Data_Visualization_Project

Angus_Brooks

3/8/2020

```
library(tidyverse)
Batting <- read.csv('Batting.csv')
Teams <- read.csv('Teams.csv')
Salaries <- read.csv('Salaries.csv')
People <- read.csv('People.csv')

glimpse(Batting)
```

```
## Observations: 107,429
## Variables: 22
## $ playerID <fct> abercda01, addybo01, allisar01, allisdo01, ansonca01, arms...
## $ yearID   <int> 1871, 1871, 1871, 1871, 1871, 1871, 1871, 1871, 1871, 1871...
## $ stint    <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ teamID   <fct> TRO, RC1, CL1, WS3, RC1, FW1, RC1, BS1, FW1, BS1, CL1, CL1...
## $ lgID     <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ G        <int> 1, 25, 29, 27, 25, 12, 1, 31, 1, 18, 22, 1, 10, 3, 20, 29,...
## $ AB       <int> 4, 118, 137, 133, 120, 49, 4, 157, 5, 86, 89, 3, 36, 15, 9...
## $ R        <int> 0, 30, 28, 28, 29, 9, 0, 66, 1, 13, 18, 0, 6, 7, 24, 26, 0...
## $ H        <int> 0, 32, 40, 44, 39, 11, 1, 63, 1, 13, 27, 0, 7, 6, 33, 32, ...
## $ X2B      <int> 0, 6, 4, 10, 11, 2, 0, 10, 1, 2, 1, 0, 0, 0, 9, 3, 0, 0, 1...
## $ X3B      <int> 0, 0, 5, 2, 3, 1, 0, 9, 0, 1, 10, 0, 0, 0, 1, 3, 0, 0, 1, ...
## $ HR       <int> 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 1, 0, 0, 0, 0, 0...
## $ RBI      <int> 0, 13, 19, 27, 16, 5, 2, 34, 1, 11, 18, 0, 1, 5, 21, 23, 0...
## $ SB       <int> 0, 8, 3, 1, 6, 0, 0, 11, 0, 1, 0, 0, 2, 2, 4, 4, 0, 0, 3, ...
## $ CS       <int> 0, 1, 1, 1, 2, 1, 0, 6, 0, 0, 1, 0, 0, 0, 0, 4, 0, 0, 1, 0...
## $ BB       <int> 0, 4, 2, 0, 2, 0, 1, 13, 0, 0, 3, 1, 2, 0, 2, 9, 0, 0, 4, ...
## $ SO       <int> 0, 0, 5, 2, 1, 1, 0, 1, 0, 0, 4, 0, 0, 0, 2, 2, 3, 0, 2, 0...
## $ IBB      <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ HBP      <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ SH       <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ SF       <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ GIDP     <int> 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 1, 2, 0, 0, 0, 0...
```

```
glimpse(Teams)
```

```
## Observations: 2,925
## Variables: 48
## $ yearID         <int> 1871, 1871, 1871, 1871, 1871, 1871, 1871, 1871, 1871...
## $ lgID           <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ teamID         <fct> BS1, CH1, CL1, FW1, NY2, PH1, RC1, TRO, WS3, BL1, BR...
```

```
## $ franchID       <fct> BNA, CNA, CFC, KEK, NNA, PNA, ROK, TRO, OLY, BLC, EC...
## $ divID          <fct> , , , , , , , , , , , , , , , , , , , , , , , , , , ,
## $ Rank           <int> 3, 2, 8, 7, 5, 1, 9, 6, 4, 2, 9, 6, 1, 7, 8, 3, 4, 5...
## $ G              <int> 31, 28, 29, 19, 33, 28, 25, 29, 32, 58, 29, 37, 48, ...
## $ Ghome          <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ W              <int> 20, 19, 10, 7, 16, 21, 4, 13, 15, 35, 3, 9, 39, 6, 5...
## $ L              <int> 10, 9, 19, 12, 17, 7, 21, 15, 15, 19, 26, 28, 8, 16,...
## $ DivWin         <fct> , , , , , , , , , , , , , , , , , , , , , , , , , , ,
## $ WCWin          <fct> , , , , , , , , , , , , , , , , , , , , , , , , , , ,
## $ LgWin          <fct> N, N, N, N, N, Y, N, N, N, N, N, N, Y, N, N, N, N, N...
## $ WSWin          <fct> , , , , , , , , , , , , , , , , , , , , , , , , , , ,
## $ R              <int> 401, 302, 249, 137, 302, 376, 231, 351, 310, 617, 15...
## $ AB             <int> 1372, 1196, 1186, 746, 1404, 1281, 1036, 1248, 1353,...
## $ H              <int> 426, 323, 328, 178, 403, 410, 274, 384, 375, 753, 24...
## $ X2B            <int> 70, 52, 35, 19, 43, 66, 44, 51, 54, 106, 29, 35, 107...
## $ X3B            <int> 37, 21, 40, 8, 21, 27, 25, 34, 26, 31, 9, 10, 30, 5,...
## $ HR             <int> 3, 10, 7, 2, 1, 9, 3, 6, 6, 14, 0, 1, 7, 0, 2, 4, 4,...
## $ BB             <int> 60, 60, 26, 33, 33, 46, 38, 49, 48, 29, 18, 19, 29, ...
## $ SO             <int> 19, 22, 25, 9, 15, 23, 30, 19, 13, 28, 40, 25, 26, 1...
## $ SB             <int> 73, 69, 18, 16, 46, 56, 53, 62, 48, 53, 8, 19, 48, 1...
## $ CS             <int> 16, 21, 8, 4, 15, 12, 10, 24, 13, 18, 13, 16, 14, 3,...
## $ HBP            <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ SF             <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ RA             <int> 303, 241, 341, 243, 313, 266, 287, 362, 303, 434, 41...
## $ ER             <int> 109, 77, 116, 97, 121, 137, 108, 153, 137, 166, 160,...
## $ ERA            <dbl> 3.55, 2.76, 4.11, 5.17, 3.72, 4.95, 4.30, 5.51, 4.37...
## $ CG             <int> 22, 25, 23, 19, 32, 27, 23, 28, 32, 48, 28, 37, 41, ...
## $ SHO            <int> 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 4, 0, 0, 3, 1, 2...
## $ SV             <int> 3, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 4, 0, 0, 1, 0, 1...
## $ IPouts         <int> 828, 753, 762, 507, 879, 747, 678, 750, 846, 1548, 7...
## $ HA             <int> 367, 308, 346, 261, 373, 329, 315, 431, 371, 573, 48...
## $ HRA            <int> 2, 6, 13, 5, 7, 3, 3, 4, 4, 3, 7, 6, 0, 6, 6, 2, 3, ...
## $ BBA            <int> 42, 28, 53, 21, 42, 53, 34, 75, 45, 63, 36, 21, 27, ...
## $ SOA            <int> 23, 22, 34, 17, 22, 16, 16, 12, 13, 77, 13, 13, 29, ...
## $ E              <int> 243, 229, 234, 163, 235, 194, 220, 198, 218, 432, 27...
## $ DP             <int> 24, 16, 15, 8, 14, 13, 14, 22, 20, 22, 9, 15, 44, 17...
## $ FP             <dbl> 0.834, 0.829, 0.818, 0.803, 0.840, 0.845, 0.821, 0.8...
## $ name           <fct> Boston Red Stockings, Chicago White Stockings, Cleve...
## $ park           <fct> South End Grounds I, Union Base-Ball Grounds, Nation...
## $ attendance     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ BPF            <int> 103, 104, 96, 101, 90, 102, 97, 101, 94, 106, 87, 11...
## $ PPF            <int> 98, 102, 100, 107, 88, 98, 99, 100, 98, 102, 96, 122...
## $ teamIDBR       <fct> BOS, CHI, CLE, KEK, NYU, ATH, ROK, TRO, OLY, BAL, EC...
## $ teamIDlahman45 <fct> BS1, CH1, CL1, FW1, NY2, PH1, RC1, TRO, WS3, BL1, BR...
## $ teamIDretro    <fct> BS1, CH1, CL1, FW1, NY2, PH1, RC1, TRO, WS3, BL1, BR...
```

```
glimpse(Salaries)
```

```
## Observations: 26,428
## Variables: 5
## $ yearID   <int> 1985, 1985, 1985, 1985, 1985, 1985, 1985, 1985, 1985, 1985...
## $ teamID   <fct> ATL, ATL, ATL, ATL, ATL, ATL, ATL, ATL, ATL, ATL, ATL, ATL...
## $ lgID     <fct> NL, NL, NL, NL, NL, NL, NL, NL, NL, NL, NL, NL, NL, NL, NL...
## $ playerID <fct> barkele01, bedrost01, benedbr01, campri01, ceronri01, cham...
```

```
## $ salary   <int> 870000, 550000, 545000, 633333, 625000, 800000, 150000, 48...
```

```
glimpse(People)
```

```
## Observations: 19,878
## Variables: 24
## $ playerID     <fct> aardsda01, aaronha01, aaronto01, aasedo01, abadan01, a...
## $ birthYear    <int> 1981, 1934, 1939, 1954, 1972, 1985, 1850, 1877, 1869, ...
## $ birthMonth   <int> 12, 2, 8, 9, 8, 12, 11, 4, 11, 10, 3, 10, 2, 8, 9, 6, ...
## $ birthDay     <int> 27, 5, 5, 8, 25, 17, 4, 15, 11, 14, 16, 22, 16, 17, 19...
## $ birthCountry <fct> USA, USA, USA, USA, USA, D.R., USA, USA, USA, USA, USA...
## $ birthState   <fct> CO, AL, AL, CA, FL, La Romana, PA, PA, VT, NE, OH, OH,...
## $ birthCity    <fct> Denver, Mobile, Mobile, Orange, Palm Beach, La Romana,...
## $ deathYear    <int> NA, NA, 1984, NA, NA, NA, 1905, 1957, 1962, 1926, 1930...
## $ deathMonth   <int> NA, NA, 8, NA, NA, NA, 5, 1, 6, 4, 2, 6, NA, NA, NA, N...
## $ deathDay     <int> NA, NA, 16, NA, NA, NA, 17, 6, 11, 27, 13, 11, NA, NA,...
## $ deathCountry <fct> , , USA, , , , USA, USA, USA, USA, USA, USA, , , , , ,...
## $ deathState   <fct> , , GA, , , , NJ, FL, VT, CA, MI, CA, , , , , , DC, , ...
## $ deathCity    <fct> , , Atlanta, , , , Pemberton, Fort Lauderdale, Colches...
## $ nameFirst    <fct> David, Hank, Tommie, Don, Andy, Fernando, John, Ed, Be...
## $ nameLast     <fct> Aardsma, Aaron, Aaron, Aase, Abad, Abad, Abadie, Abbat...
## $ nameGiven    <fct> David Allan, Henry Louis, Tommie Lee, Donald William, ...
## $ weight       <int> 215, 180, 190, 190, 184, 220, 192, 170, 175, 169, 190,...
## $ height       <int> 75, 72, 75, 75, 73, 73, 72, 71, 71, 68, 71, 70, 78, 74...
## $ bats         <fct> R, R, R, R, L, L, R, R, R, L, R, R, R, R, L, R, L, L, ...
## $ throws       <fct> R, R, R, R, L, L, R, R, R, L, R, R, R, L, L, R, L, R, ...
## $ debut        <fct> 2004-04-06, 1954-04-13, 1962-04-10, 1977-07-26, 2001-0...
## $ finalGame    <fct> 2015-08-23, 1976-10-03, 1971-09-26, 1990-10-03, 2006-0...
## $ retroID      <fct> aardd001, aaroh101, aarot101, aased001, abada001, abad...
## $ bbrefID      <fct> aardsda01, aaronha01, aaronto01, aasedo01, abadan01, a...
```

