Lahman MLB Analysis

Vishnu Rangiah

November 10, 2020

Introduction

In this report we will be exploring many aspects of baseball data through the use of R, SQL and “The Lahman Baseball Database” provided as a SQLite database. The data base contains 24 tables which we will interact with using the RSQLite and DBI packages for R. “This database covers Major League Baseball information from 1871 through 2013 about pitching, hitting, and fielding statistics.”

We will structure our report through the use of questions which reveal interesting aspects about the provided baseball data such as “finding the relationship between games won in a season and winning the World Series.”

Loading Dataset

```
##  [1] "AllstarFull"         "Appearances"         "AwardsManagers"     
##  [4] "AwardsPlayers"       "AwardsShareManagers" "AwardsSharePlayers" 
##  [7] "Batting"             "BattingPost"         "Fielding"           
## [10] "FieldingOF"          "FieldingPost"        "HallOfFame"         
## [13] "Managers"            "ManagersHalf"        "Master"             
## [16] "Pitching"            "PitchingPost"        "Salaries"           
## [19] "Schools"             "SchoolsPlayers"      "SeriesPost"         
## [22] "Teams"               "TeamsFranchises"     "TeamsHalf"          
## [25] "temp"
```

To begin our report we will connect to the “lahman2013.sqlite” database and observe the names of all 24 Tables.

1. What years does the data cover? Are there data available for each of these years?

The years that the data covers represents the range of time that is included about the MLB in the database. Taking the MAX and MIN of the year ID field in the batting table reveals the MAX and MIN of the years the data covers for batting statistics about MLB players. By comparing the MAX and MIN of the yearID field from Batting, Pitching Fielding, and Teams we find that the data covers MLB information from **1871 to 2013**.

The SQL used in the RSQLite function call to find the MAX and MIN from the various tables is given here.

```
# What years does the data cover? Are there data available for each of these
years?

dbGetQuery(db, "SELECT MIN(yearID) AS minYear, MAX(yearID) AS maxYear FROM
Teams")
dbGetQuery(db, "SELECT MIN(yearID) AS minYear, MAX(yearID) AS maxYear FROM
Batting")
dbGetQuery(db, "SELECT MIN(yearID) AS minYear, MAX(yearID) AS maxYear FROM
Pitching")
dbGetQuery(db, "SELECT MIN(yearID) AS minYear, MAX(yearID) AS maxYear FROM
Fielding")
```

To find if each of the years has data we will try to see if there are any years in the yearID field which are not present in the 1871 to 2013 range.

More particularly we will *SELECT* the *DISTINCT* yearID values then *ORDER BY* yearID. Then using the "==" operator we will compare the distinct values to a R vector which has a sequence of integers from 1871 to 2013. We will know about the availability of data by taking the sum of the logical comparison.

```
years_info = dbGetQuery(db, "SELECT DISTINCT yearID
                              FROM Teams
                              ORDER BY yearID")

years_bat =  dbGetQuery(db, "SELECT DISTINCT yearID
                              FROM Batting
                              ORDER BY yearID")

years = seq(1871,2013) # sequence of intergers in a vector from 1871 to 2013

length(years)
sum(years_info$yearID ==  years)
sum(years_bat$yearID ==  years)
```

We find that from the Teams and Batting tables there is data for each of the years from 1871 to 2013. We verify this by comparing the sequence to the distinct yearID values of each table and find that we get 143 distinct year values which means there is data for all the years.

2. How many (unique) people are included in the database? How many are players, managers, etc?

The amount of unique people included in the database can be found by counting all the distinct values of the playerID field in the MASTER TABLE. Being that the MASTER TABLE is one of the four "Main" tables this indicates that the MASTER table has the information of all the playerIDs involved in all the other Tables.

The R and SQL for this is shown here. We COUNT the DISTINCT playerIDs in the playerID field of the Master table which gives us the same result as counting the playerIDs.

```
#2: How many (unique) people are included in the database? How many are
players, managers, etc?

```

```
dbGetQuery(db, "SELECT COUNT(DISTINCT playerID) AS numP
                 FROM Master")

##    numP
## 1 18354
```

3. How many players became managers?

This questions is asking out of all the players included in our Master table are there any that are also in the Managers table indicating that a player became a manager. Since the players are represented through their playerIDs by using an INNER JOIN on the tables we find only the playerIDs that are present in both tables. Thus counting the DISTINCT playerIDs from this result gave us our answer of the total number of players that turned into a manager which is 679.

The R and SQL to do this is shown here.

```
#QUESTION 3: How many players became managers?

dbGetQuery(db, "SELECT COUNT(DISTINCT mm.playerID) AS numPM
                 FROM Master AS mm
                 INNER JOIN Managers AS m
                    ON mm.playerID = m.playerID")

##   numPM
## 1   679
```

4. How many players are there in each year, from 2000 to 2013? Do all teams have the same number of players?

We find that the position and other information about each player for each year are in the Appearances Table. By grouping by yearID then counting the number of yearIDS then only looking at yearIDs after 1999 in the Appearances table we find the amount of players by year. We can verify if this result is correct by doing the same operation on the Batting table which gives batting information for each player for each year they played.

```
#QUESTION 4: How many players are there in each year, from 2000 to 2013? Do
all teams have the same number of players?

dbGetQuery(db, "SELECT yearID, COUNT(yearID) AS Num
                 FROM Appearances
                 GROUP BY yearID
                 HAVING yearID > 1999")


dbGetQuery(db, "SELECT yearID, COUNT(yearID) AS Num
                 FROM Batting
                 GROUP BY yearID
                 HAVING yearID > 1999")
```

