MLB-Analysis SQL

Rishi Bhuva

11/16/2020

My first rule of business, was to install the proper packages used in order to run SQL on R. For this, I had to install RSQLite package and the DBI package. I then specified my working directory using the setwd() function and then used dbConnect to apply SQL within the downloaded database, "lahman2013.sqlite" I then enabled the use of listing remote tables using dbListTables and then enabled the list of the field names within this table using dbListFields and enabled a connection with the "Master" table.

This website was also very useful when working on this project:

http://www.seanlahman.com/files/database/readme2013.txt

```
library(RSQLite)
library(DBI)
library(reshape2)


setwd("~/Downloads")
db <- dbConnect(SQLite(), "lahman2013.sqlite")
dbListTables(db)
```

```
##  [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] "WS Winners"          "WorldSeries Winners" "data2010"
## [28] "temp"
```

```
dbListFields(db, "Master")
```

```
##  [1] "playerID"     "birthYear"    "birthMonth"   "birthDay"     "birthCountry"
##  [6] "birthState"   "birthCity"    "deathYear"    "deathMonth"   "deathDay"
## [11] "deathCountry" "deathState"   "deathCity"    "nameFirst"    "nameLast"
## [16] "nameGiven"    "weight"       "height"       "bats"         "throws"
## [21] "debut"        "finalGame"    "retroID"      "bbrefID"
```

```
# Question 5

# What team won the World Series in 2010?
```

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

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

In order to figure out which team won the world series in 2010, I realized I had to use the dbGetQuery() function. This function essentially enables us to select queries from within the database. Therefore, my logic toward this question was for me to select from a table that included the following criteria: year, teamID (abbreviation of the team), team name, league, division and World Series Win. After browsing the database file (the link posted above), I noticed that the Teams table included each of these. Therefore, I simply used the proper SQL commands to select each of these crtieria from the Teams table and set the conditions as to when the yearID = 2010 and where the WSWin was a 'Y.' This way, I would be able to get a single row with the World Series Winner of 2010. After running this command, we can see that the SF Giants won in 2010 and were apart of the NL League and the W Division.

```
# Question 6

# What team lost the world series each year?

dbGetQuery(db, "SELECT yearID, teamID, name, lgID, divID, WSWin FROM Teams WHERE WSWin = 'N' AND DivWin
```

```
##    yearID teamID                  name lgID divID WSWin
## 1    2013    SLN   St. Louis Cardinals   NL     C     N
## 2    2012    DET        Detroit Tigers   AL     C     N
## 3    2011    TEX         Texas Rangers   AL     W     N
## 4    2010    TEX         Texas Rangers   AL     W     N
## 5    2009    PHI Philadelphia Phillies   NL     E     N
## 6    2008    TBA        Tampa Bay Rays   AL     E     N
## 7    2004    SLN   St. Louis Cardinals   NL     C     N
## 8    2003    NYA      New York Yankees   AL     E     N
## 9    2001    NYA      New York Yankees   AL     E     N
## 10   1999    ATL        Atlanta Braves   NL     E     N
## 11   1998    SDN      San Diego Padres   NL     W     N
## 12   1997    CLE     Cleveland Indians   AL     C     N
## 13   1996    ATL        Atlanta Braves   NL     E     N
## 14   1995    CLE     Cleveland Indians   AL     C     N
## 15   1993    PHI Philadelphia Phillies   NL     E     N
## 16   1992    ATL        Atlanta Braves   NL     W     N
## 17   1991    ATL        Atlanta Braves   NL     W     N
## 18   1990    OAK     Oakland Athletics   AL     W     N
## 19   1989    SFN  San Francisco Giants   NL     W     N
## 20   1988    OAK     Oakland Athletics   AL     W     N
## 21   1987    SLN   St. Louis Cardinals   NL     E     N
## 22   1986    BOS        Boston Red Sox   AL     E     N
## 23   1985    SLN   St. Louis Cardinals   NL     E     N
## 24   1984    SDN      San Diego Padres   NL     W     N
## 25   1983    PHI Philadelphia Phillies   NL     E     N
## 26   1982    ML4     Milwaukee Brewers   AL     E     N
## 27   1981    NYA      New York Yankees   AL     E     N
## 28   1980    KCA    Kansas City Royals   AL     W     N
## 29   1979    BAL     Baltimore Orioles   AL     E     N
```

```
## 30   1978    LAN   Los Angeles Dodgers   NL     W     N
## 31   1977    LAN   Los Angeles Dodgers   NL     W     N
## 32   1976    NYA      New York Yankees   AL     E     N
## 33   1975    BOS        Boston Red Sox   AL     E     N
## 34   1974    LAN   Los Angeles Dodgers   NL     W     N
## 35   1973    NYN         New York Mets   NL     E     N
## 36   1972    CIN       Cincinnati Reds   NL     W     N
## 37   1971    BAL     Baltimore Orioles   AL     E     N
## 38   1970    CIN       Cincinnati Reds   NL     W     N
## 39   1969    BAL     Baltimore Orioles   AL     E     N
```

My logic for this question followed the same logic used for Question 5, although this time I would need to figure out which team lost the World Series and rather than for a single year, I would need to incorporate each year. I used dbGetQuery() and selected the same criteria required (yearID, teamID, name, lgID, divID and WSWin) also using the same table (Teams) After I selected each of these, I realized that within the database, per year, there is one 'Y' for WSWin while the rest were 'N.' Therefore, if I just set the condition to WSWin = 'N' I would get multiple teams per year. I then knew that in order to reach the world series, the team would have to win the division and the league. I then set the conditions to if DivWin = 'Y', lgWin = 'Y' and then WSWin = 'N' in order to establish which team actually made the World Series and then lost. I then used GROUP BY year in order to group these rows with these conditions by year and then ordered them by year for us to see in chronological order which team lost the World Series. I realized also that once I ran this, I would get a table that would start with 1969 and ascend until 2013. I wanted the opposite of this, I wanted the table to start froom 2013 and descend, which is why I added the DESC afterwards.

```
# Question 7

# Compute the table of World Series Winners by year

dbGetQuery(db, "SELECT yearID, teamID, lgID, divID, WSWin FROM Teams WHERE WSWin = 'Y' AND divID != 'NU
```

```
##    yearID teamID lgID divID WSWin
## 1    2013    BOS   AL     E     Y
## 2    2012    SFN   NL     W     Y
## 3    2011    SLN   NL     C     Y
## 4    2010    SFN   NL     W     Y
## 5    2009    NYA   AL     E     Y
## 6    2008    PHI   NL     E     Y
## 7    2007    BOS   AL     E     Y
## 8    2006    SLN   NL     C     Y
## 9    2005    CHA   AL     C     Y
## 10   2004    BOS   AL     E     Y
## 11   2003    FLO   NL     E     Y
## 12   2002    ANA   AL     W     Y
## 13   2001    ARI   NL     W     Y
## 14   2000    NYA   AL     E     Y
## 15   1999    NYA   AL     E     Y
## 16   1998    NYA   AL     E     Y
## 17   1997    FLO   NL     E     Y
## 18   1996    NYA   AL     E     Y
## 19   1995    ATL   NL     E     Y
## 20   1993    TOR   AL     E     Y
## 21   1992    TOR   AL     E     Y
## 22   1991    MIN   AL     W     Y
```

```
## 23   1990    CIN   NL     W     Y
## 24   1989    OAK   AL     W     Y
## 25   1988    LAN   NL     W     Y
## 26   1987    MIN   AL     W     Y
## 27   1986    NYN   NL     E     Y
## 28   1985    KCA   AL     W     Y
## 29   1984    DET   AL     E     Y
## 30   1983    BAL   AL     E     Y
## 31   1982    SLN   NL     E     Y
## 32   1981    LAN   NL     W     Y
## 33   1980    PHI   NL     E     Y
## 34   1979    PIT   NL     E     Y
## 35   1978    NYA   AL     E     Y
## 36   1977    NYA   AL     E     Y
## 37   1976    CIN   NL     W     Y
## 38   1975    CIN   NL     W     Y
## 39   1974    OAK   AL     W     Y
## 40   1973    OAK   AL     W     Y
## 41   1972    OAK   AL     W     Y
## 42   1971    PIT   NL     E     Y
## 43   1970    BAL   AL     E     Y
## 44   1969    NYN   NL     E     Y
```