For my project, I decided to dive into the world of baseball. I grew up playing baseball and remain an avid baseball fan to this day. As it so happens, many statisticians also love baseball and have dutifully kept in-depth statistics of teams and players since the inception of the game in the 1800's. Luckily for me, a journalist named Sean Lahman created a website where people compile different baseball datasets. I found 4 different datasets created by different authors that I used to combine together for my project. One dataset, "Batting", contains information on hitting statistics for all hitters in the major leagues since 1871. The statistics include the player's unique ID, games played, number of hits, number of homeruns, number of strikeouts, etc. The dataset "Salaries" contains the amount of money each player has been paid per year between 1985 and 2016. The "Teams" dataset contains information on the performance of each team as a whole per year since 1871. The dataset includes variables such as the rank in the division, the number of runs scored, the number of hits (including the number of doubles, triples, homeruns), ERA of pitchers, and overall fan attendance. The "People" dataset gives personal information of all the players such as birth and death year, full name, place of birth, height, weight, and major league debut date. When joined, the dataset will contain information about player performance, pay, and physical characteristics as well as team statistics. I expect to find that player performance (batting average, number of homeruns) will be positively correlated with pay because, in theory, better players are paid more money. I also expect to find that teams with high batting averages or low ERA's tend to perform better than other teams and win more world series due simply to scoring more or preventing the other team from scoring.

```
Joined_data <- left_join(Batting, Salaries, by= c("playerID", "yearID", "teamID", "lgID"))%>%
  left_join(People, by="playerID") %>% left_join(Teams, by=c("yearID", "teamID", "lgID"),
  suffix=c(".player", ".team"))
```