Inorder to find if all the teams have the same number of people we create a data frame that shows the amount of people per team per year. We find that these values range from 39 to 59 for the number of people per teams over all years. This shows that teams had did not all have the same number of people.

```
plyr_nums = dbGetQuery(db, "SELECT teamID, yearID, COUNT(teamID) AS Num FROM 
Appearances WHERE yearID > 1999 GROUP BY teamID, yearID" )
head(plyr_nums)

##   teamID yearID Num
## 1    ANA   2000  45
## 2    ARI   2000  41
## 3    ATL   2000  47
## 4    BAL   2000  50
## 5    BOS   2000  52
## 6    CHA   2000  42

c(max(plyr_nums$Num), min(plyr_nums$Num))

## [1] 59 34
```

5. What team won the World Series in 2010? Include the name of the team, the league and division.

To find the 2010 World Series winner we look at the Teams table since in this table contains yearly baseball information for each team. In order to find the 2010 winner we find the team that has a yearID value equal to 2010 and also has a 'Y' in their WSWin field which indicates if they won the World Series.

```
#QUESTION 5: What team won theWorld Series in 2010? Include the name of the 
team, the league and division.

dbGetQuery(db, "SELECT yearID, name, lgID, divID, WSWin
                 FROM Teams
                 WHERE yearID = 2010
                    AND WSWin = 'Y' ")

##   yearID                 name lgID divID WSWin
## 1   2010 San Francisco Giants   NL     W     Y
```

We observe that the San Francisco Giants won the World Series in 2010.

6. What team lost the World Series each year? Again, include the name of the team, league, and division.

To find which team lost the world series every year we will use the Teams table again and make use of the "WSWin" field or World Series Win field and the "LgWin" field. We know that only the league winners of each year are able to participate in the World Series thus we want for each year the team that won their league which is indicated by LgWin = 'Y' but lost the World Series which is indicated by WSWin = 'N'.

The R and SQL code for this shown below.

```
#QUESTION 6: What team lost the World Series each year? Again, include the
name of the team, league, and division

teamWSLs = dbGetQuery(db, "SELECT yearID, name, lgID, divID
                 FROM Teams
                 WHERE LgWin = 'Y' AND WSWin = 'N'")
head(teamWSLs)

##   yearID                    name lgID divID
## 1   1884  New York Metropolitans   AA  <NA>
## 2   1885         St. Louis Browns   AA  <NA>
## 3   1885 Chicago White Stockings   NL  <NA>
## 4   1886 Chicago White Stockings   NL  <NA>
## 5   1887         St. Louis Browns   AA  <NA>
## 6   1888         St. Louis Browns   AA  <NA>

#Find what years are not accounted for in the winners
wsL = dbGetQuery(db, "SELECT yearID, name
                        FROM Teams WHERE LgWin = 'Y' AND WSWin = 'N'" )

wsW = dbGetQuery(db, "SELECT yearID, name
                        FROM Teams WHERE WSWin = 'Y' " )

setdiff(wsL$yearID,wsW$yearID)

## [1] 1885 1890
```

We have found the losers for every World Series as well as their leagueID and divisionID. What is interesting is that there are more teams who won their league and lost the World Series (118 teams) than won the World Series (114 teams). By looking futher into the years contained in the list of winning teams of the World Series versus the list of losing teams we find that in 1885 the World Series resulted in a tie. The year 1890 is also not present in the list of winning teams.

7. Compute the table of World Series winners for all years, again with the name of the team, league and division.

Using similar techniques from Question 6 we can find the name, team, league and division of the World Series winner for each year in the Teams table. We will select the yearID, name, league ID, and division ID from Teams table then specify that we only want the tuples which have a 'Y' value in the WSWin field.

The R and SQL for finding the World Series winners for each year is shown here.

```
#Question 7: Compute the table of World Series winners for all years, again
with the name of the team, league and division.

teamWSWs = dbGetQuery(db, "SELECT yearID, name, lgID, divID
                            FROM Teams
                            WHERE WSWin = 'Y'")
head(teamWSWs)
```

```
##   yearID               name lgID divID
## 1   1884   Providence Grays   NL  <NA>
## 2   1886   St. Louis Browns   AA  <NA>
## 3   1887 Detroit Wolverines   NL  <NA>
## 4   1888    New York Giants   NL  <NA>
## 5   1889    New York Giants   NL  <NA>
## 6   1903   Boston Americans   AL  <NA>
```

8. Compute the table that has both the winner and runner-up for the World Series in each tuple/row for all years, again with the name of the team, league and division, and also the number games the losing team won in the series.

All of the relavant information to solve this question can be found in the SeriesPost table as shown below. We only require the teams that were involved in a world series so we will specify that the round feild values must be eqaul to 'WS'.

```
#QUESTION 8: Compute the table that has both the winner and runner-up for the
World Series in each tuple/row for all years, again with the name of the
team, league and division, and also the number games the losing team won in
the series.

head(dbGetQuery(db ,  "SELECT * FROM SeriesPost WHERE round = 'WS'"))
```

```
##   yearID round teamIDwinner lgIDwinner teamIDloser lgIDloser wins losses
ties
## 1   1884    WS          PRO         NL         NYP        AA    3      0
0
## 2   1885    WS          CHC         NL         STL        AA    3      3
1
## 3   1886    WS          STL         AA         CHC        NL    4      2
0
## 4   1887    WS          DTN         NL         STL        AA   10      5
0
## 5   1888    WS          NYG         NL         STL        AA    6      4
0
## 6   1889    WS          NYG         NL         BRO        AA    6      3
0
```

