Analyzing Baseball Data with R - Graphics

Tomoka Takegaki

24/11/2021

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 "Graphics".

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)
```

3.2 Character Variable

```
hof <- read_csv("../data/csv_files/hofbatting.csv")
```

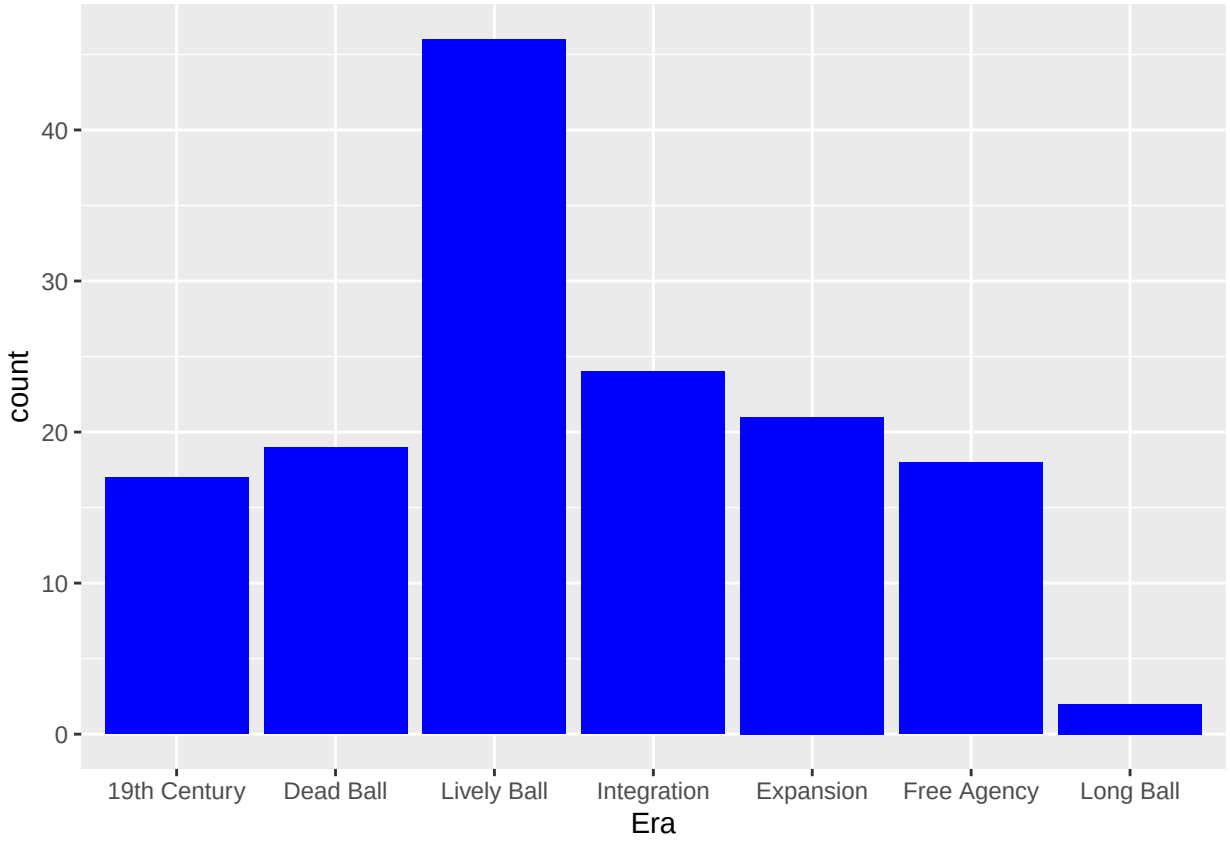
```
## New names:
## * `` -> ...2

## Rows: 147 Columns: 25

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): ...2
## dbl (24): Rk, Inducted, Yrs, From, To, ASG, WAR/pos, G, PA, AB, R, H, 2B, 3B...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# 3.2.1 Bar graph
# Create columns "MidCareer" and "Era".
hof <- hof %>%
    mutate(MidCareer = (From + To) / 2,
           Era = cut(MidCareer,
                     breaks = c(1800,1900,1919,1941,1960,1976,1993,2050),
                     labels = c("19th Century","Dead Ball",
                                "Lively Ball","Integration",
                                "Expansion","Free Agency","Long Ball")))

# Frequency table of variable Era.
hof_eras <- summarize(group_by(hof,Era), N=n())
hof_eras
```

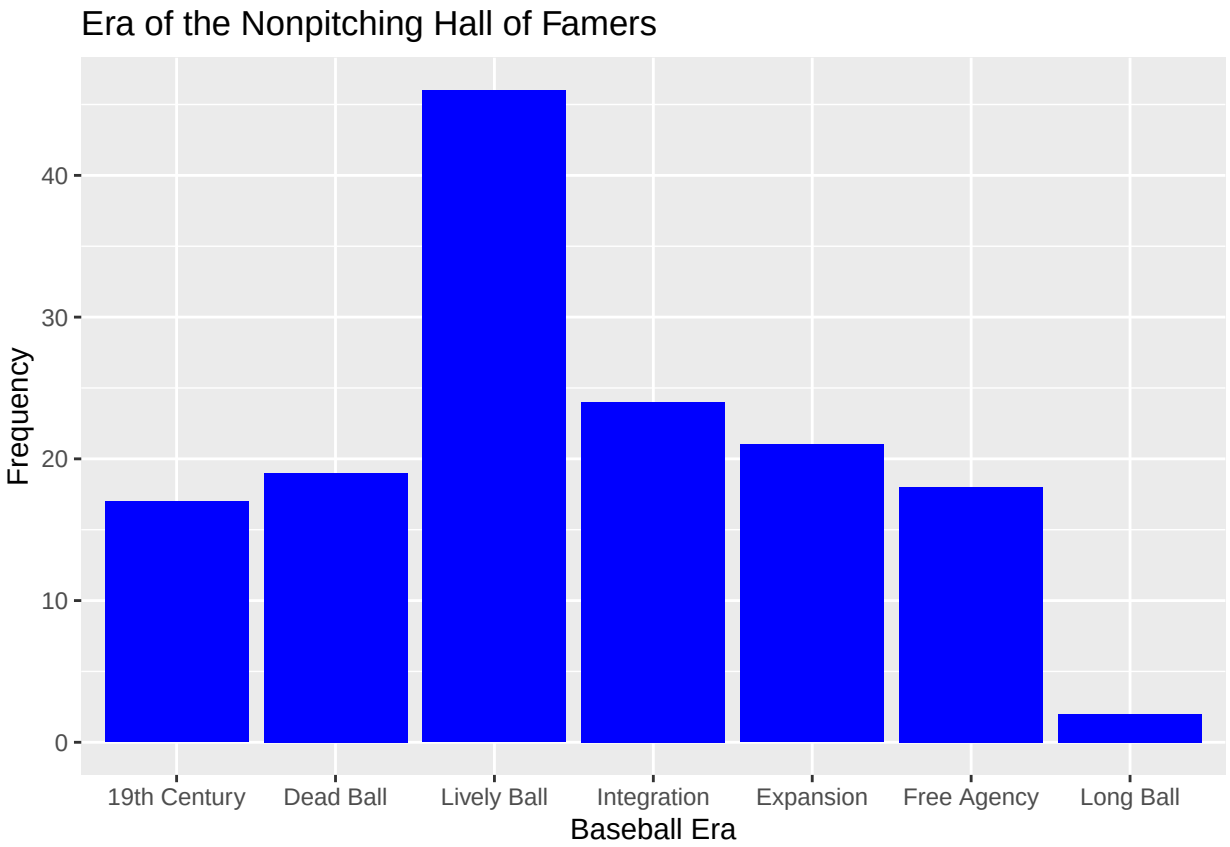
```
## # A tibble: 7 x 2
##   Era              N
##   <fct>        <int>
## 1 19th Century    17
## 2 Dead Ball       19
## 3 Lively Ball     46
## 4 Integration     24
## 5 Expansion       21
## 6 Free Agency     18
## 7 Long Ball        2
```

```
# Plot a bar graph
ggplot(hof,mapping = aes(x=Era)) +
    geom_bar(fill="blue")
```



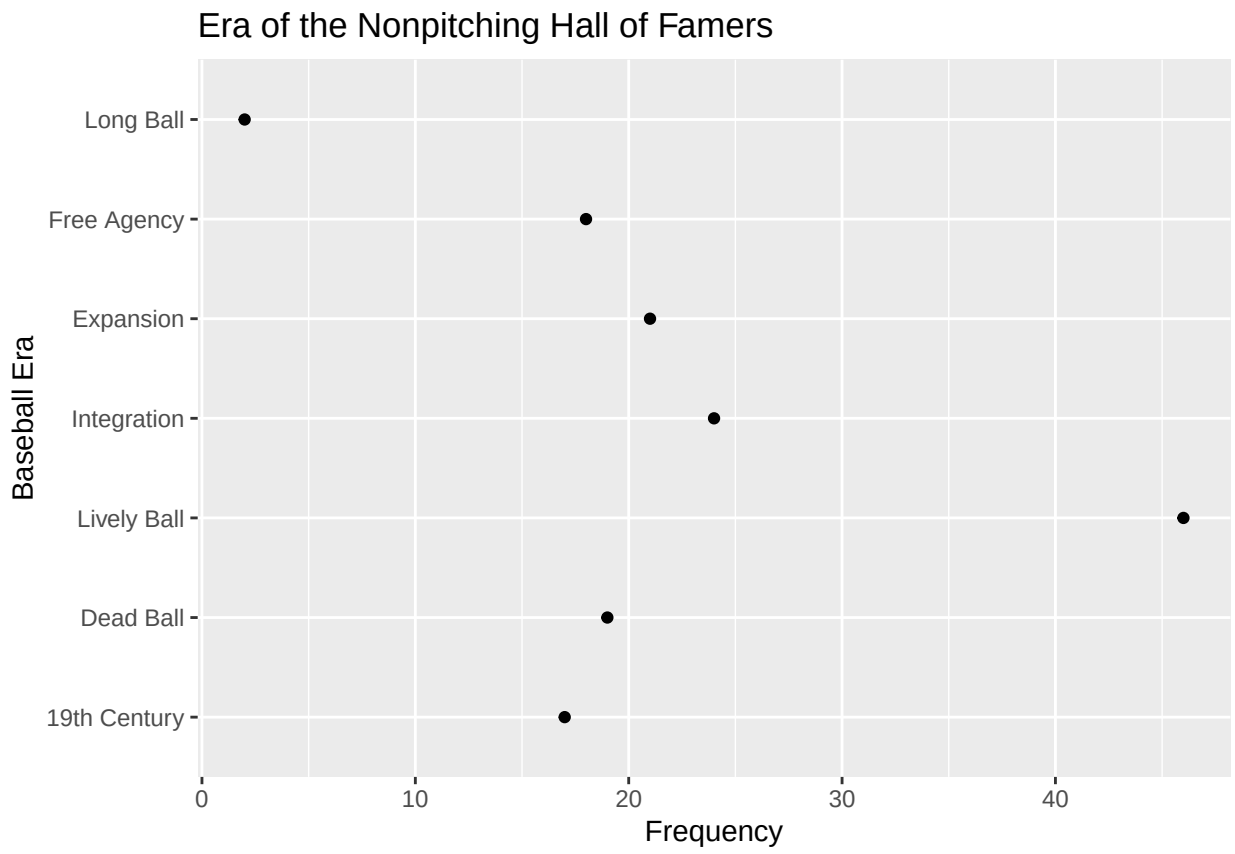
Let's add some axes labels and title to make it easy to understand the bar graph.

```
ggplot(hof,aes(Era))+
    geom_bar(fill="blue")+
    xlab("Baseball Era")+
    ylab("Frequency")+
    ggtitle("Era of the Nonpitching Hall of Famers")
```



A dot plot is helpful when there are a large number of categories of the character vector. coord_flip() function can switch x and y.

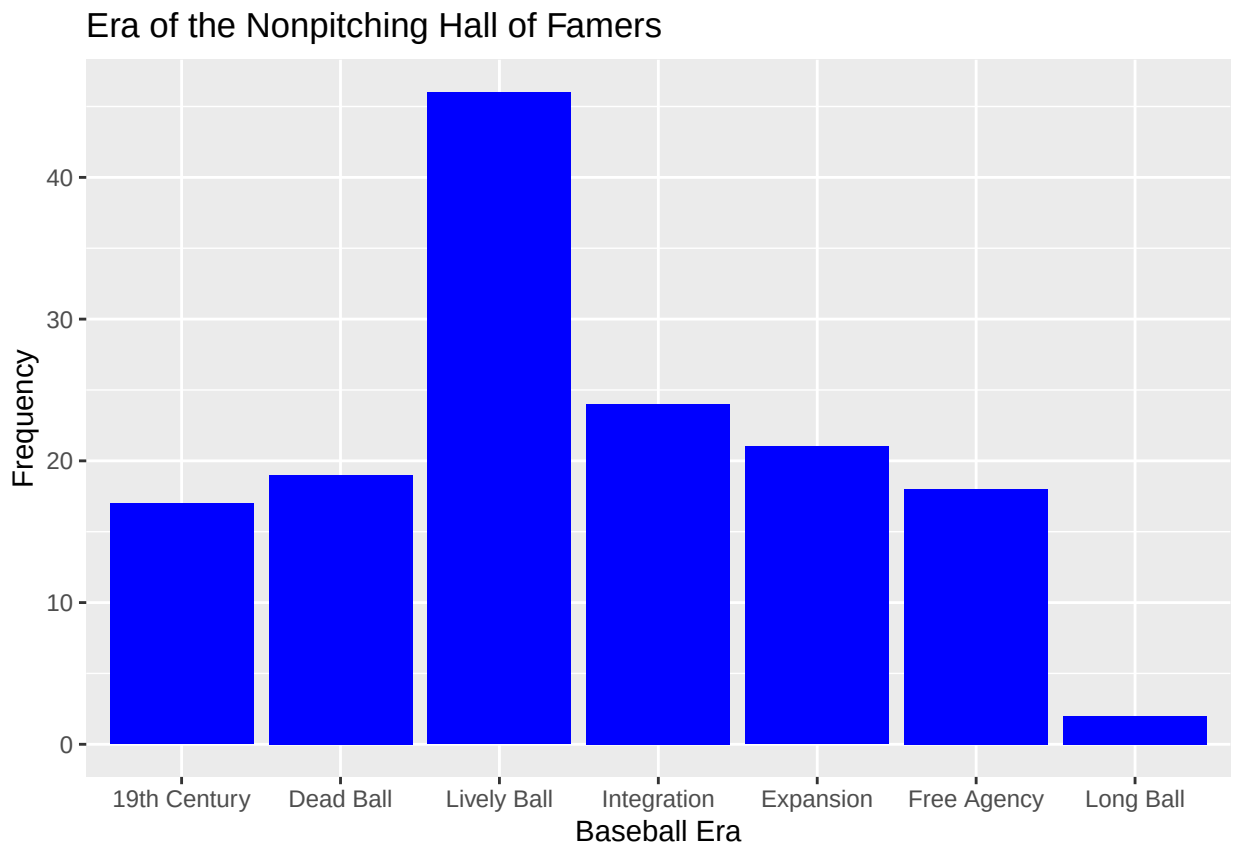
```
# 3.2.3 Other graphs of a character variable
ggplot(hof_eras,aes(Era,N))+
    geom_point() +
    xlab("Baseball Era")+
    ylab("Frequency")+
    ggtitle("Era of the Nonpitching Hall of Famers")+
    coord_flip()
```



3.3 Saving Graphs

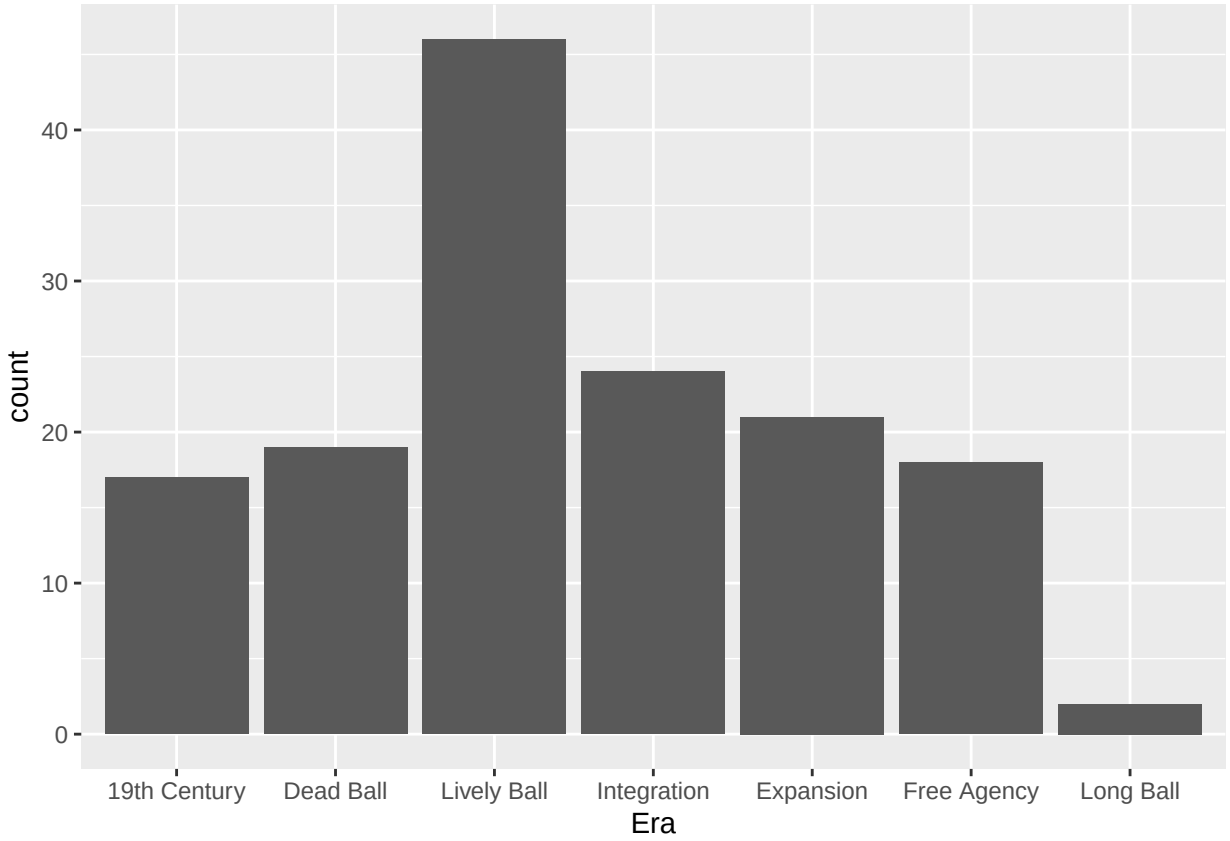
Suppose we wish to save the era bar graph in PNG format. We first type the R command to produce the graph. Then we use special ggsave() function where the argument is the name of the saved graphics file.

