# Final Project

2022-12-06

The `Lahman` dataset contains statistics using MLB data from 1871 to 2021. I will use the `Batting Table` and the `Pitching Table`. Specifically for the `Batting Table` I will be adding the columns `Singles`, `SLG`, `BA`, and `OBP`. The data only looked at players from 1950 and newer as well as the stats I chose to be must be greater than 0. Also, the number of games a player played must be greater than 150, at bats greater than 500, and batting average greater than 0.250.

```
data(Batting)

singles <- (Batting$H-Batting$X2B-Batting$X3B-Batting$HR)
SLG <- ((singles + Batting$X2B*2 + Batting$X3B*3 + Batting$HR*4)/Batting$AB)
BA <- (Batting$H/Batting$AB)
OBP <- ((Batting$H + Batting$IBB + Batting$BB +
           Batting$HBP)/(Batting$AB + Batting$IBB + Batting$BB + Batting$SF))

Batting <- Batting %>%
  add_column(Singles = singles,
             BA = BA,
             SLG =SLG,
             OBP = OBP)

Batting <- Batting %>%
  filter(yearID>=1950) %>%
  filter(AB & H & HR & Singles & BA>0) %>%
  filter(G>150) %>%
  filter(AB>500) %>%
  filter(BA>0.25)
```

Here we start to create the ranking. I choose `AB` = At Bat, `H` = Hits, `HR` = Home Runs, `Singles` = Singles, and `BA` = Batting Average. I created for loops where the loops find the current value in the column and divide it my the max of the column to get a percentage of how good that stat is compared to the max stat value. Then I added the columns to the data. Next, I summed the values in each row to get a total for each player. Finally, to get the rank we will use the `rank` function. The higher the rank the better the player.

```
for (val in Batting$playerID)
{
  x1 <- (Batting$AB/max(Batting$AB, na.rm=TRUE))
  x2 <- (Batting$H/max(Batting$H, na.rm=TRUE))
  x3 <- (Batting$HR/max(Batting$HR, na.rm=TRUE))
  x4 <- (Batting$Singles/max(Batting$Singles, na.rm=TRUE))
  x5 <- (Batting$BA/max(Batting$BA, na.rm=TRUE))
}

Batting <- Batting %>%
  add_column(x1 = x1,
             x2 = x2,
             x3 = x3,
```

```
              x4 = x4,
              x5 = x5)


for (val in Batting$playerID)
{
  total <- rowSums(Batting[ , c(27,28,29,30,31)], na.rm = TRUE)
}

Batting <- Batting %>%
  add_column(total = total)

rank <- rank(Batting$total, ties.method = "min")

Batting <- Batting %>%
  add_column(rank = rank)
```

Now we need to group by `yearID` to help make it easier to average the data.

```
Batting <- Batting %>%
  group_by(yearID)
```

Wrote the `Batting` dataset to a file as I need to manipulate the data in Excel. First I filtered the data for the specific years I wanted. Then I used the average function in Excel to average the data for the specific years.

```
write.csv(Batting,"~/STT 4890/Final Project/Batting.csv", row.names = FALSE)
```