In order to find the name, league, and division information for each the winning and losing teams we will use a double LEFT JOIN to join the Series post Table with the Teams table twice. We will left join the Teams table once by connecting the yearIDs of both table and teamID field of Teams to teamIDwinner of SeriesPost. Then we will left join again for the losers which connect yearIDs and teamIDloser to teamID. Finally the number of games the losing team won is same as the losses for the winning team in the Series Post table.

```
win_run = (dbGetQuery(db ,  "SELECT DISTINCT p.yearID, p.teamIDwinner,
t.name, p.lgIDwinner, t.divID,
                                  p.teamIDloser, t2.name, p.lgIDloser,
t2.divID, p.losses
                   FROM SeriesPost AS p
                   LEFT JOIN Teams as t ON (p.teamIDwinner = t.teamID AND
p.yearID = t.yearID)
                   LEFT JOIN Teams as t2 ON (p.teamIDloser = t2.teamID AND
p.yearID = t2.yearID )
                   WHERE p.round = 'WS'
                   ORDER BY p.yearID DESC"))
head(win_run)
```

```
##   yearID teamIDwinner                  name lgIDwinner divID teamIDloser
## 1   2013          BOS        Boston Red Sox         AL     E         SLN
## 2   2012          SFN  San Francisco Giants         NL     W         DET
## 3   2011          SLN   St. Louis Cardinals         NL     C         TEX
## 4   2010          SFN  San Francisco Giants         NL     W         TEX
## 5   2009          NYA      New York Yankees         AL     E         PHI
## 6   2008          PHI Philadelphia Phillies         NL     E         TBA
##                    name lgIDloser divID losses
## 1   St. Louis Cardinals        NL     C      3
## 2        Detroit Tigers        AL     C      0
## 3         Texas Rangers        AL     W      3
## 4         Texas Rangers        AL     W      1
## 5 Philadelphia Phillies        NL     E      2
## 6        Tampa Bay Rays        AL     E      1
```

```
dbGetQuery(db ,  "SELECT COUNT(*) FROM SeriesPost WHERE round = 'WS'")
```

```
##   COUNT(*)
## 1      116
```

To avoid duplicate yearIDs the occur after the left join we only select the DISTINCT yearIDs. We find that we have the correct result since we have 116 distinct yearIDs.

9. Do you see a relationship between the number of games won in a season and winning the World Series?

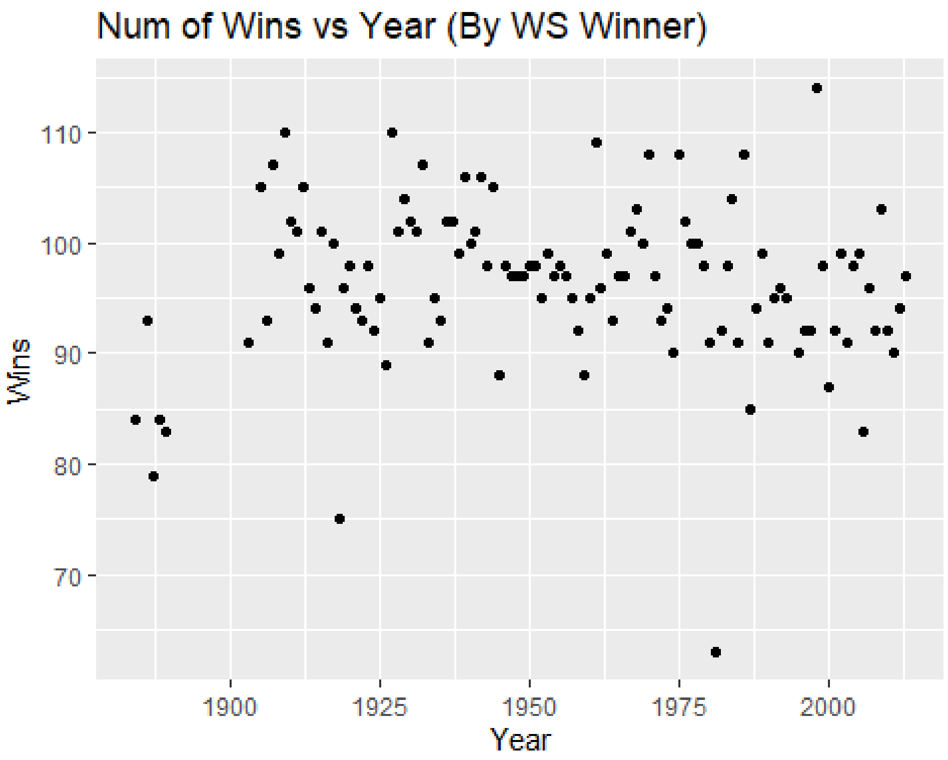
To find the number of games won in a season we look to the “W” field in Teams which shows the number of wins a team had in a particular year/season. Then by only observing the teams that have a ‘Y’ in their “WSWin” field we create a data frame that has the world series winnners for each season and the number of wins that the team had.

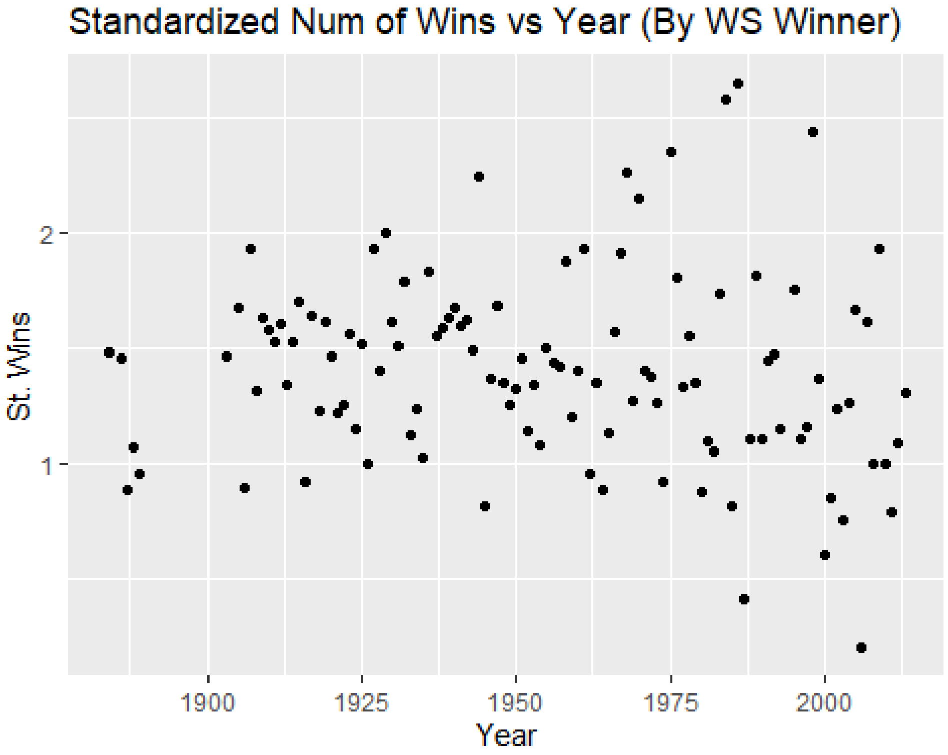
While oberving the Teams table we find that the teams ended the season playing a different amount of games. In order to get a better understand between the relationship of games won and winning the World Series we standardize the amount of wins by each season. Using a subquery we are able to find the avg and stdev of the number of wins of all the teams in each season. Then taking difference of the amount of season wins by the world series winner and the average of all the other teams in that season and dividing this number by the stdev of number of wins all the teams by season we have the standardized scores.