I followed the same logic as question 6 for this question, although this question was a bit more simple because it is easier extracting the winners of the World Series rather than the loser, as one team wins while the rest don't. Therefore, I used the same criteria and the same table (Teams). I set the condition as where WSWin = 'Y', although after I ran that I noticed that the early stages of the MLB did not include a divID, therefore I believe from 1871 - 1968 all the divID were NA. As the question asked for us to include the division ID, I thought it would be cleaner for me to exclude all those that had an NA for divID. Therefore, I added another condition where only include the rows where divID does not equal to NULL. In the ordered it by year and included it to descend for mere simplicity purposes.

```
# Question 14

# Which player has hit the most Homeruns? Show the number per year

dbGetQuery(db, "SELECT playerID, yearID, MAX(HR) FROM Batting GROUP BY yearID ORDER BY yearID DESC")
```

```
##      playerID yearID MAX(HR)
## 1   davisch02   2013      53
## 2   cabremi01   2012      44
## 3   bautijo02   2011      43
## 4   bautijo02   2010      54
## 5   pujolal01   2009      47
## 6   howarry01   2008      48
## 7   rodrial01   2007      54
## 8   howarry01   2006      58
## 9   jonesan01   2005      51
## 10  beltrad01   2004      48
## 11  rodrial01   2003      47
## 12  rodrial01   2002      57
## 13  bondsba01   2001      73
## 14   sosasa01   2000      50
```

```
## 15  mcgwima01   1999      65
## 16  mcgwima01   1998      70
## 17  griffke02   1997      56
## 18  mcgwima01   1996      52
## 19  belleal01   1995      50
## 20  willima04   1994      43
## 21  bondsba01   1993      46
## 22  gonzaju03   1992      43
## 23  cansejo01   1991      44
## 24  fieldce01   1990      51
## 25  mitchke01   1989      47
## 26  cansejo01   1988      42
## 27  dawsoan01   1987      49
## 28  barfije01   1986      40
## 29  evansda01   1985      40
## 30  armasto01   1984      43
## 31  schmimi01   1983      40
## 32  jacksre01   1982      39
## 33  schmimi01   1981      31
## 34  schmimi01   1980      48
## 35  kingmda01   1979      48
## 36   riceji01   1978      46
## 37  fostege01   1977      52
## 38  schmimi01   1976      38
## 39  schmimi01   1975      38
## 40  schmimi01   1974      36
## 41  stargwi01   1973      44
## 42  benchjo01   1972      40
## 43  stargwi01   1971      48
## 44  benchjo01   1970      45
## 45  killeha01   1969      49
## 46  howarfr01   1968      44
## 47  killeha01   1967      44
## 48  robinfr02   1966      49
## 49   mayswi01   1965      52
## 50  killeha01   1964      49
## 51  killeha01   1963      45
## 52   mayswi01   1962      49
## 53  marisro01   1961      61
## 54  bankser01   1960      41
## 55  matheed01   1959      46
## 56  bankser01   1958      47
## 57  aaronha01   1957      44
## 58  mantlmi01   1956      52
## 59   mayswi01   1955      51
## 60  kluszte01   1954      49
## 61  matheed01   1953      47
## 62  kinerra01   1952      37
## 63  kinerra01   1951      42
## 64  kinerra01   1950      47
## 65  kinerra01   1949      54
## 66  kinerra01   1948      40
## 67  kinerra01   1947      51
## 68  greenha01   1946      44
```

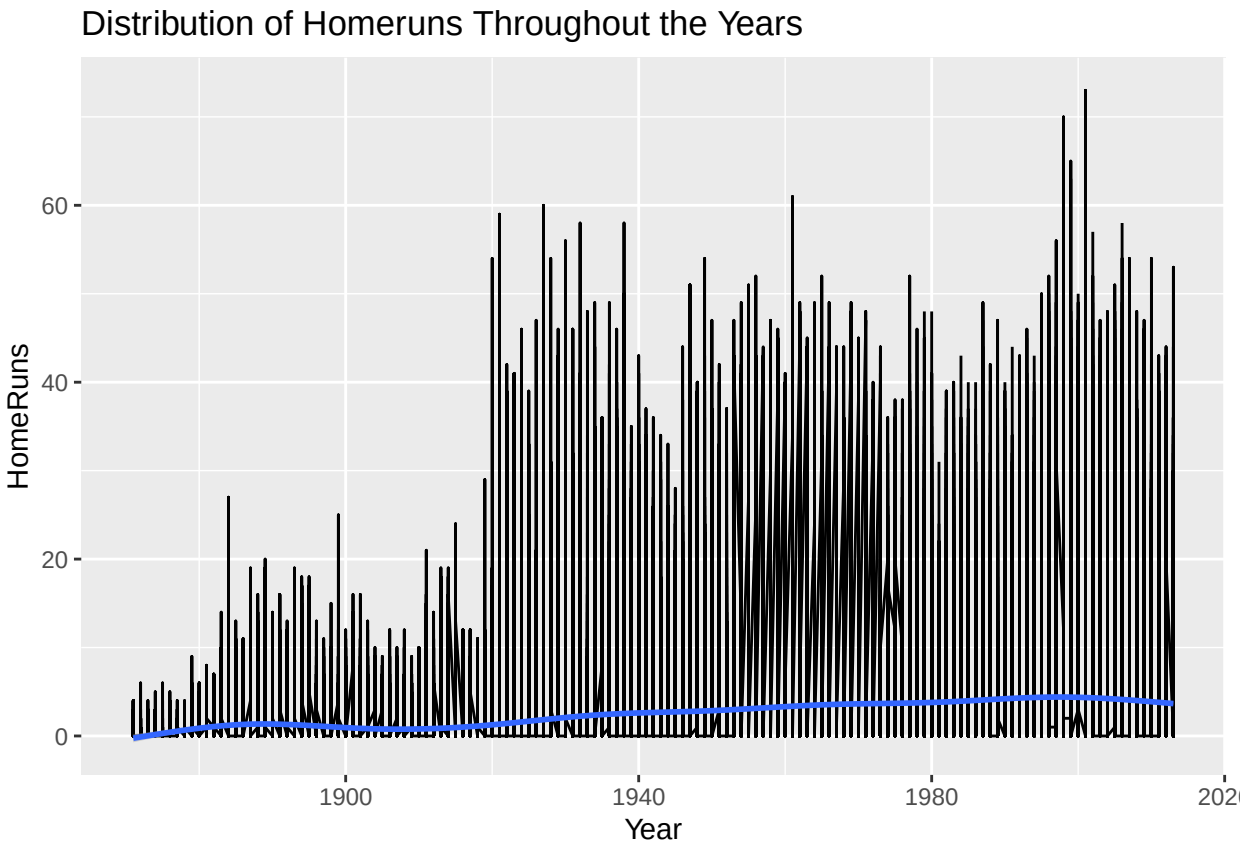
```
## 69  holmeto01   1945      28
## 70  nichobi01   1944      33
## 71   yorkru01   1943      34
## 72  willite01   1942      36
## 73  willite01   1941      37
## 74   mizejo01   1940      43
## 75   foxxji01   1939      35
## 76  greenha01   1938      58
## 77  dimagjo01   1937      46
## 78  gehrilo01   1936      49
## 79   foxxji01   1935      36
## 80  gehrilo01   1934      49
## 81   foxxji01   1933      48
## 82   foxxji01   1932      58
## 83  gehrilo01   1931      46
## 84  wilsoha01   1930      56
## 85   ruthba01   1929      46
## 86   ruthba01   1928      54
## 87   ruthba01   1927      60
## 88   ruthba01   1926      47
## 89  hornsro01   1925      39
## 90   ruthba01   1924      46
## 91   ruthba01   1923      41
## 92  hornsro01   1922      42
## 93   ruthba01   1921      59
## 94   ruthba01   1920      54
## 95   ruthba01   1919      29
## 96   ruthba01   1918      11
## 97  cravaga01   1917      12
## 98   pippwa01   1916      12
## 99  cravaga01   1915      24
## 100 cravaga01   1914      19
## 101 cravaga01   1913      19
## 102 zimmehe01   1912      14
## 103 schulfr01   1911      21
## 104  beckfr02   1910      10
## 105  cobbty01   1909       9
## 106 jordati01   1908      12
## 107 brainda01   1907      10
## 108 davisha01   1906      12
## 109 odwelfr01   1905       9
## 110 davisha01   1904      10
## 111 freembu01   1903      13
## 112 seyboso01   1902      16
## 113 crawfsa01   1901      16
## 114  longhe01   1900      12
## 115 freembu01   1899      25
## 116 colliji01   1898      15
## 117 duffyhu01   1897      11
## 118 delahed01   1896      13
## 119 thompsa01   1895      18
## 120 duffyhu01   1894      18
## 121 delahed01   1893      19
## 122 hollibu01   1892      13
```

```
## 123 stoveha01   1891      16
## 124 connoro01   1890      14
## 125 thompsa01   1889      20
## 126  ryanji01   1888      16
## 127 obriebi01   1887      19
## 128 broutda01   1886      11
## 129 stoveha01   1885      13
## 130 willine01   1884      27
## 131 stoveha01   1883      14
## 132 walkeos01   1882       7
## 133 broutda01   1881       8
## 134 orourji01   1880       6
## 135 jonesch01   1879       9
## 136 hinespa01   1878       4
## 137  pikeli01   1877       4
## 138  hallge01   1876       5
## 139 orourji01   1875       6
## 140 orourji01   1874       5
## 141  pikeli01   1873       4
## 142  pikeli01   1872       6
## 143 meyerle01   1871       4
```

