STA141 Homework 6

Shuxin Li
912525987

I certify that I have acknowledged any code that I used from any other person in the class, from Piazza or any Web site or book or other source. Any other work is my own.

1. UNIX Shell Tools

(i)

Code in shell.

```
cd F:/Airline2012_13

# find the location of ORIGIN column.
column=$(head -n 1 2012_August.csv | egrep -o .*\"ORIGIN\" | tr -dc , | wc -m)
column=$(( $column + 1 ))

# calculate the counts of outbound flights and calculate the time of running code.
time cut -d, -f$column *.csv | cut -d , -f 1 | sort | uniq -c | sort -nr | egrep "OAK|SMF|LAX|JFK|SFO"
```

Result in shell

```
222029  “LAX”
169734  “SFO”
105097  “JFK”
 44911  “OAK”
 43145  “SMF”

 real 5m34.482s
 user 5m51.873s
 sys 0m3.879s
```

Code in R.

```
maketable<-function(i){
   d = read.csv(dir[i])
   #find the index of "OAK","SMF","LAX","SFO","JFK" in table
   index=match(c("OAK","SMF","LAX","SFO","JFK"),names(table(d$ORIGIN)))
```

```
   # return table of "OAK","SMF","LAX","SFO","JFK"
   table(d$ORIGIN)[index]
}

system.time(sapply(1:length(dir), function(x) maketable(x)))

# count the five outbound flights of each csv file and make a matrix
a=sapply(1:length(dir), function(x) maketable(x))

# count the total numbers of outbound flights for the five airports
b=sapply(1:5, function(x) sum(a[x,]))

# add name for the counts and make the results readable
c=matrix(b)
rownames(c)<-c("OAK","SMF","LAX","SFO","JFK")

# sort these counts from largest to smallest
outboundf=apply(c, 2, sort, decreasing=TRUE)
```

Result in R

```
   user  system elapsed
 806.30   12.68  907.80


LAX 222029
SFO 169734
JFK 105097
OAK  44911
SMF  43145
```

Comparison of the total time for each approach

When we time both the two approaches, it turns out that the user time will be recorded on both approaches. Thus, we use the user time as an evaluation criterion.

The first approach in shell takes 351.87 seconds comparing with the second approach in R taking 806.3 seconds.

It is clearly that the approach in shell is more efficient than the approach in R.

(ii)

Code in R and shell

```
# in R, find the location of column ORIGIN and DEST.
d1 = read.csv("2012_August.csv", nrow    = 30)
```

```
match(c("ORIGIN", "DEST"), names(d))    # getting the location is 15,24

# in shell check if 15, 24 is the right location of columns ORIGIN and DEST
cut -d, -f 15,24 2012_August.csv | head -n2

# it turns out that 15, 25 is the right location of columns ORIGIN and DEST
cut -d, -f 15,25 2012_August.csv | head -n2

# creat a csv file
cut -d , -f 15,25 *.csv | egrep "OAK|SMF|LAX|JFK|SFO" > oripairsdest.csv

# in R, read the csv file created by shell
d = read.csv("F://Airline2012_13//oripairsdest.csv", header=FALSE)
names(d)=c("Origin","Destination")

# calculate the count of the lines in files which involve any of these five airports
length(d$Origin)

# compute the total number of flights for each of the 5 airports.
Airport=c("OAK","SMF","LAX","JFK","SFO")
Ori=sapply(1:5, function(x) which(d$Origin==Airport[x]))
Dest=sapply(1:5, function(x) which(d$Destination==Airport[x]))
Ori1=unlist(Ori)
Dest1=unlist(Dest)
OriDest=intersect(Ori1, Dest1)
length(OriDest)
```

Result report

First, calculate the count of the lines in files which involve any of these five airports. There are 1065141 pairs at most.

Second, compute the total number of flights in and out of the five airports. Then we conclude that there are 104690 flights in and out of the five airports.

● 2. Baseball, Databases and SQL

Question 1: What years does the data cover? Are there data for each of these years?

Answer 2.1:

The data covers years from 1871 to 2013. There are data for each of these years such as "Appearances", "Batting", "Fielding", "Managers", "Pitching", "Teams".

