STA 141B Project 3

Joseph Gonzalez

11/8/2020

I used the sqlite file from canvas to complete the questions. Codes for plots can be found in the appendix. ***Some queries were limited to save space.***

Question 1

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

For this question, we want to want to find the minimum year(use MIN()) and maximum year(use MAX()). Finding the minimum and maximum will allow us to determine the time frame for the data. There are a few tables we can use to answer this question(Batting, Pitching, Teams, etc.). However, we want to use only the yearID field to generate the years.

The SQL for this is:

```
#First, we can query the covered dates:
q_year_span="SELECT
              MIN(yearID) AS Begin_year,
              MAX(yearID) AS End_year
             FROM Batting"
dbGetQuery(Baseball_database, q_year_span)
```

```
##   Begin_year End_year
## 1       1871     2013
```

```
#The data covers from 1871 to 2013
```

From the query, we found that the data ranges from 1871 to 2013. I also checked the Pitching and Teams tables and found that the years matched.

To check that there is data for each year, we want to confirm that each year between 1871 and 2013(inclusive) is included in the data. To do this we can count(using COUNT()) the number of distinct YearID's(using DISTINCT()) listed in the batting, pitching, or teams tables. If there are data for each of these years, the count should equal 143(The max year minus the minimum year plus 1)

```
#Now, we can check to see if there are data for each of these years:
#For this we can look to see the total of unique years
#If there is data for all years, the total
#unique years should match the difference between
#the 2013 and 1871 plus 1(143)
each_year_q = "SELECT
```

```
                COUNT(DISTINCT(yearID)) AS Total_years
               FROM Batting"
dbGetQuery(Baseball_database, each_year_q)
```

```
##   Total_years
## 1         143
```

```
#It appears that all years are listed in the data
```

After executing the query, there are 143 distinct years and, therefore, there appears to be data for each of there years.

Question 2

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

To count people in the database(players, managers, etc), we want to use the functions COUNT() and DISTINCT(). We can use these functions along with the playerID in the MASTER table to count the number of unique people. If my assumption is correct, the MASTER table should contain a unique playerID for all the persons in the database. There may be a few IDs that are not in the MASTER table(errors), but we can assume most are there. To find the number of managers, we use playerID in the Managers table. If we only want managers that only managed and did not play at the same time, we can restrict the player-manager field(plyrMgr) to no("N"). However, this does not account for a person managing and playing in another season. We can also get the number of just players(no managing) by removing the managers' playerIDs from the query(restricting playerID). We can use WHERE, NOT IN and a sub-query containing the managers' playerIDs to perform this action. Lastly, we can get the number of players that played baseball and may have managed at the same time(not just manager) by restricting the playerID to not include manager playerIDs that have the player-manager field set to no("N").

```
#Number of Unique people in the database:
#We can use the master table that contains the unique code for players
person_count_q = "SELECT
                    COUNT(DISTINCT(playerID)) AS Num_unique_persons
                  FROM MASTER"
dbGetQuery(Baseball_database, person_count_q)
```

```
##   Num_unique_persons
## 1              18354
```

```
#Number of managers(Includes managers that managed the team and also managed-played):
manager_count_q = "SELECT
                    COUNT(DISTINCT(playerID)) AS Num_Managers
                   FROM Managers"
dbGetQuery(Baseball_database, manager_count_q)
```

```
##   Num_Managers
## 1          682
```

```
#Number of managers that only managed
#(plyrMgr is set to 'N', does not account for if they managed and played in another season):
manager_count2_q = "SELECT
                      COUNT(DISTINCT(playerID)) AS Num_Managers
                    FROM Managers
                    WHERE plyrMgr = 'N'"
dbGetQuery(Baseball_database, manager_count2_q)
```

```
##   Num_Managers
## 1          515
```

```
#Number of players(That played baseball and did not manage):
player_count_q = "SELECT
                    COUNT(DISTINCT(playerID)) AS Num_players
                  FROM MASTER
                  WHERE playerID NOT IN (SELECT playerID FROM Managers)"
dbGetQuery(Baseball_database, player_count_q)
```

```
##   Num_players
## 1       17675
```

```
#Number of players(That played baseball and may have managed at the same time):
player_count2_q = "SELECT
                     COUNT(DISTINCT(playerID)) AS Num_players
                   FROM MASTER
                   WHERE playerID NOT IN (
                   SELECT
                      DISTINCT(playerID)
                   FROM Managers WHERE plyrMgr = 'N')"
dbGetQuery(Baseball_database, player_count2_q)
```

```
##   Num_players
## 1       17842
```

There are 18354 unique persons in the database. There are 682 total managers(include those that managed and played at the same time). There are 515 persons that managed and did not play at the same time. There are 17675 players in the database that did not manage. There are also 17842 players, including player-managers, in the database.

```
#Some test numbers did not seem to add up correctly in previous section:
#A check here
player_check_q = "SELECT
                    COUNT(DISTINCT(M.playerID)) AS Num_players
                  FROM MASTER M
                  JOIN Managers Ma
                  ON M.playerID = Ma.playerID"
dbGetQuery(Baseball_database, player_check_q)
```

```
##   Num_players
## 1         679
```

```
#It appears that there are 3 player IDs
#from Managers that do not appear on the Master Table
missing_managers_q = "SELECT
                        playerID AS Miss_managers
                      FROM Managers
                      WHERE playerID NOT IN (SELECT DISTINCT(playerID) FROM MASTER)"
dbGetQuery(Baseball_database, missing_managers_q)
```

```
##   Miss_managers
## 1     cammebi99
## 2     hengled99
## 3     keanejo99
## 4     keanejo99
## 5     keanejo99
## 6     keanejo99
## 7     keanejo99
## 8     keanejo99
```

```
#cammebi99, hengled99, and keanejo99
#are missing from the master list(Seems that they were managers only)
missing_managers2_q = "SELECT
                          *
                       FROM Managers
                       WHERE playerID IN ('cammebi99', 'hengled99', 'keanejo99')
                       GROUP BY playerID"
dbGetQuery(Baseball_database, missing_managers2_q)
```

```
##    playerID yearID teamID lgID inseason  G  W  L rank plyrMgr
## 1 cammebi99   1876    NY3   NL        1 57 21 35    6       N
## 2 hengled99   1884    CHU   UA        1 74 34 39    5       N
## 3 keanejo99   1961    SLN   NL        2 80 47 33    5       N
```

There are 3 playerIDs in the Managers table that are not in the MASTER table. We can add these to the total unique persons in the database(18357). This shows that there could be other playerIDs in the database that are missing from the MASTER table.

Question 3

How many players became managers?

For this question, we want to see how many baseball players' playerIDs from the MASTER Table are in the manager table. I assume that the player-manager(plyrMgr) field means that the player also manages the team and not the player becomes a manager. To get the number of players that become managers, we can restrict the playerID in the master table to be in the managers and the batting table. Restricting the playerID to be in the batting table will confirm that the person played in a game(has batting data). We use DISTINCT and COUNT to get the number of unique players and we restrict playerID with WHERE and sub-querying.

```
# Count the number of player IDs from the Master List that appear in the Managers list:
player_to_manager_q = "SELECT
                         COUNT(DISTINCT playerID) AS Player_to_Manager
```

```
                        FROM MASTER
                        WHERE playerID IN (SELECT DISTINCT(playerID) FROM Managers)
                        AND playerID IN (SELECT DISTINCT(playerID) FROM Batting)"
dbGetQuery(Baseball_database, player_to_manager_q)
```

```
##   Player_to_Manager
## 1               561
```

There are 561 players that became managers.

Question 4

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

For this question, we can approximate the number of active players in a year and on a team. The number of players on the payroll for the specific year should approximate the number of active players well. I don't use the batting or fielding tables because some players move from the minor league to the major league during different times of the season(replace starters with injuries), which means using these tables would overestimate the total active players. If we use the WHERE and BETWEEN statements, we can restrict the yearID from 2000 to 2013(inclusive). We also want to use GROUP BY to group the years and count the unique playerIDs in that year. To get the number of players on the team for the specific years, we add TeamID to the GROUP BY statement.

```
#If possible, Group by year
#I believe the salaries tables should give us the
#most accurate number of players in the league but
#We can compare it will another table(Fielding)
num_players_year_q = "SELECT
                        yearID, COUNT(DISTINCT(playerID)) AS Num_Players
                      FROM Fielding
                      WHERE yearID BETWEEN 2000 AND 2013 GROUP BY yearID"
dbGetQuery(Baseball_database, num_players_year_q)
```

```
##    yearID Num_Players
## 1    2000        1226
## 2    2001        1212
## 3    2002        1216
## 4    2003        1224
## 5    2004        1244
## 6    2005        1231
## 7    2006        1236
## 8    2007        1273
## 9    2008        1280
## 10   2009        1260
## 11   2010        1249
## 12   2011        1284
## 13   2012        1278
## 14   2013        1299
```

```
#I'm thinking that this number may overestimate the number of players each year.
#Some players are brought up from the minor league
#during different times of the season(replace starters with injuries).
#So, this number may estimate the total number of unique players that play in a season.
#However, we may want the number of players that are active or start a season and
#my assumption is the Salary table would more accurately reflect this.

#We can't guarantee that certain players are in the Fielding, Batting or Pitching Tables.
#In baseball, not all players pitch, field(DH hitters), or bat(some pitchers)
#For these reasons, I believe the Salary table should provide the most accurate estimate for
#the number of players(assuming all players salaries are listed).
num_players_year2_q = "SELECT
                        yearID, COUNT(DISTINCT(playerID)) AS Num_Players
                       FROM Salaries
                       WHERE yearID BETWEEN 2000 AND 2013
                       GROUP BY yearID"
dbGetQuery(Baseball_database, num_players_year2_q)
```

```
##    yearID Num_Players
## 1    2000         835
## 2    2001         860
## 3    2002         846
## 4    2003         827
## 5    2004         831
## 6    2005         831
## 7    2006         819
## 8    2007         842
## 9    2008         856
## 10   2009         813
## 11   2010         829
## 12   2011         839
## 13   2012         848
## 14   2013         814
```

```
#Number of players per team:
#Group by team:
#limit output to 50 tuples
num_players_per_team_q = "SELECT
                            teamID, yearID, COUNT(DISTINCT(playerID)) AS Num_Players
                          FROM Salaries
                          WHERE yearID BETWEEN 2000 AND 2013
                          GROUP BY yearID, TeamID
                          LIMIT 50"
dbGetQuery(Baseball_database, num_players_per_team_q)
```

```
##    teamID yearID Num_Players
## 1     ANA   2000          30
## 2     ARI   2000          28
## 3     ATL   2000          30
## 4     BAL   2000          29
## 5     BOS   2000          30
## 6     CHA   2000          29
## 7     CHN   2000          30
```

```
## 8     CIN   2000          27
## 9     CLE   2000          26
## 10    COL   2000          28
## 11    DET   2000          27
## 12    FLO   2000          28
## 13    HOU   2000          27
## 14    KCA   2000          28
## 15    LAN   2000          26
## 16    MIL   2000          32
## 17    MIN   2000          26
## 18    MON   2000          29
## 19    NYA   2000          28
## 20    NYN   2000          25
## 21    OAK   2000          27
## 22    PHI   2000          29
## 23    PIT   2000          26
## 24    SDN   2000          30
## 25    SEA   2000          26
## 26    SFN   2000          26
## 27    SLN   2000          27
## 28    TBA   2000          31
## 29    TEX   2000          26
## 30    TOR   2000          25
## 31    ANA   2001          30
## 32    ARI   2001          28
## 33    ATL   2001          31
## 34    BAL   2001          29
## 35    BOS   2001          32
## 36    CHA   2001          27
## 37    CHN   2001          27
## 38    CIN   2001          27
## 39    CLE   2001          30
## 40    COL   2001          26
## 41    DET   2001          28
## 42    FLO   2001          31
## 43    HOU   2001          28
## 44    KCA   2001          28
## 45    LAN   2001          29
## 46    MIL   2001          28
## 47    MIN   2001          27
## 48    MON   2001          31
## 49    NYA   2001          31
## 50    NYN   2001          29
```

```
#No, It doesn't seem that all teams have the same number or players
#This makes sense because some teams may have a larger salary cap for the
#players than other teams.
```

From 2000 to 2013, we can approximate that the number of active players is between 813 and 860. No, not all teams have the same number of players. The number of players on each team ranges from the mid-twenties to the low-thirties.

Question 5

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

We want to use the teams table to find the winner. In the teams table, we can restrict the yearID to 2010 and the world-series win field(WSWin) to yes('Y'). The teams table also has the team's name(name), league(lgID), and division(divID). We use these fields in the SELECT statement.

```
#Teams table may contain the necessary info:
team_wseries_2010_q = "SELECT
                        teamIDBR AS Team_Abbrev, name AS Team_name,
                        lgID AS League, divID AS Division
                       FROM Teams
                       WHERE yearID = '2010' AND WSWin = 'Y'"
dbGetQuery(Baseball_database, team_wseries_2010_q)
```

```
##   Team_Abbrev            Team_name League Division
## 1         SFG San Francisco Giants     NL        W
```

```
#San Francisco Giants won in 2010
```

The San Francisco Giants won the world series in 2010. They are in the national league and west division.