For this question, the information I needed was essentially the player, the year and the number of homeruns per player. As I explored the database file, I realized that these fields were avaliable within the Batting table. Therefore, I selected the playerID, yearID and I included the MAX() for the HR field in order to find the max amount of homeruns. I then used GROUP BY per year in order to group what I selected within Batting per year. After this, I would get the year corresponding to the player who hit the max amount of homeruns. I then used ORDER BY yearID to get the years into chronological order and then I added the DESC for the years to start at 2013 and descend. As we can see, davisch02 (Chris Davis) had the highest amount of homeruns within 2013 and just through a quick search from this table, we can see bondsba01 (Barry Bonds) having the highest number of homeruns at 73 in the year 2001.

```
# Question 15

# Has the Distribution of Homeruns for players increased over the years?

# dbGetQuery(db, "SELECT yearID, HR FROM Batting")

HomeRunData = dbGetQuery(db, "SELECT yearID, HR FROM Batting")

library(ggplot2)

ggplot(HomeRunData, aes(x=yearID, y = HR)) + geom_line() + geom_smooth() + labs(x = "Year", y = "HomeRu
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

## Warning: Removed 6413 rows containing non-finite values (stat_smooth).
```



When I first read this question, I was quite intrigued as this question asks about a distrubtion between two variables. I immediately knew that the best way to present this distrubtion would be to somehow incorporate a plot between these two variables (Year and Homerun) My first action was to determine which table I would easily be able to find the year along with the number of homeruns for that year. After doing some research on the database file, I realized that I could easily extract the year and number of homeruns from the Batting table. The interesting part to this question, was now I was going to have to somewhat transition from SQL to R. Therefore, I first used the SQL commands in order to select the year and # of homeruns from batting. I then assigned my dbGetQuery() command to a variable. On console, just to double check I used the class() function in order to tell me the class to see if it was plottable and it the class function returned a data frame, therefore this would be doable to plot. I then used ggplot() to establish a line graph showing the distribution of homeruns throughout the year. As we can see from the graph, the distrubtion of homeruns throughout the years has increased and we can even see the smooth line in an upward trend as well, showing that the distrubtion of homeruns has increased over the years.

```
# Question 3

# How many players became managers

dbGetQuery(db, "SELECT COUNT(*) playerID, plyrMgr FROM Managers GROUP BY plyrMgr = 'Y'")
```

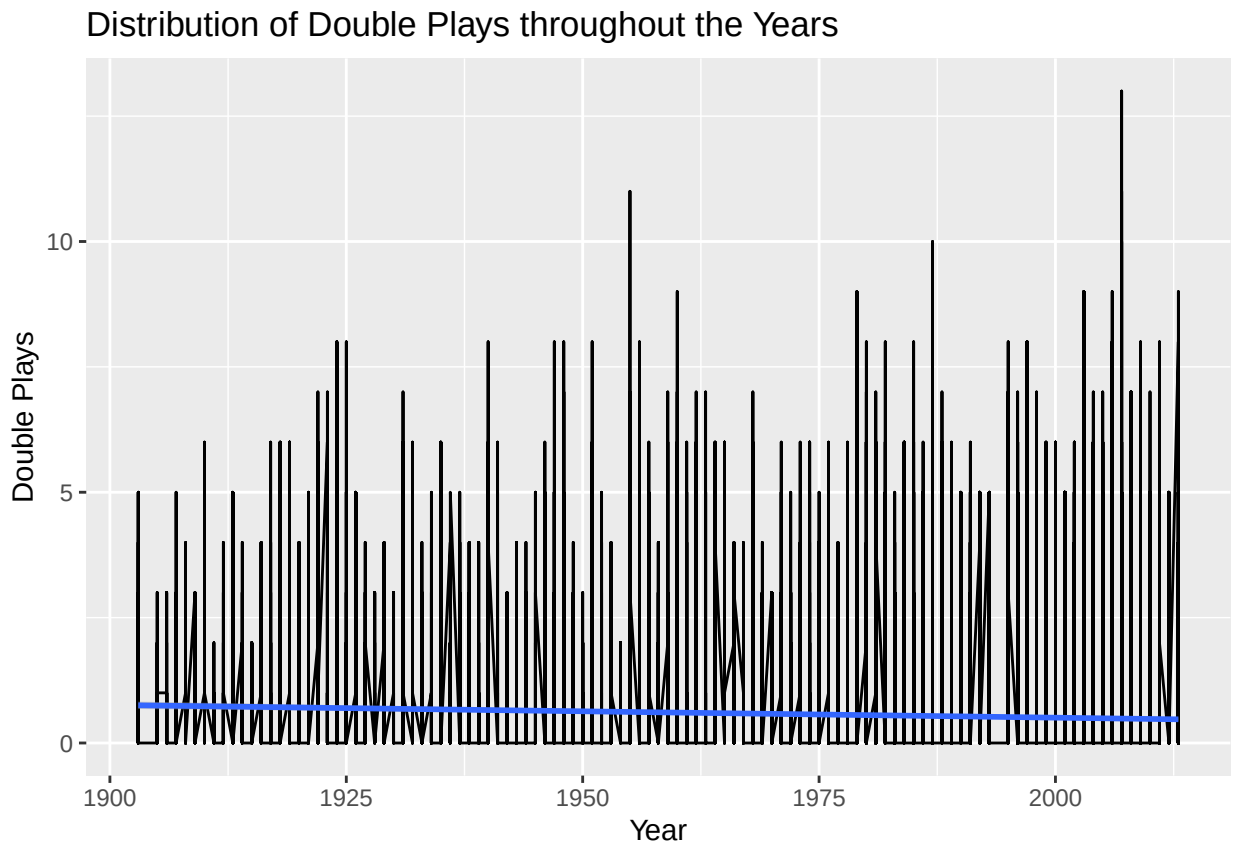
```
##   playerID plyrMgr
## 1     2692       N
## 2      645       Y
```

When I read this question, I immediately began to surf the database file to see which fields I would have to extract and from which table would those fields be located. I saw that the Managers table included the

playerID and plyrMgr which was denoted by 'Y' if a player was a manager or 'N' if a player was not a manager. My logic when it came to this question was to count which players had the 'Y' in their plyrMgr column. To do this, I used the COUNT(*) to count all the playerIDs from the table and used a condition to count the number of plyMgr as 'Y' and as 'N'. As we can see, 645 players are players that became managers while 2,692 are players that have no became a manager.

```
# Question 18


# What's the distrubtion of double plays/triple plays?

# dbGetQuery(db, "SELECT yearID, DP, TP FROM FieldingPost")



DPTP = dbGetQuery(db, "SELECT yearID, DP, TP FROM FieldingPost")



library(ggplot2)

ggplot(DPTP, aes(x = yearID, y = DP)) + geom_line() + geom_smooth(method = "lm") + labs(x = "Year", y =
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
# ggplot(DPTP, aes(x = yearID, y = TP)) + geom_line() + labs(x = "Year", y = "Triple Plays", title = "
```