```
ggplot(hof,aes(Era))+
    geom_bar(fill="blue")+
    xlab("Baseball Era")+
    ylab("Frequency")+
    ggtitle("Era of the Nonpitching Hall of Famers")
```




```
#ggsave("graphs/bargraph.png")
```

```
# Saving 2 graphs in a pdf file
#pdf("graphs/graphs.pdf")
ggplot(hof,aes(Era))+geom_bar()
```




```
#ggplot(hof_eras,aes(Era,N))+geom_point()
#dev.off()
```

PNG and PDF files are successfully saved in the graphs folder.

3.4 Numeric Varible: One-Dimensional Scatterplot and Histogram

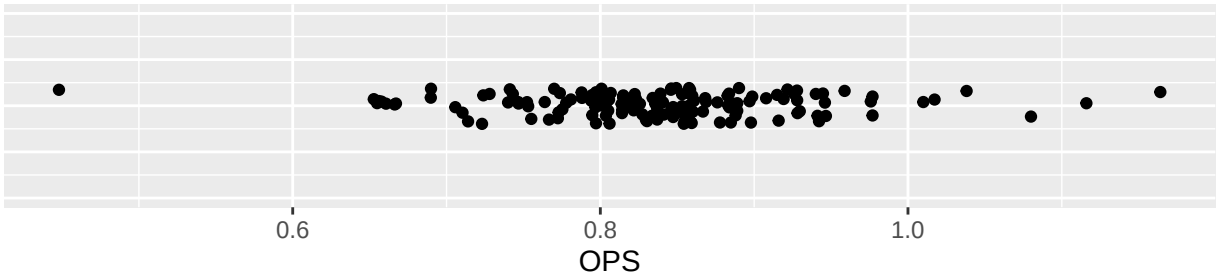
Two useful displays for visualizing a distribution are the one-dimensional scatterplot and the histgram.

We construct a graph of the OPS values for the Hall of Fame inductees in ggplot2 by the geom_jitter() function.

The theme() elements are chosen to remove the tick marks, text, and title from the y-axis.

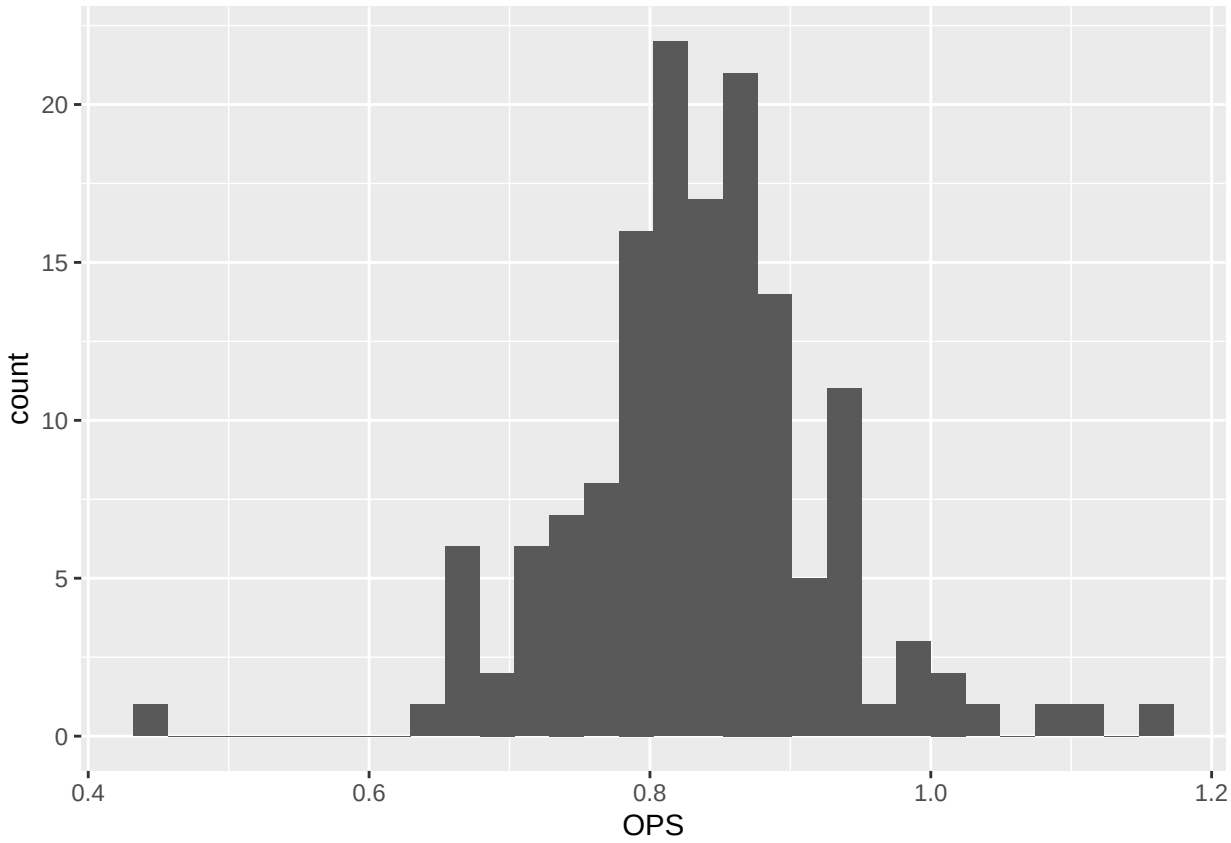
```
# One-Dimentional scatterplot
ggplot(hof, aes(x=OPS, y=1))+
    geom_jitter(hight = 0.6) + ylim(-1,3) +
    theme(axis.title.y = element_blank(),
          axis.text.y = element_blank(),
          axis.ticks.y = element_blank()) +
    coord_fixed(ratio = 0.03)
```

```
## Warning: Ignoring unknown parameters: hight
```



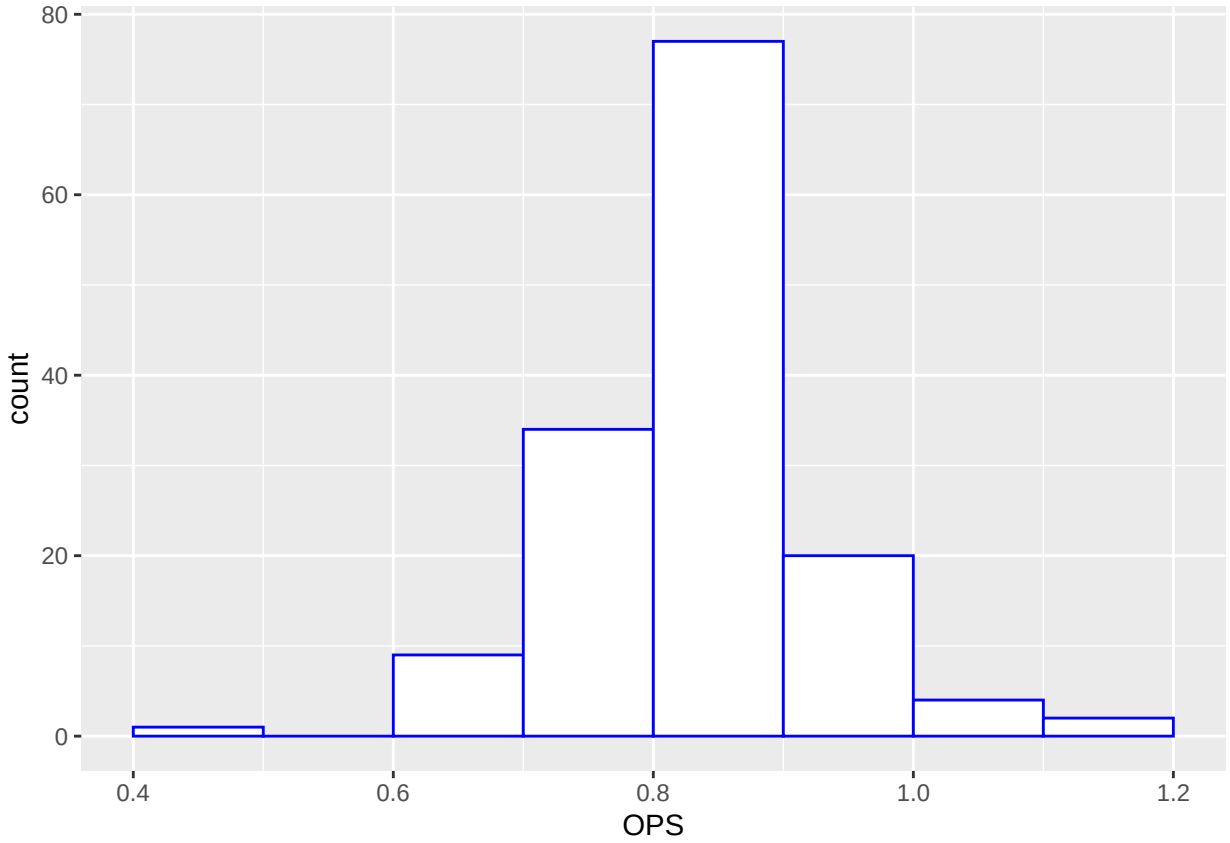

```
# Histogram
ggplot(hof,aes(x=OPS))+
    geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



One issue in constructing histogram is the choice of bins. We can use of the argument "beaks" to set number of bins.

```
ggplot(hof,aes(x=OPS))+
    geom_histogram(breaks = seq(0.4,1.2,by=0.1),
                   color = "blue", fill = 'white')
```



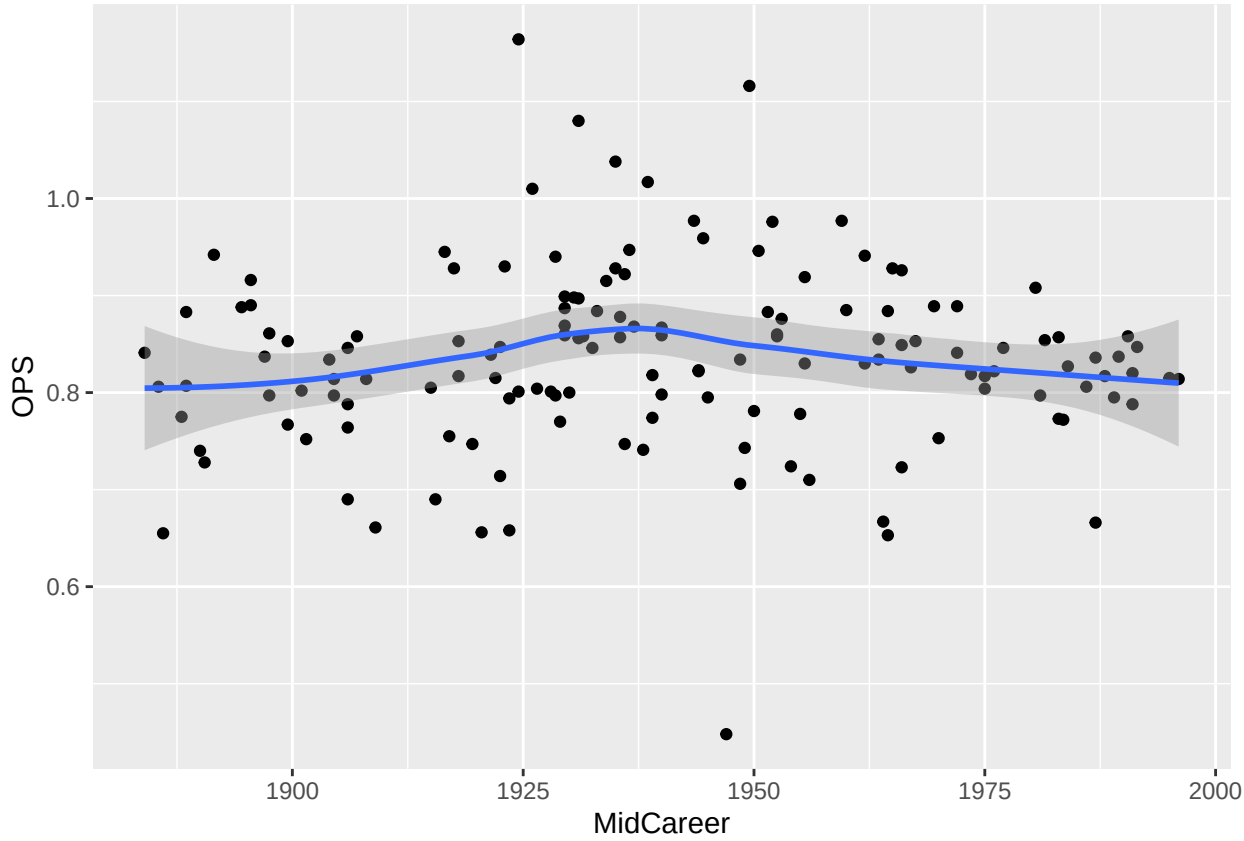
This is more readable than the previous graph without colors.