I joined the four datasets together to create a singular dataset that shows the per year performance of players and teams as well as personal player information. I performed the join through using left joins, starting with Batting as the main dataset. Starting with Batting made the most sense because all of the individual players each year join together to form teams. This means that player stats should be listed before team statistics. My first join was to attach yearly salary information to each player. Salaries were available only from 1985 forward (and only for players who spent enough time in the majors to have a major league contract) so only roughly a quarter of all total players had salary information. I then used another left join to attach biographical information for each player using each player's unique ID. I then used a left join to attach team information onto each player using the unique ID of each player and the team ID for which he played. Left joins were used in order to attach information onto player ID's and drop any extraneous information (which was not a problem since the information was nearly completely comprehensive).

```
Joined_data_2010_newer <- Joined_data %>% filter(yearID >= 2010)

Joined_data_2010_newer %>% filter(AB.player > 100) %>% group_by(teamID, yearID)%>%
  summarize(mean_salary = mean(salary, na.rm=T), median_salary=median(salary, na.rm=T)) %>%
  na.omit() %>% arrange(yearID, desc(mean_salary))
```

```
## # A tibble: 210 x 4
## # Groups:   teamID [31]
##    teamID yearID mean_salary median_salary
##    <chr>   <int>       <dbl>         <dbl>
##  1 NYA      2010   10259127.       6850000
##  2 BOS      2010    7529154.       7750000
##  3 CHN      2010    6295917.       2800000
##  4 PHI      2010    6089542        5000000
##  5 DET      2010    6056156.       2600000
##  6 NYN      2010    5809234.       5000000
##  7 SEA      2010    4824500        3000000
##  8 MIN      2010    4788782        4000000
##  9 LAN      2010    4705474.       3550000
## 10 HOU      2010    4445455.       1500000
## # ... with 200 more rows
```

```
Joined_data_2010_newer <- Joined_data_2010_newer %>% mutate(Batting_Average.player =
                                                              H.player/AB.player)
Joined_data_2010_newer <- Joined_data_2010_newer %>% mutate(Batting_Average.team =
                                                              H.team/AB.team)

Joined_data_2010_newer %>% filter(AB.player > 100) %>%
  select(teamID, yearID, playerID, Batting_Average.player, Batting_Average.team, salary) %>%
  group_by(teamID, yearID) %>% summarize(mean_batting_avg=mean(Batting_Average.player),
  sd_batting_average=sd(Batting_Average.player), salary_mean=mean(salary, na.rm=T)) %>%
  arrange(yearID, desc(salary_mean))
```

```
## # A tibble: 300 x 5
## # Groups:   teamID [31]
##    teamID yearID mean_batting_avg sd_batting_average salary_mean
##    <chr>   <int>            <dbl>              <dbl>       <dbl>
##  1 NYA      2010            0.265             0.0247   10259127.
##  2 BOS      2010            0.266             0.0308    7529154.
##  3 CHN      2010            0.264             0.0229    6295917.
##  4 PHI      2010            0.262             0.0331    6089542
```

```
##  5 DET      2010            0.264             0.0322    6056156.
##  6 NYN      2010            0.249             0.0277    5809234.
##  7 SEA      2010            0.224             0.0337    4824500
##  8 MIN      2010            0.265             0.0481    4788782
##  9 LAN      2010            0.259             0.0331    4705474.
## 10 HOU      2010            0.251             0.0301    4445455.
## # ... with 290 more rows
```

```
Joined_data_2010_newer %>%
  summarize(mean_salary=mean(salary, na.rm=T), sd_salary=sd(salary, na.rm=T))
```

```
##   mean_salary sd_salary
## 1     3819697   5091340
```

```
Joined_data_2010_newer %>%
  filter(AB.player >100) %>%
  summarize(mean_salary=mean(salary, na.rm=T), sd_salary=sd(salary, na.rm=T))
```

```
##   mean_salary sd_salary
## 1     4583398   5592986
```

```
Joined_data_2010_newer %>%select(Batting_Average.player, salary, HR.player, SB.player, weight, height) %
  drop_na(Batting_Average.player, salary) %>% cor()
```

```
##                         Batting_Average.player     salary    HR.player
## Batting_Average.player             1.00000000 0.06462612  0.404701784
## salary                             0.06462612 1.00000000  0.248953258
## HR.player                          0.40470178 0.24895326  1.000000000
## SB.player                          0.29100681 0.01269773  0.292443521
## weight                            -0.08991088 0.15377685  0.152881029
## height                            -0.20885524 0.13671492 -0.005926308
##                           SB.player      weight       height
## Batting_Average.player  0.29100681 -0.08991088 -0.208855238
## salary                  0.01269773  0.15377685  0.136714919
## HR.player               0.29244352  0.15288103 -0.005926308
## SB.player               1.00000000 -0.25815960 -0.224641926
## weight                 -0.25815960  1.00000000  0.553769069
## height                 -0.22464193  0.55376907  1.000000000
```

```
Joined_data_2010_newer %>% filter(AB.player > 100) %>% summarize(mean(HR.player))
```

```
##   mean(HR.player)
## 1        11.20708
```

```
Joined_data_2010_newer %>% filter(AB.player >100) %>% summarize(max(HR.player))
```

```
##   max(HR.player)
## 1             59
```

```
Joined_data_2010_newer %>% filter(HR.player == 59)
```

```
##    playerID yearID stint teamID lgID G.player AB.player R.player H.player
## 1 stantmi03   2017     1    MIA   NL      159       597      123      168
##   X2B.player X3B.player HR.player RBI SB.player CS.player BB.player SO.player
## 1         32          0        59 132         2         2        85       163
##   IBB HBP.player SH SF.player GIDP salary birthYear birthMonth birthDay
## 1  13          7  0         3   13     NA      1989         11        8
##   birthCountry birthState     birthCity deathYear deathMonth deathDay
## 1          USA         CA Panorama City        NA         NA       NA
##   deathCountry deathState deathCity nameFirst nameLast              nameGiven
## 1                                   Giancarlo  Stanton Giancarlo Cruz-Michael
##   weight height bats throws      debut  finalGame  retroID   bbrefID franchID
## 1    245     78    R      R 2010-06-08 2019-09-29 stanm004 stantmi03      FLA
##   divID Rank G.team Ghome  W  L DivWin WCWin LgWin WSWin R.team AB.team H.team
## 1     E    2    162    78 77 85      N     N     N     N    778    5602   1497
##   X2B.team X3B.team HR.team BB.team SO.team SB.team CS.team HBP.team SF.team
## 1      271       31     194     486    1282      91      30       67      41
##    RA  ER  ERA CG SHO SV IPouts   HA HRA BBA  SOA  E  DP    FP          name
## 1 822 772 4.82  1   7 34   4328 1450 193 627 1202 73 156 0.988 Miami Marlins
##           park attendance BPF PPF teamIDBR teamIDlahman45 teamIDretro
## 1 Marlins Park    1583014  93  93      MIA            FLO         MIA
##   Batting_Average.player Batting_Average.team
## 1               0.281407             0.267226
```

```
Joined_data %>% summarize(max(HR.player))
```

```
##   max(HR.player)
## 1             73
```

```
Joined_data %>% filter(HR.player == 73)
```

```
##    playerID yearID stint teamID lgID G.player AB.player R.player H.player
## 1 bondsba01   2001     1    SFN   NL      153       476      129      156
##   X2B.player X3B.player HR.player RBI SB.player CS.player BB.player SO.player
## 1         32          2        73 137        13         3       177        93
##   IBB HBP.player SH SF.player GIDP   salary birthYear birthMonth birthDay
## 1  35          9  0         2    5 10300000      1964          7       24
##   birthCountry birthState birthCity deathYear deathMonth deathDay deathCountry
## 1          USA         CA Riverside        NA         NA       NA
##   deathState deathCity nameFirst nameLast   nameGiven weight height bats throws
## 1                      Barry    Bonds Barry Lamar    185     73    L      L
##        debut  finalGame  retroID   bbrefID franchID divID Rank G.team Ghome  W
## 1 1986-05-30 2007-09-26 bondb001 bondsba01      SFG     W    2    162    81 90
##    L DivWin WCWin LgWin WSWin R.team AB.team H.team X2B.team X3B.team HR.team
## 1 72      N     N     N     N    799    5612   1493      304       40     235
##   BB.team SO.team SB.team CS.team HBP.team SF.team  RA  ER  ERA CG SHO SV
## 1     625    1090      57      42       50      54 748 680 4.18  3   8 47
##   IPouts   HA HRA BBA  SOA   E  DP    FP                 name         park
## 1   4390 1437 145 579 1080 118 170 0.981 San Francisco Giants PacBell Park
##   attendance BPF PPF teamIDBR teamIDlahman45 teamIDretro
## 1    3311958  93  92      SFG            SFN         SFN
```

```
Joined_data %>% filter(HR.player == 73) %>% summarize(AB.player/HR.player)
```

```
##   AB.player/HR.player
## 1            6.520548
```

```
Joined_data_2010_newer %>% group_by(teamID) %>%
  summarize(mean_attendance=mean(attendance), mean_wins=mean(W)) %>%
  arrange(desc(mean_attendance))
```

```
## # A tibble: 31 x 3
##    teamID mean_attendance mean_wins
##    <chr>            <dbl>     <dbl>
##  1 LAN           3644356.      91.9
##  2 SLN           3386725.      89.9
##  3 NYA           3369561.      91.8
##  4 SFN           3221234.      81.7
##  5 LAA           3063391.      82.0
##  6 CHN           2980256.      81.4
##  7 BOS           2953420.      86.5
##  8 COL           2790222.      74.6
##  9 MIL           2727429.      82.7
## 10 PHI           2656654.      78.0
## # ... with 21 more rows
```

```
Joined_data_2010_newer_longer <- Joined_data_2010_newer %>%
  pivot_longer(c("X2B.player", "X3B.player", "HR.player"), names_to="Extra_Base_Hit_Type",
  values_to="Extra_Base_Hit_Number")

Joined_data_2010_newer_longer %>% filter(AB.player >100)%>%
  group_by(teamID, yearID, Extra_Base_Hit_Type) %>%
  summarize(mean(Extra_Base_Hit_Number))
```

```
## # A tibble: 900 x 4
## # Groups:   teamID, yearID [300]
##    teamID yearID Extra_Base_Hit_Type `mean(Extra_Base_Hit_Number)`
##    <chr>   <int> <chr>                                       <dbl>
##  1 ARI      2010 HR.player                                   13.2
##  2 ARI      2010 X2B.player                                  20.5
##  3 ARI      2010 X3B.player                                   2.62
##  4 ARI      2011 HR.player                                   10.4
##  5 ARI      2011 X2B.player                                  18.7
##  6 ARI      2011 X3B.player                                   2.57
##  7 ARI      2012 HR.player                                   11.6
##  8 ARI      2012 X2B.player                                  20.8
##  9 ARI      2012 X3B.player                                   2.15
## 10 ARI      2013 HR.player                                    8.86
## # ... with 890 more rows
```

```
Joined_data_2010_newer %>% mutate(player_age=yearID-birthYear) %>%
  group_by(teamID, yearID, birthCountry) %>%
  summarize(mean_age=mean(player_age), n(), min(player_age), max(player_age))
```

```
## # A tibble: 2,027 x 7
## # Groups:   teamID, yearID [300]
##    teamID yearID birthCountry mean_age `n()` `min(player_age)` `max(player_age)`
##    <chr>   <int> <fct>           <dbl> <int>              <int>              <int>
##  1 ARI      2010 D.R.             25.7     6                 24                 27
##  2 ARI      2010 Germany          27       1                 27                 27
##  3 ARI      2010 Mexico           35       1                 35                 35
##  4 ARI      2010 P.R.             33       1                 33                 33
##  5 ARI      2010 USA              28.7    36                 23                 38
##  6 ARI      2010 Venezuela        25.7     3                 23                 27
##  7 ARI      2011 Cuba             32       2                 28                 36
##  8 ARI      2011 D.R.             29.3     3                 28                 31
##  9 ARI      2011 USA              29.0    40                 23                 38
## 10 ARI      2011 Venezuela        31.3     6                 24                 40
## # ... with 2,017 more rows
```

```
Joined_data_2010_newer %>% filter(AB.player > 100) %>%
  group_by(birthCountry) %>% summarize(number_of_players=n()) %>%
  mutate(proportion_of_players=number_of_players/sum(number_of_players))
```

```
## # A tibble: 21 x 3
##    birthCountry number_of_players proportion_of_players
##    <fct>                   <int>                  <dbl>
##  1 Aruba                       6                0.00135
##  2 Australia                   1                0.000225
##  3 Brazil                      9                0.00203
##  4 CAN                        55                0.0124
##  5 Colombia                   18                0.00406
##  6 Cuba                      137                0.0309
##  7 Curacao                    29                0.00653
##  8 D.R.                      458                0.103
##  9 Germany                    14                0.00315
## 10 Honduras                    1                0.000225
## # ... with 11 more rows
```