Question 6

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

For this question, we can identify teams that have never won the world series(lost each year). The first formal world series started in 1903. Therefore, I decided to restrict yearID to include 1903 and any year after. The teamID field changes when a team switches names, leagues, or discontinues for several years and, later, is re-established. This means that using teamID to identify teams that lost the World Series may not be effective because it will identify teams that won a world series but have a different name. To fix this issue, we use the franchise ID(franchID) to identify the teams that never won a world series. To get the teams that did not win a world series, we use the NOT IN statement and subquery the franchIDs that won the world series. We also GROUP BY the names to get all the names for the teams that never won. We can also GROUP BY the franchID to get just the franchises that have no World Series titles.

```
#Teams table has sufficient info to answer this question:
team_wseries_lost_q = "SELECT
                        franchID AS Franchise_ID, name AS Team_Name,
                        lgID AS League, IFNULL(divID, 'None')  AS Division
                       FROM Teams
                       WHERE franchID NOT IN (
                        SELECT
                          franchID
                        FROM Teams
                        WHERE WSWin = 'Y') AND yearID >= '1903'
                       GROUP BY name
                       ORDER BY franchID"
dbGetQuery(Baseball_database, team_wseries_lost_q)
```

```
##    Franchise_ID             Team_Name League Division
## 1           BFL         Buffalo Blues     FL     None
## 2           BFL       Buffalo Buffeds     FL     None
## 3           BLT   Baltimore Terrapins     FL     None
## 4           BTT     Brooklyn Tip-Tops     FL     None
## 5           CHH      Chicago Chi-Feds     FL     None
## 6           CHH        Chicago Whales     FL     None
## 7           COL      Colorado Rockies     NL        W
## 8           HOU        Houston Astros     NL     None
## 9           HOU    Houston Colt .45's     NL     None
## 10          KCP   Kansas City Packers     FL     None
## 11          MIL     Milwaukee Brewers     AL        W
## 12          MIL        Seattle Pilots     AL        W
## 13          NEW Indianapolis Hoosiers     FL     None
## 14          NEW         Newark Pepper     FL     None
## 15          PBS     Pittsburgh Rebels     FL     None
## 16          SDP      San Diego Padres     NL        W
## 17          SEA      Seattle Mariners     AL        W
## 18          SLI    St. Louis Terriers     FL     None
## 19          TBD  Tampa Bay Devil Rays     AL        E
## 20          TBD        Tampa Bay Rays     AL        E
## 21          TEX         Texas Rangers     AL        W
## 22          TEX   Washington Senators     AL     None
## 23          WSN        Montreal Expos     NL        E
## 24          WSN  Washington Nationals     NL        E
```

```
#See franchises that have not won a world series:
team_wseries_lost_q = "SELECT
                       franchID AS Franchise_ID, name AS Team_Name,
                       lgID AS League, IFNULL(divID, 'None') AS Division
                      FROM Teams
                      WHERE franchID NOT IN (
                       SELECT
                         franchID
                       FROM Teams
                       WHERE WSWin = 'Y') AND yearID >= '1903'
                      GROUP BY franchID
                      ORDER BY franchID"
dbGetQuery(Baseball_database, team_wseries_lost_q)
```

```
##    Franchise_ID             Team_Name League Division
## 1           BFL       Buffalo Buffeds     FL     None
## 2           BLT   Baltimore Terrapins     FL     None
## 3           BTT     Brooklyn Tip-Tops     FL     None
## 4           CHH      Chicago Chi-Feds     FL     None
## 5           COL      Colorado Rockies     NL        W
## 6           HOU    Houston Colt .45's     NL     None
## 7           KCP   Kansas City Packers     FL     None
## 8           MIL        Seattle Pilots     AL        W
## 9           NEW Indianapolis Hoosiers     FL     None
## 10          PBS     Pittsburgh Rebels     FL     None
## 11          SDP      San Diego Padres     NL        W
## 12          SEA      Seattle Mariners     AL        W
## 13          SLI    St. Louis Terriers     FL     None
```

```
## 14          TBD  Tampa Bay Devil Rays     AL        E
## 15          TEX   Washington Senators     AL     None
## 16          WSN        Montreal Expos     NL        E
```

There are about 16 franchises(with 24 team names) that have not won a world series from 1903 to 2013. The most notable teams are the Colorado Rockies, Houston Astros, Brewers, San Diego Padres, Rays, Texas Rangers and Washington Nationals.

Question 7

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

Using the Teams table, we restrict the World Series win field(WSWin) to yes('Y'). We retrieve the team name, league and division using the name, lgID, and divID fields in the SELECT statement. Since the formal world series started in 1903, we can restrict yearID to be greater than or equal to 1903. Use ORDER BY to order the world series winners by year. We condense output to 25 using LIMIT.

```
#Query the Teams table where WSWin = 'Y'
all_wseries_winners_q = "SELECT
                          yearID AS Year, name AS Team_Name, lgID AS League,
                          IFNULL(divID, 'None') AS Division, WSWin AS Won_Wseries
                         FROM Teams WHERE WSWin = 'Y' AND yearID >= 1903
                         ORDER BY yearID
                         LIMIT 25"
dbGetQuery(Baseball_database, all_wseries_winners_q)
```

```
##    Year              Team_Name League Division Won_Wseries
## 1  1903       Boston Americans     AL     None           Y
## 2  1905        New York Giants     NL     None           Y
## 3  1906      Chicago White Sox     AL     None           Y
## 4  1907           Chicago Cubs     NL     None           Y
## 5  1908           Chicago Cubs     NL     None           Y
## 6  1909     Pittsburgh Pirates     NL     None           Y
## 7  1910 Philadelphia Athletics     AL     None           Y
## 8  1911 Philadelphia Athletics     AL     None           Y
## 9  1912         Boston Red Sox     AL     None           Y
## 10 1913 Philadelphia Athletics     AL     None           Y
## 11 1914          Boston Braves     NL     None           Y
## 12 1915         Boston Red Sox     AL     None           Y
## 13 1916         Boston Red Sox     AL     None           Y
## 14 1917      Chicago White Sox     AL     None           Y
## 15 1918         Boston Red Sox     AL     None           Y
## 16 1919        Cincinnati Reds     NL     None           Y
## 17 1920      Cleveland Indians     AL     None           Y
## 18 1921        New York Giants     NL     None           Y
## 19 1922        New York Giants     NL     None           Y
## 20 1923       New York Yankees     AL     None           Y
## 21 1924    Washington Senators     AL     None           Y
## 22 1925     Pittsburgh Pirates     NL     None           Y
## 23 1926    St. Louis Cardinals     NL     None           Y
## 24 1927       New York Yankees     AL     None           Y
## 25 1928       New York Yankees     AL     None           Y
```

Question 8

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.

To query the correct output, we can join two sub-queries that retrieve the required outputs and, within those sub-queries, we can sub-query and use JOIN to further constrain the fields. First, we can sub-query the winning team's information in the FROM statement. We can call this subquery "win" and within "win" we obtain the yearID and teamIDwinner from the SeriesPost table with the playoff round field (round) constrained on World series(round = 'WS'). We also join the constrained SeriesPost table to the teams table(by YearID) with the World Series win field constrained on yes(WSWin= 'Y') to obtain the winning team's name, league, and division. Next, we join the win sub-query to the loss sub-query. In the loss sub-query, we obtain the losing team's losing year and teamID from the SeriesPost table with the playoff round field(round) set to World Series(round = 'WS'). We join this table to the teams(by YearID) that is constrained on no World Series win(WSWin = 'N') and a yes league win(LgWin = 'Y') to query the losing team's name, league, and division. The losing team's wins are acquired from the winning team's losses field in the SeriesPost table(losses). To get the full output, we join the win and loss queries on the year. Output is limited to 15 to save space in paper.

```
#Teams table,  and SeriesPost have the necessary info for this question
#Join the two tables:
#Order of join: SeriesPost, Teams table
winner_loser_wseries_q = "SELECT
                            win.*, loss.*
                          FROM
                          (SELECT
                            sp.yearID AS Year, sp.teamIDwinner AS WSWinner_ID,
                            t.name AS Winning_team,
                            t.lgID AS Winning_league, t.divID AS Winning_division
                          FROM
                          (SELECT * FROM SeriesPost WHERE round = 'WS') sp
                          JOIN (SELECT * FROM teams WHERE WSWin= 'Y') t
                          ON sp.yearID = t.yearID
                          ORDER BY sp.yearID) AS win
                          JOIN (
                          SELECT
                            s.yearID AS Year, s.teamIDloser AS WSLoser_ID,
                            te.name AS Losing_team,
                            te.lgID AS WSLosing_league, te.divID AS WSLosing_division,
                            s.losses AS Losing_Team_Wins
                          FROM
                          (SELECT * FROM SeriesPost WHERE round = 'WS') s
                          JOIN (SELECT * FROM teams WHERE WSWin = 'N' AND LgWin = 'Y') te
                          ON s.yearID = te.yearID
                          ORDER BY s.yearID) AS loss
                          ON win.Year = loss.Year
                          WHERE win.Year >= 1903
                          ORDER BY win.Year
                          LIMIT 15
                          "
dbGetQuery(Baseball_database, winner_loser_wseries_q)
```

```
##    Year WSWinner_ID           Winning_team Winning_league Winning_division Year
```

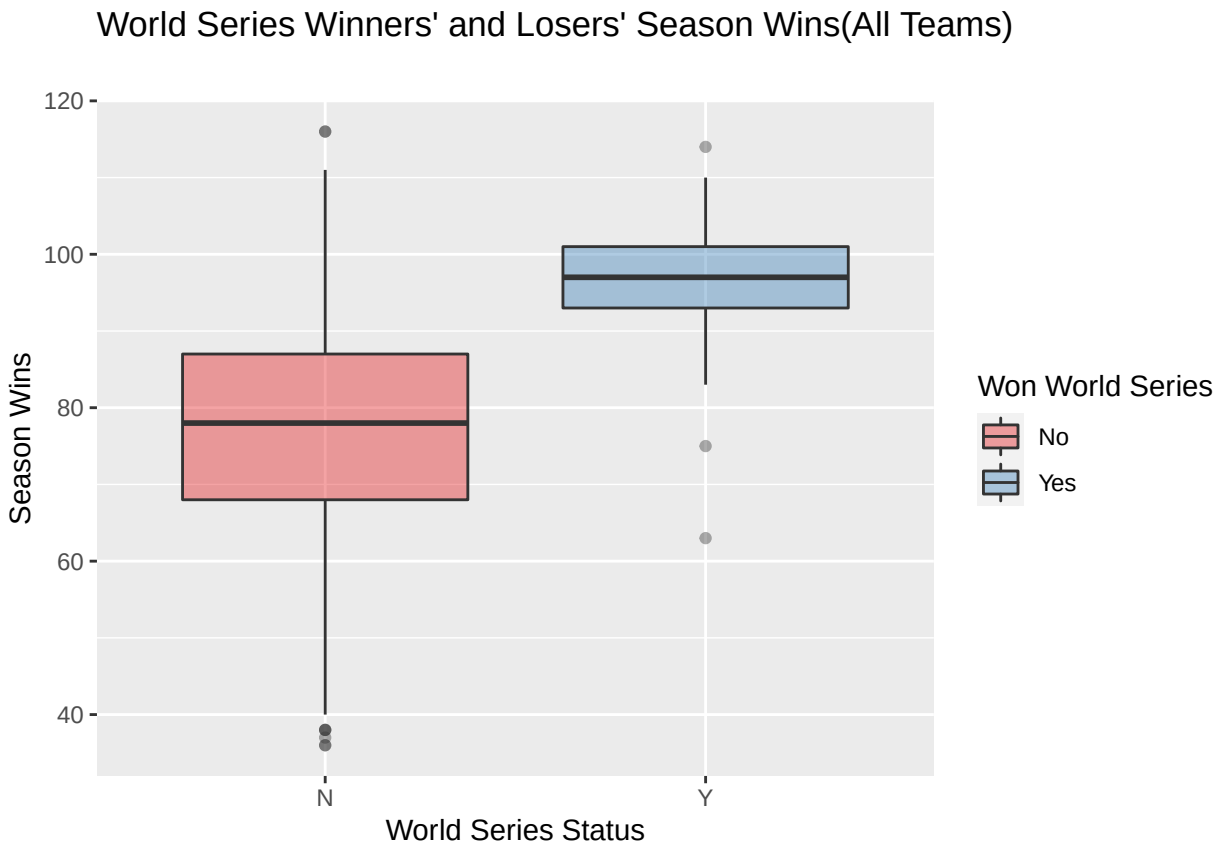
```
## 1  1903         BOS       Boston Americans             AL             <NA> 1903
## 2  1905         NY1        New York Giants             NL             <NA> 1905
## 3  1906         CHA      Chicago White Sox             AL             <NA> 1906
## 4  1907         CHN           Chicago Cubs             NL             <NA> 1907
## 5  1908         CHN           Chicago Cubs             NL             <NA> 1908
## 6  1909         PIT     Pittsburgh Pirates             NL             <NA> 1909
## 7  1910         PHA Philadelphia Athletics             AL             <NA> 1910
## 8  1911         PHA Philadelphia Athletics             AL             <NA> 1911
## 9  1912         BOS         Boston Red Sox             AL             <NA> 1912
## 10 1913         PHA Philadelphia Athletics             AL             <NA> 1913
## 11 1914         BSN          Boston Braves             NL             <NA> 1914
## 12 1915         BOS         Boston Red Sox             AL             <NA> 1915
## 13 1916         BOS         Boston Red Sox             AL             <NA> 1916
## 14 1917         CHA      Chicago White Sox             AL             <NA> 1917
## 15 1918         BOS         Boston Red Sox             AL             <NA> 1918
##    WSLoser_ID            Losing_team WSLosing_league WSLosing_division
## 1         PIT     Pittsburgh Pirates              NL              <NA>
## 2         PHA Philadelphia Athletics              AL              <NA>
## 3         CHN           Chicago Cubs              NL              <NA>
## 4         DET         Detroit Tigers              AL              <NA>
## 5         DET         Detroit Tigers              AL              <NA>
## 6         DET         Detroit Tigers              AL              <NA>
## 7         CHN           Chicago Cubs              NL              <NA>
## 8         NY1         New York Giants             NL              <NA>
## 9         NY1         New York Giants             NL              <NA>
## 10        NY1         New York Giants             NL              <NA>
## 11        PHA Philadelphia Athletics              AL              <NA>
## 12        PHI  Philadelphia Phillies              NL              <NA>
## 13        BRO         Brooklyn Robins             NL              <NA>
## 14        NY1         New York Giants             NL              <NA>
## 15        CHN           Chicago Cubs              NL              <NA>
##    Losing_Team_Wins
## 1                 3
## 2                 1
## 3                 2
## 4                 0
## 5                 1
## 6                 3
## 7                 1
## 8                 2
## 9                 3
## 10                1
## 11                0
## 12                1
## 13                1
## 14                2
## 15                2
```