The SQL and R code is shown here.

```
#QUESTION 9. Do you see a relationship between the number of games won in a
season and winning the World Series?

gWdf = dbGetQuery(db, "SELECT t.yearID, t.name, t.WSWin, t.W, t.G, a.avg,
((t.W - a.avg)/a.std) as fW, (t.W / t.G) as nW
                        FROM Teams as t
                        JOIN (SELECT t2.yearID, AVG(t2.W) AS avg, STDEV(t2.W)
AS std
                                FROM Teams AS t2
                                GROUP BY t2.yearID) AS a
                        ON t.yearID = a.yearID
                        WHERE WSWin = 'Y'
                        ORDER BY t.yearID DESC ") #contains subquery
head(gWdf)
```

```
##   yearID                  name WSWin   W   G      avg        fW nW
## 1   2013        Boston Red Sox     Y  97 162 81.03333 1.3029293  0
## 2   2012  San Francisco Giants     Y  94 162 81.00000 1.0893497  0
## 3   2011   St. Louis Cardinals     Y  90 162 80.96667 0.7913352  0
## 4   2010  San Francisco Giants     Y  92 162 81.00000 0.9995728  0
## 5   2009      New York Yankees     Y 103 162 81.00000 1.9239222  0
## 6   2008 Philadelphia Phillies     Y  92 162 80.93333 0.9982863  0
```





The first graph shows the number of wins the World series winner had per year. The second graph shows the standardized score of wins the World Series winning team has per year. Both graphs show variablity in the amount of World Series wins, and since there is no evident pattern it may be possible that the number of season wins does not have a strong relationship with World Series wins. However, most teams that won the world series had over 85 wins.

10. In 2003, what were the three highest salaries? (We refer here to unique salaries, i.e., there may be several players getting the exact same amount.) Find the players who got any of these 3 salaries with all of their details?

To find information about player salaries per year we can look to the Salaries table. Here we are able to ORDER By salary and find the highest salaries given in a particular year. We can then join the Master table to gain information about the player name and then join the Appearances table to gain information about the player's position and lastly join the Teams table to find information about the team name.

```
#QUESTION 10: In 2003, what were the three highest salaries? (We refer here
to unique salaries, i.e., there may
#be several players getting the exact same amount.) Find the players who got
any of these 3 salaries
#with all of their details?

dbGetQuery(db, "SELECT * FROM Salaries WHERE yearID = 2003 ORDER BY salary
DESC LIMIT 4" )

##   yearID teamID lgID  playerID   salary
## 1   2003    TEX   AL rodrial01 22000000
## 2   2003    BOS   AL ramirma02 20000000
## 3   2003    TOR   AL delgaca01 18700000
## 4   2003    NYN   NL vaughmo01 17166667

dbGetQuery(db, "SELECT s.yearID, s.teamID, s.lgID, m.nameFirst,m.nameLast,
s.playerID, s.salary, t.name as team,
                      a.G_all,
                      a.G_batting,
                      a.G_defense,
                      a.G_p,
                      a.G_c,
                      a.G_1b,
                      a.G_2b,
                      a.G_3b,
                      a.G_ss,
                      a.G_lf,
                      a.G_cf,
                      a.G_rf,
                      a.G_of,
                      a.G_dh,
                      a.G_ph,
                      a.G_pr
                  FROM Salaries as s
                   JOIN Master as m
                   JOIN Teams as t
                   JOIN Appearances as a
                  ON s.playerID = m.playerID
                    AND s.teamID = t.teamID
                    AND s.playerID = a.playerID
                    AND s.yearID = t.yearID
                    AND s.yearID = a.yearID
                  WHERE s.yearID = 2003 ORDER BY salary DESC LIMIT 4" )
```

```
##   yearID teamID lgID nameFirst  nameLast  playerID   salary
team
## 1   2003    TEX   AL      Alex Rodriguez rodrial01 22000000     Texas
Rangers
## 2   2003    BOS   AL     Manny   Ramirez ramirma02 20000000    Boston Red
Sox
## 3   2003    TOR   AL    Carlos   Delgado delgaca01 18700000 Toronto Blue
Jays
## 4   2003    NYN   NL        Mo    Vaughn vaughmo01 17166667     New York
Mets
##   G_all G_batting G_defense G_p G_c G_1b G_2b G_3b G_ss G_lf G_cf G_rf
G_of
## 1   161       161       158   0   0    0    0    0  158    0    0    0
0
## 2   154       154       128   0   0    0    0    0    0  128    0    0
128
## 3   161       161       147   0   0  147    0    0    0    0    0    0
0
## 4    27        27        25   0   0   25    0    0    0    0    0    0
0
##   G_dh G_ph G_pr
## 1    1    2    0
## 2   26    2    0
## 3   14    0    0
## 4    0    2    0
```

We observe that Alex Rodriguez recieved the largest salary of $22,000,000 who batted for 161 games in 2003 as part of the Texas Rangers.