3.5 Two Numeric Variables

Is there any relationship between a player's OPS and the baseball era? Were there particular seasons where the Hall of Fame OPS values were unusually high or low?

```
# 3.5.1 Scatterplot
ggplot(hof, aes(MidCareer, OPS)) +
    geom_point() + geom_smooth()
```

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



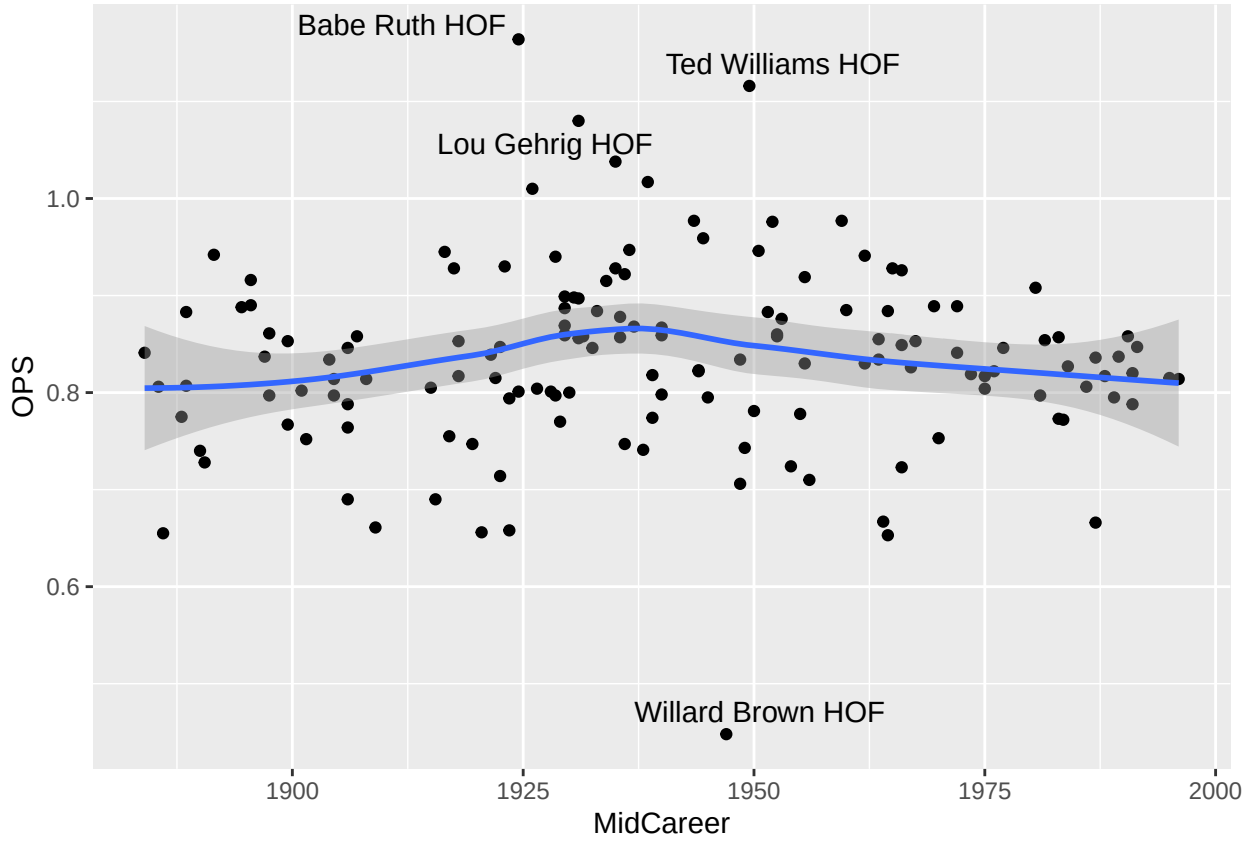
We can see that there are four unusual career OPS values. Three large values and one small value. We would like to identify the players with these extreme values.

```
library(ggrepel)
```

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

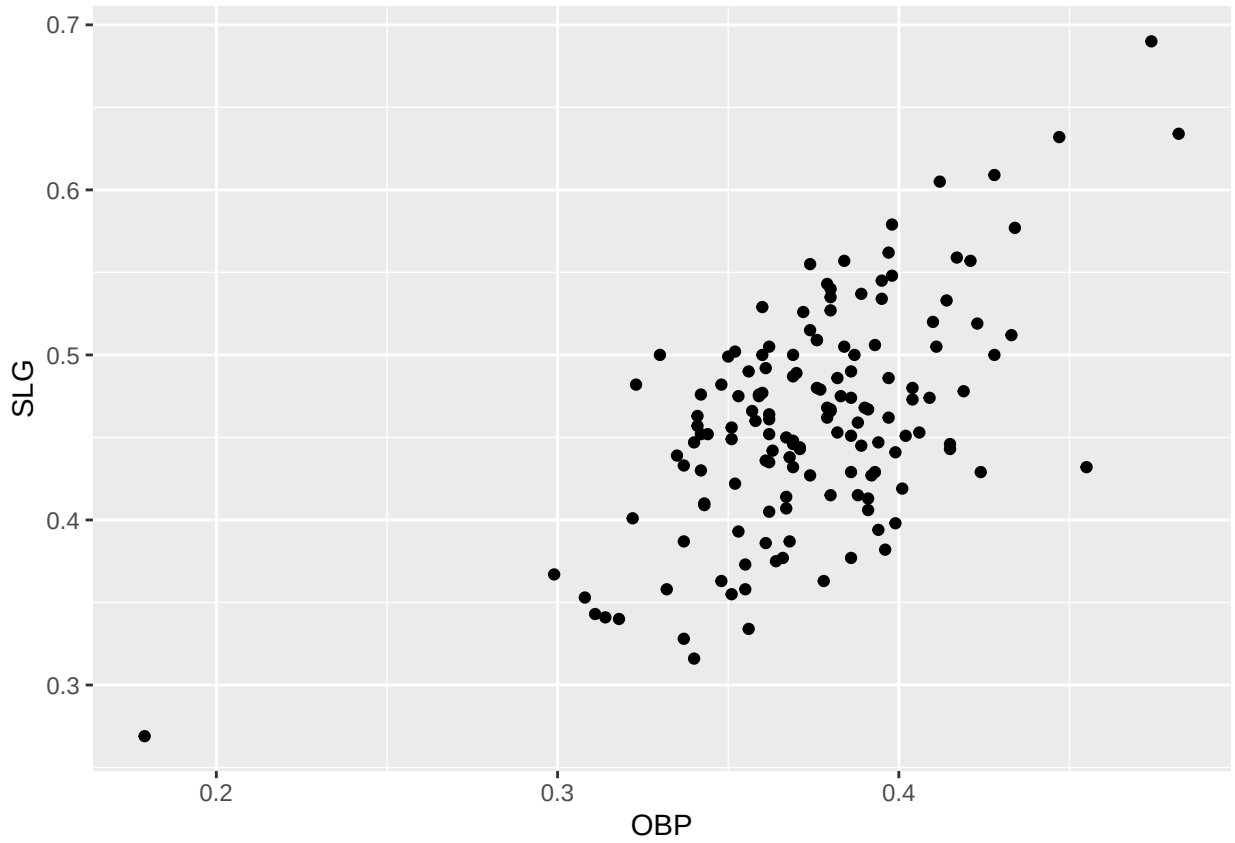
```
ggplot(hof, aes(MidCareer, OPS)) +
    geom_point() +
    geom_smooth() +
    geom_text_repel(data = filter(hof,OPS > 1.05 | OPS <.5),
                    aes(MidCareer,OPS,label=...2))
```

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



There was an increase when Babe Ruth and Lou Gehrig were in their primes. There has been steady decline in the average OPS over the last 30 years.

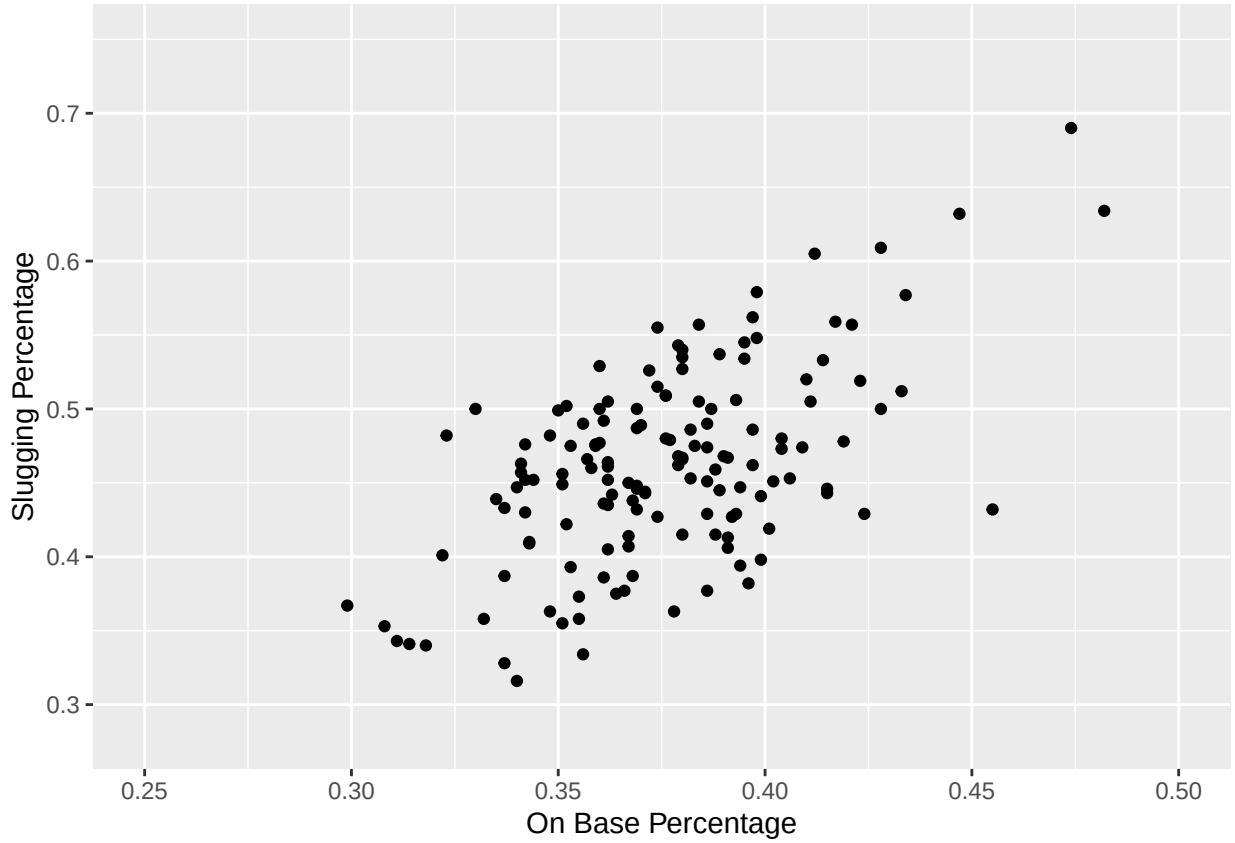
```
# 3.5.2 Building a graph, step-by-step
p <- ggplot(hof,aes(OBP,SLG))+geom_point()
p
```



Due to the outlier in the bottom left, most of the points fall in relatively small region of the plotting section. By use of xlim() and ylim() functions, we change the limit of horizontal and vertacal axes.

```
p <- p +
    xlim(0.25,0.50) + ylim(0.28, 0.75) +
    xlab("On Base Percentage") +
    ylab("Slugging Percentage")
p
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



Equivalently, we can change the limits and the labels by appealing to the scale_x_continuous() and scale_y_continuous() functions.

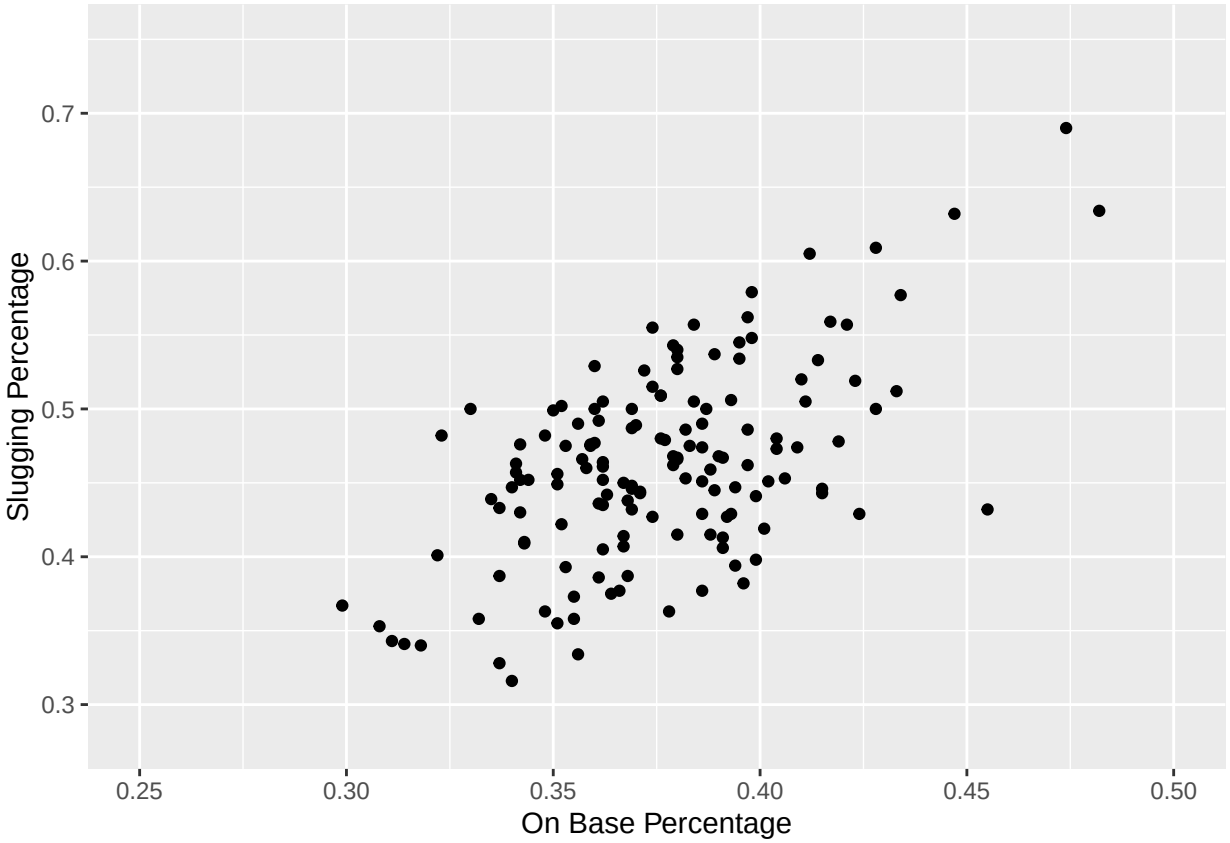
```
p <- p +
    scale_x_continuous("On Base Percentage",
                       limits = c(0.25, 0.50))+
    scale_y_continuous("Slugging Percentage",
                       limits = c(0.28, 0.75))
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
```