My logic when approaching 18 was very similar to that of question 15. I knew that because the question was asking me for a distribution, I had to incorporate some plots within my answer. I began to scavenge through the database file and came across the FieldingPost table. This table includes the fields DP (double play) and TP (triple play) and as this question was asking what's the distrubtion of them, I decided to plot each against year. Although, after checking out the data, I noticed that the triple play data within this table seemed to be a little off... I noticed that the highest would be one, although with a google search of triple plays I could tell that this dataset had no properly incorporated the correct amount of triple plays within the MLB. Therefore, the TP plot against year can be seen as inaccurate, which is why I have commented it out and not shown the graph. The DP data does seem quite accurate, and according to the plot above, we can see a general increase in double plays throughout the year. As I did add a smooth line to this graph, we can see that the line doesn't show a proper upward trend indicating that the distrubtion of double plays is not necessarily in an upward trend. After some more analyzing of this plot, I can see that between the years there are a lot of fluctuations and this is possibly why there isn't an upward trend. The max's for double plays can be seen toward the later years, although if we look closely we can also see there are lows within those years as well. Therefore, the distrubtion of double plays hasn't necessarily increased over the years, due to the fluctuation within each year.

```
# Question 17

# Are Certain Baseball Parks better for scoring homeruns?

# dbGetQuery(db, "SELECT HR FROM Teams")
avgHR = dbGetQuery(db, "SELECT HR FROM Teams")
summary(avgHR)
```

```
##        HR
##  Min.   :  0
##  1st Qu.: 40
##  Median :105
##  Mean   :100
##  3rd Qu.:148
##  Max.   :264
```

```
dbGetQuery(db, "SELECT DISTINCT yearID, HR, park FROM Teams WHERE HR > 148 GROUP BY park ORDER BY yearI
```

```
##    yearID  HR                              park
## 1    2013 151                      Target Field
## 2    2013 171                 Progressive Field
## 3    2012 195                     O.co Coliseum
## 4    2012 187           Angel Stadium of Anaheim
## 5    2011 149                   Sun Life Stadium
## 6    2009 244                 Yankee Stadium III
## 7    2009 156                    Nationals Park
## 8    2007 179      Rangers Ballpark in Arlington
## 9    2006 199                     Rogers Centre
## 10   2006 164                     R.F.K. Stadium
## 11   2006 161                         Petco Park
## 12   2006 182                   Dolphin Stadium
## 13   2006 160                       Chase Field
## 14   2006 184                 Busch Stadium III
```

```
## 15   2006 159                         Angel Stadium
## 16   2006 163                            AT&T Park
## 17   2005 155                       McAfee Coliseum
## 18   2005 260                      Ameriquest Field
## 19   2004 151 Stade Olympique/Hiram Bithorn Stadium
## 20   2004 183                               SBC Park
## 21   2004 189           Network Associates Coliseum
## 22   2004 215                    Citizens Bank Park
## 23   2004 162             Angels Stadium of Anaheim
## 24   2003 220                  U.S. Cellular Field
## 25   2003 182              Great American Ball Park
## 26   2002 167                     Minute Maid Park
## 27   2001 161                              PNC Park
## 28   2001 209                         Miller Park
## 29   2000 162                      Tropicana Field
## 30   2000 198                         Safeco Field
## 31   2000 160                   Pro Player Stadium
## 32   2000 226                         PacBell Park
## 33   2000 249                          Enron Field
## 34   2000 177                       Comerica Park
## 35   1999 244               Kingdome / Safeco Field
## 36   1999 209                       Cinergy Field
## 37   1998 223                     Busch Stadium II
## 38   1998 159                    Bank One Ballpark
## 39   1998 166                            Astrodome
## 40   1997 174                       Turner Field
## 41   1997 172                     Stade Olympique
## 42   1997 152                    Qualcomm Stadium
## 43   1997 158                    Kauffman Stadium
## 44   1997 161            Edison International Field
## 45   1997 172                            3Com Park
## 46   1996 221             The Ballpark at Arlington
## 47   1996 150                    Joe Robbie Stadium
## 48   1995 200                           Coors Field
## 49   1994 167                         Jacobs Field
## 50   1993 157           Oriole Park at Camden Yards
## 51   1993 162                     Comiskey Park II
## 52   1990 167                              Skydome
## 53   1987 192                         Shea Stadium
## 54   1986 196           Hubert H Humphrey Metrodome
## 55   1986 184                     Arlington Stadium
## 56   1985 154                        Royals Stadium
## 57   1985 171                              Kingdome
## 58   1983 167                    Exhibition Stadium
## 59   1979 164                       Anaheim Stadium
## 60   1977 184                     Yankee Stadium II
## 61   1977 186                     Veterans Stadium
## 62   1977 181                    Riverfront Stadium
## 63   1977 191                        Dodger Stadium
## 64   1977 192                         Comiskey Park
## 65   1971 154                  Three Rivers Stadium
## 66   1970 171                      Oakland Coliseum
## 67   1970 172                   Jack Murphy Stadium
## 68   1970 191      Crosley Field/Riverfront Stadium
```

```
## 69   1966 207         Atlanta-Fulton County Stadium
## 70   1961 189                   Wrigley Field (LA)
## 71   1961 180                        Tiger Stadium
## 72   1961 167                  Metropolitan Stadium
## 73   1961 149                      Memorial Stadium
## 74   1961 183                     Candlestick Park
## 75   1959 163                   Griffith Stadium II
## 76   1958 170                        Seals Stadium
## 77   1958 172         Los Angeles Memorial Coliseum
## 78   1957 166                   Municipal Stadium I
## 79   1956 150                        Briggs Stadium
## 80   1953 166                         Crosley Field
## 81   1953 156                        County Stadium
## 82   1950 161                        Fenway Park II
## 83   1949 152                          Ebbets Field
## 84   1948 155                     Cleveland Stadium
## 85   1947 221                       Polo Grounds IV
## 86   1947 156                          Forbes Field
## 87   1937 150                           Navin Field
## 88   1932 172                            Shibe Park
## 89   1930 171                         Wrigley Field
## 90   1929 153                            Baker Bowl
## 91   1927 158                       Yankee Stadium I
```

I feel like I answered Question 17 in a non-traditional way, although it seems to work. I found that the Teams table contains homeruns along with the park name. What I did first was select the homeruns from the table and use the summary() function to get a summary of the homeruns within this table. This summary claims that the 3rd Quartile is 148 HR, meaning that the highest 25% of homeruns will be greater than 148. I applied this logic and created another Query, where I selected the year, HR, park from the Teams table and only selected HR $>$ 148. I then grouped it by parks in order to see which parks acquire these higher-than-normal homerun statistics. I then ordered by year in order to see a good breakdown throughout the years of homeruns within the parks. Through this, we can see these the parks that have a high number of homeruns.

```
# Question 1

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

dbGetQuery(db, "SELECT min(yearID), max(yearID) from Pitching")
```

