]] <- c("0 outs", "1 out", "2 outs",
                                   "0 outs", "1 out", "2 outs")
> dimnames(double_STATE)[[1]] <- c("000", "001", "010", "011",
                                   "100", "101", "110", "111")
> double_STATE
    0 outs 1 out 2 outs 0 outs 1 out 2 outs
000   2194  1443   1132   0.27  0.17   0.14
001     25    79    111   0.00  0.01   0.01
010    158   196    266   0.02  0.02   0.03
011     19    50     59   0.00  0.01   0.01
100    501   553    545   0.06  0.07   0.07
101     37    88     99   0.00  0.01   0.01
110    101   191    220   0.01  0.02   0.03
111     29    70     88   0.00  0.01   0.01
```

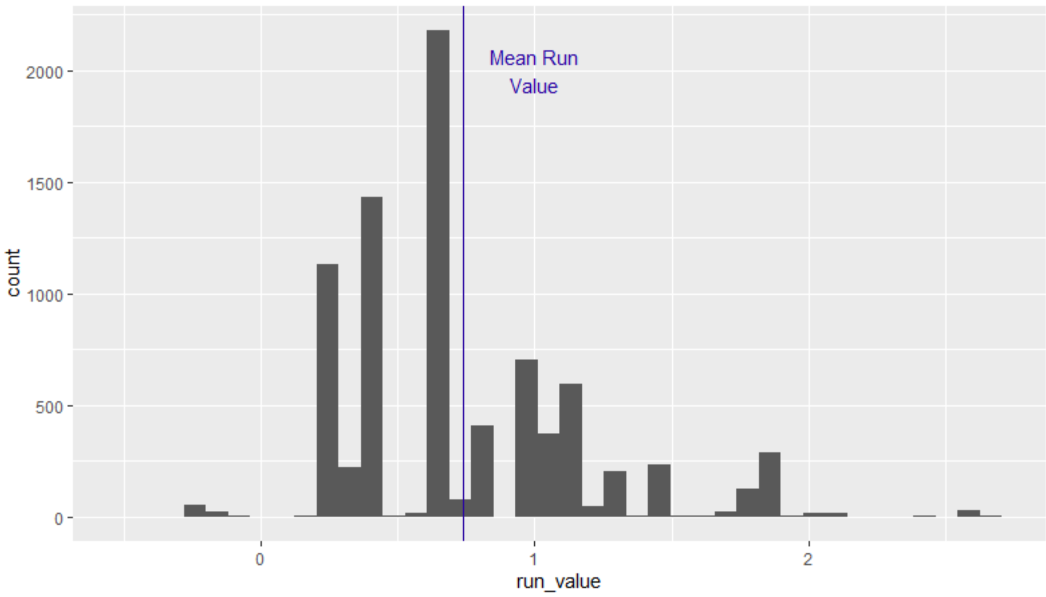
`double_STATE` counts states for each double. The numbers are on the left, and the proportions are on the right. More than half of the doubles were hit when no runners were on base.

```
> mean_doubles <- doubles %>%
    summarize(mean_run_value = mean(run_value))
> med_doubles <- doubles %>%
    summarize(median_run_value = median(run_value))
> c(mean_doubles, med_doubles)
$mean_run_value
[1] 0.739

$median_run_value
[1] 0.635
```

The run value of a double varies on situations. Difference in state is one factor of the variation and unexpected errors by fielders could be a another. Thus, the run value is estimated in point, in this case, using the classic mean and median.

```
> ggplot(doubles, aes(run_value)) +
    geom_histogram(bins = 40) +
    geom_vline(data = mean_doubles, color = crcblue,
               aes(xintercept = mean_run_value)) +
    annotate("text", 1, 2000,
             label = "Mean Run\nValue", color = crcblue)
```




```
> doubles %>%
    group_by(STATE) %>%
    summarize(mean_run_value = mean(run_value)) -> double_RV_STATE
> double_RV_STATE <- matrix(double_RV_STATE$mean_run_value, 8, 3, byrow = T)
> dimnames(double_RV_STATE)[[2]] <- c("0 outs", "1 out", "2 outs")
> dimnames(double_RV_STATE)[[1]] <- c("000", "001", "010", "011",
                                      "100", "101", "110", "111")
> double_RV_STATE
     0 outs 1 out 2 outs
000   0.631 0.404  0.206
001   0.795 0.722  0.938
010   0.979 0.986  0.992
011   1.205 1.276  1.764
100   1.120 0.933  0.689
101   1.260 1.210  1.469
110   1.532 1.534  1.551
```

```
111  1.901 1.908  2.123
```

Rather than following the codes introduced in the lecture notes, I created a `double_RV_STATE` matrix, which demonstrates the run value of doubles per each state.

- The most valuable double occurs when there are **bases loaded with two outs**.
- The least valuable double occurs when there is **no runner on base with two outs**. This value is close to the difference between two values in `RUNS_out` in the lecture note.