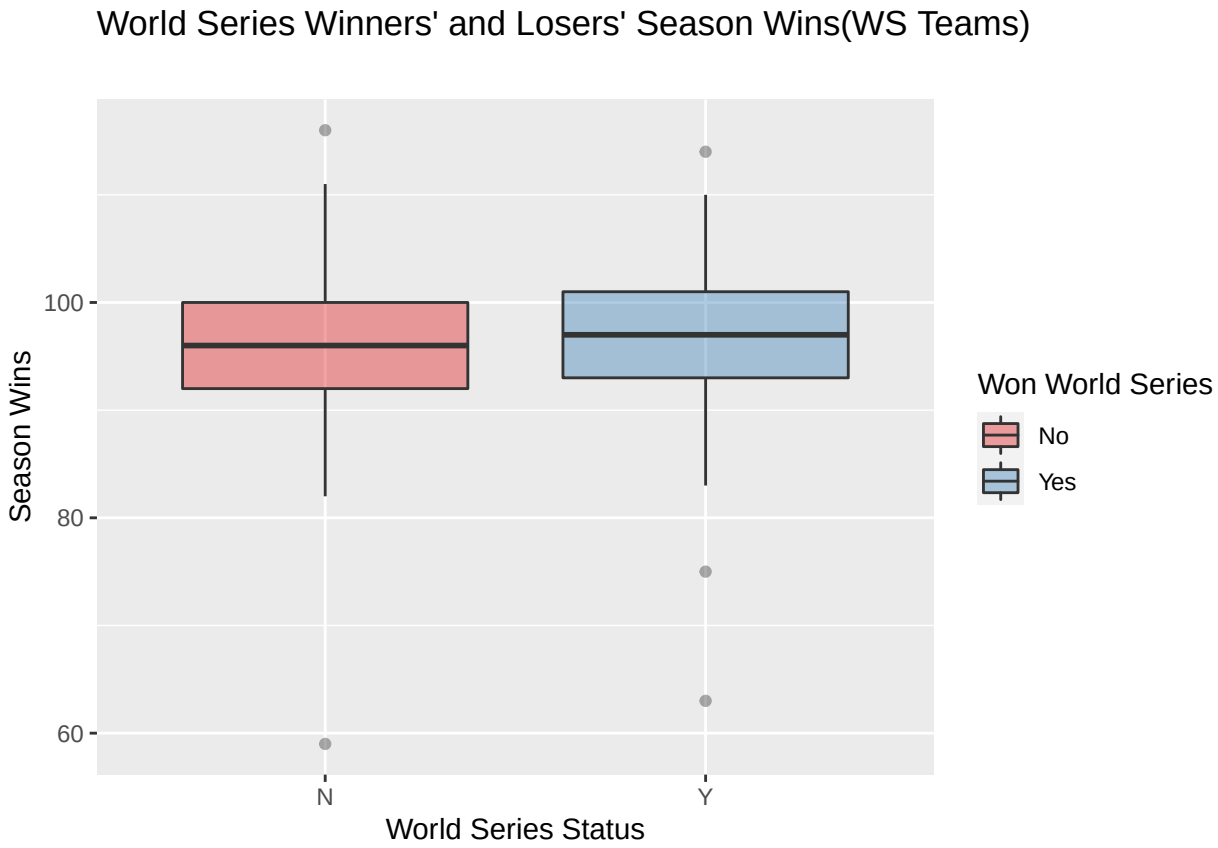
Question 9

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

We can use the win(W) and World Series win(WSWin) fields in the team table to get the games won in a season and whether the team won the World Series or did not win the World Series. Some values in the WSWin field are null and, as a result, I used IFNULL function to indicate that they did not win. Since the formal World Series started in 1903, I also constrained yearID to include all years after 1902. First, we can interpret losing the World Series as teams that loss the in World Series and teams that did not make the World Series. Then, we can also compare the wins for teams that make it to the World Series.

```
#Teams table: look at wins and world series data
#First modern world series occurred in 1903
wins_wseries_q = "SELECT
                    W AS Wins, IFNULL(WSWin, 'N') AS World_series_status
                  FROM Teams
                  WHERE yearID > '1902'"
wins_data = dbGetQuery(Baseball_database, wins_wseries_q)

wins_wseries_q2 = "SELECT
                    W AS Wins, IFNULL(WSWin, 'N') AS World_series_status
                  FROM Teams
                  WHERE yearID > '1902' AND LgWin = 'Y'"
wins_data2 = dbGetQuery(Baseball_database, wins_wseries_q2)
```





In the first plot above(All teams), there is a difference between the two data distributions and this suggests that the World Series winners have, in general, more wins than teams that did not win the world series or did not make it to the world series. However, the second plot(World Series teams only) shows there is no apparent difference between the number of season wins for teams that won the world series and teams that loss the world series. For all teams, we can assume that there may be a relationship between the number of games won in a season and making it to the World Series(with the possibility of winning). For teams that make it to the world series, we can also assume that there does not appear to be a relationship between season wins and winning the World Series.

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?

We can JOIN the Master, Salaries, Fielding, and Teams tables to query the players with the top three highest salaries and their details. First, we use FROM to query the players' first and last names. Next, we JOIN the MASTER table to the salaries table ON playerID. This connects the players' names to their salaries. We also JOIN the salaries table to the Fielding table on playerID and yearID. We join on playerID and yearID to get the players' position in 2003. Then, we JOIN the Fielding table to the Teams table ON yearID and teamID to get the players' 2003 team. To further structure the query output, we GROUP BY the playerID just in case there are repeated salary information(Contract switch, etc.) and use MAX on the Salaries table's salary column. This ensures we are getting the players' max salary. To get the correct player position, we use HAVING to specify that we want the number of games played to be the max(assuming that the player's position is where they played the most games at). Lastly, we ORDER BY the salaries in descending order(DESC) and LIMIT the output to 3(top 3 salaries).

```
#Join the MASTER, Salaries, Fielding and teams tables:
high_salaries_q = "SELECT
                    s.yearID AS Year, m.playerID AS ID,
                    m.nameFirst AS First_name, m.nameLast AS Last_name,
                    f.Pos AS Position, t.name AS Team_name,
                    MAX(s.salary) AS Top_Salaries
                   FROM MASTER m
                   JOIN Salaries s
                   ON m.playerID = s.playerID
                   JOIN Fielding f
                   ON s.playerID = f.playerID AND s.yearID = f.yearID
                   JOIN Teams t
                   ON f.yearID = t.yearID AND f.teamID = t.teamID
                   WHERE s.yearID = '2003'
                   GROUP BY m.playerID
                   HAVING f.G = max(f.G)
                   ORDER BY Top_Salaries DESC
                   LIMIT 3"
dbGetQuery(Baseball_database, high_salaries_q)
```

```
##   Year        ID First_name Last_name Position         Team_name Top_Salaries
## 1 2003 rodrial01       Alex Rodriguez       SS     Texas Rangers     22000000
## 2 2003 ramirma02      Manny   Ramirez       LF    Boston Red Sox     20000000
## 3 2003 delgaca01     Carlos   Delgado       1B Toronto Blue Jays     18700000
```

In 2003, the top 3 highest paid players were Alex Rodriguez($22 million), Manny Ramirez($20 milli0n), and Carlos Delgado($18.7 million).

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.

We JOIN the Salaries table to the teams table to get the team names and their payroll information. When we JOIN the tables, we can sub-query the Teams table by restricting the yearID to include years after 1985(Earliest recorded salary year). We JOIN the two tables on teamID and yearID to make sure the players' salaries match with the year and team. Using the WHERE statement and playerID, we can restrict the output to only include 2010 data. We GROUP BY the teamID to group the teams' salary information and use sum to add the players' salaries. To compute the payrolls for all years, we take a similar approach as the steps mentioned before. However, we add yearID to the GROUP BY statement to group the teams and salary information into the correct years.

```
#Total payroll sum of all player and manager salaries on the team:
#First, check the year where the salary data starts
payroll_2010_q = "SELECT
                    MIN(yearID)
                  FROM Salaries
                  "
dbGetQuery(Baseball_database, payroll_2010_q) #1985 is the earliest recording
```

```
##   MIN(yearID)
## 1        1985
```

```
#For 2010:
payroll_2010_q = "SELECT
                    s.teamID AS Team_ID, t.name AS Team_Name,
                    SUM(s.salary) AS Total_Payroll
                  FROM Salaries s
                  JOIN (SELECT yearID, teamID, name FROM Teams WHERE yearID >= '1985') t
                  ON s.teamID = t.teamID AND s.yearID = t.yearID
                  WHERE s.yearID = 2010
                  GROUP BY s.teamID"
dbGetQuery(Baseball_database, payroll_2010_q)
```

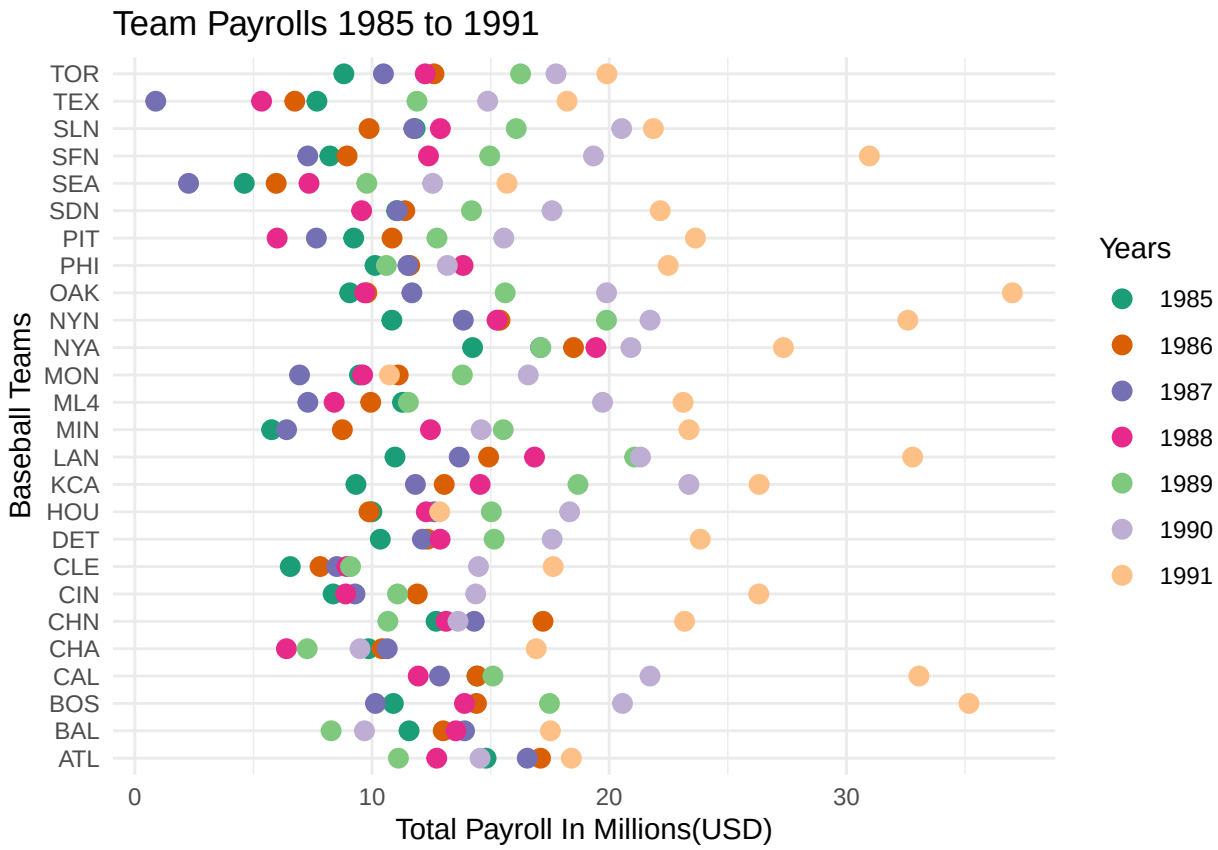
```
##    Team_ID                     Team_Name Total_Payroll
## 1      ARI          Arizona Diamondbacks      60718166
## 2      ATL                Atlanta Braves      84423666
## 3      BAL             Baltimore Orioles      81612500
## 4      BOS                Boston Red Sox     162447333
## 5      CHA             Chicago White Sox     105530000
## 6      CHN                  Chicago Cubs     146609000
## 7      CIN               Cincinnati Reds      71761542
## 8      CLE             Cleveland Indians      61203966
## 9      COL              Colorado Rockies      84227000
## 10     DET                Detroit Tigers     122864928
## 11     FLO               Florida Marlins      57029719
## 12     HOU                Houston Astros      92355500
## 13     KCA            Kansas City Royals      71405210
## 14     LAA Los Angeles Angels of Anaheim     104963866
## 15     LAN           Los Angeles Dodgers      95358016
## 16     MIL             Milwaukee Brewers      81108278
## 17     MIN               Minnesota Twins      97559166
## 18     NYA              New York Yankees     206333389
## 19     NYN                 New York Mets     134422942
## 20     OAK             Oakland Athletics      55254900
## 21     PHI         Philadelphia Phillies     141928379
## 22     PIT            Pittsburgh Pirates      34943000
## 23     SDN              San Diego Padres      37799300
## 24     SEA              Seattle Mariners      86510000
## 25     SFN          San Francisco Giants      98641333
## 26     SLN           St. Louis Cardinals      93540751
## 27     TBA                Tampa Bay Rays      71923471
## 28     TEX                 Texas Rangers      55250544
## 29     TOR            Toronto Blue Jays      62234000
## 30     WAS          Washington Nationals      61400000
```