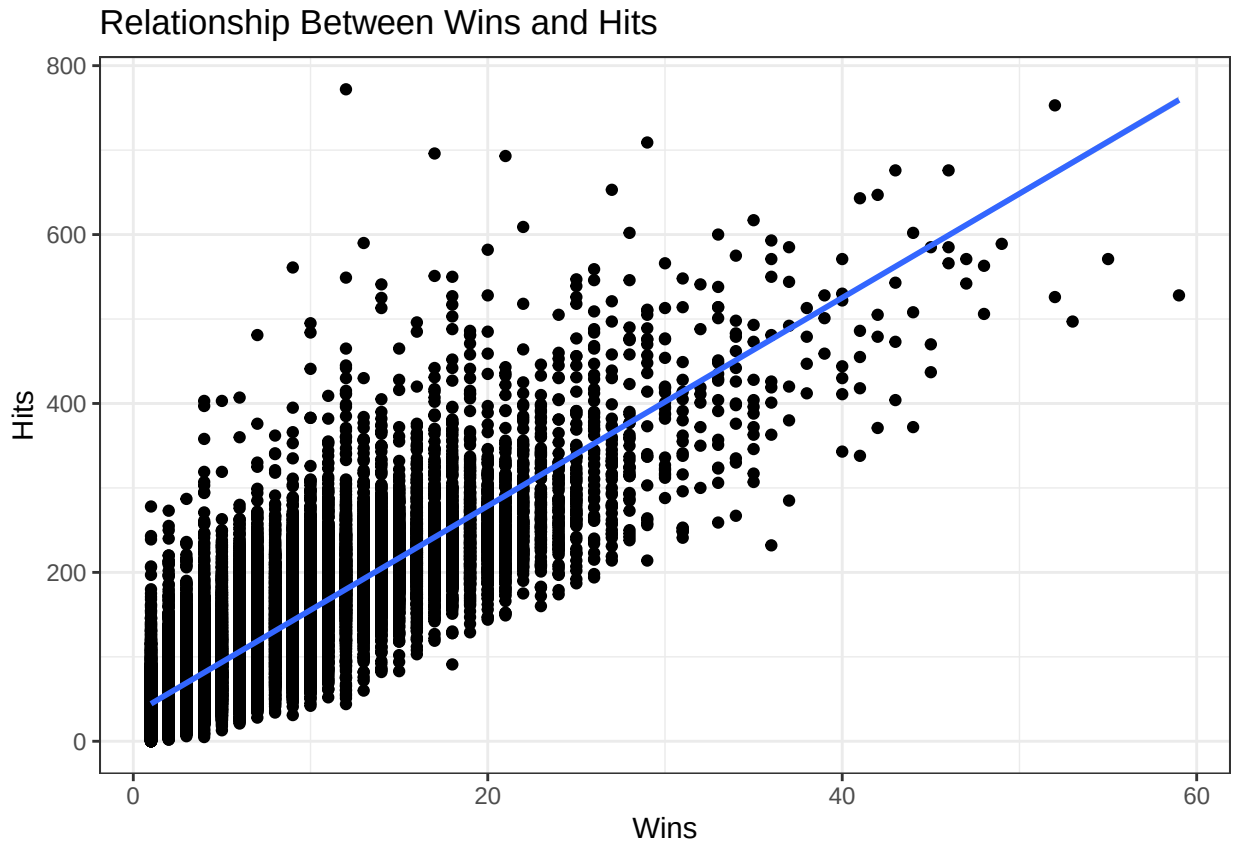
```
##   min(yearID) max(yearID)
## 1        1871        2013
```

This question seemed quite simple to me, what I did was essentially search for a table that included a yearID. In this case, I used Pitching. I then selected the minimum year and the maximum year provided within the yearID. I get 1871 for the minumum year and 2013 for the maximum year. This shows that data ranges from 1871 to 2013, as we can see from the years we have been extracting from other tables, this proves accurate. I believe that there is data for a certain specific field from 1871-2013, although not all. For example, divID is not available until 1969. Therefore, divID does not have data until then. Although for the most part, there should be at least one specific data (probably more) for each year from 1871 - 2013.

```
# Question 22

# How are wins related to hits, strikeouts, walks, homeruns and earned runs
```

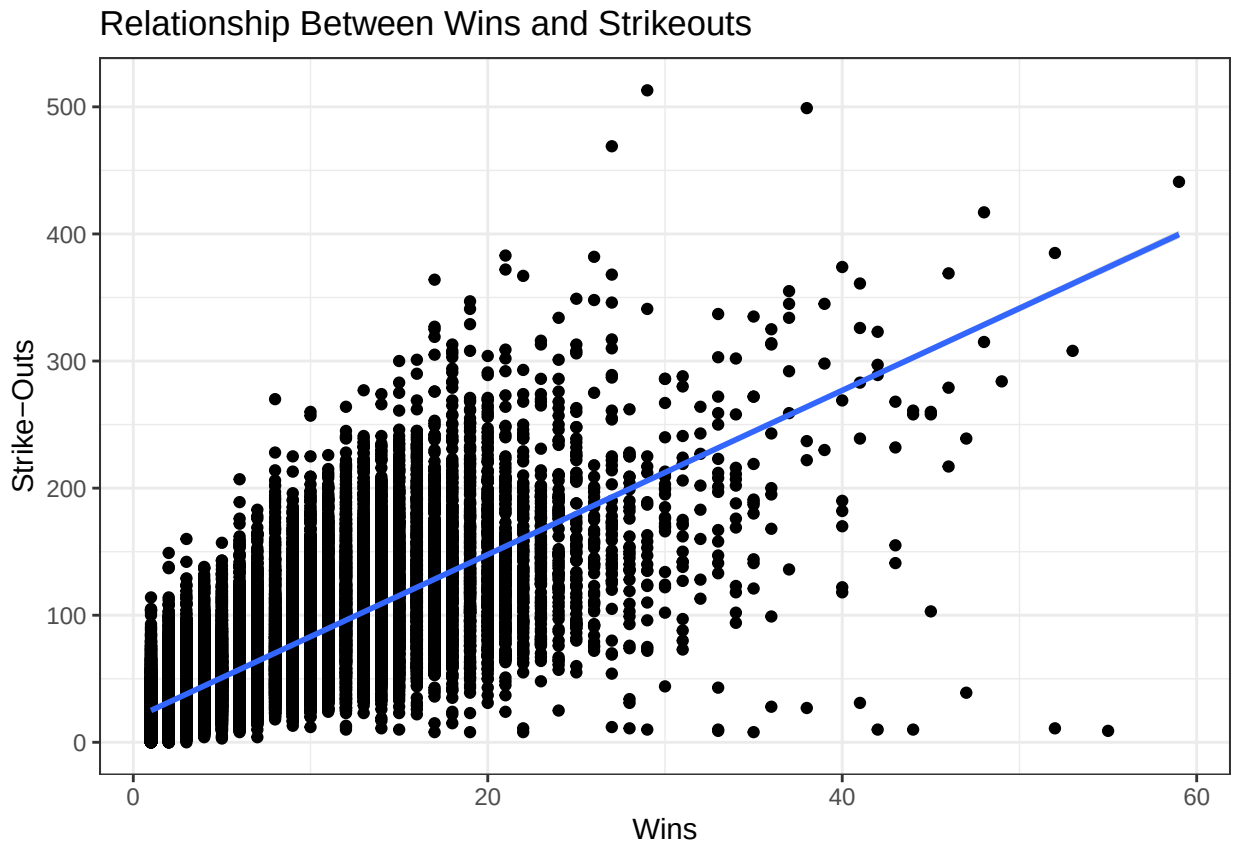
```
# dbGetQuery(db, "SELECT W, SO, BB, H, ER, HR FROM Pitching WHERE W!= 0")

HitData = dbGetQuery(db, "SELECT W, SO, BB, H, ER, HR FROM Pitching WHERE W != 0")

library(ggplot2)

HitWin = ggplot(HitData, aes(x=(W), y =H))
HitWin + aes() + geom_point() + geom_smooth(method = "lm") + theme_bw() + labs(x = "Wins", y = "Hits", t
```

```
## `geom_smooth()` using formula 'y ~ x'
```