Before exploring the data, I first filtered the data to only be from 2010 onwards in order to have a better understanding of today's game (salaries were much different before 2010, homeruns in the steroid era were common, etc.). Also, most of the statistics were calculated after first filtering out players with less than 100 at bats (I did this because players with few at bats were generally rookie players who were called up to the major leagues for only a few games and would have stats that would be skewed). The first statistic I calculated was the yearly mean and median player salary for each team. Unsurprisingly, the Yankees tended to spend the most money in baseball. Also, the mean tended to be much higher than the median, which indicated that there was a large skew (most likely due to a few players being paid lots of money skewing right). I then attempted to determine if mean salary was related to mean batting average, but it appeared that there was not a strong relationship between the two. I then calculated the mean and standard deviation of salary with and without the players with less than 100 at bats factored in. When players with less than 100 at bats were included, the mean salary decreased, which was in line with my prediction that players with less than 100 at bats would be low payed rookies or recently called up minor leaguers. I then calculated a correlation matrix to figure out the relationships between different numeric variables. Interestingly, it appears that batting average and salary have almost no relationship. Although number of homeruns and salary have a weak positive correlation, I thought that the relationship would be stronger. I then calculated the mean number of homeruns per player, which made me interested to figure out what was the most homeruns hit in a single season by a player. I discovered that in the previous 10 seasons, Giancarlo Stanton holds the record