```
> RUNS_out
    0 outs 1 out 2 outs
000   0.50  0.27   0.11
001   1.35  0.94   0.37
010   1.13  0.67   0.31
011   1.93  1.36   0.55
100   0.86  0.51   0.22
101   1.72  1.20   0.48
110   1.44  0.92   0.41
111   2.11  1.54   0.70
```

0.31 (from "010", 2 outs) **− 0.11** (from "000", 2 outs) **= 0.2**
Also, check the following for interpreting the run value of triples on "000", 2 outs.
0.37 (from "001", 2 outs) **− 0.11** (from "000", 2 outs) **= 0.26**

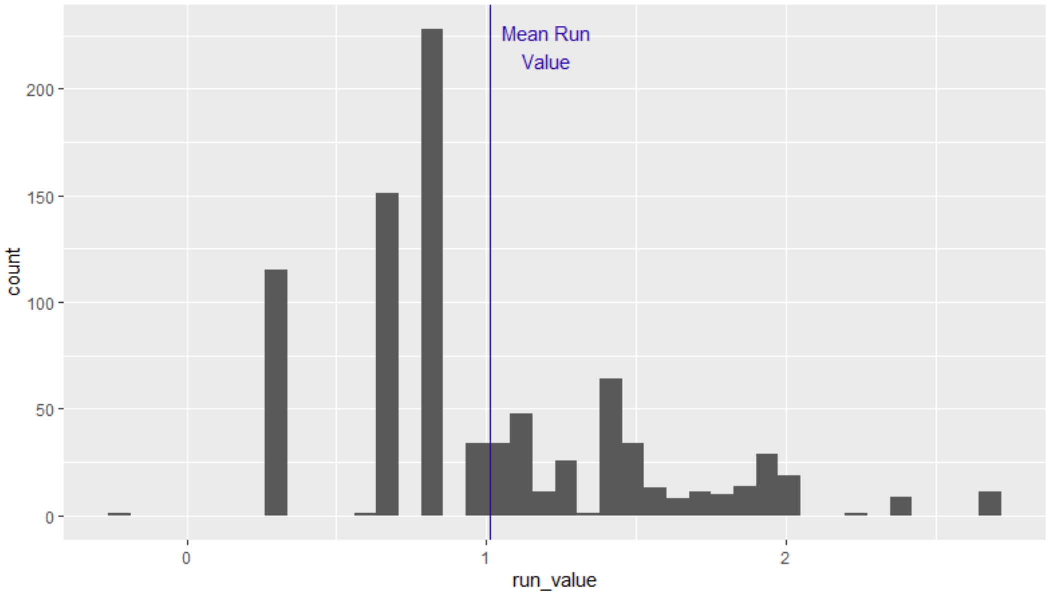
Triples

Similar analysis can be applied to triples. Of course, a triple is superior to a double, but there seems to be no substantial difference in interpreting the two. The explanation for doubles holds for triples, so only the results are excerpted. Please check `Sports_HW2_LeeJH.R` for full codes.

```
> triple_STATE
    0 outs 1 out 2 outs 0 outs 1 out 2 outs
000    233   150    115   0.27  0.17   0.13
001      3    11     16   0.00  0.01   0.02
010     12    27     35   0.01  0.03   0.04
011      4    12     10   0.00  0.01   0.01
100     34    61     48   0.04  0.07   0.05
101      8    11     14   0.01  0.01   0.02
110      6    19     23   0.01  0.02   0.03
111      1     9     11   0.00  0.01   0.01
```

```
> c(mean_triples, med_triples)
$mean_run_value
[1] 1.01

$median_run_value
[1] 0.849
```




```
> triple_RV_STATE
     0 outs 1 out 2 outs
000  0.847 0.669  0.266
001  1.000 1.000  1.000
010  1.226 1.276  1.049
011  1.418 1.579  1.824
100  1.489 1.411  1.152
101  1.624 1.741  1.894
110  1.902 2.016  1.958
111  2.241 2.400  2.677
```