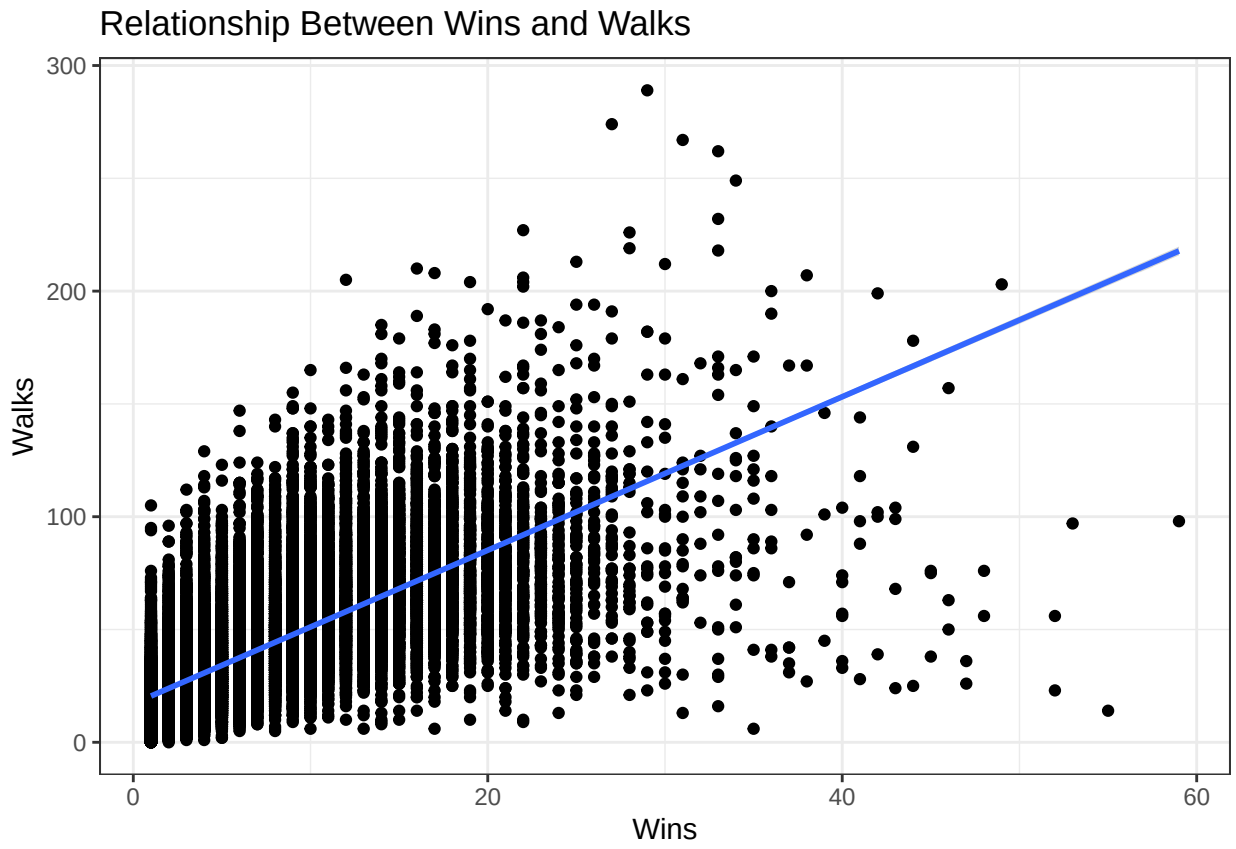

```
SOWin = ggplot(HitData, aes(x=(W), y =SO))
SOWin + aes() + geom_point() + geom_smooth(method = "lm") + theme_bw() + labs(x = "Wins", y = "Strike-Ou
```

```
## `geom_smooth()` using formula 'y ~ x'
```



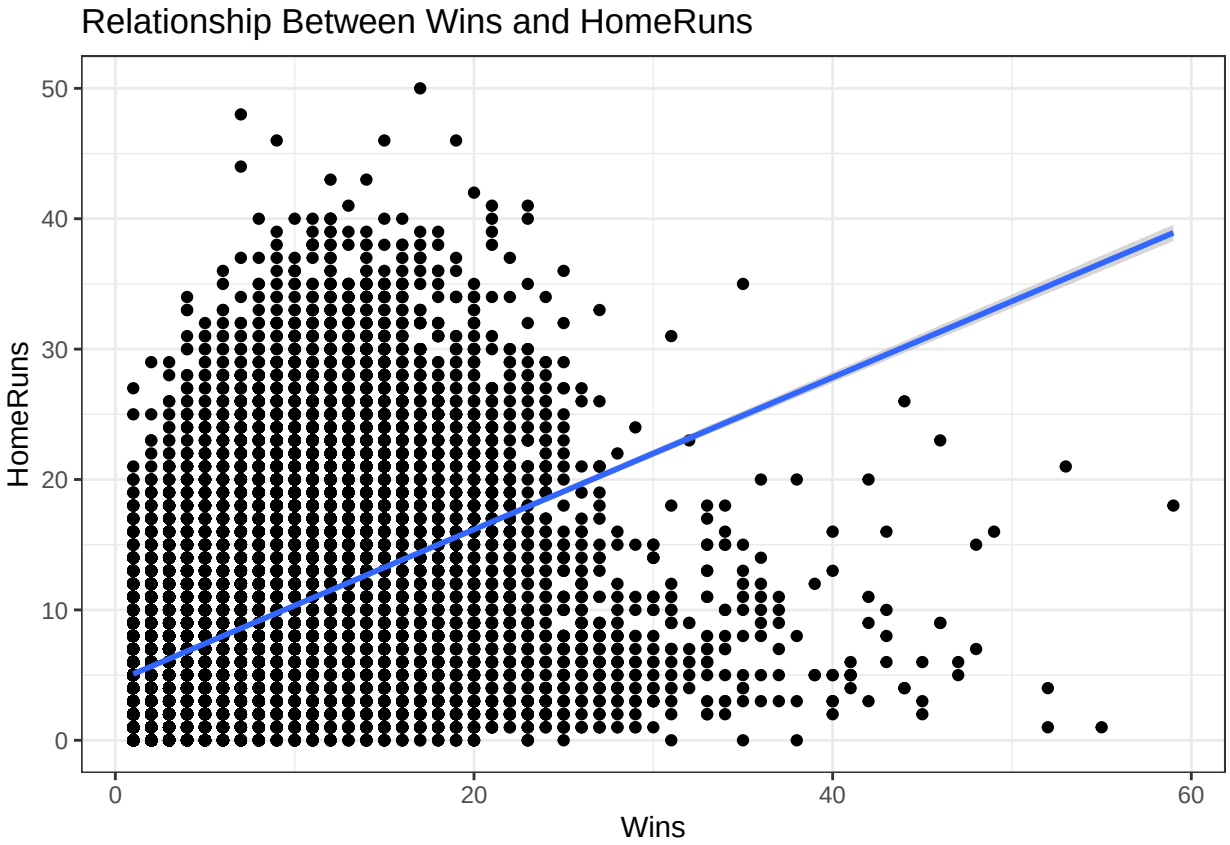

```
WWin = ggplot(HitData, aes(x=(W), y =BB))
WWin + aes() + geom_point() + geom_smooth(method = "lm") + theme_bw() + labs(x = "Wins", y = "Walks", t
```

```
## `geom_smooth()` using formula 'y ~ x'
```



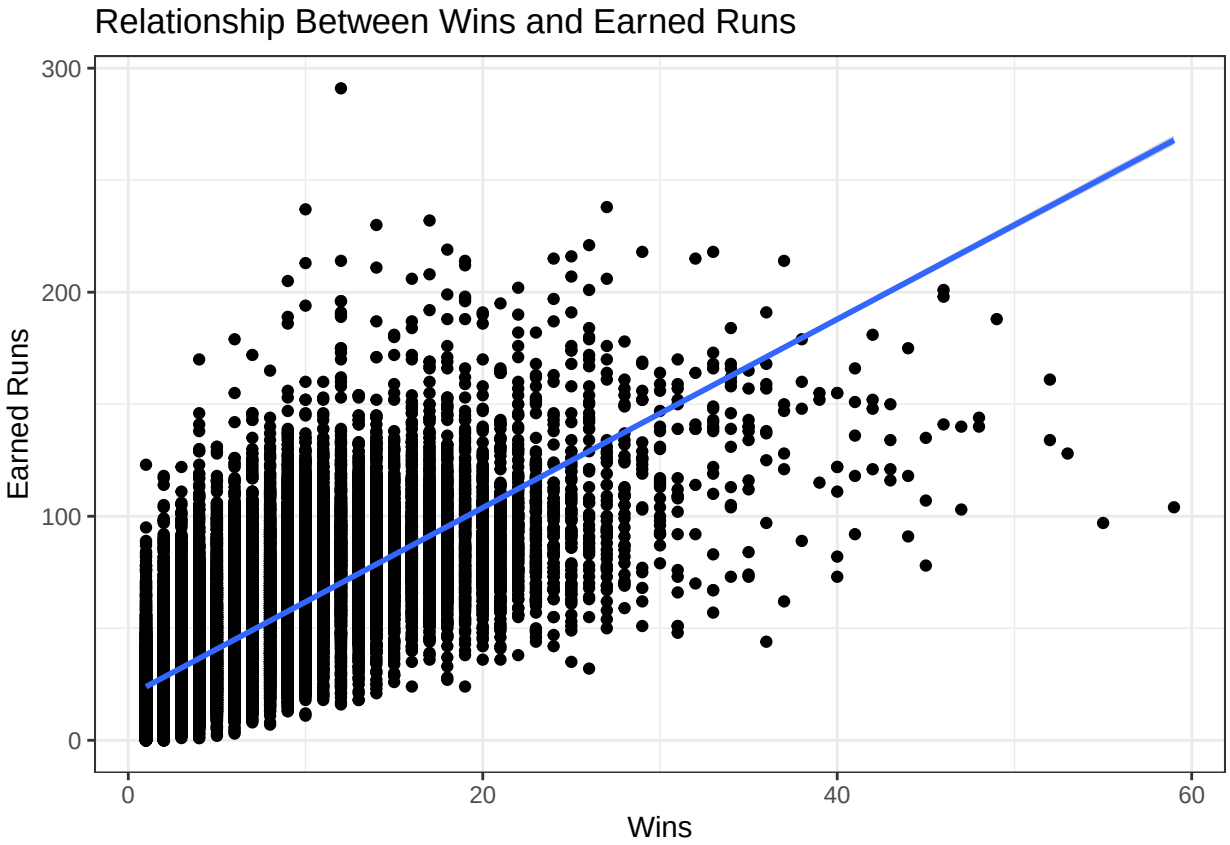

```
HRWin = ggplot(HitData, aes(x=(W), y =HR))
HRWin + aes() + geom_point() + geom_smooth(method = "lm") + theme_bw() + labs(x = "Wins", y = "HomeRuns
```

```
## `geom_smooth()` using formula 'y ~ x'
```