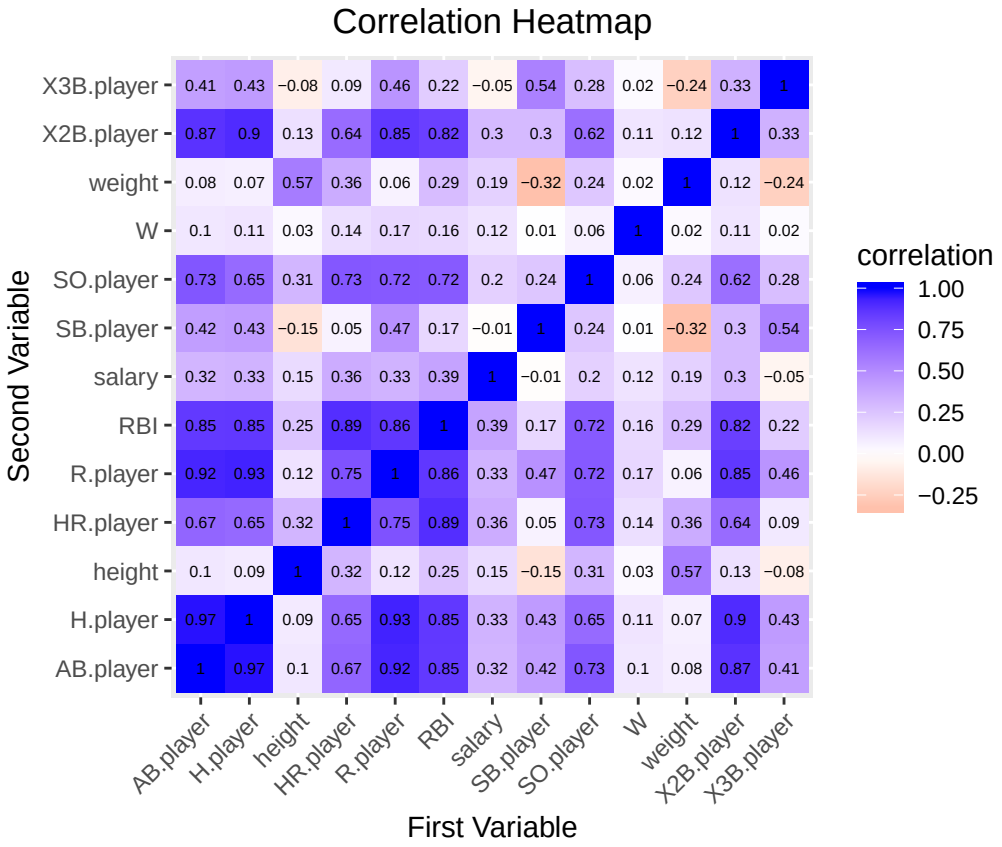
with 59 homeruns hit in 2017. I then figured out who hit the most homeruns in a single season ever. That title is held by Barry Bonds, who hit 73 homeruns in 2001. I calculated that he hit a homerun every 6.52 at bats.

Next, I wanted to visualize how teams hit extra base hits (doubles, triples, homeruns). I did this by using the pivot_longer function. I took the "X2B.player", "X3B.player", and "HR.player" variables and pivoted them longer to form the categories of the Extra_Base_Hit_Type variable. THe previous values were pivoted into the "Extra_Base_Hit_Number"" variable. I was then able to group by team, year, and type of extra base hit, then easily calculate the mean of each type of hit. Lastly, I wanted to understand the age and nationality distribution of the major leagues. I grouped by team, year, and birth country of players and then found the mean age, minimum age, maximum age, and the number of players from each country. I was interested to see several countries that are not traditionally associated with baseball (Germany for example) have players in the major leagues. I then calculated the proportions of players across the previous 10 years based on their nationalities. I found that the vast majority (nearly 70%) of players are from the United States.

```
Joined_data_cor <- Joined_data_2010_newer %>%
  filter(AB.player > 100) %>%
  select(7:14, salary, height, weight, SO.player, W) %>% drop_na(salary)

tidycor<-cor(Joined_data_cor)%>%as.data.frame%>%
  rownames_to_column%>%
  pivot_longer(-1,names_to="name",values_to="correlation")

tidycor%>%ggplot(aes(rowname,name,fill=correlation))+
  geom_tile()+
  scale_fill_gradient2(low="red",mid="white",high="blue")+
  geom_text(aes(label=round(correlation,2)),color = "black", size = 2)+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  coord_fixed()+
  xlab("First Variable") + ylab("Second Variable")+
  ggtitle("Correlation Heatmap")+
  theme(plot.title = element_text(hjust=0.5))
```