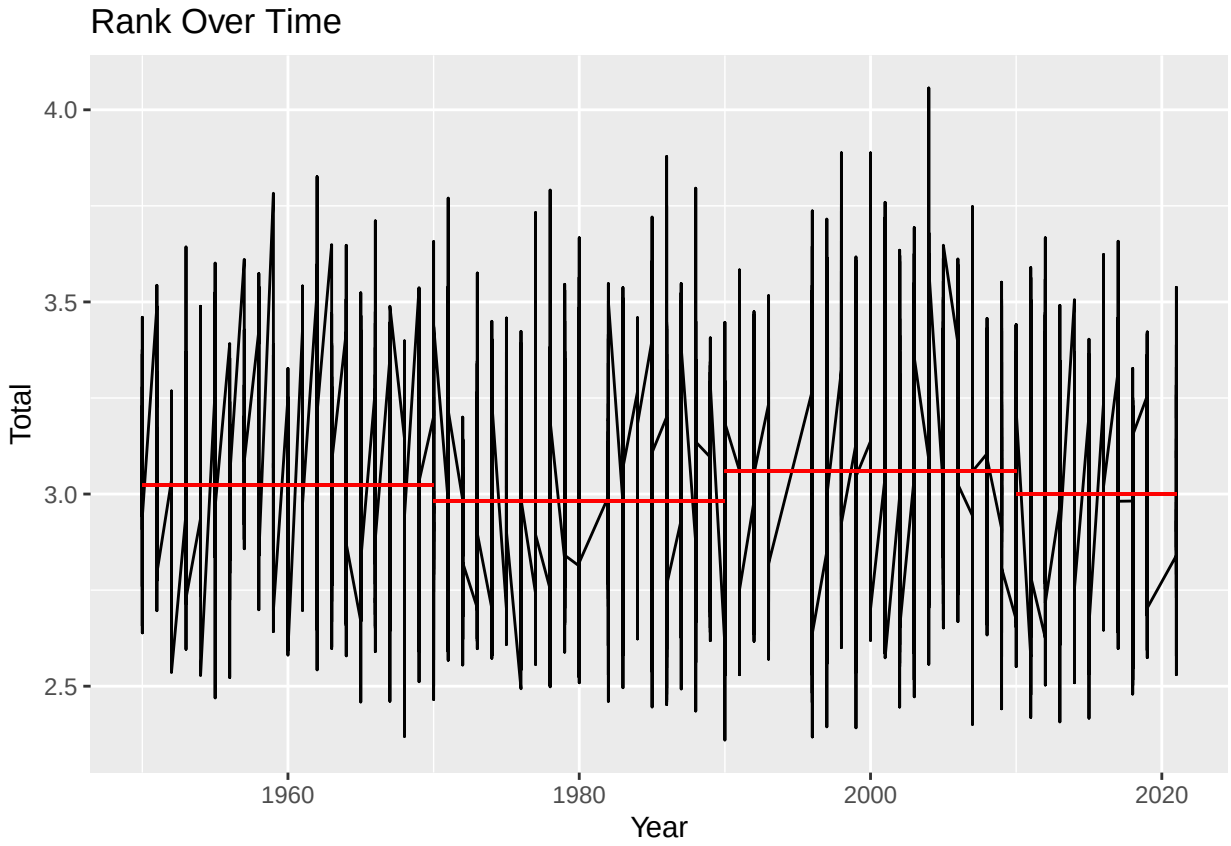
Here we read in the file with the averages. Then we view them to see the averages for each group of years.

```
bat_year_avg <- read_excel("bat_year_avg.xlsx")
head(bat_year_avg)
```

```
## # A tibble: 4 x 2
##   years     Average
##   <chr>       <dbl>
## 1 1950-1970    3.02
## 2 1970-1990    2.98
## 3 1990-2010    3.06
## 4 2010-2021    3.00
```

This graph is a line graph that shows how all the players ranked over the time from 1950 to 1970. The higher the total the better they were ranked. The red lines are the averages for each group of years players above the average were better and players below the average were worse.

```
p1 <- ggplot(Batting, aes(x=yearID, y=total)) + geom_line() +
  ggtitle("Rank Over Time") +
  geom_segment(x=1950, xend=1970, y=3.022714, yend=3.022714, colour = "red") +
  geom_segment(x=1970, xend=1990, y=2.981726, yend=2.981726, colour = "red") +
  geom_segment(x=1990, xend=2010, y=3.059210, yend=3.059210, colour = "red") +
  geom_segment(x=2010, xend=2021, y=3.000778, yend=3.000778, colour = "red") +
  labs(x="Year", y= "Total")
p1
```