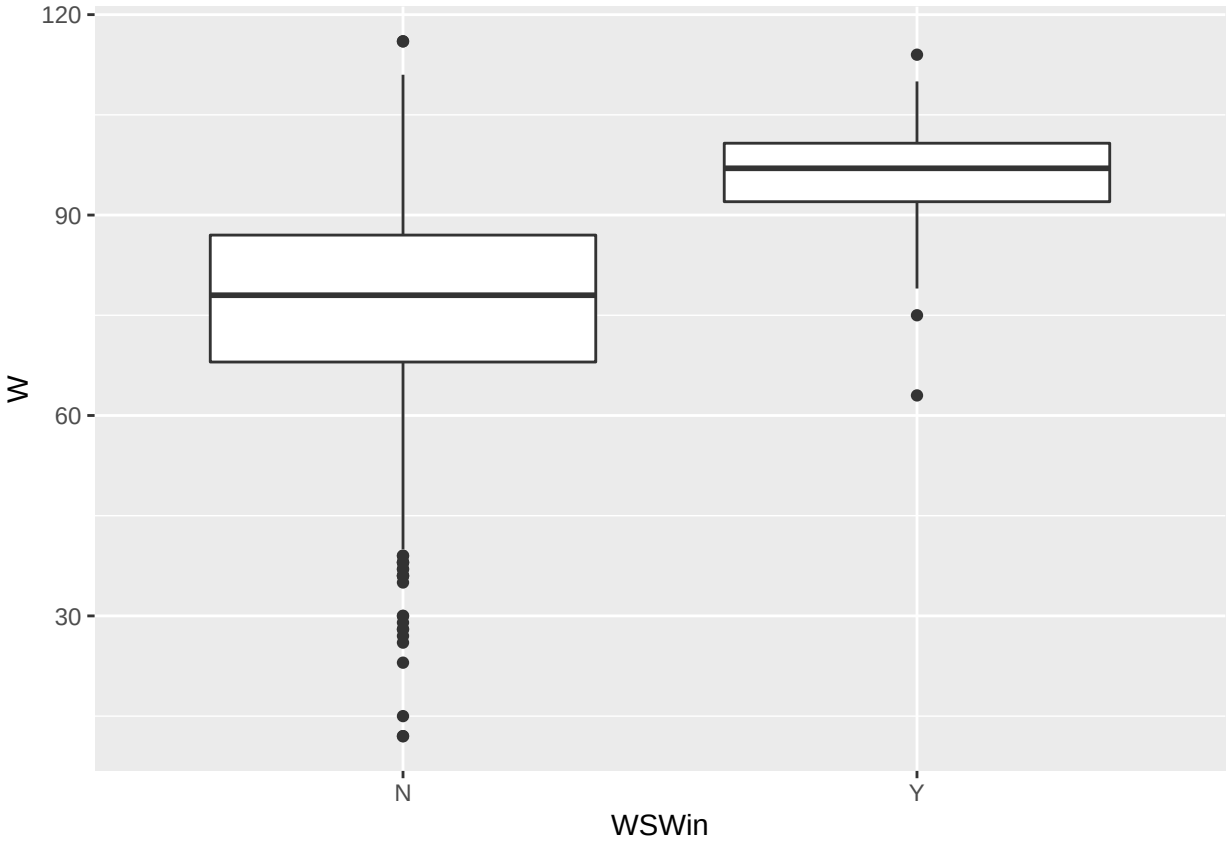

```
ERWin = ggplot(HitData, aes(x=(W), y =ER))
ERWin + aes() + geom_point() + geom_smooth(method = "lm") + theme_bw() + labs(x = "Wins", y = "Earned Ru
```

```
## `geom_smooth()` using formula 'y ~ x'
```



This question asks to see the relationship between hits, strikeouts, walks, homeruns and earned runs.I knew that the best way to portray this relationship was to individually plot Wins with the other respected variables (hits, strikeouts, walks, homeruns and earned runs) I first figured out that Pitching would be the most ideal table to select from as it contained all the variables I needed. I then set a condition to cut out any NA Wins or any Wins that were 0 as they were considered a lost, not a win. I then established a variable to this query and made sure that it was a dataframe by checking the class() within my console. Afterwards, I decided to plot Wins against each variable. The first from wins and hits, the second from wins and strikeouts, the third from wins and walks, fourth from wins and homeruns and fifth from wins and earned runs. Throughout each graph, we can see a positive correlation between wins and each respected variable. The smooth line also helps satisfy this claim as the smooth line is in an upward direction for each graph. Therefore, wins have a positive correlation when it comes to hits, strikeouts, walks, homeruns and earned runs.

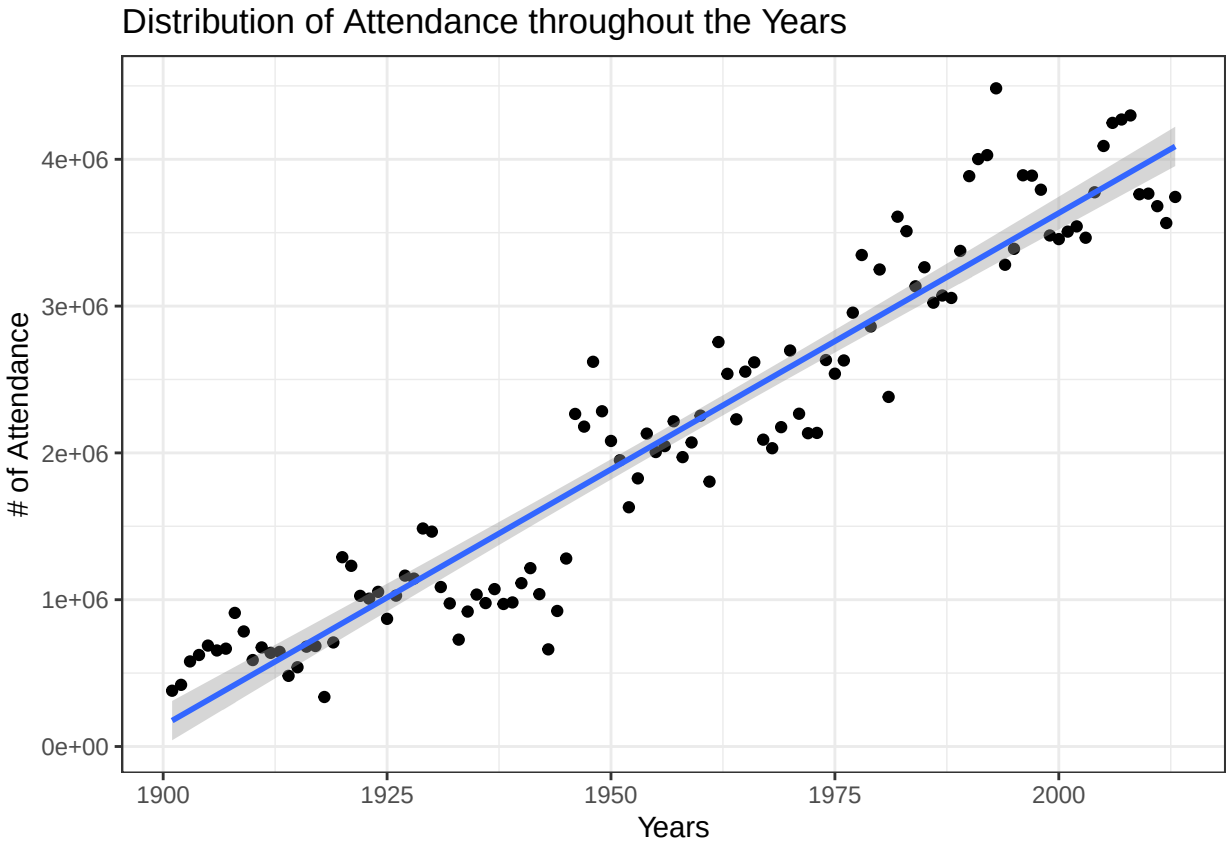
```
# Question 9

# Do you see a relationship between games won in a season and winning the World Series?

# dbGetQuery(db, "SELECT yearID, W, WSWin FROM Teams WHERE WSWin != 'NULL'")

WSWinData = dbGetQuery(db, "SELECT yearID, W, WSWin FROM Teams WHERE WSWin!= 'NULL'")

library(ggplot2)

ggplot(data = WSWinData, aes(x = WSWin, y = W)) +  geom_boxplot()
```