```
p
```

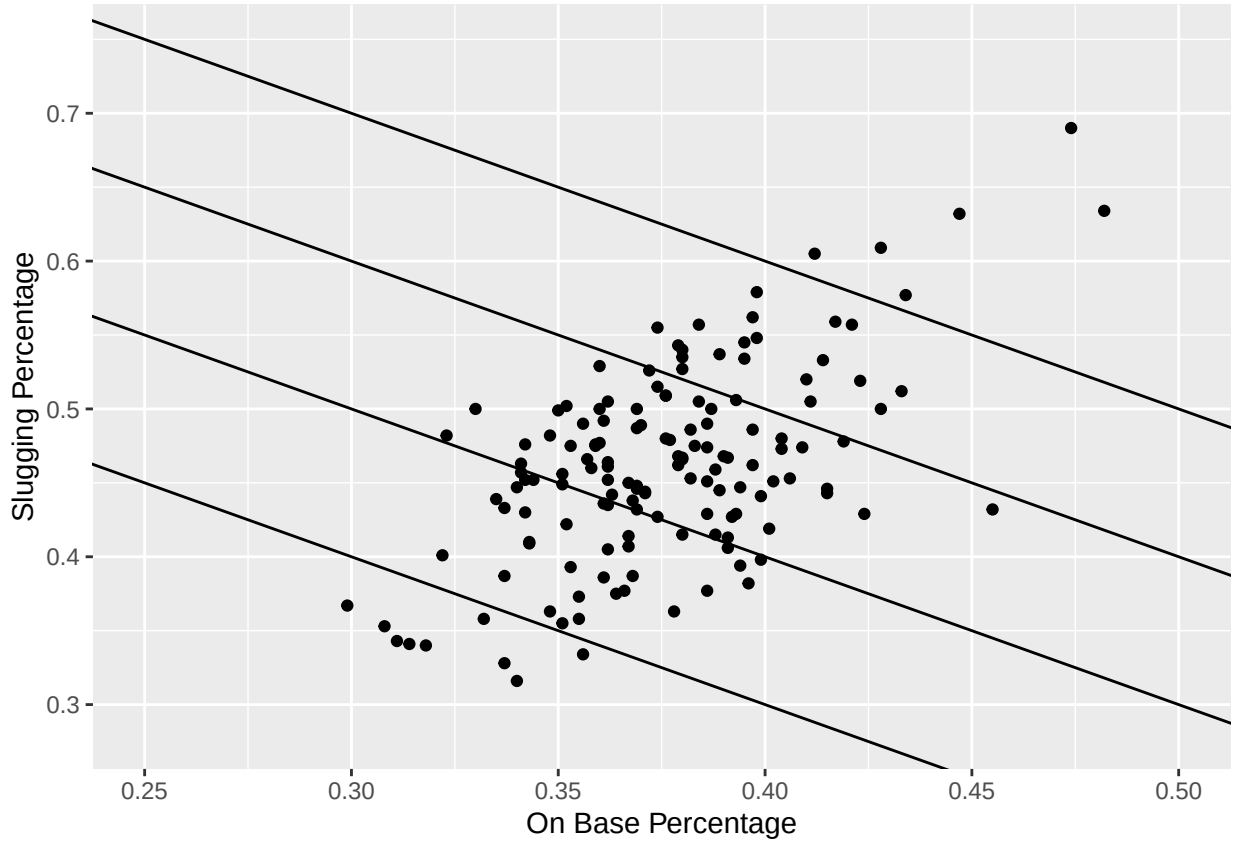
```
## Warning: Removed 1 rows containing missing values (geom_point).
```



OPS = OBP + SLG. It is helpful to draw constant values of OPS on the graph to evaluate hitters. Suppose we wish to draw a line where OPS = 0.7. We want to draw the function y = 0.7 - x on the graph. We can use geom_abline() to accomplish that.

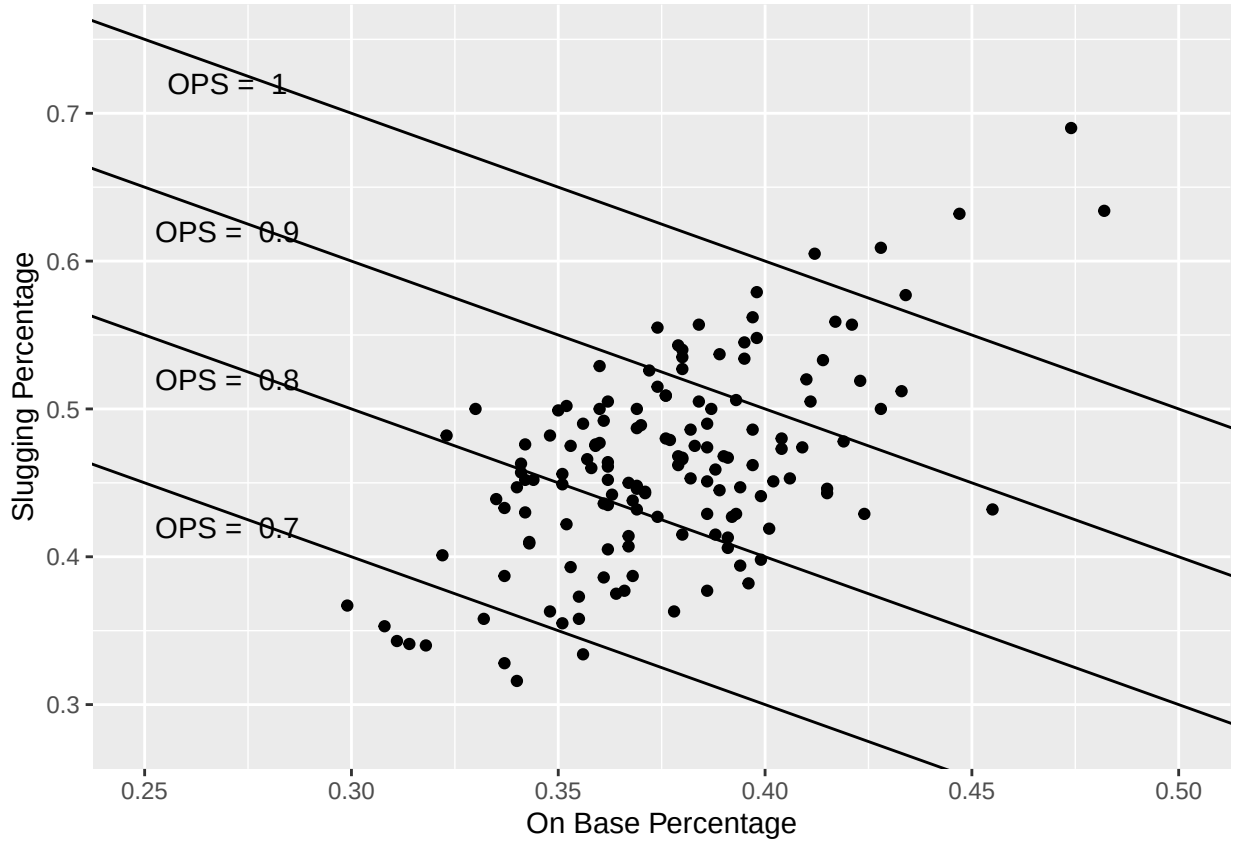
```
p <- p + geom_abline(slope = -1,
                     intercept = seq(0.7, 1, by=0.1))

p
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



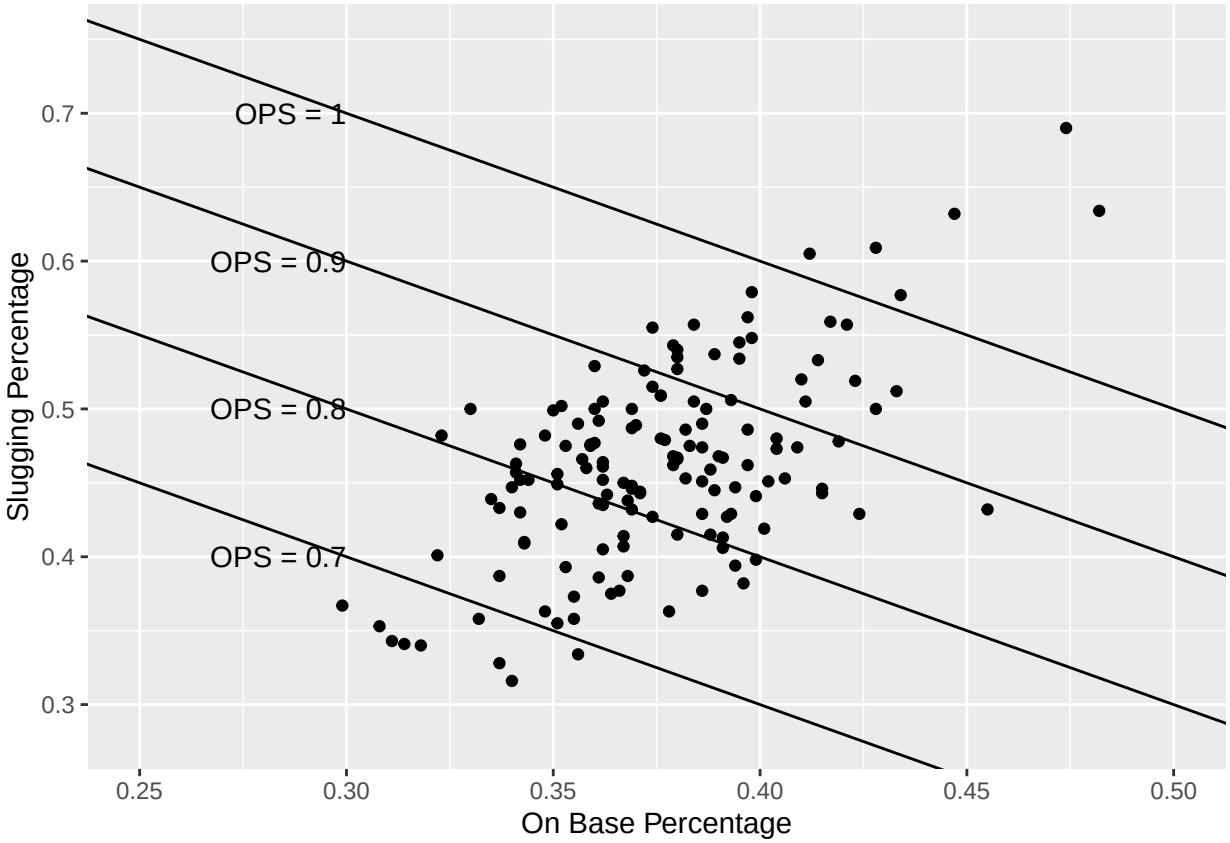

```
# annotate the ops values
p + annotate("text",
             x = rep(.27,4),c(.42,.52,.62,.72),
             label =paste("OPS = ",
                          c(0.7,0.8,0.9,1.0)))
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
# Another way of creating the graph.
ops_labels <- tibble(
    OBP = rep(0.3,4),
    SLG = seq(0.4,0.7,by = 0.1),
    label = paste("OPS =", OBP + SLG)
)

p + geom_text(data = ops_labels, hjust = "right",
              aes(label=label))
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



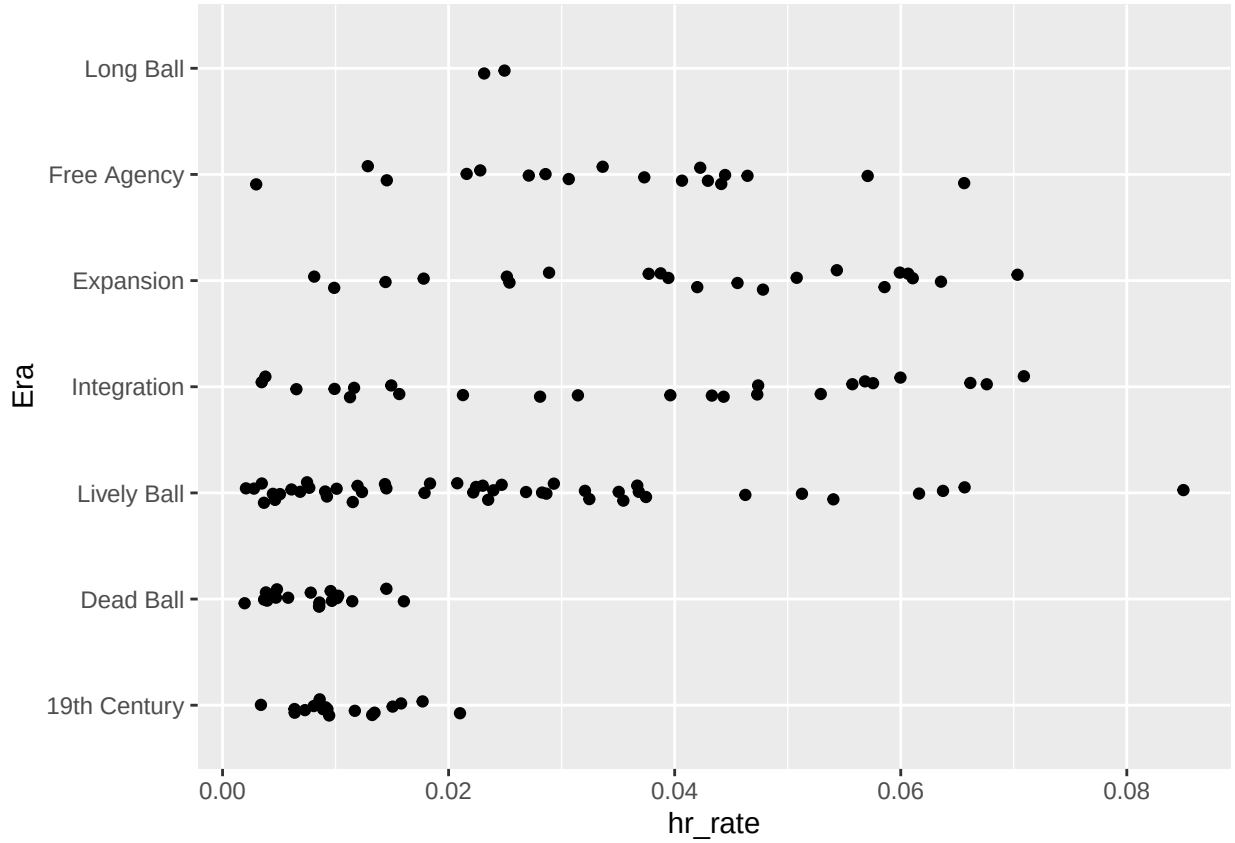
3.6 A Numeric Variable and a Factor Variable

Adding a new column "hr_rate", Home run rates.

```
hof <- hof %>%
    mutate(hr_rate = HR/AB)
```

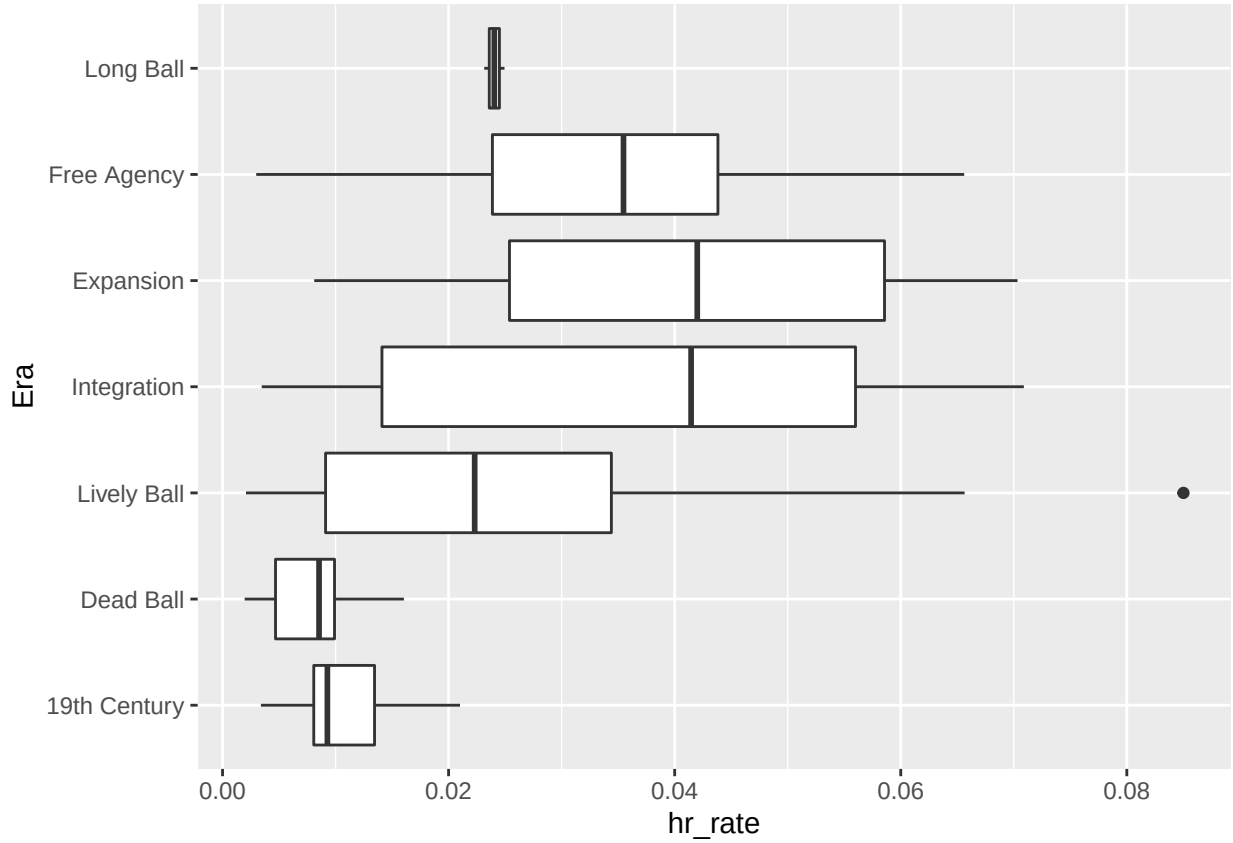
Construct parallel stripcharts of hr_rate by Era by using the geom_jitter() function.

```
# 3.6.1 Parallel stripcharts
ggplot(hof,aes(hr_rate,Era)) +
    geom_jitter(height = 0.1)
```



This graph shows how the rate of hitting home runs has changed over eras. Home runs were rare in the 19 century and Dead Ball eras.

```
# 3.6.2 Parallel boxplots
ggplot(hof, aes(Era,hr_rate)) +
    geom_boxplot() + coord_flip()
```



3.7 Comparing Ruth, Aaron, Bonds, and A-Rod

```
# 3.7.1 Getting the data
library(Lahman)
```

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

From the Lahman package, the relevant data frames are Master and Batting. From the Master data frame, we obtain player ids and birth years for the four players. The Batting data frame is used to extract the home run and at-bats information.

We will create a new function get_birthyear() to get information of the players.