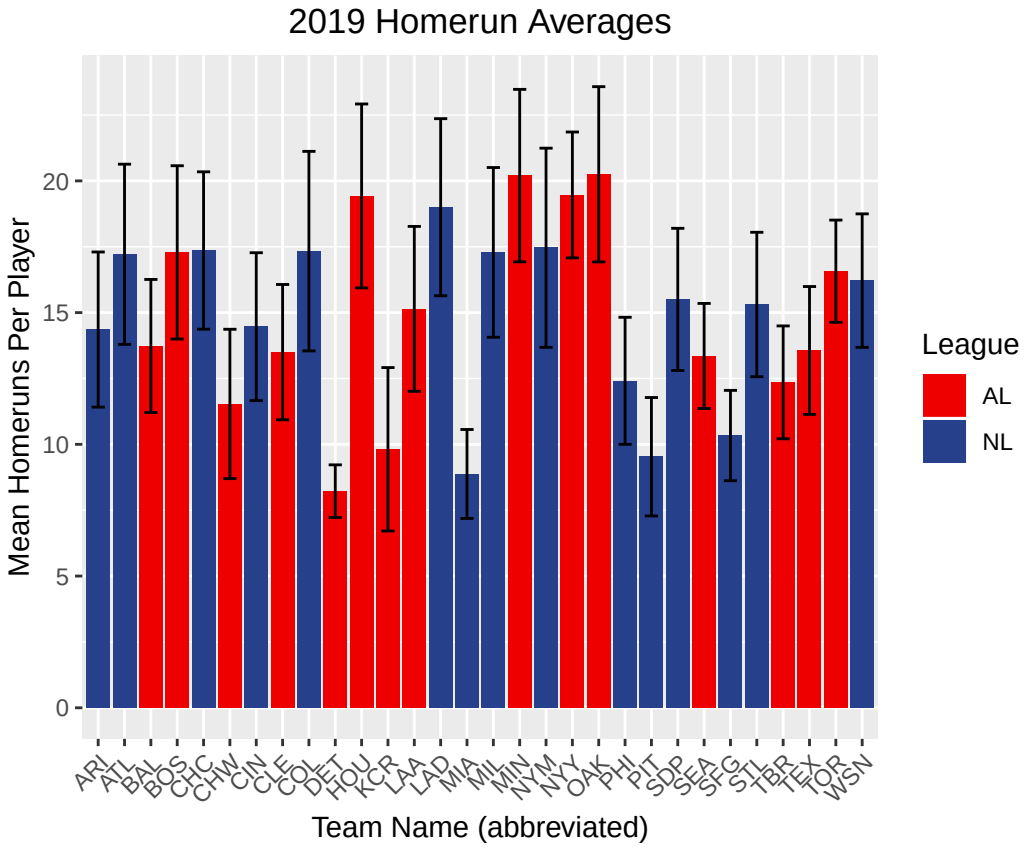


Because there are so many numeric variables in my dataset, I decided to narrow the correlation heatmap down to 13 varaibles. Some of the obvious correlations that result are that height and weight have a moderate positive correlation of 0.57, hits and at bats have a very strong positive correlation of almost 1 (because the more at bats you have, the more hits you are likely to get), and that hits are strongly positively correlated with RBI (because the more hits you get, the more runners you will drive in). I found it interesting that the most negative correlation on the chart is between weight and number of stolen bases. It makes sense because larger players generally tend to move more slowly and would thus be caught stealing nearly every time. While I thought that salary would be most strongly positively correlated to number of homeruns (due to fans of this era of baseball loving homeruns), it actually turns out that the number of runs batted in by a player is most strongly correlated with salary (homeruns came in second). Lastly, I found that the number of stolen bases and the number of triples hit by a player are moderately positively correlated. This makes sense because only the most fast players are able to hit triples and therefore would have more stolen bases simply due to being very fast.

```
Joined_data_2010_newer_filtered_AB <- Joined_data_2010_newer %>%
  filter(AB.player >100, yearID ==2019)

ggplot(Joined_data_2010_newer_filtered_AB, aes(x = teamIDBR, fill = lgID))+
  geom_bar(aes(y=HR.player), stat="summary", fun.y="mean")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  coord_fixed()+
  geom_errorbar(fun.data='mean_se',stat="summary", aes(y=HR.player, width = 0.5))+
  xlab("Team Name (abbreviated)") + ylab("Mean Homeruns Per Player")+
  ggtitle("2019 Homerun Averages")+
  theme(plot.title = element_text(hjust=0.5))+
```

```
  labs(fill = "League")+
  scale_fill_manual(values=c('red2','royalblue4'))
```

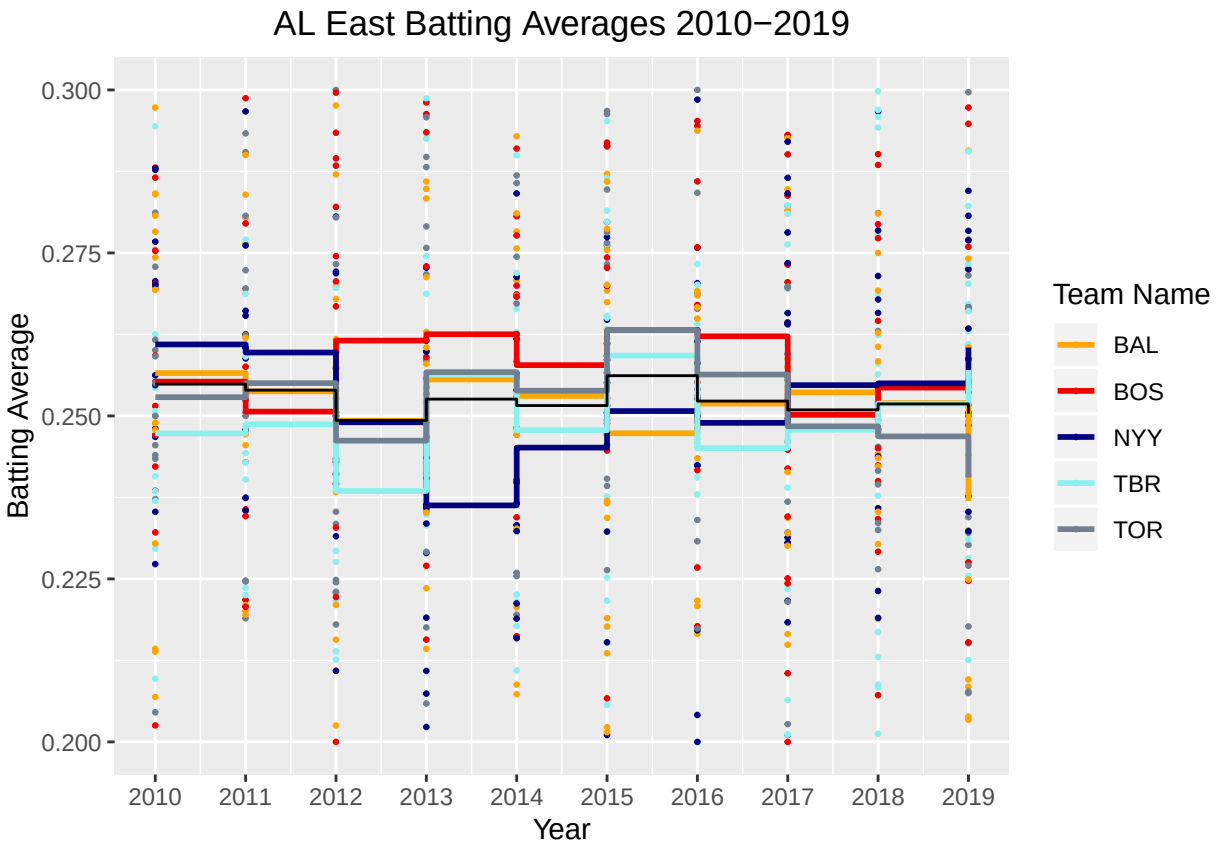


I decided to see how many homeruns the average player on each team hit in the 2019 season. While the Oakland A's did not lead the league in total homeruns (that distinction belongs to the Minnesota Twins), they did have the most homeruns per player. This is likely due to not having a few outstanding sluggers and instead having players throughout the lineup who can hit homeruns. It was interesting to note that the top 5 teams in mean homeruns per player all made it to the playoffs. This means that perhaps homeruns are the way for teams to win games. It is interesting to note that the eventual world series champion, the Washington Nationals, came in 11th place in terms of mean homeruns per player. This may indicate that while homeruns are an effective way to win in the regular season, relying on homeruns when the other team's pitching is good (which is the case in the playoffs) may be counterproductive. I wanted to see how the mean homeruns varied based on the league the team is in. In the National League, pitchers (who are not great hitter) are required to bat, which could result in less homeruns per player. This appears to be a factor since 4 of the top 5 teams in the major leagues are from the American League (where pitchers are replaced with a designated hitter who is usually a power hitter).

```
Joined_data_2010_newer_filtered <- Joined_data_2010_newer %>% filter (AB.player >100, lgID =='AL', divI

ggplot(data=Joined_data_2010_newer_filtered, aes(yearID, Batting_Average.player))+
  geom_point(aes(color = teamIDBR), size=.5)+
  geom_step(aes(color = teamIDBR),size=1, stat = "summary", fun.y = "mean")+
  geom_step( stat = 'summary', fun.y = 'mean')+
  labs(y="Batting Average", x="Year", color="Team Name")+
  ggtitle("AL East Batting Averages 2010-2019")+
```