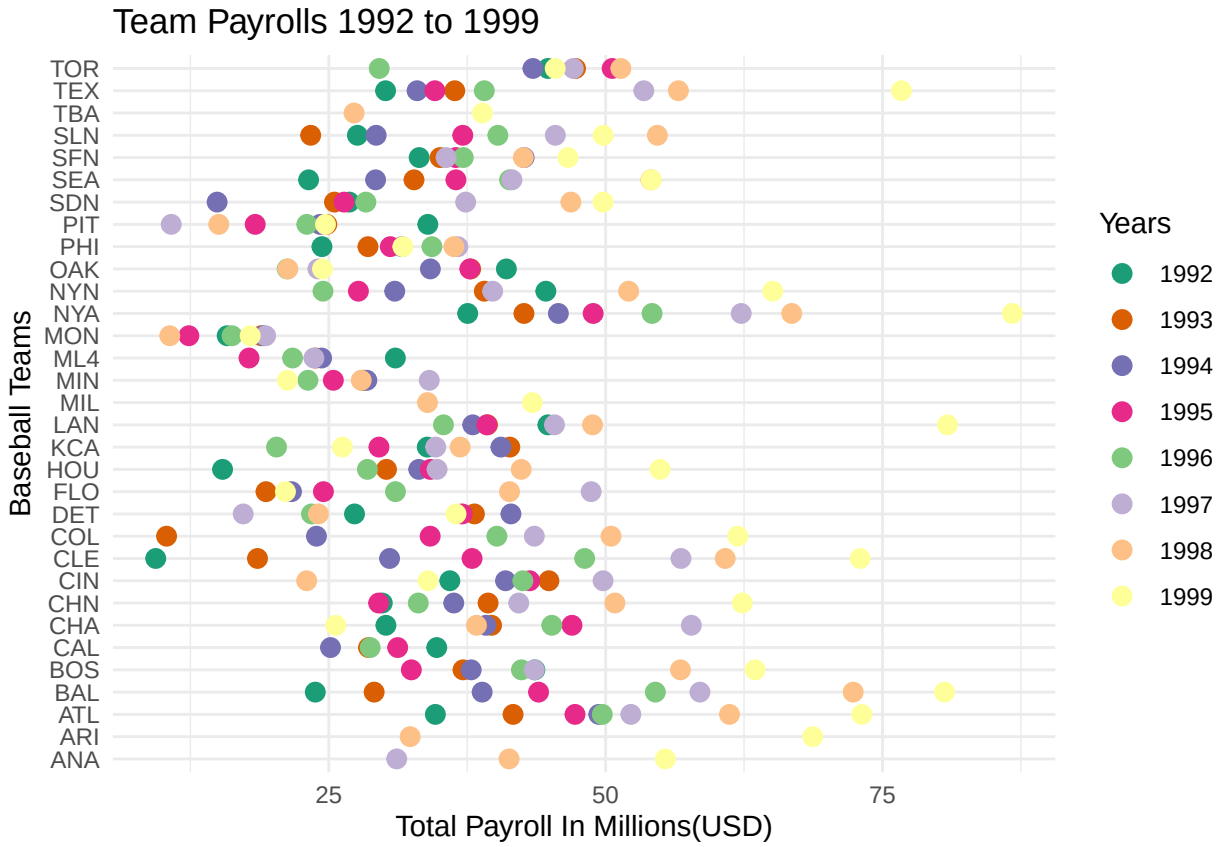
```
#For all the years:
payroll_all_q = "SELECT
                   s.yearID AS Year, s.teamID AS Team_ID, t.name AS Team_Name,
                  SUM(s.salary) AS Total_Payroll
                  FROM Salaries s
                  JOIN (SELECT yearID, teamID, name FROM Teams WHERE yearID >= '1985') t
                  ON s.teamID = t.teamID AND s.yearID = t.yearID
                  GROUP BY s.teamID, s.yearID"
payroll_all_data = dbGetQuery(Baseball_database, payroll_all_q)
```

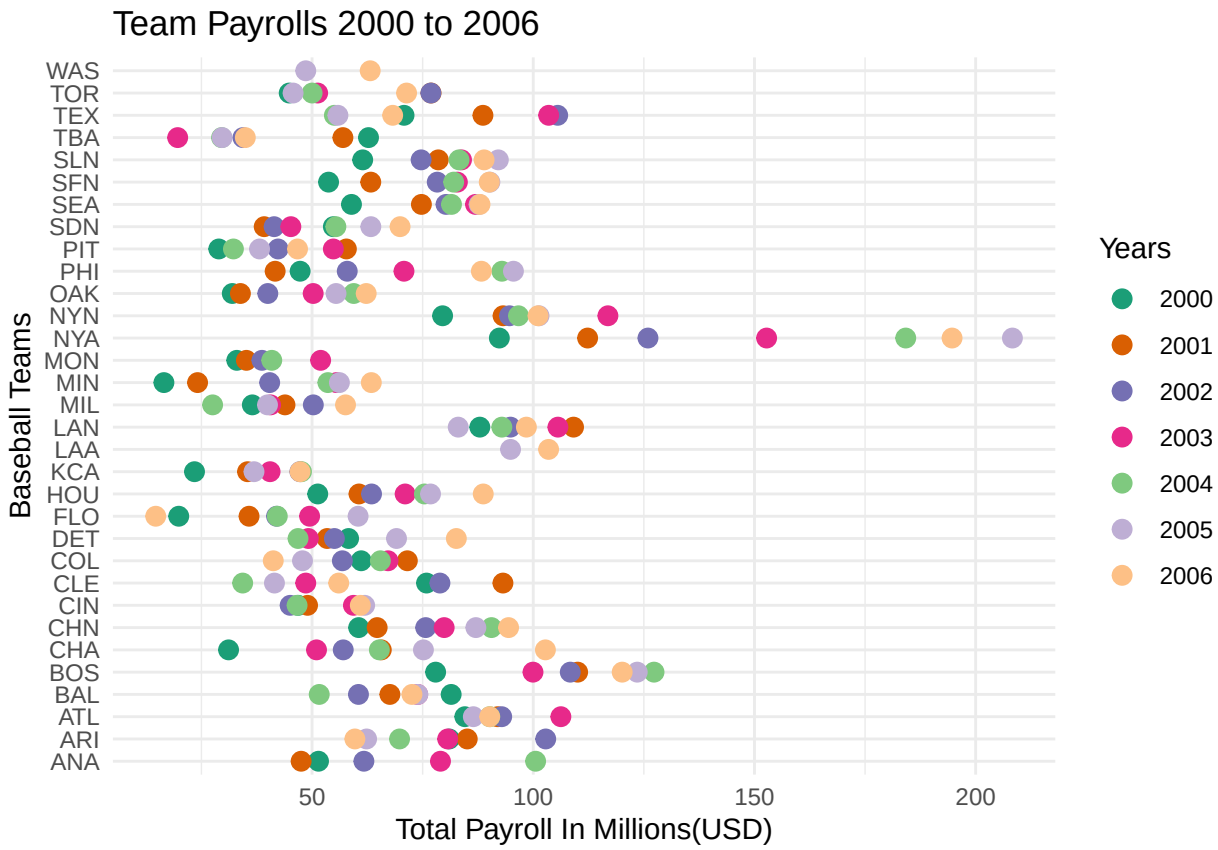
```
#For a teamID identification table:
Team_ident_q = "SELECT
                 teamID AS Team_ID, name AS Team_Name
                FROM Teams
                WHERE yearID >= '1985'
                GROUP BY teamID
                "
Team_ident_data = dbGetQuery(Baseball_database, Team_ident_q)
```

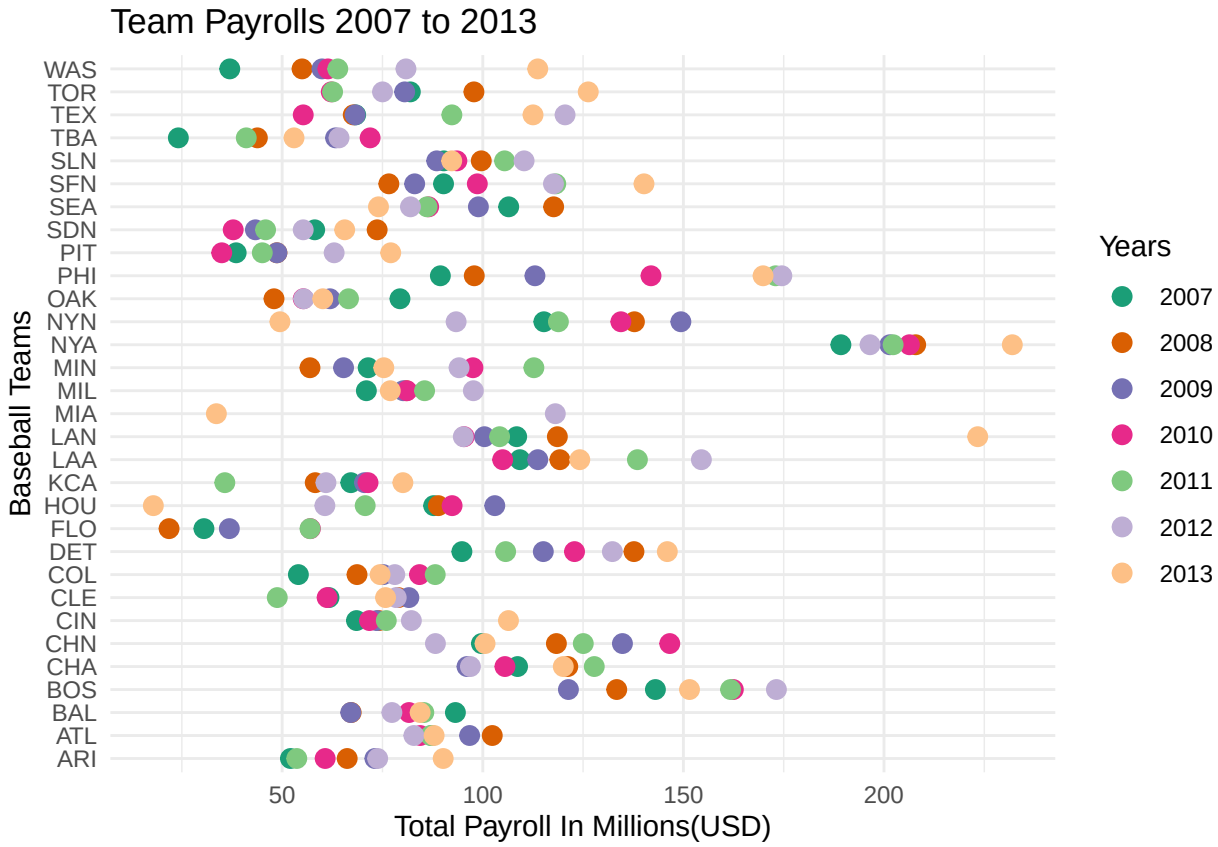
To make the plots more organized, we can use the teamID to identify teams. We can created a teamID identification table to show the full team names. In the SQL query, we use the Teams table and GROUP BY teamID .