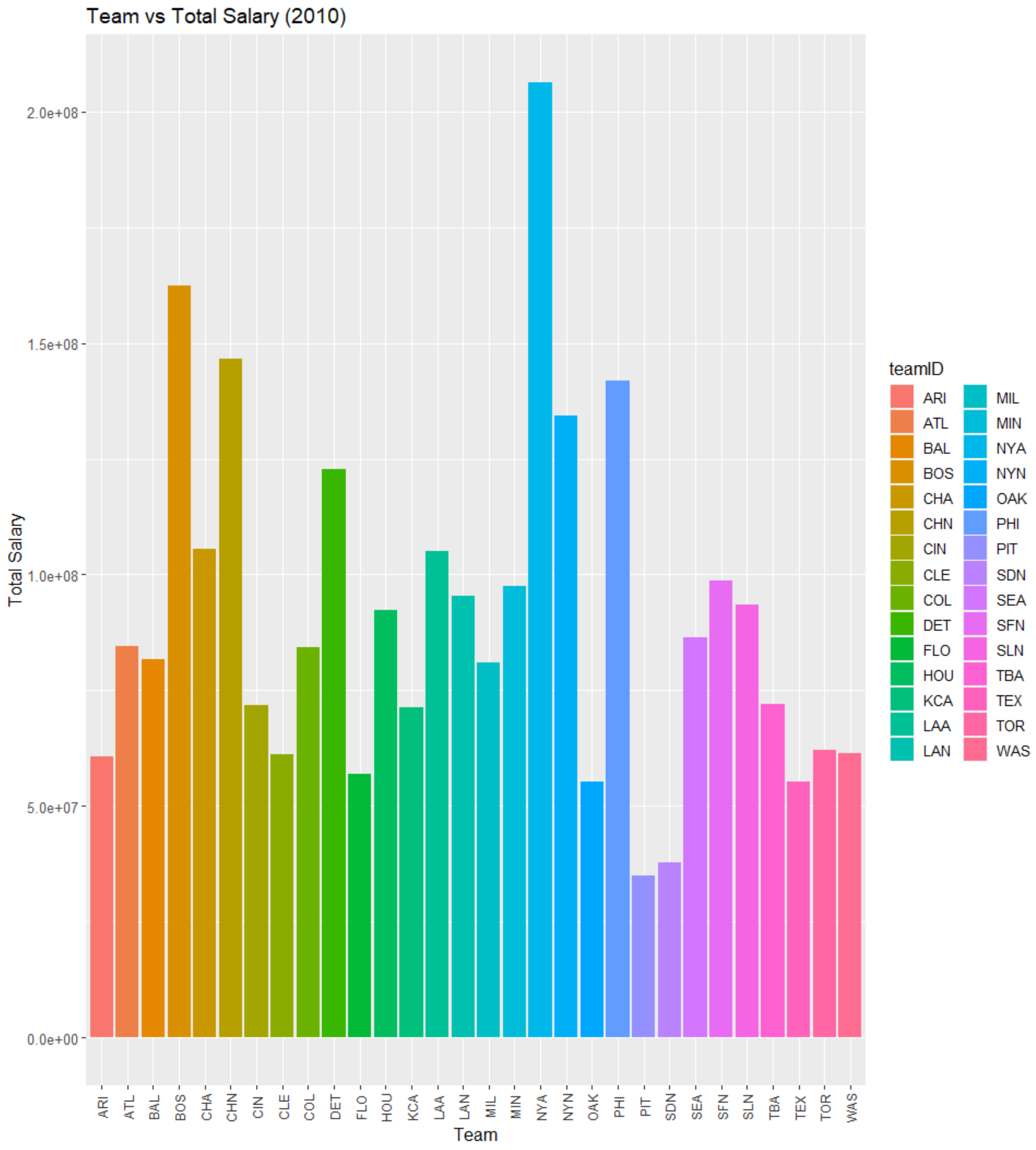
11. For 2010, compute the total payroll of each of the different teams. Next compute the team payrolls for all years in the database for which we have salary information. Display these in a plot.

To compute the total payroll of each of the different teams in 2010 we find the salaries for each player in Salaries table. We then use GROUP BY to group the Salaries table by both teamID and yearID then use SUM(salary) which gives the salaries of each team for each year.

The SQL and R code for this is shown here.

```
#QUESTION 11: For 2010, compute the total payroll of each of the different
teams. Next compute the team
#payrolls for all years in the database for which we have salary information.
```