Lastly, we want to see what  of players were above and below the average for each group of years. This is done by first finding the total number of rows in the specific data we need. Then finding the number of rows above and below and dividing them by the total number of rows. The results are shown in the table below with `year1` being 1950-1970, `year2` being 1970-1990, `year3` being 1990-2010, and `year4` being 2010-2021.

```
#1950-1970
a <- nrow(Batting[Batting$yearID>=1950 & Batting$yearID<= 1970, ])

b <- nrow(Batting[Batting$yearID>=1950 & Batting$yearID<= 1970 & Batting$total >= 3.022714, ])

c <- nrow(Batting[Batting$yearID>=1950 & Batting$yearID<= 1970 & Batting$total <= 3.022714, ])

#1970-1990
d <- nrow(Batting[Batting$yearID>=1970 & Batting$yearID<= 1990, ])

e <- nrow(Batting[Batting$yearID>=1970 & Batting$yearID<= 1990 & Batting$total >= 2.981726, ])

f <- nrow(Batting[Batting$yearID>=1970 & Batting$yearID<= 1990 & Batting$total <= 2.981726, ])

#1990-2010
g <- nrow(Batting[Batting$yearID>=1990 & Batting$yearID<= 2010, ])

h <- nrow(Batting[Batting$yearID>=1990 & Batting$yearID<= 2010 & Batting$total >= 3.059210, ])

i <- nrow(Batting[Batting$yearID>=1990 & Batting$yearID<= 2010 & Batting$total <= 3.059210, ])

#2010-2021
j <- nrow(Batting[Batting$yearID>=2010 & Batting$yearID<= 2021, ])
```

```
k <- nrow(Batting[Batting$yearID>=2010 & Batting$yearID<= 2021 & Batting$total >= 3.000778, ])

l <- nrow(Batting[Batting$yearID>=2010 & Batting$yearID<= 2021 & Batting$total <= 3.000778, ])


b/a
```

```
## [1] 0.4815466
```

```
c/a
```

```
## [1] 0.5184534
```

```
e/d
```

```
## [1] 0.4625144
```

```
f/d
```

```
## [1] 0.5374856
```

```
h/g
```

```
## [1] 0.4899905
```

```
i/g
```

```
## [1] 0.5100095
```

```
k/j
```

```
## [1] 0.4704684
```

```
l/j
```

```
## [1] 0.5295316
```

```
##       % below average % above average
## year1       0.4815466       0.5184534
## year2       0.4625144       0.5374856
## year3       0.4899905       0.5100095
## year4       0.4704684       0.5295316
```

Now we will look at ranking pitchers using the `Pitching` table from the `Lahman` dataset. We will filter the data to only include data from 1950 and newer as well as `IPouts` greater than 500. Then add the columns `nonHRhits` which is $H - HR$ and `FIP` which stands for Fielding Independent Pitching where it measures a pitcher's effectiveness taking plays that would involve the defense trying to field the ball out of the equation. `FIP` is composed of $(13 * Pitching\text{HR}) + 5nonHRhits + 3(\text{Pitching}BB + Pitching\text{HBP}$ - 2*Pitching$SO)/(Pitching\text{IPouts}))$. The higher the `FIP` the higher ranked the player is which implies they are better. After we add the `FIP` column, we rank the players with the higher the rank the better the player.

```
data(Pitching)

Pitching <- Pitching %>%
  filter(yearID >= 1950, IPouts > 500)
Pitching <- Pitching %>%
  mutate(nonHRhits = Pitching$H - Pitching$HR,
         FIP = (13*Pitching$HR) + 5*nonHRhits +
           3*(Pitching$BB + Pitching$HBP - 2*Pitching$SO)/(Pitching$IPouts))
```

```
rank <- rank(Pitching$FIP, ties.method = "min")

Pitching <- Pitching %>%
  add_column(rank = rank)
```

Then we group the data by `yearID` to to help make it easier to average the data.

```
Pitching <- Pitching %>%
  group_by(yearID)
```