Team_ID	Team_Name
ANA	Anaheim Angels
ARI	Arizona Diamondbacks
ATL	Atlanta Braves
BAL	Baltimore Orioles
BOS	Boston Red Sox
CAL	California Angels
CHA	Chicago White Sox
CHN	Chicago Cubs
CIN	Cincinnati Reds
CLE	Cleveland Indians
COL	Colorado Rockies
DET	Detroit Tigers
FLO	Florida Marlins
HOU	Houston Astros
KCA	Kansas City Royals
LAA	Los Angeles Angels of Anaheim
LAN	Los Angeles Dodgers
MIA	Miami Marlins
MIL	Milwaukee Brewers
MIN	Minnesota Twins
ML4	Milwaukee Brewers
MON	Montreal Expos
NYA	New York Yankees
NYN	New York Mets
OAK	Oakland Athletics
PHI	Philadelphia Phillies
PIT	Pittsburgh Pirates
SDN	San Diego Padres
SEA	Seattle Mariners
SFN	San Francisco Giants
SLN	St. Louis Cardinals
TBA	Tampa Bay Devil Rays
TEX	Texas Rangers
TOR	Toronto Blue Jays
WAS	Washington Nationals


Team Payrolls 1985 to 1991
Baseball Teams
TOR
TEX
SLN
SFN
SEA
SDN
PIT
PHI
OAK
NYN
NYA
MON
ML4
MIN
LAN
KCA
HOU
DET
CLE
CIN
CHN
CHA
CAL
BOS
BAL
ATL
0
10
20
30
Total Payroll In Millions(USD)
Years
1985
1986
1987
1988
1989
1990
1991


Team Payrolls 1992 to 1999
Baseball Teams
TOR
TEX
TBA
SLN
SFN
SEA
SDN
PIT
PHI
OAK
NYN
NYA
MON
ML4
MIN
MIL
LAN
KCA
HOU
FLO
DET
COL
CLE
CIN
CHN
CHA
CAL
BOS
BAL
ATL
ARI
ANA
25
50
75
Total Payroll In Millions(USD)
Years
1992
1993
1994
1995
1996
1997
1998
1999


Team Payrolls 2000 to 2006
Baseball Teams
WAS
TOR
TEX
TBA
SLN
SFN
SEA
SDN
PIT
PHI
OAK
NYN
NYA
MON
MIN
MIL
LAN
LAA
KCA
HOU
FLO
DET
COL
CLE
CIN
CHN
CHA
BOS
BAL
ATL
ARI
ANA
50
100
150
200
Total Payroll In Millions(USD)
Years
2000
2001
2002
2003
2004
2005
2006

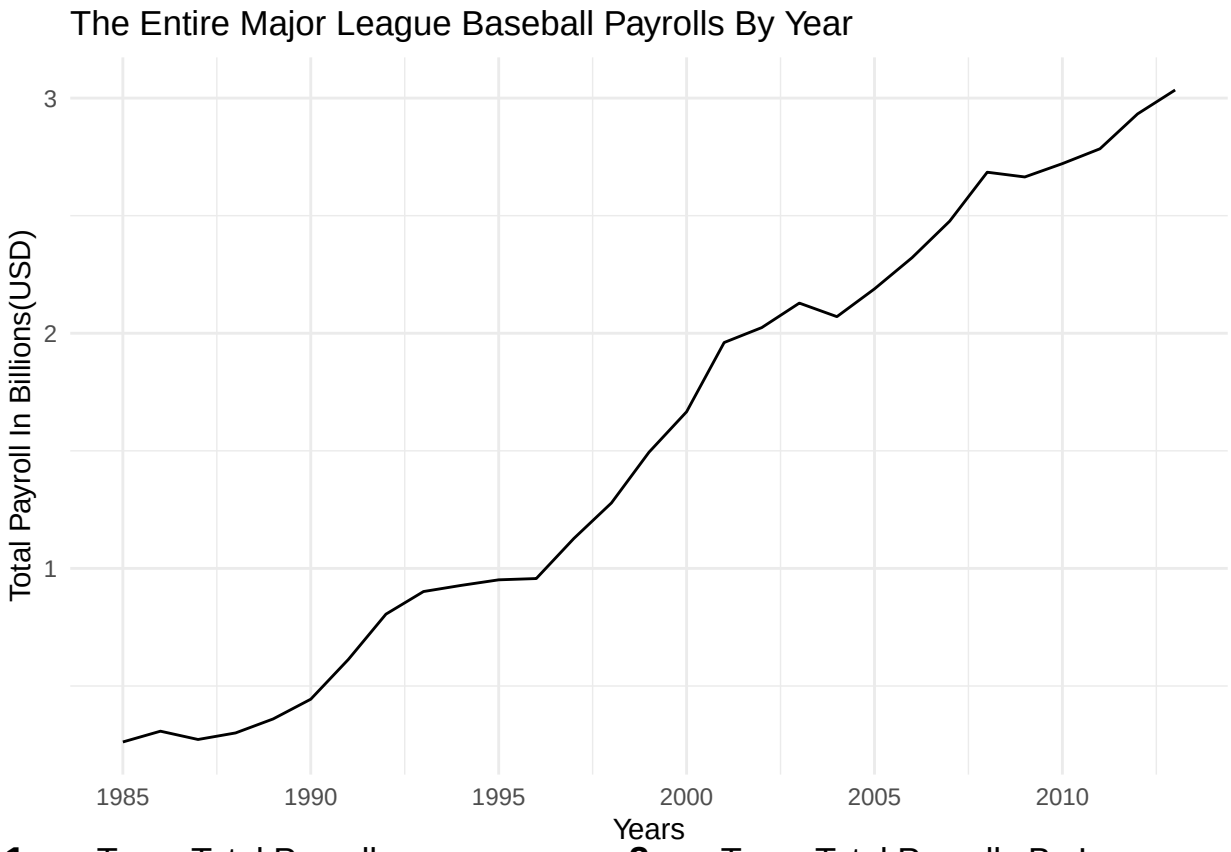

Team Payrolls 2007 to 2013
Baseball Teams
WAS
TOR
TEX
TBA
SLN
SFN
SEA
SDN
PIT
PHI
OAK
NYN
NYA
MIN
MIL
MIA
LAN
LAA
KCA
HOU
FLO
DET
COL
CLE
CIN
CHN
CHA
BOS
BAL
ATL
ARI
50
100
150
200
Total Payroll In Millions(USD)
Years
2007
2008
2009
2010
2011
2012
2013

Question 12

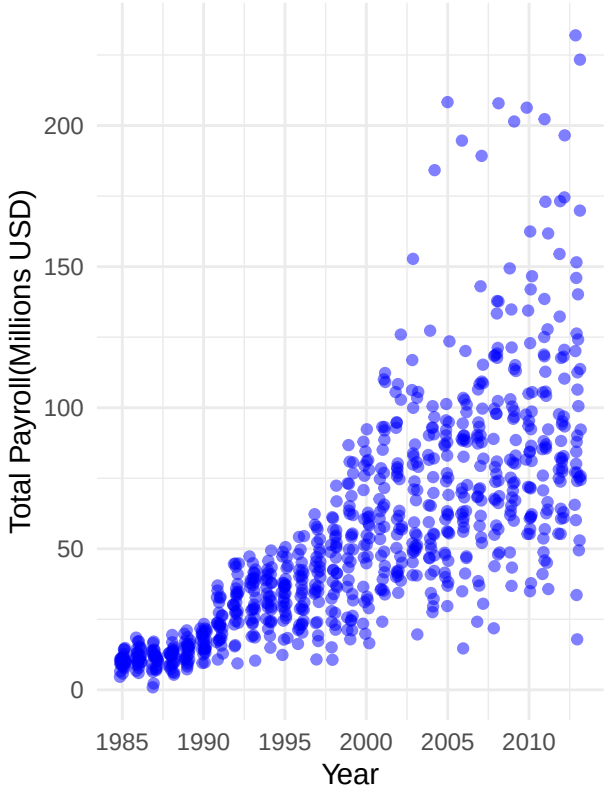
Explore the change in salary over time. Use a plot. Identify the teams that won the world series or league on the plot. How does salary relate to winning the league and/or world series.

For this query, we can use the same or similar statement from the query on question 11. We add the league ID(lgID), division ID(divID), league win(LgWin), and World Series win(WSWin) fields to the output. Some values for LgWin and WSWin may be NULL. Therefore, we use IFNULL to set NULL values from the LgWin and WSWin fields to no('N').

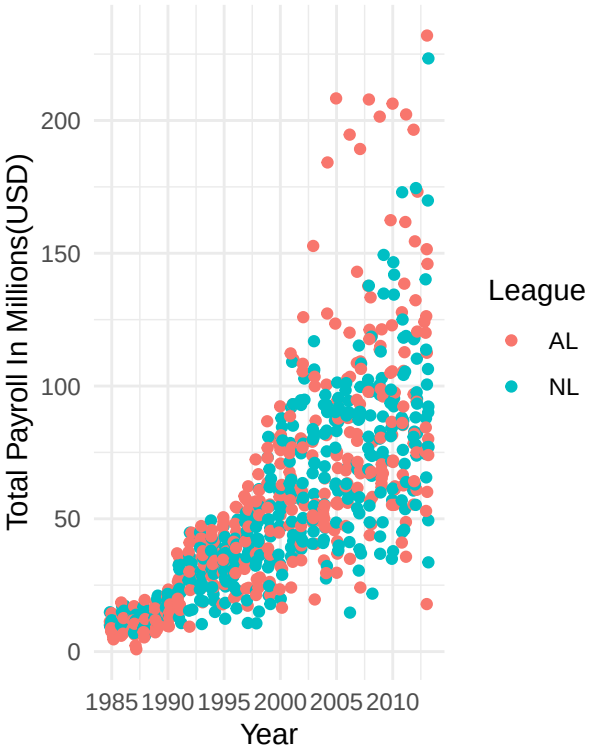
```
#We need to add LgWin, and WSWin to data structure
payroll_wins_q = "SELECT
                    s.yearID AS Year, s.teamID AS Team_ID, t.name AS Team_Name,
                    t.lgID AS League, t.divID AS Division,
                    IFNULL(t.LgWin, 'N') AS Won_League,
                    IFNULL(t.WSWin, 'N') AS Won_Wseries,
                    SUM(s.salary) AS Total_Payroll
                  FROM Salaries s
                  JOIN (SELECT yearID, teamID, lgID, divID, name, LgWin, WSWin
                  FROM Teams
                  WHERE yearID >= '1985') t
                  ON s.teamID = t.teamID AND s.yearID = t.yearID
                  GROUP BY s.teamID, s.yearID"
payroll_Wins_data = dbGetQuery(Baseball_database, payroll_wins_q)

Total_payrollt_q = "SELECT
                    yearID AS Year, SUM(salary) AS Total_Payroll
                  FROM Salaries
                  GROUP BY yearID"
Total_payroll_data = dbGetQuery(Baseball_database, Total_payrollt_q)
```