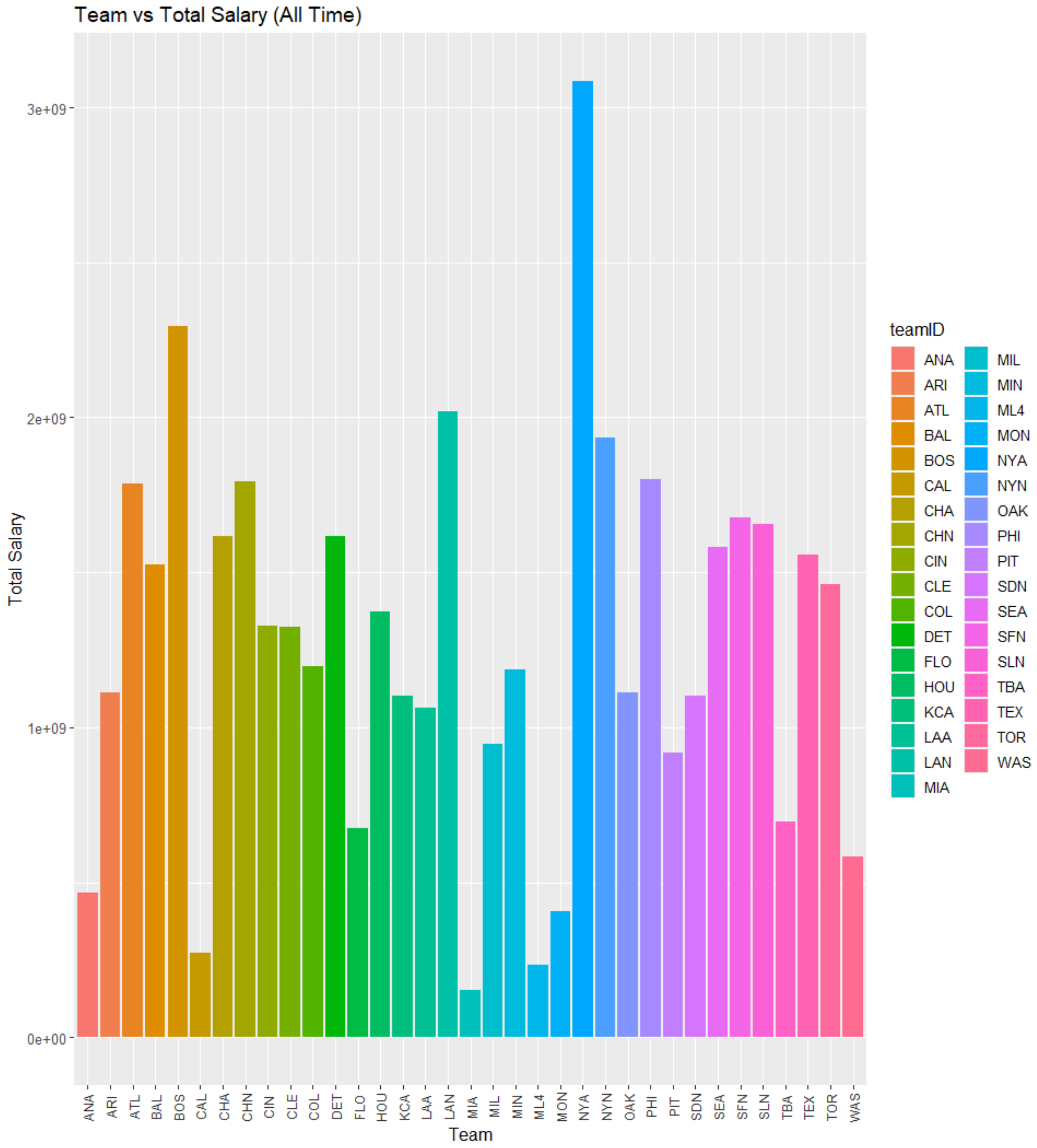
```
Display these in a plot.

#2010
sals_2010 = dbGetQuery(db, "SELECT yearID, teamID, SUM(salary) as sal_total
                             FROM Salaries WHERE yearID = 2010
                             GROUP BY teamID, yearID
                             ORDER BY yearID" )
```



Here we can see that some teams such as (NYA) and (BOS) appear to have a costlier payroll in 2010 than other teams.

To find the payrolls for all years in the database for which we have salary information we will use the same approach in finding just the 2010 payroll however, we will not use a WHERE in our SQL call.

```
#everything
sals_all =  dbGetQuery(db, "SELECT yearID, teamID, SUM(salary) as sal_total
                              FROM Salaries
                              GROUP BY teamID, yearID
                              ORDER BY yearID" )
```



By looking at the graph of all the payrolls from all the teams over all the years team NYA, New York Yankees appears to have the largest payroll.

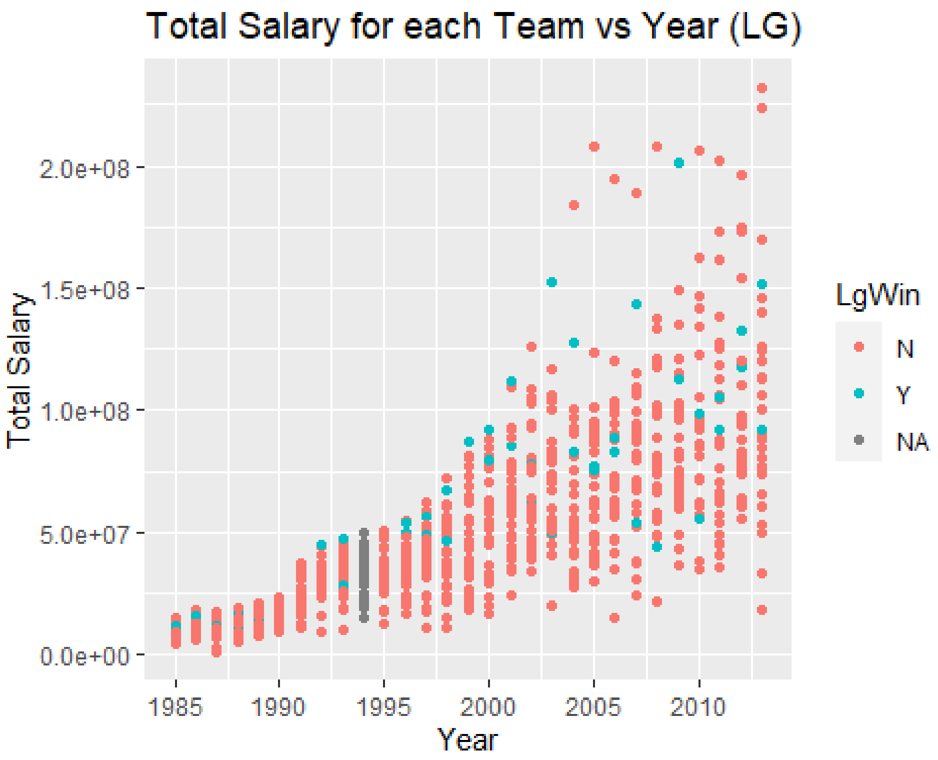
12. Explore the change in salary over time. Use a plot. Identify the teams that won the world series or league on the plot. How does salary relate to winning the league and/or world series.