```
# Creating a function to get birth year.
get_birthyear <- function(Name) {
    Names <- unlist(strsplit(Name," "))
    Master %>%
        filter(nameFirst == Names[1],
               nameLast == Names[2]) %>%
        mutate(birthyear = ifelse(birthMonth >= 7,
                                  birthYear + 1, birthYear),
               Player = paste(nameFirst, nameLast)) %>%
        select(playerID,Player, birthyear)
}
```

Beak down the code above:

1. Takes the given Name and split the Name by a space.
2. filter() function to look for the data that first name and last name are the same.
3. Create a column "birthyear" which has specific condition. if birthMonth >= 7 is True, we add 1 to the birtYear. if not, birthYear stays the same.
4. Create a column "Player" that is combining the first name and the last name of the player.
5. select only those columns (playerID,Player, birthyear)

```
# Run the function to get information of the players
PlayerInfo <- bind_rows(get_birthyear("Babe Ruth"),
                        get_birthyear("Hank Aaron"),
                        get_birthyear("Barry Bonds"),
                        get_birthyear("Alex Rodriguez"))

PlayerInfo
```

```
##    playerID         Player birthyear
## 1  ruthba01      Babe Ruth      1895
## 2 aaronha01     Hank Aaron      1934
## 3 bondsba01    Barry Bonds      1965
## 4 rodrial01 Alex Rodriguez      1976
```

We will now use Batting data set from Lahman to create data frame of those four players.

```
Batting %>%
    inner_join(PlayerInfo, by = "playerID") %>%
    mutate(Age = yearID - birthyear) %>%
    select(Player, Age, HR) %>%
    group_by(Player) %>%
    mutate(CHR = cumsum(HR)) -> HRdata

HRdata
```

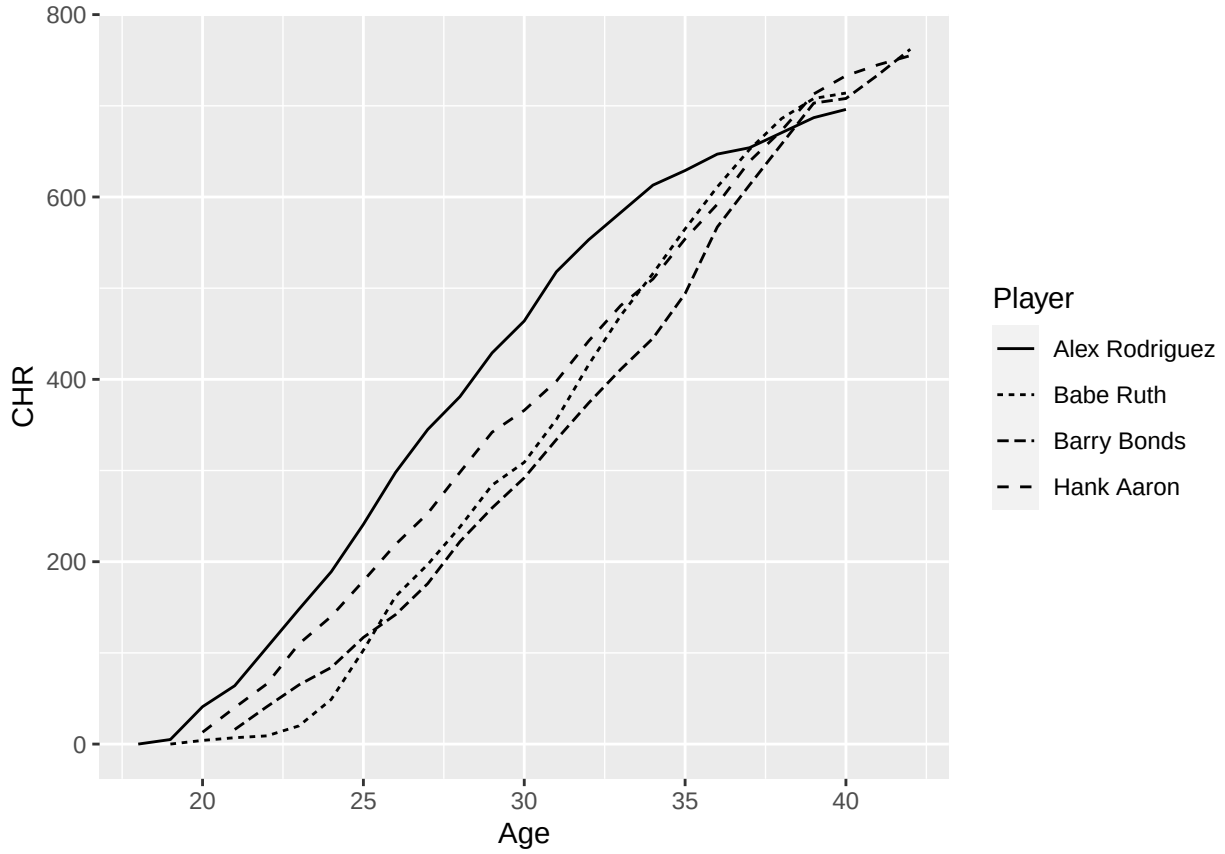
```
## # A tibble: 89 x 4
## # Groups:   Player [4]
##    Player      Age    HR   CHR
##    <chr>     <dbl> <int> <int>
##  1 Babe Ruth    19     0     0
##  2 Babe Ruth    20     4     4
##  3 Babe Ruth    21     3     7
##  4 Babe Ruth    22     2     9
##  5 Babe Ruth    23    11    20
##  6 Babe Ruth    24    29    49
##  7 Babe Ruth    25    54   103
##  8 Babe Ruth    26    59   162
##  9 Babe Ruth    27    35   197
## 10 Babe Ruth    28    41   238
## # ... with 79 more rows
```

Break down the code above:

1.Merge Batting data set and PlayerInfo that we created earlier by "playerID" column.

2. Create a new column "Age" (yearID - birthyear)
3. Select only the columns (Player, Age, HR)
4. Calculate cumulative sum of HR for each player.

```
# Cumulative home run counts against age for four players.
ggplot(HRdata, aes(x=Age, y=CHR, linetype=Player))+
    geom_line()
```



3.8 The 1998 Home Run Race

We illustrate the use of R to read in the files for the 1998 season and graphically view the famous home run duel between Mark McGwire and Sammy Sosa.

```
# 3.8.1 Read Data
fields <- read_csv("../data/csv_files/fields.csv")
```

```
## Rows: 97 Columns: 3
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Description, Header
## dbl (1): Field number

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
data1998 <- read_csv("../data/csv_files/all1998.csv", col_names = pull(fields,Header))
```

```
## Warning: One or more parsing issues, see `problems()` for details

## Rows: 195011 Columns: 97

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (36): GAME_ID, AWAY_TEAM_ID, PITCH_SEQ_TX, BAT_ID, BAT_HAND_CD, RESP_BAT...
## dbl (34): INN_CT, BAT_HOME_ID, OUTS_CT, BALLS_CT, STRIKES_CT, AWAY_SCORE_CT,...
## lgl (27): LEADOFF_FL, PH_FL, BAT_EVENT_FL, AB_FL, SH_FL, SF_FL, DP_FL, TP_FL...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

BAT_ID in data1998 is the identification code for player who is batting.

Extract playerID for both players from Master data set.

```
sosa_id <- Master %>%
    filter(nameFirst == "Sammy", nameLast == "Sosa") %>%
    pull(retroID)

mac_id <- Master %>%
    filter(nameFirst == "Mark", nameLast == "McGwire") %>%
    pull(retroID)
```

```
# extract only the two players data
hr_race <- data1998 %>%
    filter(BAT_ID %in% c(sosa_id, mac_id))
```

We create a function to extract home run count and date. For date variable we will extract it from GAME_ID which identifies the game location and date.EVENT_CD of 23 indicates that a home run has been hit. So, we create ifelse statement to cumulate the home run count.

```
# 3.8.2 Extracting the variables
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
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)
}
```

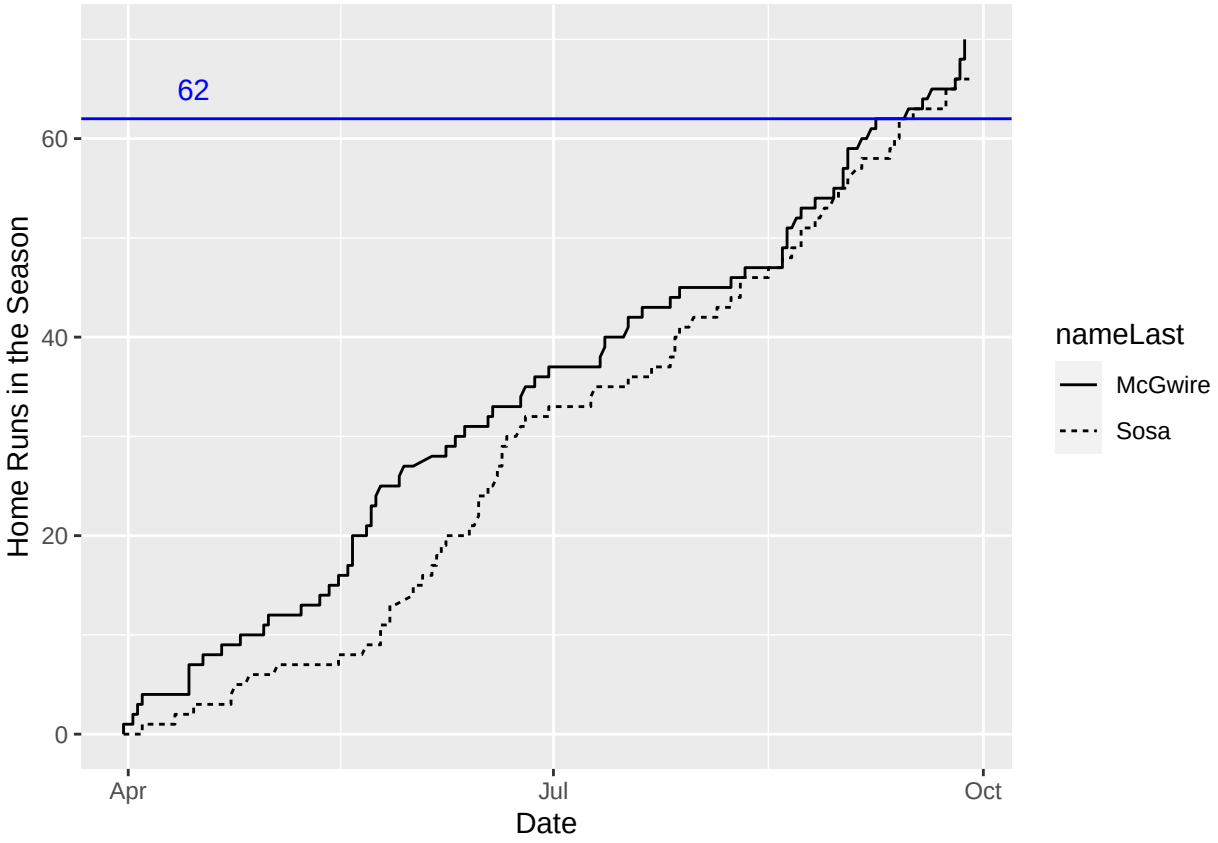
We use map_df() function to iterate cum_hr() twice. Obtaining the new data frame hr_ytd.

```
hr_ytd <- hr_race %>%
    split(pull(., BAT_ID)) %>%
    map_df(cum_hr,.id="BAT_ID") %>%
    inner_join(Master, by = c("BAT_ID" = "retroID"))
hr_ytd
```

```
## # A tibble: 1,447 x 28
##    BAT_ID   Date       cumHR playerID  birthYear birthMonth birthDay birthCountry
##    <chr>    <date>     <dbl> <chr>         <int>      <int>    <int> <chr>
##  1 mcgwm001 1998-03-31     0 mcgwima01      1963         10        1 USA
##  2 mcgwm001 1998-03-31     0 mcgwima01      1963         10        1 USA
##  3 mcgwm001 1998-03-31     1 mcgwima01      1963         10        1 USA
##  4 mcgwm001 1998-03-31     1 mcgwima01      1963         10        1 USA
##  5 mcgwm001 1998-04-02     1 mcgwima01      1963         10        1 USA
##  6 mcgwm001 1998-04-02     1 mcgwima01      1963         10        1 USA
##  7 mcgwm001 1998-04-02     1 mcgwima01      1963         10        1 USA
##  8 mcgwm001 1998-04-02     1 mcgwima01      1963         10        1 USA
##  9 mcgwm001 1998-04-02     1 mcgwima01      1963         10        1 USA
## 10 mcgwm001 1998-04-02     1 mcgwima01      1963         10        1 USA
## # ... with 1,437 more rows, and 20 more variables: birthState <chr>,
## #   birthCity <chr>, deathYear <int>, deathMonth <int>, deathDay <int>,
## #   deathCountry <chr>, deathState <chr>, deathCity <chr>, nameFirst <chr>,
## #   nameLast <chr>, nameGiven <chr>, weight <int>, height <int>, bats <fct>,
## #   throws <fct>, debut <chr>, finalGame <chr>, bbrefID <chr>,
## #   deathDate <date>, birthDate <date>
```

Creating a line plot with the hr_ytd data. Set x axis as Date and y axis as cumHR. geom_hline() function allows us to put an horizontal line on the graph.

```
#3.8.3 Constructing the graph
ggplot(hr_ytd,aes(Date, cumHR, linetype = nameLast))+
    geom_line() +
    geom_hline(yintercept = 62, color ='blue') +
    annotate("text",ymd("1998-04-15"),65,
             label='62', color='blue') +
    ylab("Home Runs in the Season")
```



3.10 Exercises

1. Hall of Fame Pitching Dataset

```
hofpitching <- read_csv("../data/csv_files/hofpitching.csv")
```

```
## New names:
## * `` -> ...2
```

```
## Rows: 70 Columns: 30
```

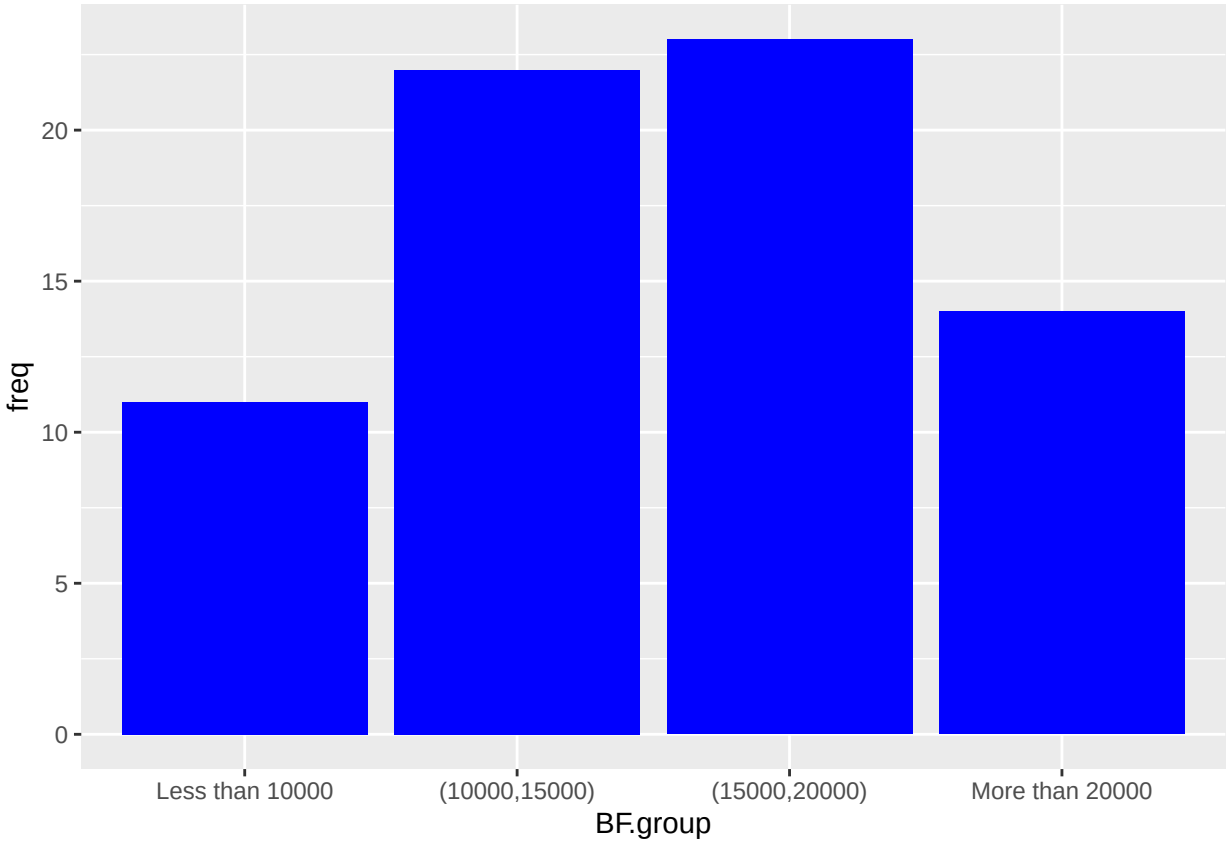
```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): ...2
## dbl (29): Rk, Inducted, Yrs, From, To, ASG, WAR, W, L, W-L%, ERA, G, GS, GF,...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# BF = the number of batters faced by a pitcher in his career.
hofpitching <- hofpitching %>%
    mutate(BF.group = cut(BF, c(0, 10000, 15000, 20000, 30000),
                          labels = c("Less than 10000","(10000,15000)",
                                     "(15000,20000)","More than 20000")))


# (a) Construct a freq table of BF.group using summarize() function.
BF_group_freq <- hofpitching %>%
    group_by(BF.group) %>%
    summarize(freq = n())
BF_group_freq
```