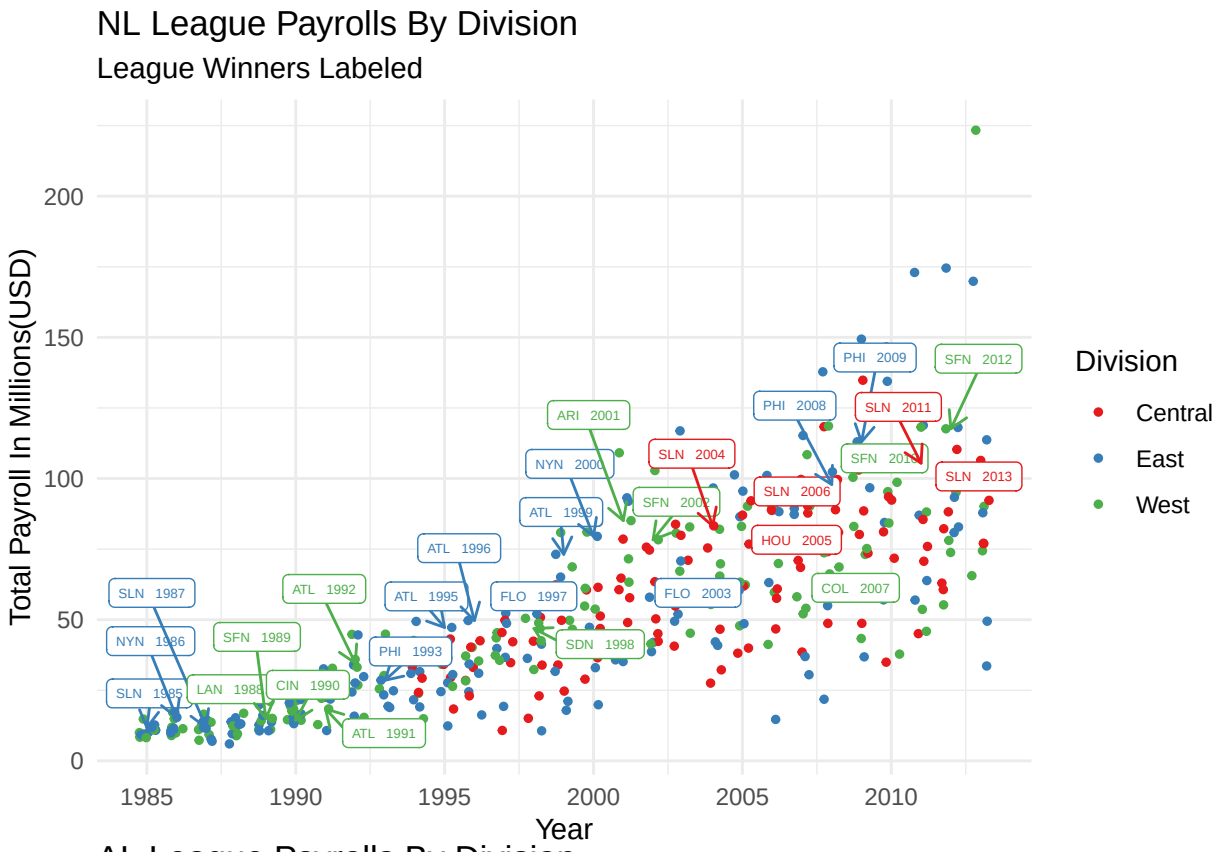


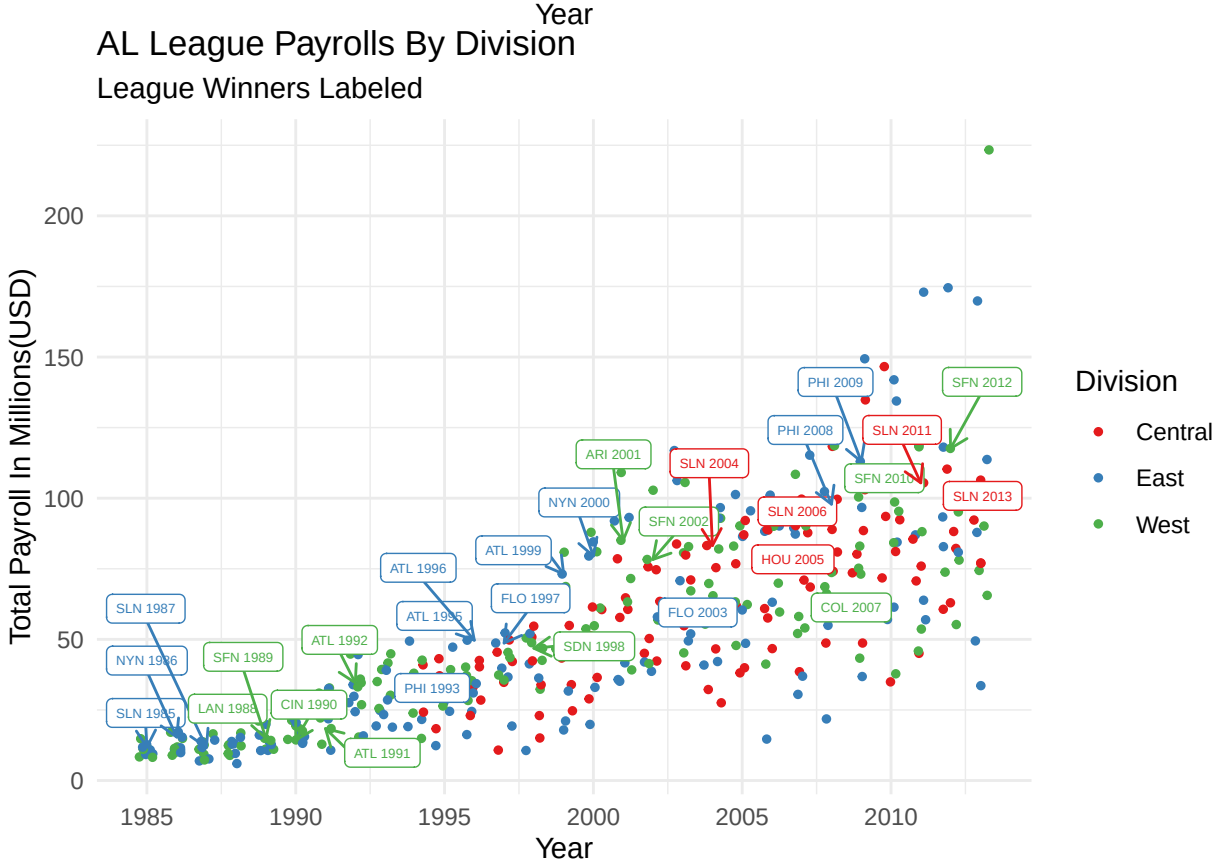
1 Team Total Payrolls

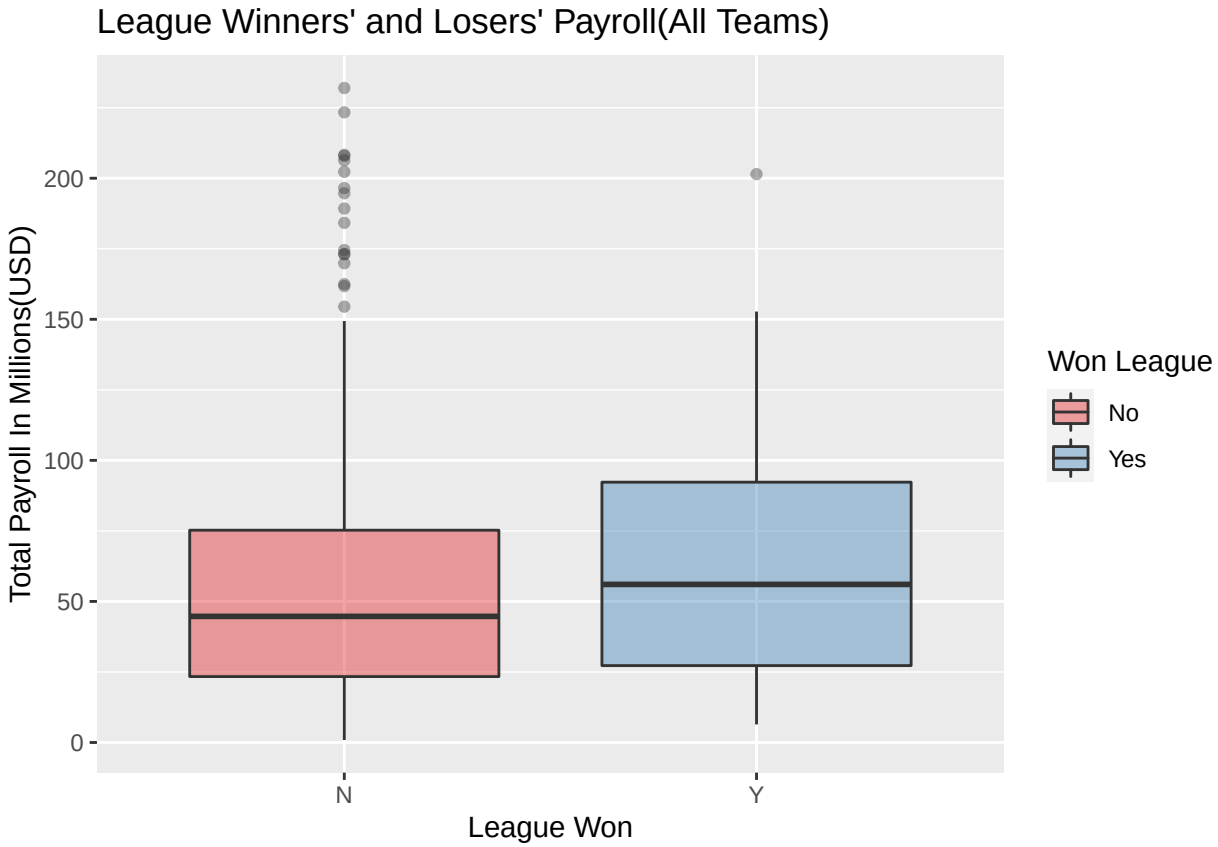


2 Team Total Payrolls By League




NL League Payrolls By Division
League Winners Labeled
Total Payroll In Millions(USD)
Year
Division
Central
East
West
SLN 1987
NYN 1986
SLN 1985
SFN 1989
LAN 1988
CIN 1990
ATL 1992
ATL 1991
ATL 1996
ATL 1995
PHI 1993
FLO 1997
SDN 1998
ARI 2001
NYN 2000
ATL 1999
SFN 2002
SLN 2004
FLO 2003
SLN 2006
HOU 2005
PHI 2008
COL 2007
PHI 2009
SLN 2011
SFN 2012
SLN 2013


AL League Payrolls By Division
League Winners Labeled
Total Payroll In Millions(USD)
Year
Division
Central
East
West
SLN 1987
NYN 1986
SLN 1985
SFN 1989
LAN 1988
CIN 1990
ATL 1992
ATL 1991
ATL 1996
ATL 1995
PHI 1993
FLO 1997
SDN 1998
ARI 2001
NYN 2000
ATL 1999
SFN 2002
SLN 2004
FLO 2003
SLN 2006
HOU 2005
PHI 2008
COL 2007
PHI 2009
SLN 2011
SFN 2012
SLN 2013



From the plots, we see that teams' total salary has increased over time. In the League Winners' and Losers' Payroll(All Teams) boxplot, there doesn't seem to be a difference in total payroll for teams that win their league and teams that don't win their league. Therefore, team payroll may not be a great indicator for the team winning their league.

Question 14

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

We want to find the player with the most career home runs. To query the information about the player with the most home runs, we JOIN the Batting and Master table on playerID. Next, we want to GROUP BY the playerID and SUM the number of home runs(HR field in Batting) to get the total home runs. We ORDER the output by home runs in descending order(DESC) and LIMIT the output to 1. To get the number of home runs per year, we add a WHERE statement to the previous steps and remove the SUM on the home runs field. In the WHERE statement, we add a subquery to restrict playerID to the player with the most home runs. In the subquery, we use the Batting table and GROUP BY playerID to get the single playerID. With the GROUP BY statement, we use HAVING to restrict the SUM of home runs(HR) to be equal to the max total number of home runs. To get the max number of homeruns, we use a subquery to sum the player homeruns in Batting, order the sum in descending order, and LIMITING the number output to one. Lastly, we ORDER BY yearID to put the output in chronological order.

```
#Join the Master and Batting tables
#First, we can get the player with the most homeruns:
most_homers_q = "SELECT
                    b.playerID AS ID,
                    m.nameFirst AS First_name,
```

```
                    m.nameLast AS Last_name,
                    SUM(b.HR) AS Homeruns
                  FROM Batting b
                  JOIN Master m
                  ON b.playerID = m.playerID
                  GROUP BY b.playerID
                  ORDER BY Homeruns DESC
                  LIMIT 1
                  "
dbGetQuery(Baseball_database, most_homers_q)
```

```
##          ID First_name Last_name Homeruns
## 1 bondsba01      Barry     Bonds      762
```

```
#Number per year:
most_homers_q = "SELECT
                    b.yearID AS Year,
                    m.nameFirst || ' ' || m.nameLast AS Name,
                    b.HR AS Homeruns
                  FROM Batting b
                  JOIN Master m
                  ON b.playerID = m.playerID
                  WHERE b.playerID IN (
                    SELECT
                      playerID AS ID
                    FROM Batting
                    GROUP BY playerID
                    HAVING SUM(HR) =
                      (SELECT
                        SUM(HR) AS Homeruns
                      FROM Batting
                      GROUP BY playerID
                      ORDER BY Homeruns DESC LIMIT 1))
                  ORDER BY b.yearID
                  "
dbGetQuery(Baseball_database, most_homers_q)
```

```
##    Year        Name Homeruns
## 1  1986 Barry Bonds       16
## 2  1987 Barry Bonds       25
## 3  1988 Barry Bonds       24
## 4  1989 Barry Bonds       19
## 5  1990 Barry Bonds       33
## 6  1991 Barry Bonds       25
## 7  1992 Barry Bonds       34
## 8  1993 Barry Bonds       46
## 9  1994 Barry Bonds       37
## 10 1995 Barry Bonds       33
## 11 1996 Barry Bonds       42
## 12 1997 Barry Bonds       40
## 13 1998 Barry Bonds       37
## 14 1999 Barry Bonds       34
## 15 2000 Barry Bonds       49
```

```
## 16 2001 Barry Bonds       73
## 17 2002 Barry Bonds       46
## 18 2003 Barry Bonds       45
## 19 2004 Barry Bonds       45
## 20 2005 Barry Bonds        5
## 21 2006 Barry Bonds       26
## 22 2007 Barry Bonds       28
```

Barry Bonds has the most homeruns.