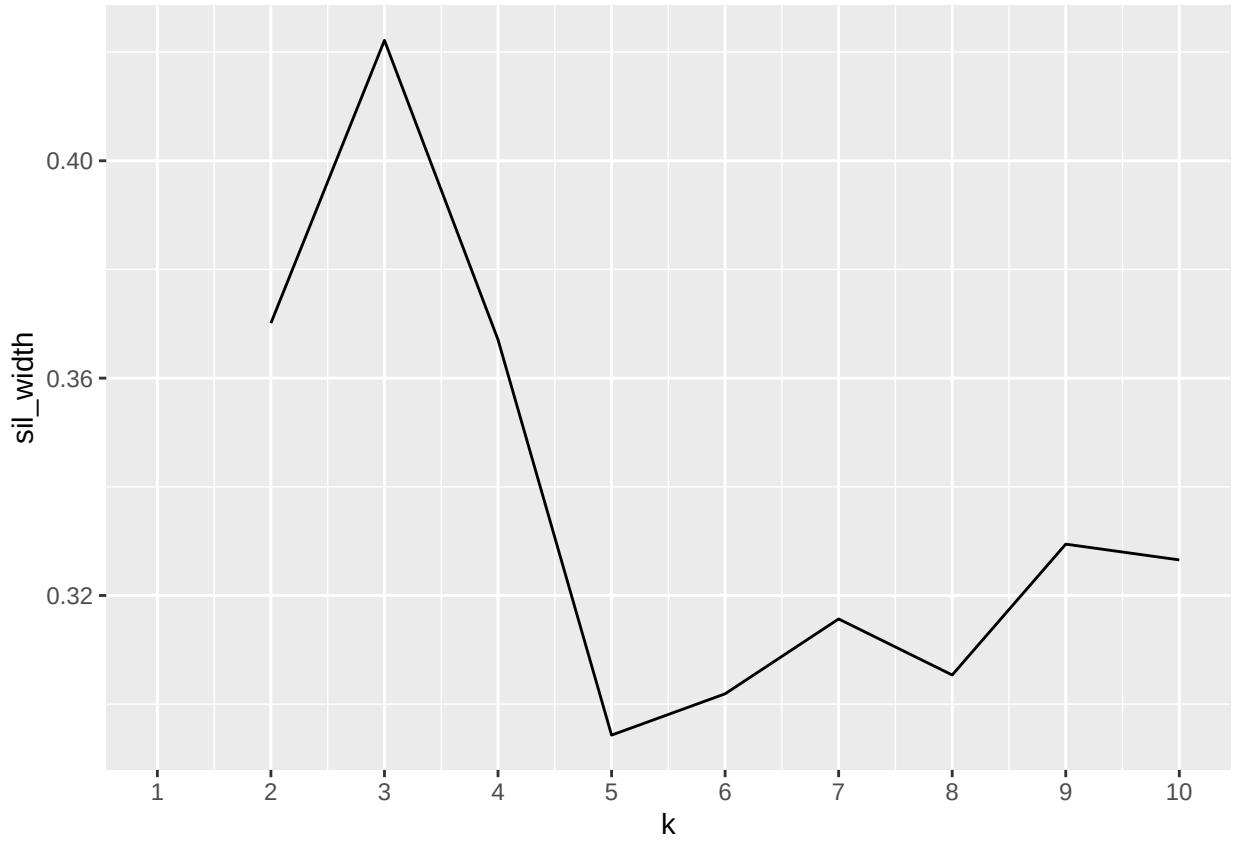
```
  scale_color_manual(values=c('orange','red2','navyblue','darkslategray2','slategray'))+
  scale_x_continuous(breaks=seq(2010,2019,1))+
  scale_y_continuous(breaks=seq(.200,.300,.010))+
  ylim(.200,.300)+
  theme(plot.title = element_text(hjust=0.5))
```



In this plot, I wanted to see how the batting averages of teams in the American League East differed over the 2010-2019 seasons. Interestingly, the team with the highest batting average in the division only won the division 60% of the time (the 2010 Tampa Bay Rays had the lowest batting average and won the division). It makes sense that a rebuilding Yankees team between 2013 and 2015 would have the lowest batting average in the division. The Red Sox, with 4 divison wins in the previous 10 season (the most of any AL East team), tend to trend toward the top of the division in terms of batting average. Although one would think the team with the best batting average would be the team that wins the division every year, this does not appear to be the best predictor of success based on the data.

```
library(cluster)
Joined_data_2010_newer_filtered <- Joined_data_2010_newer %>%
  filter (AB.player >100) %>% drop_na(salary)
pam_dat <-Joined_data_2010_newer_filtered %>%
  select(SB.player, HR.player, salary) %>% scale
sil_width<-vector()
for(i in 2:10){
  pam_fit <- pam(pam_dat, k = i)
  sil_width[i] <- pam_fit$silinfo$avg.width
```

```
}
ggplot()+geom_line(aes(x=1:10,y=sil_width))+scale_x_continuous(name="k",breaks=1:10)
```