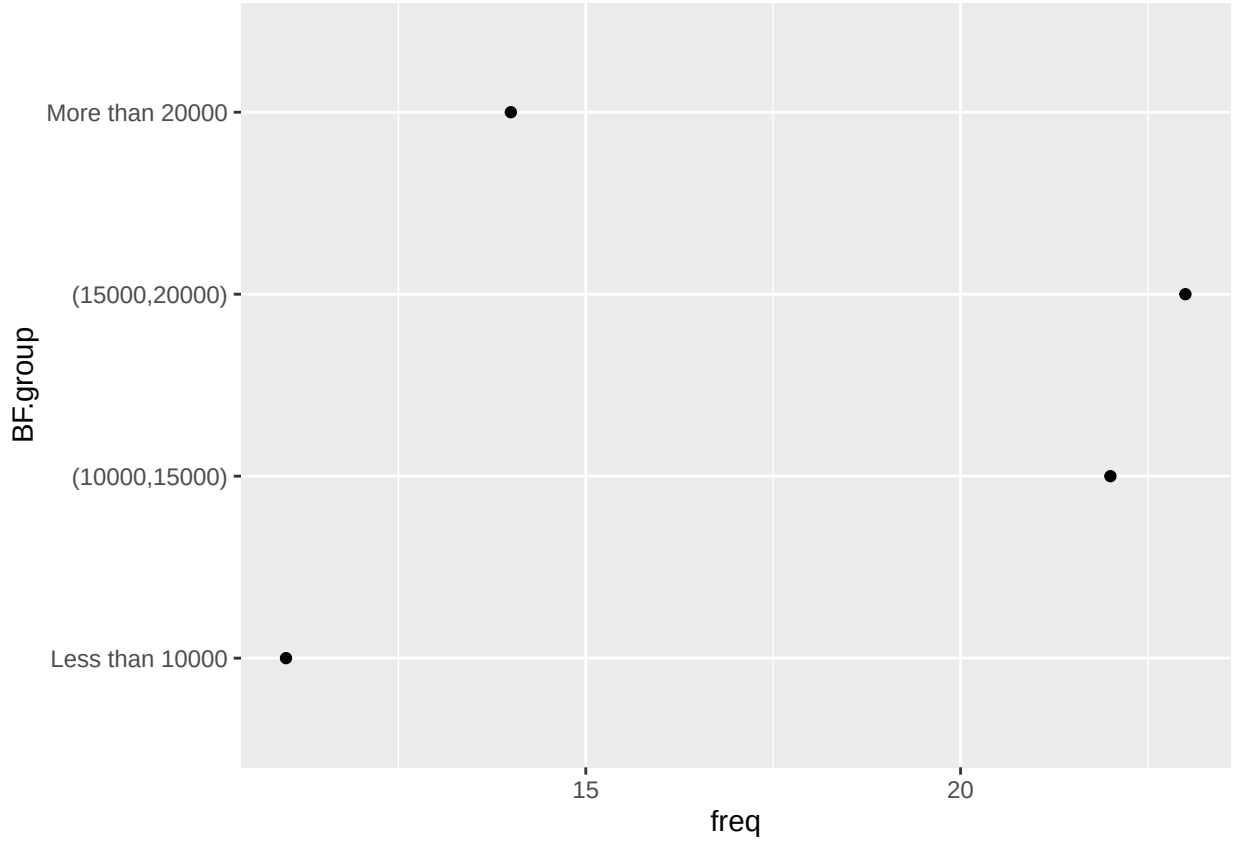
```
## # A tibble: 4 x 2
##   BF.group         freq
##   <fct>           <int>
## 1 Less than 10000    11
## 2 (10000,15000)      22
## 3 (15000,20000)      23
## 4 More than 20000    14
```

```r
# (b) Construct a bar graph of the output from summarize().
ggplot(BF_group_freq, aes(BF.group, freq))+
    geom_col(fill = 'blue')
```



Q. How many HOF pitchers faced more than 20,000 batters in their career? A. 14 HOF pitchers faced more than 20,000 batters in their career.

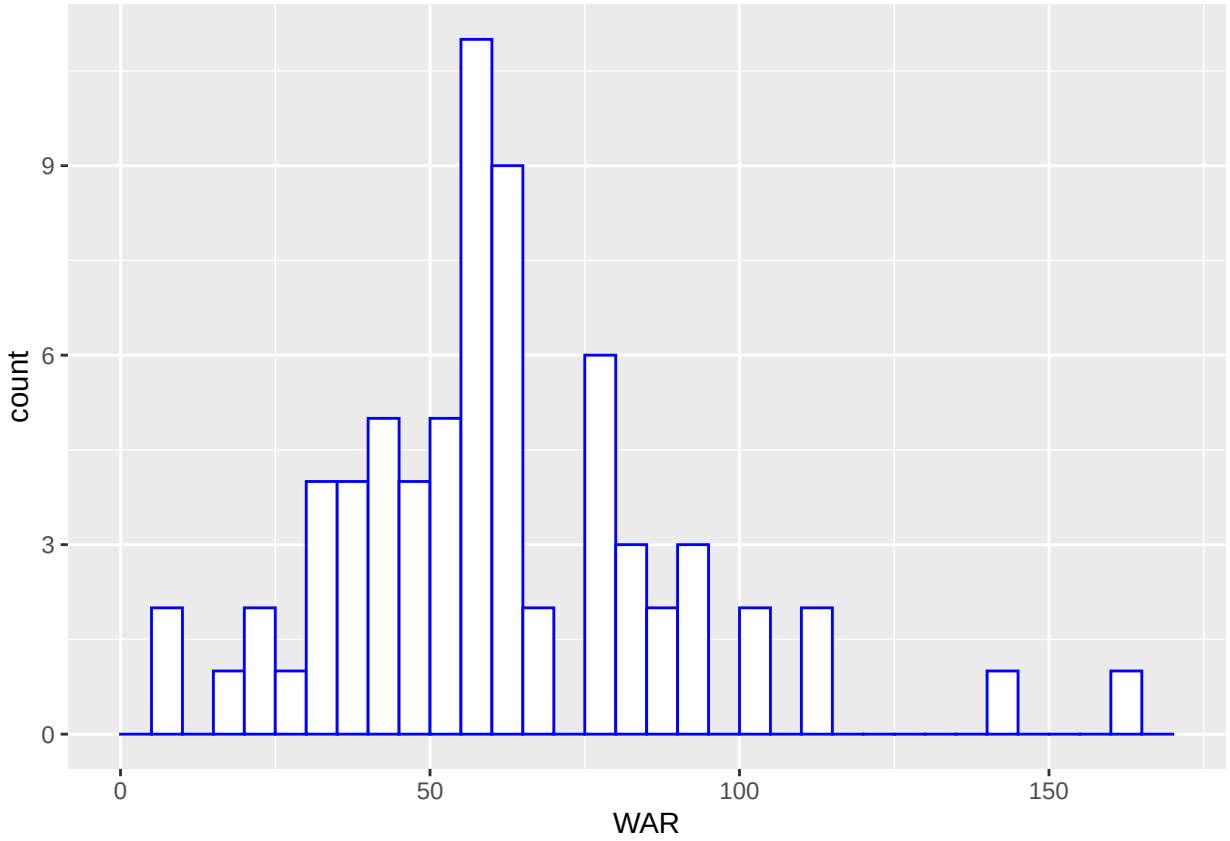
```
# (c) Construct an alternative graph of the BF.group variable.
# Compare the effectiveness of the bar graph and the new graph in compare the frequencies in the four i
ggplot(BF_group_freq, aes(freq, BF.group))+
    geom_point()
```



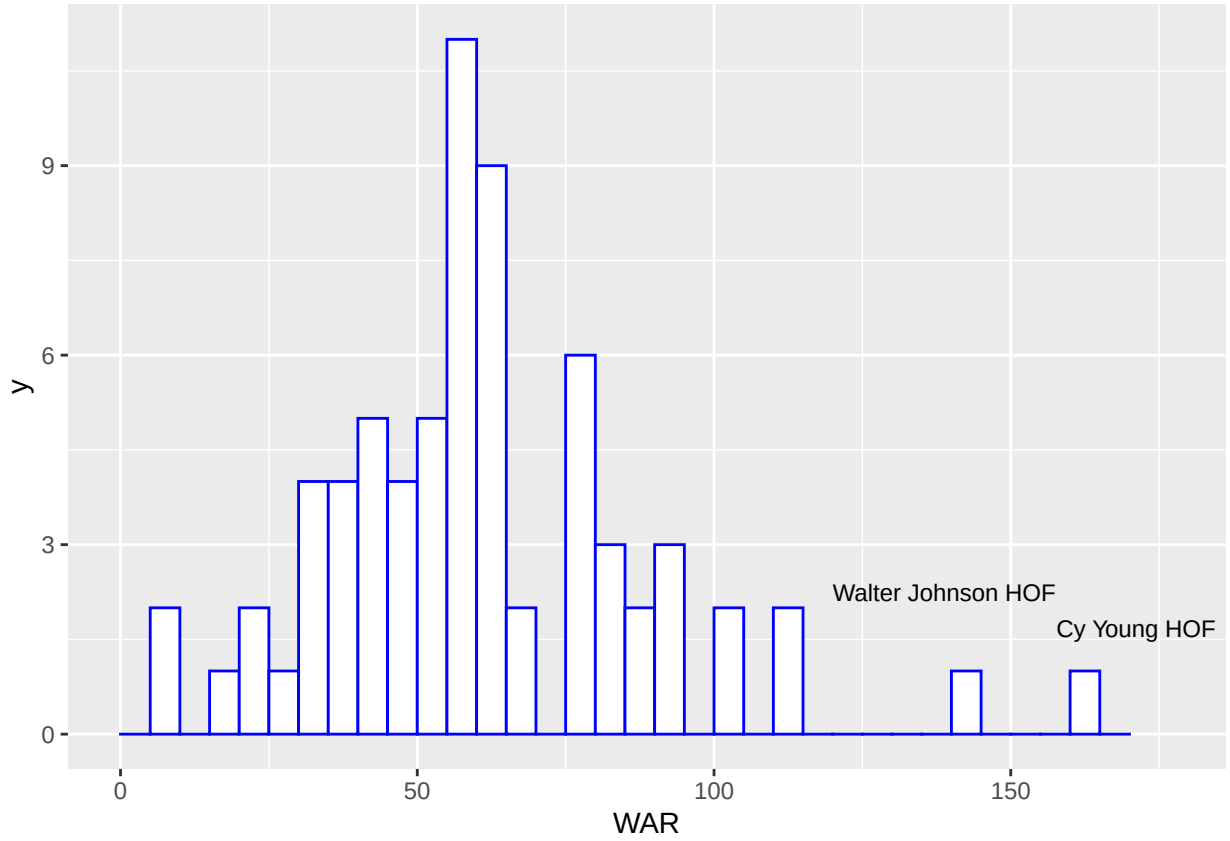
2. Hall of Fame Pitching Dataset (Continued)

The variable WAR is the total wins above replacement of the pitcher during his career.

```
# (a) Using the geom_histogram() function, construct a histogram of WAR
# for the pitchers in the HOF dataset.
ggplot(hofpitching,aes(WAR)) +
    geom_histogram(breaks = seq(0,170,by=5),
                   color = "blue", fill = 'white')
```



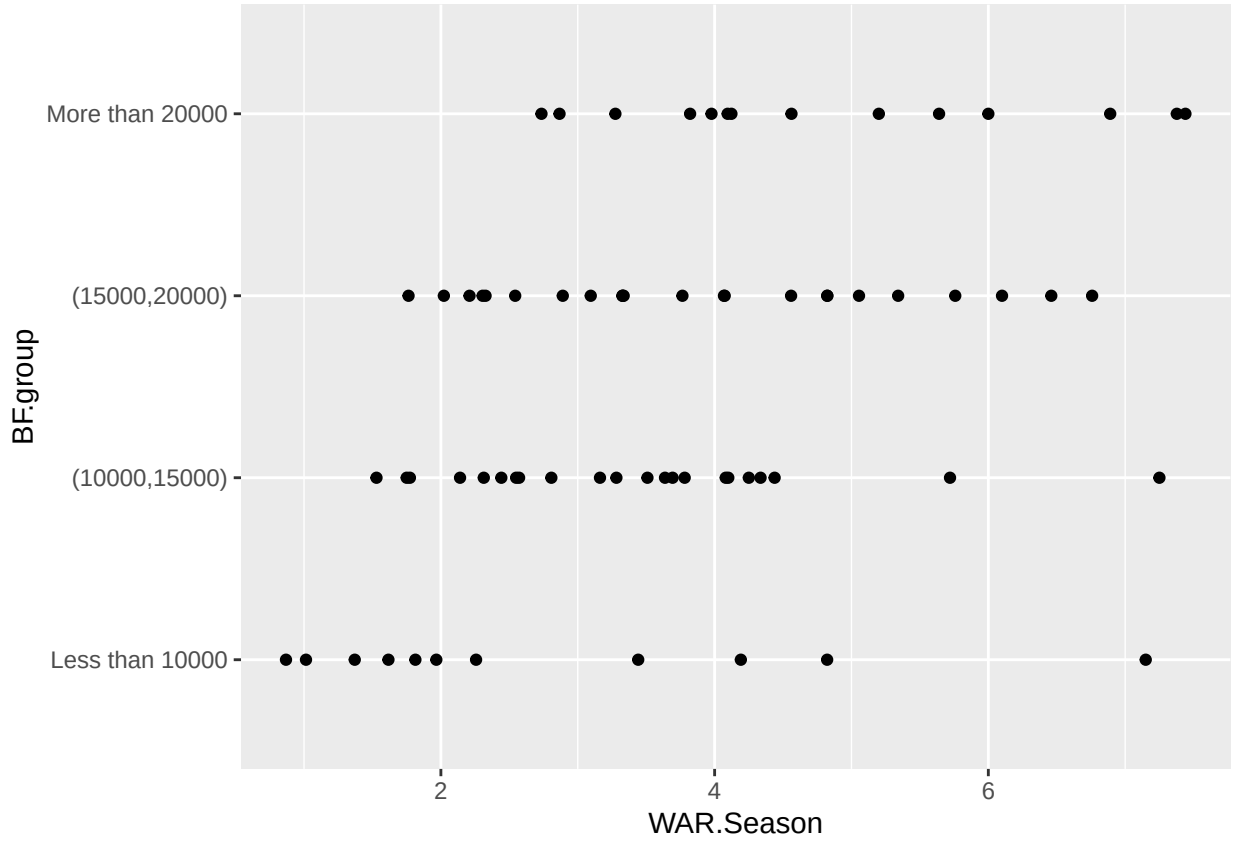

```
# (b) There are two pitchers who stand out among all of the HOF on the total WAR variable.
# Identify those two pitchers.
library(ggrepel)
q <- ggplot(hofpitching,aes(WAR)) +
    geom_histogram(breaks = seq(0,170,by=5),color = "blue", fill = 'white') +
    geom_text_repel(data=filter(hofpitching,WAR > 140),
                        aes(WAR+15,2,label = ...2), size=3)

q
```