Wrote the `Pitching` data set to a file as I need to manipulate the data in Excel. First I filtered the data for the specific years I wanted. Then I used the average function in Excel to average the data for the specific years.

```
write.csv(Pitching,"~/STT 4890/Final Project/Pitching.csv", row.names = FALSE)
```

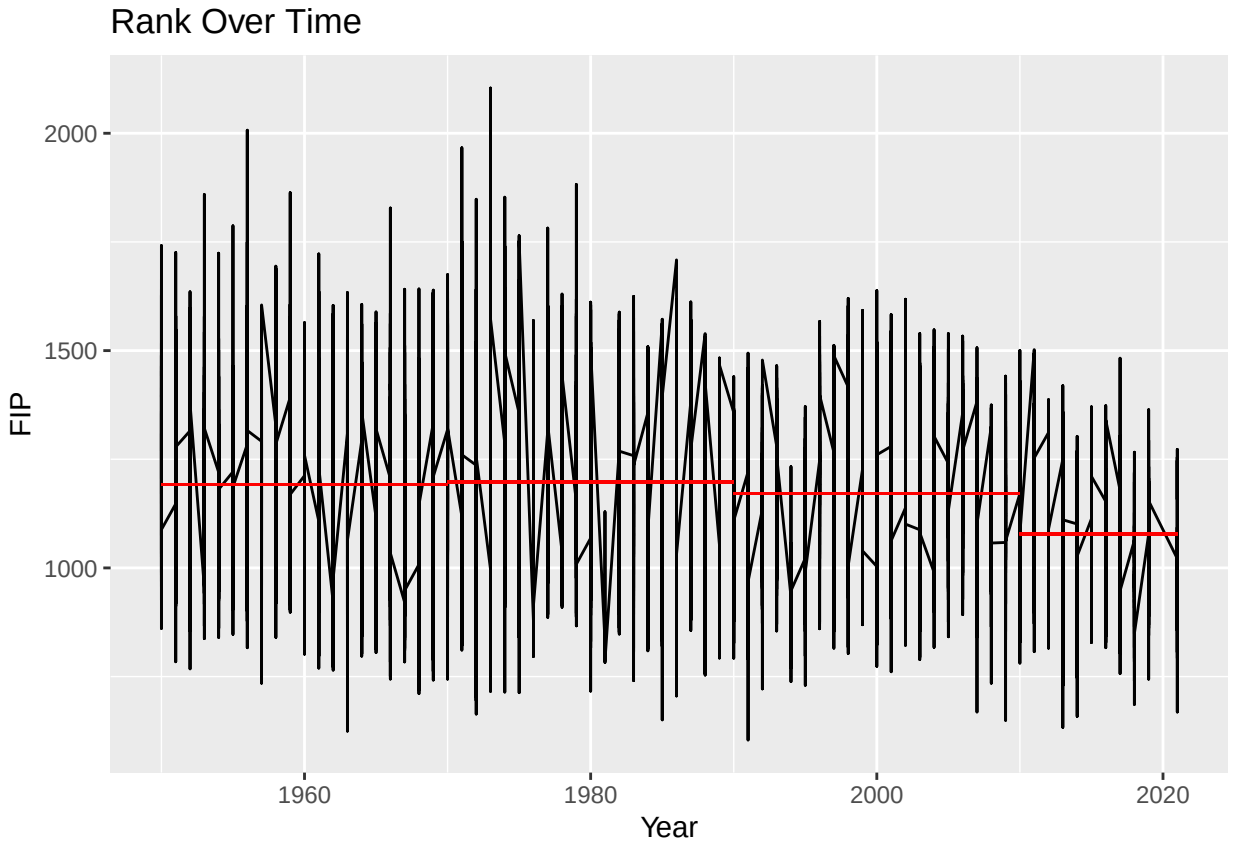
Here we read in the file with the averages. Then we view them to see the averages for each group of years.

```
pitch_year_avg <- read_excel("pitch_year_avg.xlsx")
as.tbl(pitch_year_avg)
```

```
## # A tibble: 4 x 2
##   Year      Average
##   <chr>       <dbl>
## 1 1950-1970   1192.
## 2 1970-1990   1198.
## 3 1990-2010   1171.
## 4 2010-2021   1079.
```