For this question, I immedediately looked at the Teams table and selected the year, the number of wins, and world series winner denoted by 'Y' or 'N'. I then established a boxplot between the number of wins that had not won a World Series (N) vs the number of wins that had a won a world series (Y), as we can see from this boxplot, there are more teams that have not won a World Series, therefore more data to plot within N, although we can see that on average, teams that have won the world series tend to have a higher number of wins in a season.

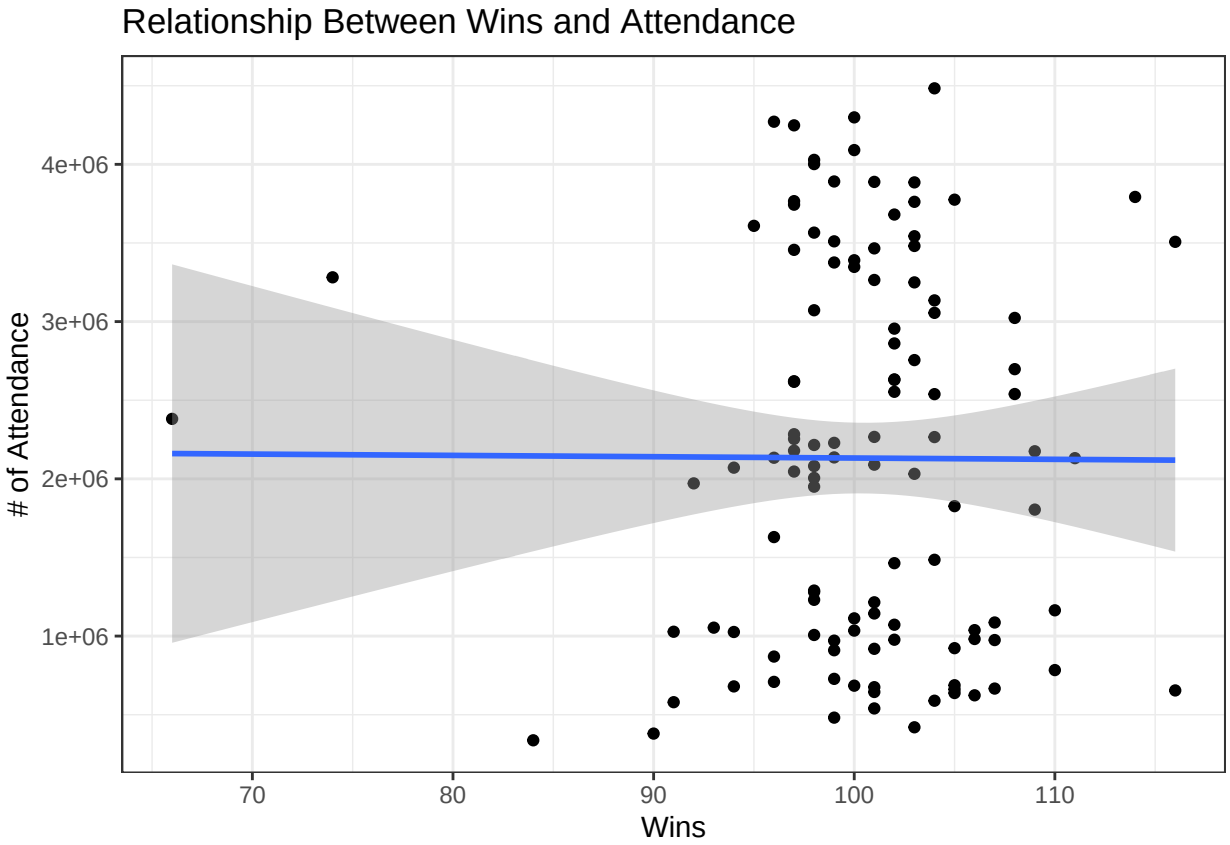
```
# Has attendance increased over the years in baseball?

# Does attendance effect the number of wins in baseball?


attendancedub = dbGetQuery(db, "SELECT yearID, teamID, name, G, GHome, max(W) AS Wins, max(attendance) A

library(ggplot2)

ggplot(data = attendancedub, aes(x = yearID, y = Attendance)) + geom_point() + geom_smooth(method = "lm
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
ggplot(data = attendancedub, aes(x = Wins, y = Attendance)) + geom_point() + geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



This was a question I posed for myself, as the fields within the dataset seemed interesting and I wanted to see some sort of relationship. I was looking for whether attendance as increased over the years in baseball and if attendance effects the number of wins in baseball. What I did first, was extract fields from the Teams table. The fields that I extracted were the year, team, name, games played, games at home played, the max amount of wins and the highest number of attendance. I then applied conditions that take out the 'NULL' of the data and then I used GROUP BY yearID to see these variables grouped by the year. I then plotted attendance by year, and as we can see attendance did increase as the years went on. The next plot shows the correlation of attendance and wins, and as we can see through the plot, there really is no correlation. I think one thing that would help me properly extract this data and get a more proper answer would be if any field within the table included games won at home, therefore I would properly be able to tell if "home-court" advantage would be a thing. Although, as we can see with the second plot, the number of wins and attendance don't show any relationship.

```
# Question 23

# What are the top ten college producers of MLB players? How many colleges are represented in the datab

dbGetQuery(db, "SELECT schoolName, COUNT(SPName.schoolID) AS NumCount FROM Schools AS SName LEFT JOIN (S
```

```
##                                schoolName NumCount
## 1          University of Southern California      102
## 2              University of Texas at Austin      100
## 3                   Arizona State University       98
## 4                        Stanford University       82
## 5                      University of Michigan       77
## 6                  College of the Holy Cross       75
```

```
## 7                    University of Notre Dame       70
## 8  University of Illinois at Urbana-Champaign       68
## 9       University of California, Los Angeles       66
## 10                      University of Arizona       66
```

```
# dbGetQuery(db, "SELECT DISTINCT schoolname FROM Schools")

dbGetQuery(db, "SELECT DISTINCT COUNT(schoolname) FROM Schools")
```

```
##   COUNT(schoolname)
## 1               749
```

This question was quite interesting, what I did was first find the tables that include schoolname, which I found from Schools. I then found schoolID from SchoolsPlayers which tell us the schoolID in regard to the player. After this, I established a count for how many schoolID's occur for each player and then correspond them to schoolName. I then used GROUP BY schoolID from SchoolsPlayers and ORDER it by the count of schools from the table SchoolsPlayers. I then used the LIMIT command to enable the top 10 colleges that produce MLB players, as we can see USC is the top with 102 players. After this, I wanted to count how many colleges are within this database, I did this by using SELECT DISTINCT (which does not give any repeated queries) of the schoolname field from the Schools table. This count gives us 749 colleges. Above the count code, I have the code without the COUNT command, this commented code gives us a list of all of these colleges, I decided to comment it out simply because of the extensive list it would produce.