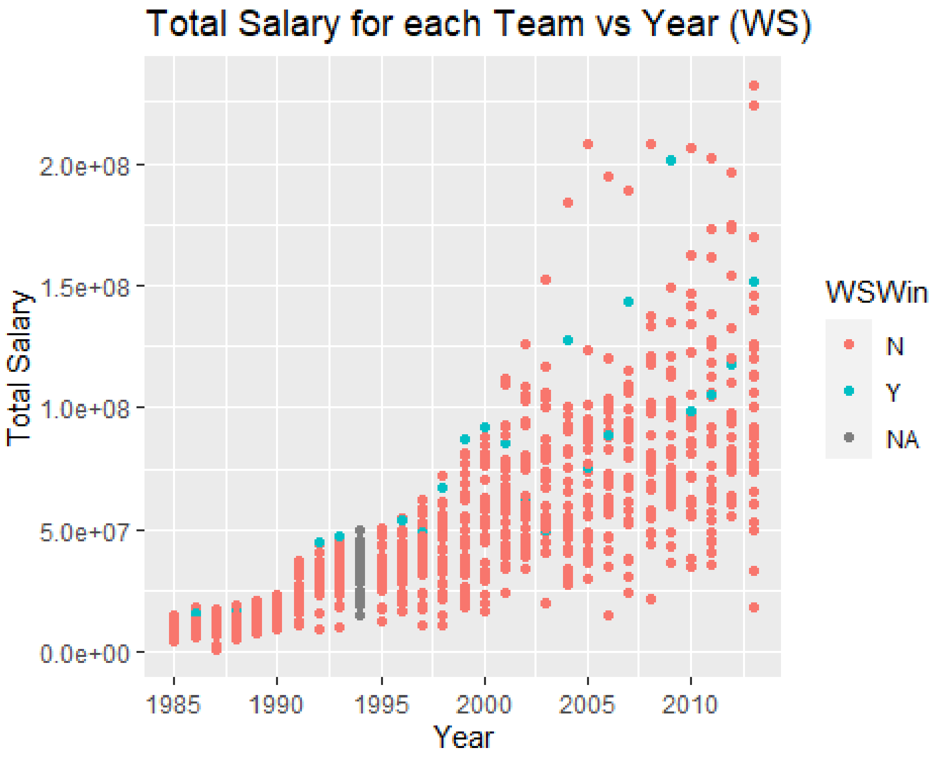
To explore chnge in salary over time we will create scatter plot that shows the salaries for each team on the y-axis and year on the x-axis, then we highlight the World Series and League winners within the plot. First we notice that we need to sum the salaries in the Salary table by grouping by team and year. Then by joining the Teams table we are able to find the teams that won the world series and won thier league for each year using "WSWin" and "LGWin". When joining we align the yearIDs and teamIDs of both tables.

The SQL and R code for this is shown below.

```
# QUESTION 12. Explore the change in salary over time. Use a plot. Identify
the teams that won the world
# series or league on the plot. How does salary relate to winning the league
and/or world series.

sal_Wins = dbGetQuery(db, "SELECT s.yearID, s.teamID, SUM(s.salary) as
sal_total, t.LgWin, t.WSWin
                            FROM Salaries as s
                            LEFT JOIN Teams as t
                              ON (s.yearID = t.yearID AND s.teamID =
t.teamID)
                            GROUP BY s.teamID, s.yearID
                            ORDER BY s.yearID")
head(sal_Wins)

##   yearID teamID sal_total LgWin WSWin
## 1   1985    ATL  14807000     N     N
## 2   1985    BAL  11560712     N     N
## 3   1985    BOS  10897560     N     N
## 4   1985    CAL  14427894     N     N
## 5   1985    CHA   9846178     N     N
## 6   1985    CHN  12702917     N     N
```

Then we may create plots from this data table shown here.





The first graph highlights league winners and we can observe that there is not a clear pattern between the salaries given by each team per year and their ability to win the league. Where as for the second graph it appears that teams that payed higher in salaries tend to win the World Series over the years.

14. Which player has hit the most home runs? Show the number per year.

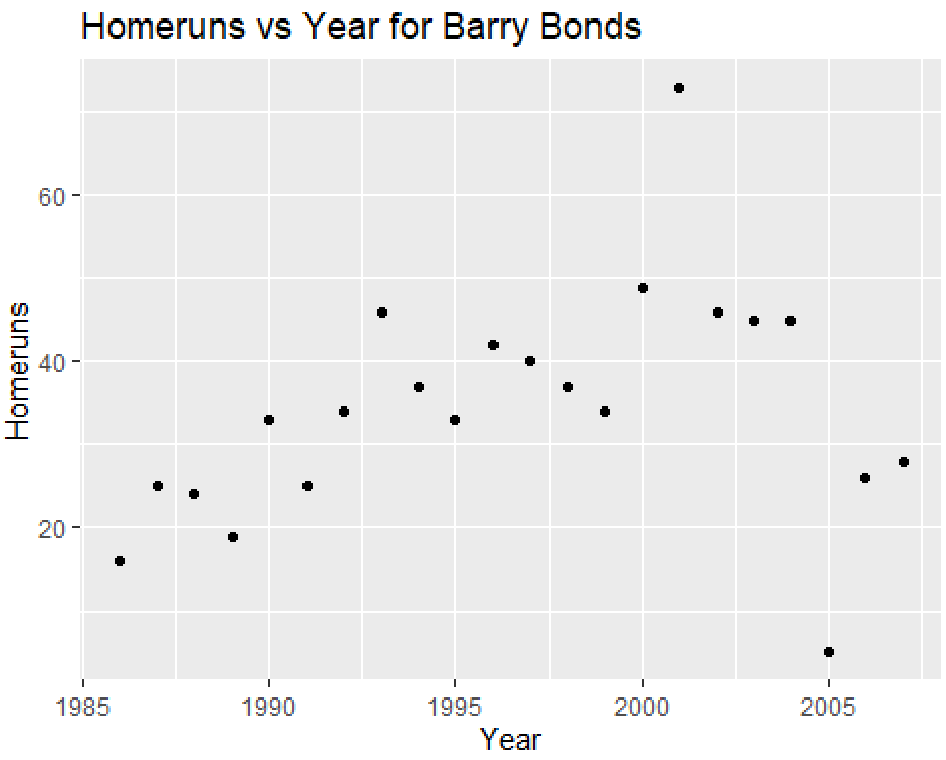
To find the all time statistics about home run hits we observe the Batting table which displays each player's Batting statistic per year. By suming "HR" (the number of homeruns hits per year) by each player using playerID we find the total amount of homeruns hit by each player over all the years. In order to get player name information we JOIN with the Master table and lastly ORDER BY HR. We find that Barry Bonds has the most homeruns with 762 compared with all other player over the years the data covers.