This graph is a line graph that shows how all the players ranked over the time from 1950 to 1970. The higher the `totalFIP‘` the better they were ranked. The red lines are the averages for each group of years players above the average were better and players below the average were worse.

```
p2 <- ggplot(Pitching, aes(x=yearID, y=FIP)) + geom_line() + ggtitle("Rank Over Time") +
  geom_segment(x=1950, xend=1970, y=1192.023, yend=1192.023, colour = "red") +
  geom_segment(x=1970, xend=1990, y=1197.672, yend=1197.672, colour = "red") +
  geom_segment(x=1990, xend=2010, y=1171.328, yend=1171.328, colour = "red") +
  geom_segment(x=2010, xend=2021, y=1078.646, yend=1078.646, colour = "red") +
  labs(x="Year", y= "FIP")
p2
```




Lastly, we want to see what  of players were above and below the average for each group of years. This is done by first finding the total number of rows in the specific data we need. Then finding the number of rows above and below and dividing them by the total number of rows. The results are shown in the table below with `year1.2` being 1950-1970, `year2.2` being 1970-1990, `year3.2` being 1990-2010, and `year4.2` being 2010-2021.

```
#1950-1970
a1 <- nrow(Pitching[Pitching$yearID>=1950 & Pitching$yearID<= 1970, ])

b1 <- nrow(Pitching[Pitching$yearID>=1950 & Pitching$yearID<= 1970 & Pitching$FIP >= 1192.023, ])

c1 <- nrow(Pitching[Pitching$yearID>=1950 & Pitching$yearID<= 1970 & Pitching$FIP <= 1192.023, ])

#1970-1990
d1 <- nrow(Pitching[Pitching$yearID>=1970 & Pitching$yearID<= 1990, ])

e1 <- nrow(Pitching[Pitching$yearID>=1970 & Pitching$yearID<= 1990 & Pitching$FIP >= 1197.672, ])

f1 <- nrow(Pitching[Pitching$yearID>=1970 & Pitching$yearID<= 1990 & Pitching$FIP <= 1197.672, ])

#1990-2010
g1 <- nrow(Pitching[Pitching$yearID>=1990 & Pitching$yearID<= 2010, ])

h1 <- nrow(Pitching[Pitching$yearID>=1990 & Pitching$yearID<= 2010 & Pitching$FIP >= 1171.328, ])

i1 <- nrow(Pitching[Pitching$yearID>=1990 & Pitching$yearID<= 2010 & Pitching$FIP <= 1171.328, ])

#2010-2021
```

```r
j1 <- nrow(Pitching[Pitching$yearID>=2010 & Pitching$yearID<= 2021, ])

k1 <- nrow(Pitching[Pitching$yearID>=2010 & Pitching$yearID<= 2021 & Pitching$FIP >= 1078.646, ])

l1 <- nrow(Pitching[Pitching$yearID>=2010 & Pitching$yearID<= 2021 & Pitching$FIP <= 1078.646, ])


b1/a1
```

```
## [1] 0.4726661
```

```r
c1/a1
```

```
## [1] 0.5273339
```

```r
e1/d1
```

```
## [1] 0.5101394
```

```r
f1/d1
```

```
## [1] 0.4898606
```

```r
h1/g1
```

```
## [1] 0.5137795
```

```r
i1/g1
```

```
## [1] 0.4862205
```

```r
k1/j1
```

```
## [1] 0.503639
```

```r
l1/j1
```

```
## [1] 0.496361
```

```
##         % below average % above average
## year1.2       0.4726661       0.5273339
## year2.2       0.5101394       0.4898606
## year3.2       0.5137795       0.4862205
## year4.2       0.5036390       0.4963610
```