Homework 1 - You Can Be A Baseball Analytics Star Too!

The Lahman Package

Sean Lahman is an author and journalist, who led the fight to make sports data open to the public. One of his most famous efforts was creating an open-source database of baseball data. A version of his database is in R through the **Lahman** package. In this homework, we will practice our skills at making plots! Much of the "grunt" work will be done by me, but you will still have plenty to do and change!!

Loading packages

```
# You may have to install these packages before the document will knit. I have
# included the installation lines below but commented out! Get rid of the
# hashtags before running. You only have to install the package once, but every
# time you knit the document you need to have the library statements because it
# opens up a fresh session of R behind the scenes.

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

# loading the tidyverse and setting the theme to black-white (it's my favorite theme)
library("tidyverse");theme_set(theme_bw())
# loading the Lahman package
library("Lahman")
```

Now that we have everything installed and loaded into R, we can take a look at all of the datasets in the Lahman package... and there are a lot of them! Running the code `data(package = "Lahman")` will open a list of data sets in the Lahman package. There are 30 datasets in total! We will focus on the `Batting` dataset. Here it is below:

```
# original data is a data frame but tibbles are nicer to work with
(batting <- as_tibble(Batting))
```

```
## # A tibble: 112,184 x 22
##    playerID  yearID stint teamID lgID      G    AB     R     H   X2B   X3B    HR
##    <chr>      <int> <int> <fct>  <fct> <int> <int> <int> <int> <int> <int> <int>
##  1 abercda01   1871     1 TRO    NA        1     4     0     0     0     0     0
##  2 addybo01    1871     1 RC1    NA       25   118    30    32     6     0     0
##  3 allisar01   1871     1 CL1    NA       29   137    28    40     4     5     0
##  4 allisdo01   1871     1 WS3    NA       27   133    28    44    10     2     2
##  5 ansonca01   1871     1 RC1    NA       25   120    29    39    11     3     0
##  6 armstbo01   1871     1 FW1    NA       12    49     9    11     2     1     0
##  7 barkeal01   1871     1 RC1    NA        1     4     0     1     0     0     0
##  8 barnero01   1871     1 BS1    NA       31   157    66    63    10     9     0
##  9 barrebi01   1871     1 FW1    NA        1     5     1     1     1     0     0
## 10 barrofr01   1871     1 BS1    NA       18    86    13    13     2     1     0
## # i 112,174 more rows
## # i 10 more variables: RBI <int>, SB <int>, CS <int>, BB <int>, SO <int>,
## #   IBB <int>, HBP <int>, SH <int>, SF <int>, GIDP <int>
```

```
# let's look at the range of years we have!
batting$yearID %>% range()
```

```
## [1] 1871 2022
```

```
# from 1871 to 2018! That's a lot!
```

Notes about the code above:

- parentheses around an assignment statement will print the result
- the $ operator will access the specific variable in the tibble
- The %>% is called the pipe operator and takes the result of the left side and puts it in the first argument of the function on the right.

That's a lot of data (147 years worth) in there! Let's hone in something interesting in baseball's past that would be cool to look at! In 1969, the MLB lowered the pitching mound by 10 inches and shrunk the strike zone to what it is today. This really benefited the batters and took a huge advantage away from pitchers. Thus, let's look at batting averages for those two seasons and see if there is a difference graphically. Specifically, let's make a side-by-side boxplot to carry out our comparison. The code below carries out this task (almost!).

Description and comments of the code below:

- This code below is often called a data analysis pipeline because I can change the years or the minimum at-bats, and get a whole different analysis (here just a plot) without having to re-write any code.
- I also hope that you notice/think this code is pretty readable! Without any R knowledge, you could probably tell me what is happening. That is what the tidyverse is trying to do!
- Code steps explanation:
  - The first step filters the data to only include the years 1968 and 1969. The | represents or and to check equality is ==. A single = is assigning a value to a variable.
  - The second step changes the `yearID` variable to be a factor (categorical variable) because R will yell at you if year is still a numeric value when plotting.
  - The third step filters the dataset so that only players with more than 50 at-bats are included. This was a personal choice that you can change if you want! It gets rid of some outliers due to players only batting a few times that year.
  - The fourth step creates a new variable, `avg` that holds the batting average, because it was not present in the original dataset.