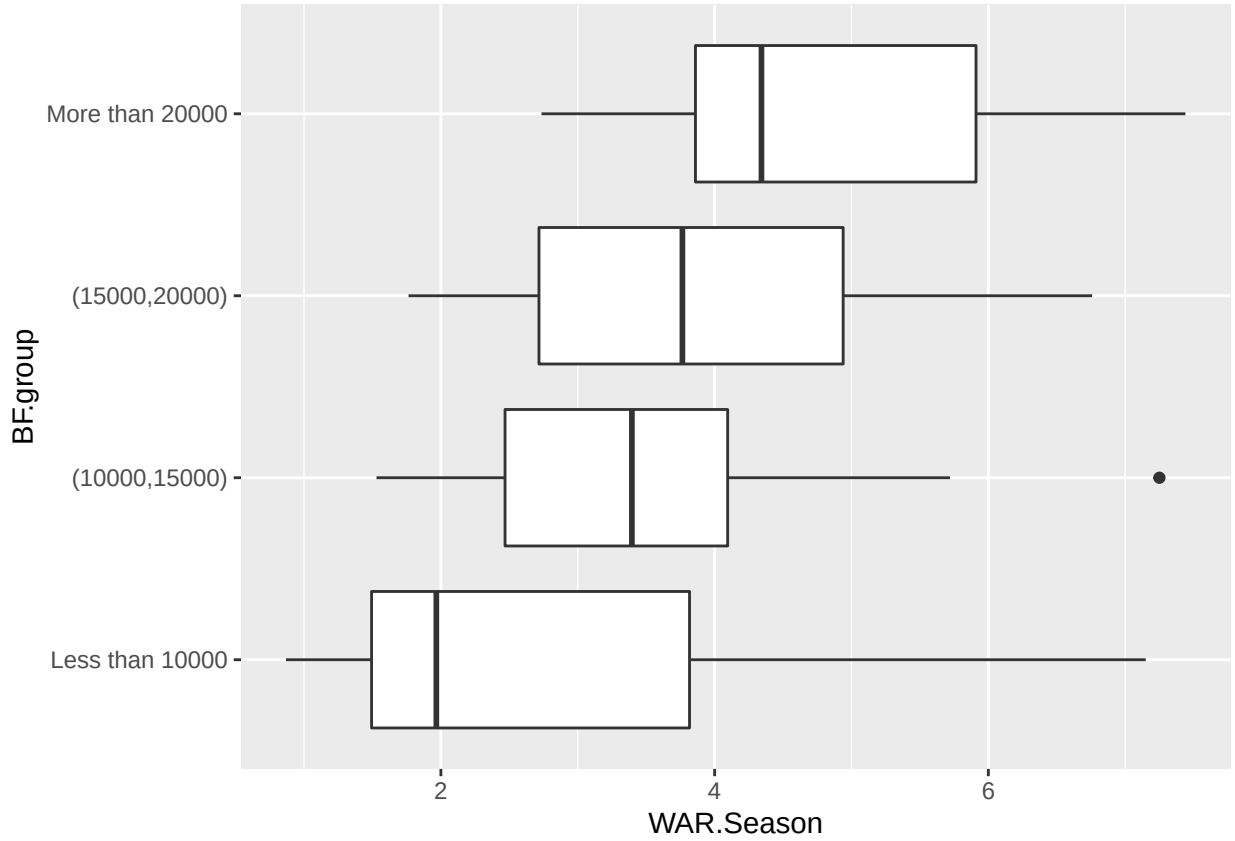
A. The pitcher who had the highest WAR is Cy Young. The second is Walter Johnson.

3. Hall of Fame Pitching Dataset (Continued)

```
# Create a column "WAR.Season".
hofpitching <- hofpitching %>%
    mutate(WAR.Season = WAR / Yrs)

# (a) One-dimension scatter plot for each BF.group.
ggplot(hofpitching,aes(x=WAR.Season,y=BF.group))+
    geom_point()
```




```
# (b) Boxplot for each BF.group.
ggplot(hofpitching,aes(WAR.Season, BF.group)) +
    geom_boxplot()
```



Q. Based on your graph, how does wins above replacement per season depend on the number of batters faced?

A. Pitchers who have faced more than 20,000 batters have the highest WAR per season. Those who have faced less than 10,000 have the lowest WAR per season. Therefore, the more pitchers face batters, the higher WAR those pitchers have.

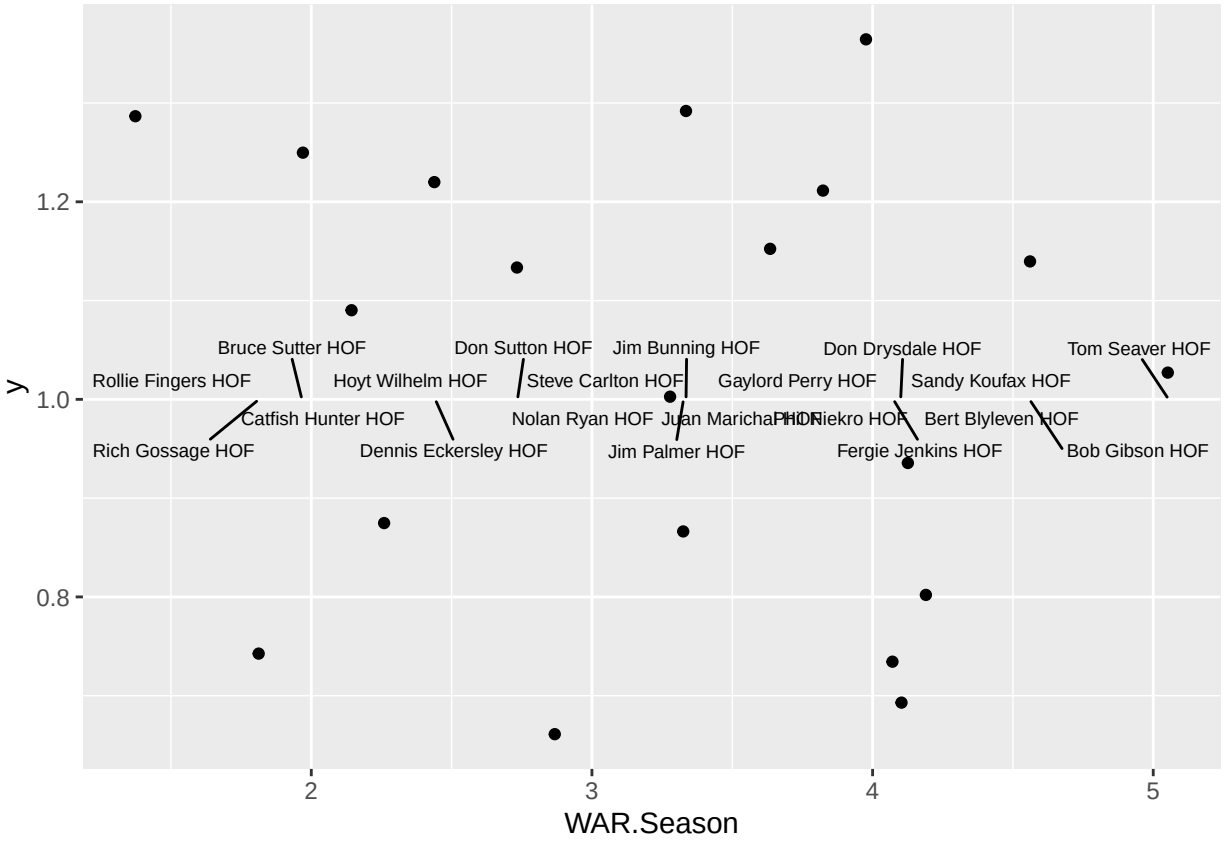
4. Hall of Fame Pitching Dataset (Continued)

Suppose we limit our exploration to pitchers whose mid-career was 1960 or later. We first define the MidYear variable and use the filter function to construct a data frame.

```
hofpitching <- hofpitching %>%
    mutate(midYear = (From + To) / 2)

hofpitching.recent <- hofpitching %>%
    filter(midYear >= 1960)
```

```
# (a) Sort the data by WAR.Season
hofpitching.recent %>%
    arrange(WAR.Season) -> hofpitching.recent
```

```
# (b) Construct a dot plot of the values of WAR.Season where the labels are players name.
ggplot(hofpitching.recent, aes(WAR.Season, y=1))+
    geom_jitter() +
    geom_text_repel(aes(WAR.Season, label = ...2), max.overlaps = Inf, size=2.5)
```




```
# (c) Which two 1960+ pitchers stand out with respect to wins above replacement per season?
tail(hofpitching.recent,2)[2]
```

```
## # A tibble: 2 x 1
##   ...2
##   <chr>
## 1 Bob Gibson HOF
## 2 Tom Seaver HOF
```

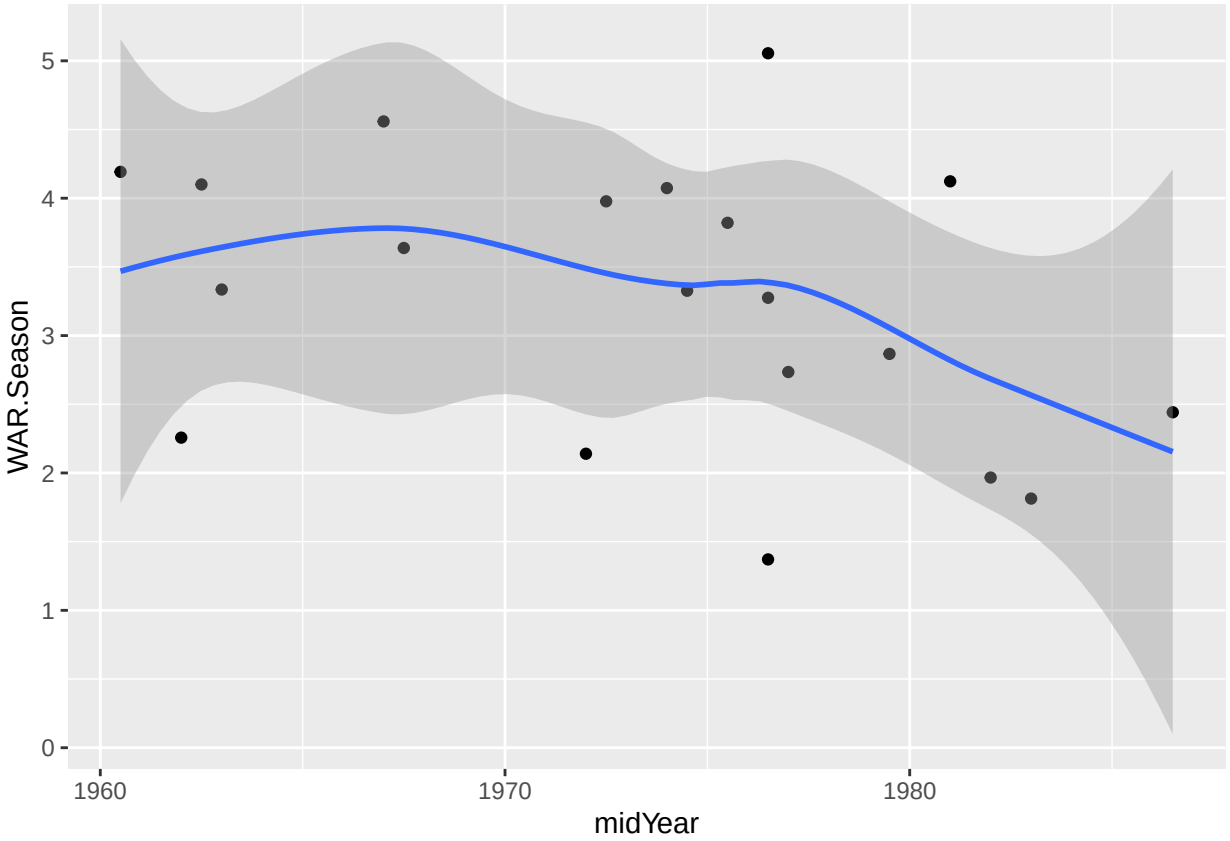
A.

1. Tom Seaver
2. Bob Gibson

5. Hall of Fame Pitching Dataset (Continued)

```
# (a) Construct a scatterplot of MidYear(horizontal)
#against WAR.Season(vertical)
ggplot(hofpitching.recent, aes(midYear, WAR.Season)) +
    geom_point() + geom_smooth()
```

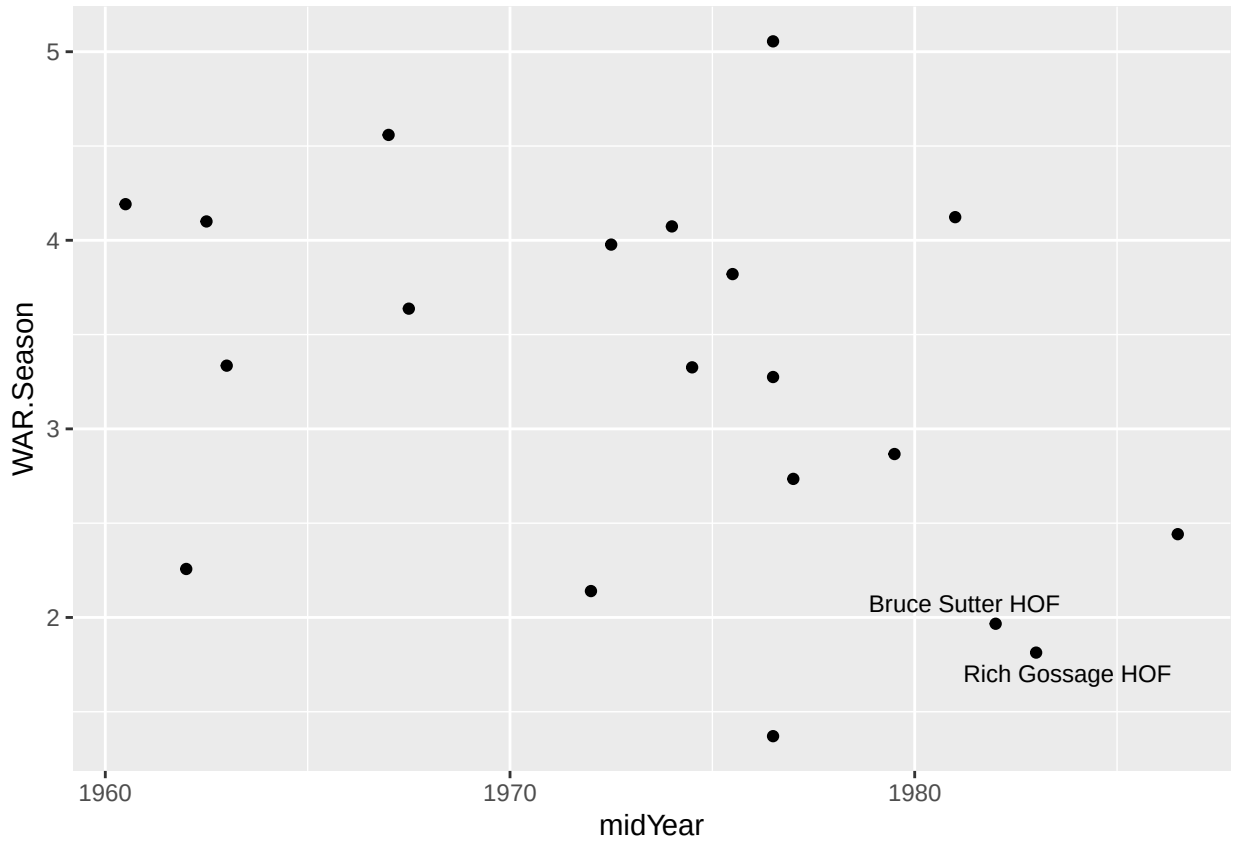
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



(b) Is there a general pattern in this scatterplot? Explain.