The SQL and R code for this is shown here:

```
#14. Which player has hit the most home runs? Show the number per year.

dbGetQuery(db, "SELECT b.playerID, m.nameFirst, m.nameLast, SUM(b.HR) as HRs
                  FROM Batting as b
                  Join Master as m
                    ON b.playerID = m.playerID
                  GROUP BY b.playerID
                  ORDER BY HRs DESC
                  LIMIT 5")

##    playerID nameFirst  nameLast HRs
## 1 bondsba01     Barry     Bonds 762
## 2 aaronha01      Hank     Aaron 755
## 3  ruthba01      Babe      Ruth 714
## 4  mayswi01    Willie      Mays 660
## 5 rodrial01      Alex Rodriguez 654
```

To find the number of homeruns Barry Bonds hit per year we find the Batting statistics from the Batting table and specify we are only looking for information with the playerID = 'bondsa01' for Barry Bonds. Then we plot the number of hits (HR) per year (yearID) shown below.

```
bb_HR = dbGetQuery(db, "SELECT yearID, playerID, HR
                          FROM Batting
                          WHERE playerID = 'bondsba01' ")
```

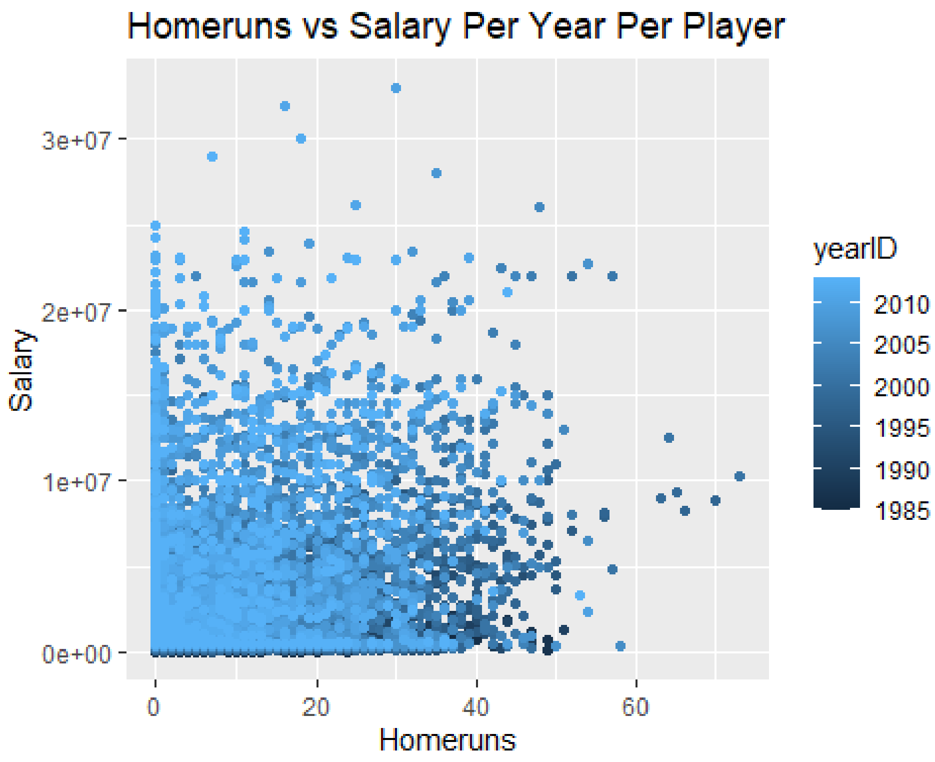


It is interesting to see that Barry Bonds increased the amount of homeruns he hit per year quite steadily. In the time frame of 2000-2005 Bonds it appears hit more home runs than in any other 5 year time frame.

16. Do players who hit more home runs receive higher salaries?

We will be comparing the yearly salary for each player in the Salary table by the number of homeruns the player hit that same year by using the Batting table. We will be observing the number of homeruns a player hit compared to their salary. Below we can see there is not a clear pattern that shows as the number of homeruns increases the salary increases so this relationship may not have a linear relationship.

```
#QUESTION 16: Do players who hit more home runs receive higher salaries?

bats = dbGetQuery(db, "SELECT s.yearID, s.teamID, s.playerID, s.salary, b.HR
                       FROM Salaries as s
                       LEFT JOIN Batting as b
                       ON s.yearID = b.yearID AND s.playerID = b.playerID")

## Warning: Removed 3583 rows containing missing values (geom_point).
```



Conclusion

In this report we used R and SQL to explore interesting aspects of "The Lahman Baseball" SQLite Database. While comparing many aspects of baseball together it was noteable to find most of the teams that won the World Series across all the years of given data showed to have higher team salaries.

Code of Conduct

For The SQL Lahman MLB Project part of the STA141B Fall 2020 I did not look for or use code that addresses this dataset. I implemented the computational approach myself. I used ideas from lecture, office hours, and piazza in my code.