Code Appendix

```
knitr::opts_chunk$set(echo = TRUE)
#Libraries:
library(knitr)
library(RSQLite)
library(DBI)
library(ggplot2)
library(dplyr)
library(reshape2)
library(ggrepel)
library(RColorBrewer)
library(ggpubr)

#Connect to the database:
Baseball_database = dbConnect(SQLite(), "lahman2013.sqlite")

#Check for tables:
dbListTables(Baseball_database)
#To check the years, I will check the yearIDs for the Batting and Pitching tables
#If my assumption is correct, there will be batting and pitching info for each year
dbListFields(Baseball_database, "Batting")
dbListFields(Baseball_database, "Pitching") #We want to use yearID
#First, we can check the format
q_year_check = "SELECT
                  yearID
                FROM Batting
                LIMIT 5"
dbGetQuery(Baseball_database, q_year_check)
#The YearID contains the actual year and no month or day information
#First, we can query the covered dates:
q_year_span="SELECT
              MIN(yearID) AS Begin_year,
              MAX(yearID) AS End_year
             FROM Batting"
dbGetQuery(Baseball_database, q_year_span)
#The data covers from 1871 to 2013
#Check to confirm span:
q_year_span2 = "SELECT
                  MIN(yearID) AS Begin_year,
                  MAX(yearID) AS End_year
                FROM Pitching"
```

```
dbGetQuery(Baseball_database, q_year_span2)

q_year_span3= "SELECT
                MIN(yearID) AS Begin_year,
                MAX(yearID) AS End_year
               FROM Teams"
dbGetQuery(Baseball_database, q_year_span3)
#Year spans match!
#Now, we can check to see if there are data for each of these years:
#For this we can look to see the total of unique years
#If there is data for all years, the total
#unique years should match the difference between
#the 2013 and 1871 plus 1(143)
each_year_q = "SELECT
                COUNT(DISTINCT(yearID)) AS Total_years
               FROM Batting"
dbGetQuery(Baseball_database, each_year_q)
#It appears that all years are listed in the data
#Check result:
each_year_q2 = "SELECT
                  COUNT(DISTINCT(yearID))
                FROM Pitching"
dbGetQuery(Baseball_database, each_year_q2)
each_year_q2 = "SELECT
                  COUNT(DISTINCT(yearID))
                FROM Teams"
dbGetQuery(Baseball_database, each_year_q2)
#Check resulted in the same number of years
#Number of Unique people in the database:
#We can use the master table that contains the unique code for players
person_count_q = "SELECT
                    COUNT(DISTINCT(playerID)) AS Num_unique_persons
                  FROM MASTER"
dbGetQuery(Baseball_database, person_count_q)

#Number of managers(Includes managers that managed the team and also managed-played):
manager_count_q = "SELECT
                    COUNT(DISTINCT(playerID)) AS Num_Managers
                   FROM Managers"
dbGetQuery(Baseball_database, manager_count_q)

#Number of managers that only managed
#(plyrMgr is set to 'N', does not account for if they managed and played in another season):
manager_count2_q = "SELECT
                      COUNT(DISTINCT(playerID)) AS Num_Managers
                    FROM Managers
                    WHERE plyrMgr = 'N'"
dbGetQuery(Baseball_database, manager_count2_q)

#Number of players(That played baseball and did not manage):
player_count_q = "SELECT
                    COUNT(DISTINCT(playerID)) AS Num_players
                  FROM MASTER
```

```
                  WHERE playerID NOT IN (SELECT playerID FROM Managers)"
dbGetQuery(Baseball_database, player_count_q)

#Number of players(That played baseball and may have managed at the same time):
player_count2_q = "SELECT
                   COUNT(DISTINCT(playerID)) AS Num_players
                  FROM MASTER
                  WHERE playerID NOT IN (
                  SELECT
                     DISTINCT(playerID)
                  FROM Managers WHERE plyrMgr = 'N')"
dbGetQuery(Baseball_database, player_count2_q)

#Some test numbers did not seem to add up correctly in previous section:
#A check here
player_check_q = "SELECT
                    COUNT(DISTINCT(M.playerID)) AS Num_players
                  FROM MASTER M
                  JOIN Managers Ma
                  ON M.playerID = Ma.playerID"
dbGetQuery(Baseball_database, player_check_q)
#It appears that there are 3 player IDs
#from Managers that do not appear on the Master Table
missing_managers_q = "SELECT
                        playerID AS Miss_managers
                      FROM Managers
                      WHERE playerID NOT IN (SELECT DISTINCT(playerID) FROM MASTER)"
dbGetQuery(Baseball_database, missing_managers_q)
#cammebi99, hengled99, and keanejo99
#are missing from the master list(Seems that they were managers only)
missing_managers2_q = "SELECT
                          *
                       FROM Managers
                       WHERE playerID IN ('cammebi99', 'hengled99', 'keanejo99')
                       GROUP BY playerID"
dbGetQuery(Baseball_database, missing_managers2_q)
# Count the number of player IDs from the Master List that appear in the Managers list:
player_to_manager_q = "SELECT
                        COUNT(DISTINCT playerID) AS Player_to_Manager
                       FROM MASTER
                       WHERE playerID IN (SELECT DISTINCT(playerID) FROM Managers)
                       AND playerID IN (SELECT DISTINCT(playerID) FROM Batting)"
dbGetQuery(Baseball_database, player_to_manager_q)

#If possible, Group by year
#I believe the salaries tables should give us the
#most accurate number of players in the league but
#We can compare it will another table(Fielding)
num_players_year_q = "SELECT
                        yearID, COUNT(DISTINCT(playerID)) AS Num_Players
                      FROM Fielding
                      WHERE yearID BETWEEN 2000 AND 2013 GROUP BY yearID"
dbGetQuery(Baseball_database, num_players_year_q)
```

```
#I'm thinking that this number may overestimate the number of players each year.
#Some players are brought up from the minor league
#during different times of the season(replace starters with injuries).
#So, this number may estimate the total number of unique players that play in a season.
#However, we may want the number of players that are active or start a season and
#my assumption is the Salary table would more accurately reflect this.

#We can't guarantee that certain players are in the Fielding, Batting or Pitching Tables.
#In baseball, not all players pitch, field(DH hitters), or bat(some pitchers)
#For these reasons, I believe the Salary table should provide the most accurate estimate for
#the number of players(assuming all players salaries are listed).
num_players_year2_q = "SELECT
                        yearID, COUNT(DISTINCT(playerID)) AS Num_Players
                       FROM Salaries
                       WHERE yearID BETWEEN 2000 AND 2013
                       GROUP BY yearID"
dbGetQuery(Baseball_database, num_players_year2_q)

#Number of players per team:
#Group by team:
#limit output to 50 tuples
num_players_per_team_q = "SELECT
                            teamID, yearID, COUNT(DISTINCT(playerID)) AS Num_Players
                          FROM Salaries
                          WHERE yearID BETWEEN 2000 AND 2013
                          GROUP BY yearID, TeamID
                          LIMIT 50"
dbGetQuery(Baseball_database, num_players_per_team_q)
#No, It doesn't seem that all teams have the same number or players
#This makes sense because some teams may have a larger salary cap for the
#players than other teams.
#Teams table may contain the necessary info:
team_wseries_2010_q = "SELECT
                        teamIDBR AS Team_Abbrev, name AS Team_name,
                        lgID AS League, divID AS Division
                       FROM Teams
                       WHERE yearID = '2010' AND WSWin = 'Y'"
dbGetQuery(Baseball_database, team_wseries_2010_q)
#San Francisco Giants won in 2010
#Teams table has sufficient info to answer this question:
team_wseries_lost_q = "SELECT
                        franchID AS Franchise_ID, name AS Team_Name,
                        lgID AS League, IFNULL(divID, 'None')  AS Division
                       FROM Teams
                       WHERE franchID NOT IN (
                        SELECT
                          franchID
                        FROM Teams
                        WHERE WSWin = 'Y') AND yearID >= '1903'
                       GROUP BY name
                       ORDER BY franchID"
dbGetQuery(Baseball_database, team_wseries_lost_q)
```

```
#See franchises that have not won a world series:
team_wseries_lost_q = "SELECT
                        franchID AS Franchise_ID, name AS Team_Name,
                        lgID AS League, IFNULL(divID, 'None') AS Division
                       FROM Teams
                       WHERE franchID NOT IN (
                        SELECT
                          franchID
                        FROM Teams
                        WHERE WSWin = 'Y') AND yearID >= '1903'
                       GROUP BY franchID
                       ORDER BY franchID"
dbGetQuery(Baseball_database, team_wseries_lost_q)

#Query the Teams table where WSWin = 'Y'
all_wseries_winners_q = "SELECT
                          yearID AS Year, name AS Team_Name, lgID AS League,
                          IFNULL(divID, 'None') AS Division, WSWin AS Won_Wseries
                         FROM Teams WHERE WSWin = 'Y' AND yearID >= 1903
                         ORDER BY yearID
                         LIMIT 25"
dbGetQuery(Baseball_database, all_wseries_winners_q)

#Teams table,  and SeriesPost have the necessary info for this question
#Join the two tables:
#Order of join: SeriesPost, Teams table
winner_loser_wseries_q = "SELECT
                            win.*, loss.*
                           FROM
                           (SELECT
                             sp.yearID AS Year, sp.teamIDwinner AS WSWinner_ID,
                             t.name AS Winning_team,
                             t.lgID AS Winning_league, t.divID AS Winning_division
                           FROM
                           (SELECT * FROM SeriesPost WHERE round = 'WS') sp
                           JOIN (SELECT * FROM teams WHERE WSWin= 'Y') t
                           ON sp.yearID = t.yearID
                           ORDER BY sp.yearID) AS win
                           JOIN (
                           SELECT
                             s.yearID AS Year, s.teamIDloser AS WSLoser_ID,
                             te.name AS Losing_team,
                             te.lgID AS WSLosing_league, te.divID AS WSLosing_division,
                             s.losses AS Losing_Team_Wins
                           FROM
                           (SELECT * FROM SeriesPost WHERE round = 'WS') s
                           JOIN (SELECT * FROM teams WHERE WSWin = 'N' AND LgWin = 'Y') te
                           ON s.yearID = te.yearID
                           ORDER BY s.yearID) AS loss
                           ON win.Year = loss.Year
                           WHERE win.Year >= 1903
                           ORDER BY win.Year
                           LIMIT 15
```

```
                          "
dbGetQuery(Baseball_database, winner_loser_wseries_q)

#Teams table: look at wins and world series data
#First modern world series occurred in 1903
wins_wseries_q = "SELECT
                    W AS Wins, IFNULL(WSWin, 'N') AS World_series_status
                  FROM Teams
                  WHERE yearID > '1902'"
wins_data = dbGetQuery(Baseball_database, wins_wseries_q)

wins_wseries_q2 = "SELECT
                    W AS Wins, IFNULL(WSWin, 'N') AS World_series_status
                  FROM Teams
                  WHERE yearID > '1902' AND LgWin = 'Y'"
wins_data2 = dbGetQuery(Baseball_database, wins_wseries_q2)

#Create a box plot to see how the winning teams' distribution of wins differs from the
#Losing teams distribution of wins(teams that appeared and didn't apear in the World Series):
ggplot(wins_data, aes(x = World_series_status, y = Wins, fill = World_series_status)) +
                  geom_boxplot(alpha=0.4) +
                  scale_fill_brewer(name = "Won World Series",
                                    labels = c("No", "Yes"), palette = "Set1") +
                  labs(title = "World Series Winners' and Losers' Season Wins(All Teams)",
                       subtitle = "", y = "Season Wins", x = "World Series Status")

#Create another box plot to see how the winning teams' distribution of wins differs from the
#Losing teams distribution of wins(only teams that appeared in World Series):
ggplot(wins_data2, aes(x = World_series_status, y = Wins, fill = World_series_status)) +
                  geom_boxplot(alpha=0.4) +
                  scale_fill_brewer(name = "Won World Series",
                                    labels = c("No", "Yes"), palette = "Set1") +
                  labs(title = "World Series Winners' and Losers' Season Wins(WS Teams)",
                       subtitle = "", y = "Season Wins", x = "World Series Status")


#Join the MASTER, Salaries, Fielding and teams tables:
high_salaries_q = "SELECT
                     s.yearID AS Year, m.playerID AS ID,
                     m.nameFirst AS First_name, m.nameLast AS Last_name,
                     f.Pos AS Position, t.name AS Team_name,
                     MAX(s.salary) AS Top_Salaries
                   FROM MASTER m
                   JOIN Salaries s
                   ON m.playerID = s.playerID