Problem 1: The code below doesn't create the side-by-side boxplot of batting average for the two years mentioned above. You need to complete the code so that it produces the correct plot! I have included the ggplot call but you need to choose the right aesthetics and finish the `geom_` call. Make sure to remove the hashtags before the last %>% and the `ggplot` call or it won't make the graph.

```
batting %>%
   filter(yearID == 1968 | yearID == 1969) %>%
   mutate(yearID = as_factor(yearID)) %>%
   filter(AB > 50) %>%
   mutate(avg = H / AB)  # %>%
```

```
## # A tibble: 871 x 23
##    playerID  yearID stint teamID lgID      G    AB     R     H   X2B   X3B    HR
##    <chr>     <fct>  <int> <fct>  <fct> <int> <int> <int> <int> <int> <int> <int>
##  1 aaronha01 1968       1 ATL    NL      160   606    84   174    33     4    29
##  2 aaronto01 1968       1 ATL    NL       98   283    21    69    10     3     1
##  3 adairje01 1968       1 BOS    AL       74   208    18    45     1     0     2
##  4 adlesda01 1968       1 HOU    NL       40   104     3    19     1     1     0
##  5 ageeto01  1968       1 NYN    NL      132   368    30    80    12     3     5
##  6 alcarlu01 1968       1 LAN    NL       41   106     4    16     1     0     2
```

```
##  7 allenbe01 1968       1 WS2    AL      120   373    31    90    12     4     6
##  8 allendi01 1968       1 PHI    NL      152   521    87   137    17     9    33
##  9 allenha02 1968       1 WS2    AL       68   128    16    28     2     2     1
## 10 alleyge01 1968       1 PIT    NL      133   474    48   116    20     2     4
## # i 861 more rows
## # i 11 more variables: RBI <int>, SB <int>, CS <int>, BB <int>, SO <int>,
## #   IBB <int>, HBP <int>, SH <int>, SF <int>, GIDP <int>, avg <dbl>
```

```
    # ggplot(aes()) + geom_()
```

Problem 2: Comment on the plot's findings.

Answer:

Another interesting time in baseball was the steroid era, which I will define as the 1990's! During this time, an increased number of major league players used performance enhancing drugs. One way that we could explore the data to see if there were any side-effects is looking at home run production. Specifically, I want to make a plot of the total home run count for every year between 1980 and 2000 with dots and a path between them. This will give us some reference for before and during the time when steroids were present. For this plot, I also want to put the actual home run totals on the plot above the points.

Problem 3: Again, fill in the correct aesthetics that are missing and the correct geoms to make the points and the path between them. Again make sure to take away the hashtags to produce the plot.

```
batting %>%
    filter(yearID >= 1980 & yearID <= 2000) %>%
    group_by(yearID) %>%
    summarize(hr_total = sum(HR)) # %>%
```

```
## # A tibble: 21 x 2
##    yearID hr_total
##     <int>    <int>
##  1   1980     3087
##  2   1981     1781
##  3   1982     3379
##  4   1983     3301
##  5   1984     3258
##  6   1985     3602
##  7   1986     3813
##  8   1987     4458
##  9   1988     3180
## 10   1989     3083
## # i 11 more rows
```

```
     # ggplot(aes( , , label = hr_total)) + geom_() + geom_() + geom_label(size = 3, vjust = -.25) + la
```

Problem 4: Comment on the plot. Does it seem to support increased doping in that time span? Do you think any external factors other than doping could have affected the increase?

Answer:

One of the most famous players in the doping era was Barry Bonds. Barry Bonds played for the Pittsburgh Pirates early in his career and was very successful there, but it wasn't until he started playing for the San Francisco Giants that he started putting up huge home run numbers. If you look up "Barry Bonds Doping", you will find many pictures of the difference in body mass between Pittsburgh and in San Francisco. Many people point to that as evidence toward doping. Let's look at his career trajectory through his home run production.

Problem 5: I have done the filtering for you, but I want you to create the entire ggplot by yourself! I

have the ggplot call there but nothing else. Remember to chain ggplot2 statements together with +'s Do the following:

- Create a point plot of home runs over the years he was active.
- Color the points with the team he was playing for at the time. You'll need to add another aesthetic.
- Change the labels to be more descriptive.
- Change the legend so that the legend title says Team and the team name is what's describing the color (Pirates, Giants). You will accomplish this with the function `scale_color_discrete()`. Make sure to look it up for reference.

```
batting %>%
    filter(playerID == "bondsba01") #%>%
```

```
## # A tibble: 22 x 22
##    playerID  yearID stint teamID lgID      G    AB     R     H   X2B   X3B    HR
##    <chr>      <int> <int> <fct>  <fct> <int> <int> <int> <int> <int> <int> <int>
##  1 bondsba01   1986     1 PIT    NL      113   413    72    92    26     3    16
##  2 bondsba01   1987     1 PIT    NL      150   551    99   144    34     9    25
##  3 bondsba01   1988     1 PIT    NL      144   538    97   152    30     5    24
##  4 bondsba01   1989     1 PIT    NL      159   580    96   144    34     6    19
##  5 bondsba01   1990     1 PIT    NL      151   519   104   156    32     3    33
##  6 bondsba01   1991     1 PIT    NL      153   510    95   149    28     5    25
##  7 bondsba01   1992     1 PIT    NL      140   473   109   147    36     5    34
##  8 bondsba01   1993     1 SFN    NL      159   539   129   181    38     4    46
##  9 bondsba01   1994     1 SFN    NL      112   391    89   122    18     1    37
## 10 bondsba01   1995     1 SFN    NL      144   506   109   149    30     7    33
## # i 12 more rows
## # i 10 more variables: RBI <int>, SB <int>, CS <int>, BB <int>, SO <int>,
## #   IBB <int>, HBP <int>, SH <int>, SF <int>, GIDP <int>
```

```
    # ggplot() +
```

Problem 6: Comment on the plot.

For this last plot, I want you to think of your favorite baseball player. If you don't have one or don't care about baseball, then search for the list of the greatest baseball hitters of all time and choose from one of those. We are going to plot their batting average over time and compare it to every other player in the `Batting` dataset! Specifically, we are going to do the following:

- Search for the `playerID` of your person.
- Create a separate dataset just for them and make the batting average variable.
- Create a dataset for everyone else that only contains batting averages for seasons with more than 50 at-bats. Also, we need to standardize the years so that they all fit on one plot.
- Plot the averages over the career years with a line in the background.
- Overlay your favorite players in red.

To find your player, we need to look in the `Master` dataset that contains personal information about them. For instance, if I was looking for a player named Roberto Clemente, I would do the following:

```
filter(People, nameLast == "Clemente")
```

```
##    playerID birthYear birthMonth birthDay birthCountry birthState birthCity
## 1 clemeed02      1975         12       15         P.R.       <NA>  Santurce
## 2 clemero01      1934          8       18         P.R.       <NA>  Carolina
##   deathYear deathMonth deathDay deathCountry deathState deathCity nameFirst
## 1        NA         NA       NA         <NA>       <NA>      <NA>    Edgard
## 2      1972         12       31         P.R.       <NA>  San Juan   Roberto
##   nameLast               nameGiven weight height bats throws      debut
```

```
## 1 Clemente Edgard Alexis Velazquez    188     71    R      R 1998-09-10
## 2 Clemente                  Roberto    175     71    R      R 1955-04-17
##    finalGame  retroID   bbrefID  deathDate  birthDate
## 1 2000-07-31 cleme001 clemeed02       <NA> 1975-12-15
## 2 1972-10-03 clemr101 clemero01 1972-12-31 1934-08-18
```

Next, make a dataset with just that player. Also, add two new variables. One is the batting average and two is the career years that the person played. For instance, if I were to make one for Roberto, it would look like this:

```
clem_bat <- batting %>%
    filter(playerID == "clemero01") %>%
    mutate(avg = H / AB, career = yearID - min(yearID))
```

Now we are going to do the same thing for everyone else, and add in the more than 50 AB condition!

```
other_bats <- batting %>%
    filter(AB > 50) %>%
    group_by(playerID) %>%
    arrange(playerID, yearID, teamID) %>%
    mutate(avg = H / AB, career = yearID - min(yearID))
```

Lastly, we need to make the plot! Here are the steps you should follow to make the plot:

Problem 7:

- First call `ggplot()` using the `other_bats` data with aesthetics to show change in batting average over time.
- Add on a geom that tracks the path from one year to another. Don't use dots. In your call to the geom, make sure to add the statement `alpha = 0.02`. This is called alpha blending and it help you avoid overplotting!
- Make the x axis go from 0 to 20 by 5 and the y axis go from 0 to 0.4 by 0.1. This is accomplished by adding the two lines `scale_x_continuous("Career Year", breaks = seq(0, 20, 5), limits = c(0, 20))` and `scale_y_continuous("Batting Average", breaks = seq(0, 0.4, 0.1), limits = c(0, 0.4))`. This will also fix the labels.
- Lastly, make another line/path using just the data from your player's small dataset and color it red.

Put the code in the empty code chunk below.

Extra Credit: If you are really interested in making these plots, I challenge you to try and recreate the plot below about Pete Rose.