A. There is no clear pattern, however, the WAR.Season is slightly decreasing between 1970 and 1980.

```
# (c) There are two pitchers whose mid Careers were in the 1800s
# who had relatively low WAR.Season values. Find out by using filter and geom_text functions.
ggplot(hofpitching.recent, aes(midYear, WAR.Season)) +
    geom_point() +
    geom_text_repel(data=filter(hofpitching, midYear >=1980, WAR.Season < 2.0),aes(midYear, WAR.Season,
```



A. Two pitchers less than 2.0 WAR.Season in 1980s. 1. Rich Gossage 2. Bruce Sutter

6. Working with the Lahman batting dataset

```
# (a) Read Master and Batting datasets

# (b) Collect a single dataframe the season batting statistics for the great hitters.
# Ty Cobb, Ted Williams, and Pete Rose.

cobb_id <- Master %>% filter(nameFirst=="Ty",nameLast=="Cobb") %>%
    pull(playerID)

williams_id <- Master %>% filter(nameFirst=="Ted",nameLast=="Williams") %>%
    pull(playerID)

rose_id <- "rosepe01"

season_batting_stats <- Batting %>%
    filter(playerID %in% c(cobb_id,williams_id,rose_id))

season_batting_stats
```

```
##     playerID yearID stint teamID lgID   G  AB   R   H X2B X3B HR RBI SB CS  BB
## 1   cobbty01   1905     1    DET   AL  41 151  19  36   6   0  1  15  2 NA  10
```

```
## 2   cobbty01   1906     1    DET   AL  98 358  45 113  15   5  1  41 23 NA  19
## 3   cobbty01   1907     1    DET   AL 150 605  99 212  28  14  5 119 53 NA  24
## 4   cobbty01   1908     1    DET   AL 150 580  88 188  36  20  4 108 39 NA  34
## 5   cobbty01   1909     1    DET   AL 156 573 115 216  33  10  9 107 76 NA  48
## 6   cobbty01   1910     1    DET   AL 140 508 106 194  35  13  8  91 65 NA  64
## 7   cobbty01   1911     1    DET   AL 146 591 147 248  47  24  8 127 83 NA  44
## 8   cobbty01   1912     1    DET   AL 140 553 120 226  30  23  7  83 61 NA  43
## 9   cobbty01   1913     1    DET   AL 122 428  70 167  18  16  4  67 51 NA  58
## 10  cobbty01   1914     1    DET   AL  98 345  69 127  22  11  2  57 35 17  57
## 11  cobbty01   1915     1    DET   AL 156 563 144 208  31  13  3  99 96 38 118
## 12  cobbty01   1916     1    DET   AL 145 542 113 201  31  10  5  68 68 24  78
## 13  cobbty01   1917     1    DET   AL 152 588 107 225  44  24  6 102 55 NA  61
## 14  cobbty01   1918     1    DET   AL 111 421  83 161  19  14  3  64 34 NA  41
## 15  cobbty01   1919     1    DET   AL 124 497  92 191  36  13  1  70 28 NA  38
## 16  cobbty01   1920     1    DET   AL 112 428  86 143  28   8  2  63 15 10  58
## 17  cobbty01   1921     1    DET   AL 128 507 124 197  37  16 12 101 22 15  56
## 18  cobbty01   1922     1    DET   AL 137 526  99 211  42  16  4  99  9 13  55
## 19  cobbty01   1923     1    DET   AL 145 556 103 189  40   7  6  88  9 10  66
## 20  cobbty01   1924     1    DET   AL 155 625 115 211  38  10  4  78 23 14  85
## 21  cobbty01   1925     1    DET   AL 121 415  97 157  31  12 12 102 13  9  65
## 22  cobbty01   1926     1    DET   AL  79 233  48  79  18   5  4  62  9  4  26
## 23  cobbty01   1927     1    PHA   AL 134 490 104 175  32   7  5  93 22 16  67
## 24  cobbty01   1928     1    PHA   AL  95 353  54 114  27   4  1  40  5  8  34
## 25 willite01   1939     1    BOS   AL 149 565 131 185  44  11 31 145  2  1 107
## 26 willite01   1940     1    BOS   AL 144 561 134 193  43  14 23 113  4  4  96
## 27 willite01   1941     1    BOS   AL 143 456 135 185  33   3 37 120  2  4 147
## 28 willite01   1942     1    BOS   AL 150 522 141 186  34   5 36 137  3  2 145
## 29 willite01   1946     1    BOS   AL 150 514 142 176  37   8 38 123  0  0 156
## 30 willite01   1947     1    BOS   AL 156 528 125 181  40   9 32 114  0  1 162
## 31 willite01   1948     1    BOS   AL 137 509 124 188  44   3 25 127  4  0 126
## 32 willite01   1949     1    BOS   AL 155 566 150 194  39   3 43 159  1  1 162
## 33 willite01   1950     1    BOS   AL  89 334  82 106  24   1 28  97  3  0  82
## 34 willite01   1951     1    BOS   AL 148 531 109 169  28   4 30 126  1  1 144
## 35 willite01   1952     1    BOS   AL   6  10   2   4   0   1  1   3  0  0   2
## 36 willite01   1953     1    BOS   AL  37  91  17  37   6   0 13  34  0  1  19
## 37 willite01   1954     1    BOS   AL 117 386  93 133  23   1 29  89  0  0 136
## 38 willite01   1955     1    BOS   AL  98 320  77 114  21   3 28  83  2  0  91
## 39 willite01   1956     1    BOS   AL 136 400  71 138  28   2 24  82  0  0 102
## 40 willite01   1957     1    BOS   AL 132 420  96 163  28   1 38  87  0  1 119
## 41 willite01   1958     1    BOS   AL 129 411  81 135  23   2 26  85  1  0  98
## 42 willite01   1959     1    BOS   AL 103 272  32  69  15   0 10  43  0  0  52
## 43 willite01   1960     1    BOS   AL 113 310  56  98  15   0 29  72  1  1  75
## 44  rosepe01   1963     1    CIN   NL 157 623 101 170  25   9  6  41 13 15  55
## 45  rosepe01   1964     1    CIN   NL 136 516  64 139  13   2  4  34  4 10  36
## 46  rosepe01   1965     1    CIN   NL 162 670 117 209  35  11 11  81  8  3  69
## 47  rosepe01   1966     1    CIN   NL 156 654  97 205  38   5 16  70  4  9  37
## 48  rosepe01   1967     1    CIN   NL 148 585  86 176  32   8 12  76 11  6  56
## 49  rosepe01   1968     1    CIN   NL 149 626  94 210  42   6 10  49  3  7  56
## 50  rosepe01   1969     1    CIN   NL 156 627 120 218  33  11 16  82  7 10  88
## 51  rosepe01   1970     1    CIN   NL 159 649 120 205  37   9 15  52 12  7  73
## 52  rosepe01   1971     1    CIN   NL 160 632  86 192  27   4 13  44 13  9  68
## 53  rosepe01   1972     1    CIN   NL 154 645 107 198  31  11  6  57 10  3  73
## 54  rosepe01   1973     1    CIN   NL 160 680 115 230  36   8  5  64 10  7  65
## 55  rosepe01   1974     1    CIN   NL 163 652 110 185  45   7  3  51  2  4 106
```

```
## 56  rosepe01   1975     1    CIN   NL 162 662 112 210  47   4  7  74  0  1  89
## 57  rosepe01   1976     1    CIN   NL 162 665 130 215  42   6 10  63  9  5  86
## 58  rosepe01   1977     1    CIN   NL 162 655  95 204  38   7  9  64 16  4  66
## 59  rosepe01   1978     1    CIN   NL 159 655 103 198  51   3  7  52 13  9  62
## 60  rosepe01   1979     1    PHI   NL 163 628  90 208  40   5  4  59 20 11  95
## 61  rosepe01   1980     1    PHI   NL 162 655  95 185  42   1  1  64 12  8  66
## 62  rosepe01   1981     1    PHI   NL 107 431  73 140  18   5  0  33  4  4  46
## 63  rosepe01   1982     1    PHI   NL 162 634  80 172  25   4  3  54  8  8  66
## 64  rosepe01   1983     1    PHI   NL 151 493  52 121  14   3  0  45  7  7  52
## 65  rosepe01   1984     1    MON   NL  95 278  34  72   6   2  0  23  1  1  31
## 66  rosepe01   1984     2    CIN   NL  26  96   9  35   9   0  0  11  0  0   9
## 67  rosepe01   1985     1    CIN   NL 119 405  60 107  12   2  2  46  8  1  86
## 68  rosepe01   1986     1    CIN   NL  72 237  15  52   8   2  0  25  3  0  30
##     SO IBB HBP SH SF GIDP
## 1   23  NA   0  4 NA   NA
## 2   40  NA   3 14 NA   NA
## 3   55  NA   5  8 NA   NA
## 4   42  NA   6 15 NA   NA
## 5   45  NA   6 24 NA   NA
## 6   46  NA   4 16 NA   NA
## 7   NA  NA   8 11 NA   NA
## 8   NA  NA   5  8 NA   NA
## 9   31  NA   4 11 NA   NA
## 10 22  NA   6  6 NA   NA
## 11 43  NA  10  9 NA   NA
## 12 39  NA   2 14 NA   NA
## 13 34  NA   4 16 NA   NA
## 14 21  NA   2  9 NA   NA
## 15 22  NA   1  9 NA   NA
## 16 28  NA   2  7 NA   NA
## 17 19  NA   3 15 NA   NA
## 18 24  NA   4 27 NA   NA
## 19 14  NA   3 22 NA   NA
## 20 18  NA   1 15 NA   NA
## 21 12  NA   5  5 NA   NA
## 22  2  NA   1 13 NA   NA
## 23 12  NA   5 12 NA   NA
## 24 16  NA   4  2 NA   NA
## 25 64  NA   2  3 NA   10
## 26 54  NA   3  1 NA   13
## 27 27  NA   3  0 NA   10
## 28 51  NA   4  0 NA   12
## 29 44  NA   2  0 NA   12
## 30 47  NA   2  1 NA   10
## 31 41  NA   3  0 NA   10
## 32 48  NA   2  0 NA   22
## 33 21  NA   0  0 NA   12
## 34 45  NA   0  0 NA   10
## 35  2  NA   0  0 NA    0
## 36 10  NA   0  0 NA    1
## 37 32  NA   1  0  3   10
## 38 24  17   2  0  4    8
## 39 39  11   1  0  0   13
## 40 43  33   5  0  2   11
```

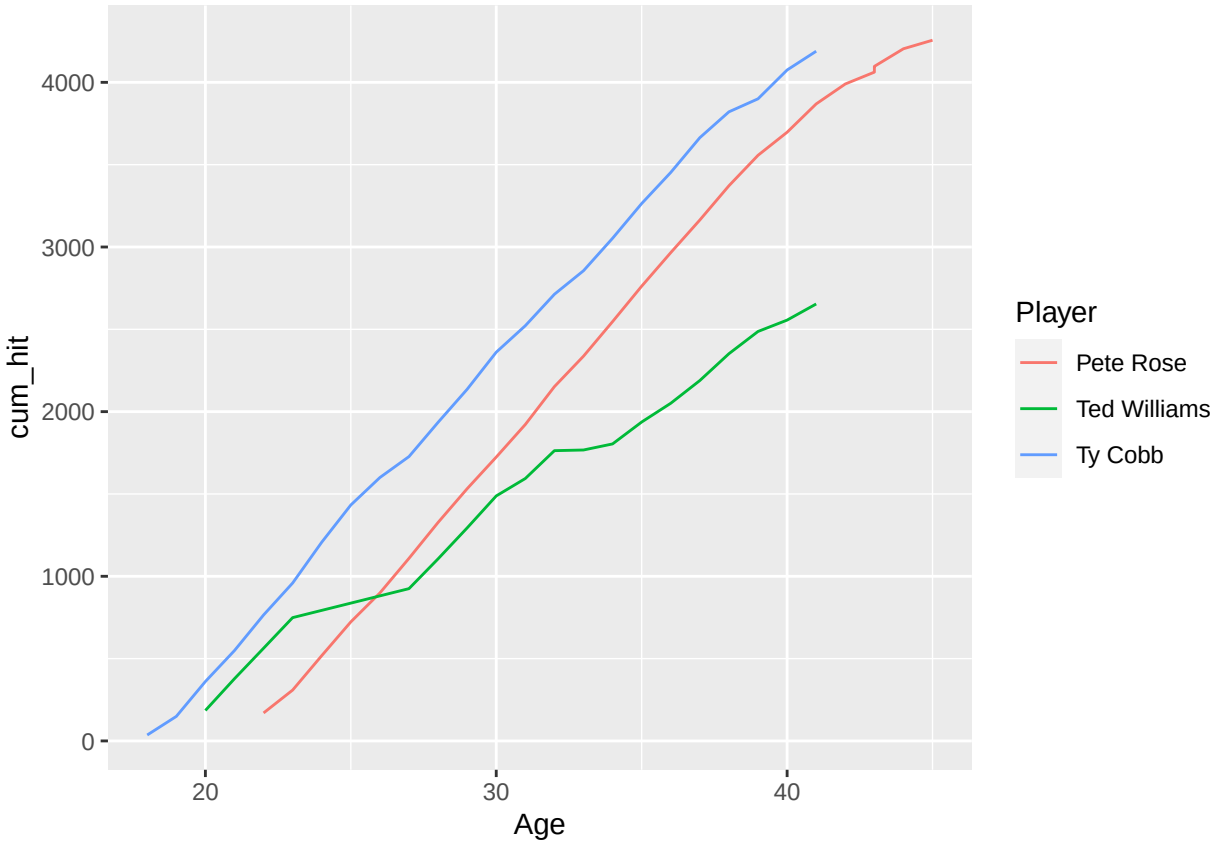
```
## 41 49  12   4  0  4   19
## 42 27   6   2  0  5    7
## 43 41   7   3  0  2    7
## 44 72   0   5  6  6    8
## 45 51   0   2  3  1    6
## 46 76   2   8  8  2   10
## 47 61   3   1  7  1   12
## 48 66   9   3  1  2    9
## 49 76  15   4  2  4   11
## 50 65  18   5  2  6   13
## 51 64  10   2  0  4    7
## 52 50  15   3  1  3    9
## 53 46   4   7  2  2    7
## 54 42   6   6  1  0   14
## 55 54  14   5  1  6    9
## 56 50   8  11  1  1   13
## 57 54   7   6  0  2   17
## 58 42   7   5  1  4    9
## 59 30   6   3  2  7    8
## 60 32  10   2  0  5   18
## 61 33   5   6  4  4   13
## 62 26   5   3  1  3    8
## 63 32   9   7  8  3   12
## 64 28   5   2  1  7   11
## 65 20   3   1  3  1   10
## 66  7   1   2  0  0    1
## 67 35   5   4  1  4   10
## 68 31   0   4  0  1    2
```

```
# (c) Add the variable Age to each data frame

players_info <- bind_rows(get_birthyear("Ty Cobb"),
                          get_birthyear("Ted Williams"),
                          get_birthyear("Pete Rose")
)

players_info
```

```
##    playerID       Player birthyear
## 1  cobbty01      Ty Cobb      1887
## 2 willite01 Ted Williams      1919
## 3  rosepe01    Pete Rose      1941
## 4  rosepe02    Pete Rose      1970
```

```
season_batting_stats %>% inner_join(players_info, by="playerID") %>%
    mutate(Age = yearID - birthyear) -> season_batting_stats


season_batting_stats %>%
    group_by(playerID) %>%
    mutate(cum_hit = cumsum(H)) -> season_batting_stats

# (d) Create a line plot for three players.
```

```
ggplot(season_batting_stats, aes(Age, cum_hit, color=Player))+
    geom_line()
```



From the line graph, we can see that Pete Rose and Ty Cobb had constant number of hits throughout their careers. On the other hands, Ted Williams had different pattern.He had good hitting number in his early 20s and he did not get hits so much in his late 20s.

7. Working with the Retrosheet Play-by-Play Dataset

```
# (a) Create two data frames mac.data and sosa.data.
mac.data <- hr_race %>% filter(BAT_ID==mac_id)
sosa.data <- hr_race %>% filter(BAT_ID==sosa_id)

# (b) Filter the 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 appearances 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 return the number of the plate appearances when the player 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 diff(), the following commands compute
# the spacings between the occurrences of home runs.
mac.spacings <- diff(c(0,mac.HR.PA))
sosa.spacings <- diff(c(0,sosa.HR.PA))

# Create a new data frame HR_Spacing with two variable Player and Spacing.

Player <- c("Mark McGwire", "Sammy Sosa")
Spacings <- c(mac.spacings, sosa.spacings)

HR_Spacings <- data.frame(Player,Spacings)

ggplot(HR_Spacings) +
    geom_histogram(aes(Spacings, fill = Player))+
    scale_x_continuous(breaks = seq(0,50,5))+
    scale_y_continuous(breaks = seq(0,30,5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```