                   JOIN Fielding f
                   ON s.playerID = f.playerID AND s.yearID = f.yearID
                   JOIN Teams t
                   ON f.yearID = t.yearID AND f.teamID = t.teamID
                   WHERE s.yearID = '2003'
                   GROUP BY m.playerID
                   HAVING f.G = max(f.G)
                   ORDER BY Top_Salaries DESC
```

```
                    LIMIT 3"
dbGetQuery(Baseball_database, high_salaries_q)
#Total payroll sum of all player and manager salaries on the team:
#First, check the year where the salary data starts
payroll_2010_q = "SELECT
                    MIN(yearID)
                  FROM Salaries
                  "
dbGetQuery(Baseball_database, payroll_2010_q) #1985 is the earliest recording

#For 2010:
payroll_2010_q = "SELECT
                    s.teamID AS Team_ID, t.name AS Team_Name,
                    SUM(s.salary) AS Total_Payroll
                  FROM Salaries s
                  JOIN (SELECT yearID, teamID, name FROM Teams WHERE yearID >= '1985') t
                  ON s.teamID = t.teamID AND s.yearID = t.yearID
                  WHERE s.yearID = 2010
                  GROUP BY s.teamID"
dbGetQuery(Baseball_database, payroll_2010_q)

#For all the years:
payroll_all_q = "SELECT
                   s.yearID AS Year, s.teamID AS Team_ID, t.name AS Team_Name,
                  SUM(s.salary) AS Total_Payroll
                  FROM Salaries s
                  JOIN (SELECT yearID, teamID, name FROM Teams WHERE yearID >= '1985') t
                  ON s.teamID = t.teamID AND s.yearID = t.yearID
                  GROUP BY s.teamID, s.yearID"
payroll_all_data = dbGetQuery(Baseball_database, payroll_all_q)

#For a teamID identification table:
Team_ident_q = "SELECT
                   teamID AS Team_ID, name AS Team_Name
                  FROM Teams
                  WHERE yearID >= '1985'
                  GROUP BY teamID
                  "
Team_ident_data = dbGetQuery(Baseball_database, Team_ident_q)

#TeamID plot:
kable(Team_ident_data)

#PLot the data:
payroll_all_data = payroll_all_data %>%
                    arrange(Year, Team_ID)
team_labels = names(table(payroll_all_data$Team_Name))
year_labels = as.character(1985:1999)

mycolors = c(brewer.pal(name = "Dark2", n = 4),
             brewer.pal(name = "Accent", n = 4))

#First plot:
```

```
ggplot(payroll_all_data[which(payroll_all_data$Year >= 1985 &
                                payroll_all_data$Year<=1991),]) +
  geom_point(mapping = aes(x = as.factor(Team_ID) ,
                           y = Total_Payroll/10^6,
                           colour = as.factor(Year)), size = 3) +
   coord_flip() +
  #scale_y_continuous() +
  #scale_color_brewer(name="Years") +
  scale_color_manual(name="Years", values = mycolors) +
  xlab("Baseball Teams") +
  ylab("Total Payroll In Millions(USD)") +
  theme_minimal() +
  theme(axis.text.x= element_text(size = 9),
        axis.text.y=element_text(size = 9)) +
  labs(title = "Team Payrolls 1985 to 1991")

#Second plot:
ggplot(payroll_all_data[which(payroll_all_data$Year >= 1992 &
                                payroll_all_data$Year<2000),]) +
  geom_point(mapping = aes(x = as.factor(Team_ID) ,
                           y = Total_Payroll/10^6,
                           colour = as.factor(Year)), size = 3) +
   coord_flip() +
  #scale_y_continuous() +
  #scale_color_brewer(name="Years") +
  scale_color_manual(name="Years", values = mycolors) +
  xlab("Baseball Teams") +
  ylab("Total Payroll In Millions(USD)") +
  theme_minimal() +
  theme(axis.text.x= element_text(size = 9),
        axis.text.y=element_text(size = 9)) +
  labs(title = "Team Payrolls 1992 to 1999")

#Plot 3
ggplot(payroll_all_data[which(payroll_all_data$Year < 2007 &
                                payroll_all_data$Year>=2000),]) +
  geom_point(mapping = aes(x = as.factor(Team_ID),
                           y = Total_Payroll/10^6,
                           colour = as.factor(Year)), size = 3) +
  coord_flip() +
  #scale_y_continuous() +
  #scale_color_brewer(name="Years") +
  scale_color_manual(name="Years", values = mycolors) +
  xlab("Baseball Teams") +
  ylab("Total Payroll In Millions(USD)") +
  theme_minimal() +
  theme(axis.text.x= element_text(size = 9),
        axis.text.y=element_text(size = 9)) +
  labs(title = "Team Payrolls 2000 to 2006")

#Plot 4:
ggplot(payroll_all_data[which(payroll_all_data$Year >=2007),]) +
  geom_point(mapping = aes(x = as.factor(Team_ID),
```

```
                          y = Total_Payroll/10^6,
                          colour = as.factor(Year)), size = 3) +
  coord_flip() +
  #scale_y_continuous() +
  #scale_color_brewer(name="Years") +
  scale_color_manual(name="Years", values = mycolors) +
  xlab("Baseball Teams") +
  ylab("Total Payroll In Millions(USD)") +
  theme_minimal() +
  theme(axis.text.x= element_text(size = 9),
        axis.text.y=element_text(size = 9)) +
  labs(title = "Team Payrolls 2007 to 2013")
#We need to add LgWin, and WSWin to data structure
payroll_wins_q = "SELECT
                    s.yearID AS Year, s.teamID AS Team_ID, t.name AS Team_Name,
                    t.lgID AS League, t.divID AS Division,
                    IFNULL(t.LgWin, 'N') AS Won_League,
                    IFNULL(t.WSWin, 'N') AS Won_Wseries,
                    SUM(s.salary) AS Total_Payroll
                  FROM Salaries s
                  JOIN (SELECT yearID, teamID, lgID, divID, name, LgWin, WSWin
                  FROM Teams
                  WHERE yearID >= '1985') t
                  ON s.teamID = t.teamID AND s.yearID = t.yearID
                  GROUP BY s.teamID, s.yearID"
payroll_Wins_data = dbGetQuery(Baseball_database, payroll_wins_q)

Total_payrollt_q = "SELECT
                    yearID AS Year, SUM(salary) AS Total_Payroll
                  FROM Salaries
                  GROUP BY yearID"
Total_payroll_data = dbGetQuery(Baseball_database, Total_payrollt_q)

#Total Baseball payrolls over time:
ggplot(Total_payroll_data, mapping = aes(x = Year,
                          y = Total_Payroll/10^9)) +
  geom_line()+
  #scale_y_continuous() +
  #scale_color_brewer(name="Years") +
  xlab("Years") +
  ylab("Total Payroll In Billions(USD)") +
  theme_minimal() +
  theme(axis.text.x= element_text(size = 9),
        axis.text.y=element_text(size = 9)) +
  labs(title = "The Entire Major League Baseball Payrolls By Year")

#All salaries over time
league_proll2 = ggplot(payroll_Wins_data) +
  geom_jitter(mapping = aes(x = Year ,
                          y = Total_Payroll/10^6, colour = League),
              width = 0.2) +
  #color = alpha("blue", 0.5)
  #coord_flip() +
```

```
  #scale_y_continuous() +
  #scale_color_brewer(name="Years") +
  #scale_color_manual(name="Years", values = mycolors) +
  xlab("Year") +
  ylab("Total Payroll In Millions(USD)") +
  theme_minimal() +
  theme(axis.text.x= element_text(size = 9),
        axis.text.y=element_text(size = 9)) +
  labs(title = "Team Total Payrolls By League",
       subtitle = "")

league_proll1 = ggplot(payroll_Wins_data) +
                geom_jitter(mapping = aes(x = Year ,
                           y = Total_Payroll/10^6),
                           width = 0.2, color = alpha("blue", 0.5)) +

  #coord_flip() +
  #scale_y_continuous() +
  #scale_color_brewer(name="Years") +
  #scale_color_manual(name="Years", values = mycolors) +
  xlab("Year") +
  ylab("Total Payroll(Millions USD)") +
  theme_minimal() +
  theme(axis.text.x= element_text(size = 9),
        axis.text.y=element_text(size = 9)) +
  labs(title = "Team Total Payrolls")

ggarrange(league_proll1, league_proll2,
          ncol = 2, nrow=1, labels = c("1", "2"))

#Payroll by league and division:
#League Winner Labels:
NL_league = payroll_Wins_data[which(payroll_Wins_data$League == 'NL' &
                                    payroll_Wins_data$Won_League == 'Y'), ]
NL_League_winners = cbind(ifelse(NL_league$Won_League == 'Y',
                                 NL_league$Team_ID, " "),
                          ifelse(NL_league$Won_League == 'Y',
                                 NL_league$Year, " "))
NL_League_winners = paste(NL_League_winners[,1], " ",
                          paste(NL_League_winners[,2]))
NL_league = data.frame(NL_league, NL_League_winners)
#NL league
colors_set = c(brewer.pal(name = "Set1", n = 3))
ggplot(payroll_Wins_data[which(payroll_Wins_data$League == "NL"),],
       aes(x = Year, y = Total_Payroll/10^6, colour = as.factor(Division))) +
  geom_jitter(width = 0.3, size = 1) +
  #scale_y_continuous() +
  #scale_color_brewer(name="Years") +
  scale_color_manual(name="Division",
                     labels = c("Central", "East", "West"),
                     values = colors_set) +
  xlab("Year") +
  ylab("Total Payroll In Millions(USD)") +
```

```
  theme_minimal() +
  theme(axis.text.x = element_text(size = 9),
        axis.text.y = element_text(size = 9)) +
  geom_label_repel(aes(label = NL_League_winners),
                   size = 1.75,  nudge_x = 0.5,
                   nudge_y = 10, arrow= arrow(length = unit(0.1, "inches")),
                   show.legend = FALSE, data = NL_league) +
  labs(title = "NL League Payrolls By Division",
       subtitle = "League Winners Labeled")

#AL League
#League Winner Labels:
AL_league = payroll_Wins_data[which(payroll_Wins_data$League == 'NL' &
                                      payroll_Wins_data$Won_League == 'Y'), ]
AL_League_winners = cbind(ifelse(AL_league$Won_League == 'Y',
                                 AL_league$Team_ID, " "),
                          ifelse(AL_league$Won_League == 'Y',
                                 AL_league$Year, " "))
AL_League_winners = paste(AL_League_winners[,1], AL_League_winners[,2])
AL_league = data.frame(AL_league, AL_League_winners)

#AL Plot:
ggplot(payroll_Wins_data[which(payroll_Wins_data$League == "NL"),]) +
  geom_jitter(mapping = aes(x = Year,
                            y = Total_Payroll/10^6,
                            colour = as.factor(Division)),
                            width = 0.3, size = 1) +
  scale_color_manual(name="Division",
                     labels = c("Central", "East", "West"),
                     values = colors_set) +
  xlab("Year") +
  ylab("Total Payroll In Millions(USD)") +
  theme_minimal() +
  geom_label_repel(aes(x = Year,
                            y = Total_Payroll/10^6,
                            colour = as.factor(Division),
                   label = AL_League_winners), size = 1.75, nudge_x = 0.5,
                   nudge_y = 10, arrow= arrow(length = unit(0.1, "inches")),
                   show.legend = FALSE, data = AL_league)+
  theme(axis.text.x= element_text(size = 9),
        axis.text.y=element_text(size = 9)) +
  labs(title = "AL League Payrolls By Division",
       subtitle = "League Winners Labeled")



#Side-by-side box plot for winning the League and Payroll
options(scipen = 5)
ggplot(payroll_Wins_data, aes(x =  Won_League, y = Total_Payroll/10^6,
                              fill =  Won_League)) +
               geom_boxplot(alpha=0.4) +
               scale_fill_brewer(name = "Won League",
                                 labels = c("No", "Yes"),
```

```
                                    palette = "Set1") +
                labs(title = "League Winners' and Losers' Payroll(All Teams)",
                       y = "Total Payroll In Millions(USD)",
                       x = "League Won")


#Join the Master and Batting tables
#First, we can get the player with the most homeruns:
most_homers_q = "SELECT
                    b.playerID AS ID,
                    m.nameFirst AS First_name,
                    m.nameLast AS Last_name,
                    SUM(b.HR) AS Homeruns
                 FROM Batting b
                 JOIN Master m
                 ON b.playerID = m.playerID
                 GROUP BY b.playerID
                 ORDER BY Homeruns DESC
                 LIMIT 1
                 "
dbGetQuery(Baseball_database, most_homers_q)

#Number per year:
most_homers_q = "SELECT
                   b.yearID AS Year,
                   m.nameFirst || ' ' || m.nameLast AS Name,
                   b.HR AS Homeruns
                FROM Batting b
                JOIN Master m
                ON b.playerID = m.playerID
                WHERE b.playerID IN (
                   SELECT
                     playerID AS ID
                   FROM Batting
                   GROUP BY playerID
                   HAVING SUM(HR) =
                     (SELECT
                       SUM(HR) AS Homeruns
                     FROM Batting
                     GROUP BY playerID
                     ORDER BY Homeruns DESC LIMIT 1))
                 ORDER BY b.yearID
                 "
dbGetQuery(Baseball_database, most_homers_q)

#AN ATTEMPT WAS NOT FINISHED
#15. Has the distribution of home runs for players increased over the years?
#Batting table:
dist_homers_q = "SELECT
                    yearID AS Year,
                    playerID AS ID,
                    HR AS Homeruns
                 FROM Batting
```

```
                  "
dist_homers_data = dbGetQuery(Baseball_database, dist_homers_q)

#Plot boxplots for each year
ggplot(dist_homers_data,
       aes(x = as.factor(Year), y = Homeruns, fill = as.factor(Year))) +
                  geom_boxplot(alpha=0.4) +
                  #ylim(0, 25)
                  labs(title = "Yearly Boxplots For Homeruns",
                       subtitle = "", y = "Homeruns", x = "Year")
```