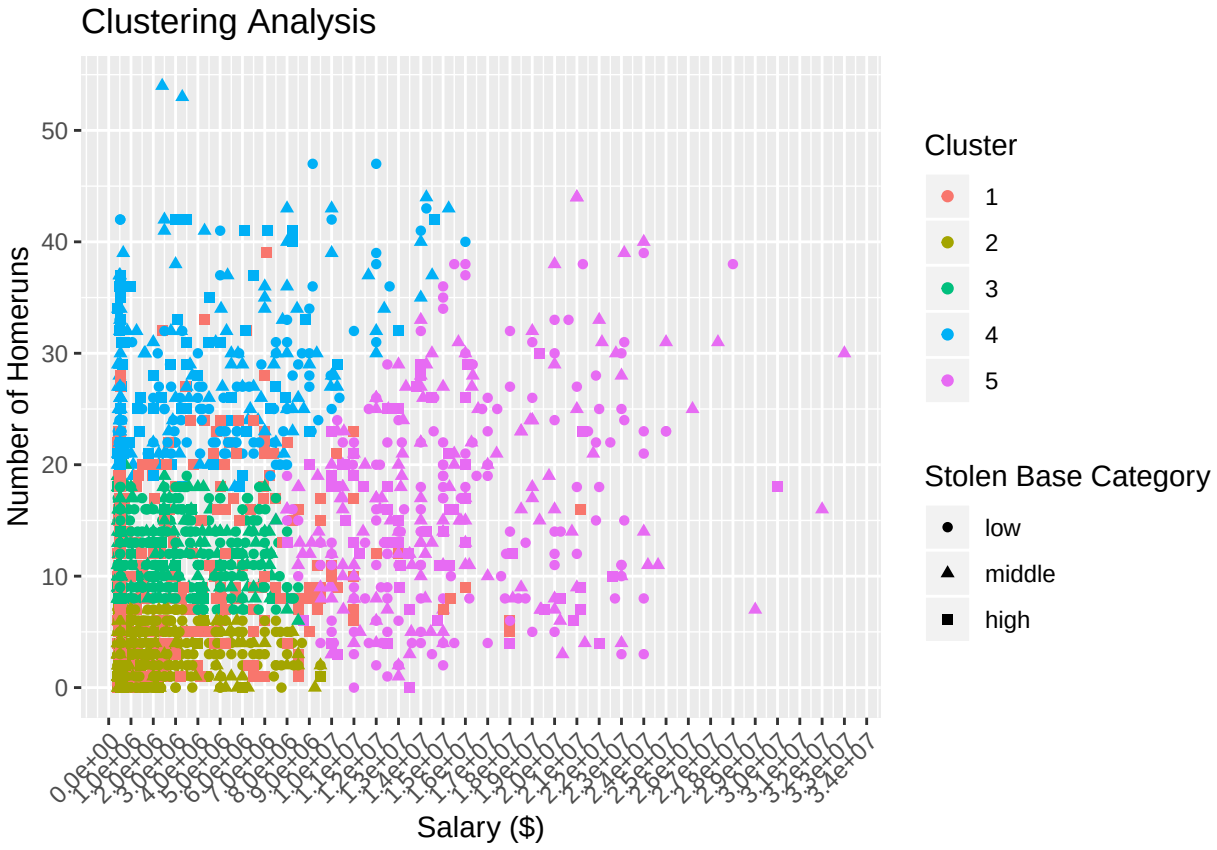



```
pam2 <- Joined_data_2010_newer_filtered %>%
  select(SB.player, HR.player, salary) %>%
  scale %>% pam(5)
head(pam2)
```

```
## $medoids
##       SB.player   HR.player     salary
## 1497  1.9032506 -0.36622197 -0.3099236
## 360  -0.5016009 -0.89081361 -0.5959962
## 1523 -0.5016009  0.05345135 -0.4618996
## 2157 -0.1736666  1.62722628 -0.1937066
## 3107 -0.2829781  0.36820634  1.6836448
##
## $id.med
## [1] 1195  290 1212 1730 2484
##
## $clustering
##   1   2   4   5   7   8   9  10  11  12  13  15  16  17  18  19  20  21  23  24
##   1   2   2   1   2   2   1   1   2   2   3   3   2   3   4   5   3   2   4   5
##  25  28  30  31  33  34  35  36  38  39  42  43  44  45  47  48  49  50  52  53
##   5   3   3   2   2   2   2   2   2   1   1   5   2   3   4   3   4   3   2   3
```

```
##  54  55  56  57  58  59  62  64  65  66  67  69  70  72  73  75  76  77  78  79
##   3   2   2   2   5   2   3   2   4   3   2   2   2   2   3   3   2   5   1   2
##  80  81  82  83  84  85  86  87  88  89  91  92  95  97  99 100 102 103 105 106
##   2   2   4   3   2   2   2   1   1   2   4   5   5   2   1   2   1   2   2   2
## 108 109 110 112 113 114 116 117 118 119 122 124 126 127 128 129 130 131 133 134
##   3   5   3   4   2   3   2   4   2   2   4   2   4   1   2   2   2   3   2   5
##  [ reached getOption("max.print") -- omitted 2385 entries ]
##
## $objective
##     build      swap
## 0.8741687 0.8176341
##
## $isolation
##  1  2  3  4  5
## no no no no no
## Levels: no L L*
##
## $clusinfo
##      size max_diss   av_diss diameter separation
## [1,]  317 4.921706 1.3523484 6.172431  0.1093128
## [2,]  850 1.785415 0.4723604 2.228954  0.1049183
## [3,]  600 1.434919 0.5976447 2.131132  0.1049183
## [4,]  326 3.168994 1.0612630 4.069141  0.1049291
## [5,]  392 3.743947 1.2680135 5.089628  0.1049183
```

```
Joined_data_2010_newer_filtered <- Joined_data_2010_newer_filtered %>%
  mutate(SB_category=cut(SB.player, breaks=c(-Inf,2,10,Inf),labels=c('low','middle','high')))
final<-Joined_data_2010_newer_filtered%>%mutate(cluster=as.factor(pam2$clustering))
ggplot(final, aes(x=salary,y=HR.player, color=cluster, shape=SB_category))+geom_point()+
  scale_x_continuous(breaks=seq(0,35000000,1000000, labels=scales::comma))+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  scale_y_continuous(breaks=seq(0,70,10))+
  ggtitle("Clustering Analysis")+
  labs(x="Salary ($)", y="Number of Homeruns", shape="Stolen Base Category", color="Cluster")
```



Because my data had so many numeric variables, I chose to cluster based on three that I thought would be interesting: Number of homeruns, salary, and number of stolen bases. To begin, I needed to determine the number of clusters that would best suit the data. I filtered by players with more than 100 at bats, dropped the NA's that were present in salary, and scaled to generate my pam data. I then used that data in a for loop to determine the silhouette width within clusters for clusters between 2 and 10. I found that the elbow was at 5 clusters, which means that using 5 clusters would produce the tightest results. I then ran pam with 5 clusters and and plotted the clusters on a ggplot. The ggplot, which had salary on the X-axis, homeruns on the Y-axis, color by cluster, and shape by stolen base category (I divided the number of stolen bases of each player into low, medium, and high categories to better visualize it. Low was 2 or less stolen bases, medium was 2 to 10 stolen bases, and high was anything more than 10 stolen bases). The results indicated that the five clusters consisted of players with a very high salary, players with lots of stolen bases, players with lots of home runs, players who are very low on all three values, and players who make a small salary, hit a moderate number of homeruns, and steal a few bases. Overall, the data indicates that players being paid a lot of money tend to steal few bases and hit a good number of homeruns. Stolen bases are not compensated very well in terms of salary. Interestingly, there does not appear to be an incredibly strong connection between players hitting lots of homeruns getting paid well. This could be because a power hitting rookie can hit many homeruns but still be paid a low amount of money. Overall, it appears that a number of other factors (fielding, situational hitting, jersey sales, etc.) play a role in determining the salary of a player.