Code 2.1:

```
library(RSQLite)
db = dbConnect( SQLite(), dbname = "C:\\Users\\Administrator\\Desktop\\141
                    HW6\\lahman2013.sqlite")
alltable = dbListTables(db)

getallnames = function(db, index, database) {
   # index is the index of the table we want to get all names in database
   # database is the database in which we want to find the table

   # the fuction is to get all names of the table we choose in database
   table = database[index]
   query = 'SELECT * FROM '
   query = paste0(query, table)
   names(dbGetQuery(db, query))
}

# get all col names of all tables
nameall=sapply(1:length(alltable), function(x) getallnames(db, x, alltable))

# check if the col names include yearID data
namematchyearID = sapply(1:length(nameall), function(x)
                                    match(nameall[[x]],"yearID"))

# select tables with yearID data
namewithyearID = sapply(1:length(nameall), function(x)

any(namematchyearID[[x]]==1,na.rm=TRUE))
sometable = alltable[namewithyearID]

getyearIDs = function(db, index, database) {
   # index is the index of the table we want to get the data years in database
   # database is the database in which we want to find the table to get years

   # the fuction is to get the data years in the table we choose
   table = database[index]
   query = 'SELECT yearID FROM '
   query = paste0(query, table)
   dbGetQuery(db, query)
}

allyear = sapply(1:length(sometable), function(x) getyearIDs(db, x, sometable))

# the years covered
range(unlist(allyear))
```

```
# check if there are data for each of these years
number = range(unlist(allyear))[2]-range(unlist(allyear))[1]+1
yearnumber = sapply(1:length(allyear), function(x) length(table(allyear[[x]])))
data_for_each_yea r= sometable[which(yearnumber==number)]
```

Question 2: How many people are included in the database? How many are players, managers, etc?

Answer 2.2:

From the introduction on http://seanlahman.com/files/database/readme2013.txt.I find that all managers in other table are included in the "Manager" table and all players in other table are included in "Master" table.

Thus after taking a calculation, there are 18354 players and 682 managers who are recorded.

Since some people who were not only players but were also managers, there are 18357 people which are included in the database.

Code 2.2:

```
# number of managers
allmanager = dbGetQuery(db, 'SELECT DISTINCT playerID
                                FROM Managers')
numallmanager = length(allmanager$playerID)

# number of players
allplayer = dbGetQuery(db, 'SELECT playerID
                              FROM MASTER')
numallplayer = length(allplayer$playerID)

# Unique people for some people who were not only players but were also manaagers
length(union(allmanager$playerID,allplayer$playerID))
```

Question 3: What team won the World Series in 2000

Answer 2.3:

In the team table, I notice that the variable "WSWin" means whether a team won World Series or not. In 2000, I found New York Yankees won the World Series and its team ID is NYA.

Code 2.3:

 # win the World Series in 2000

```
WinSeries2000 = dbGetQuery(db, 'SELECT    teamID, name FROM Teams
                                WHERE    WSWin = "Y" AND    yearID = 2000 ')
```

Question 4: What team lost the World Series each year

Answer 2.4:

Since each year there are two teams competing for World Series and both teams have won the League Champion. Therefore we can get a table which describe the teams losing World Series each year. The table below shows part of results.

	yearID	name
1	1884	New York Metropolitans
2	1885	St. Louis Browns
3	1885	Chicago White Stockings
4	1886	Chicago White Stockings
5	1887	St. Louis Browns
6	1888	St. Louis Browns
7	1889	Brooklyn Bridegrooms
8	1890	Louisville Colonels
9	1890	Brooklyn Bridegrooms
10	1903	Pittsburgh Pirates
11	1905	Philadelphia Athletics
112	2007	Colorado Rockies
113	2008	Tampa Bay Rays
114	2009	Philadelphia Phillies
115	2010	Texas Rangers
116	2011	Texas Rangers
117	2012	Detroit Tigers
118	2013	St. Louis Cardinals

Code 2.4:

```
teamlostSeries = dbGetQuery(db, 'SELECT    yearID, name
                                FROM      Teams
                                WHERE    WSWin = "N" AND LgWin = "Y" ')
```

Question 5: Do you see a relationship between the number of games won in a season and wining the World Series?

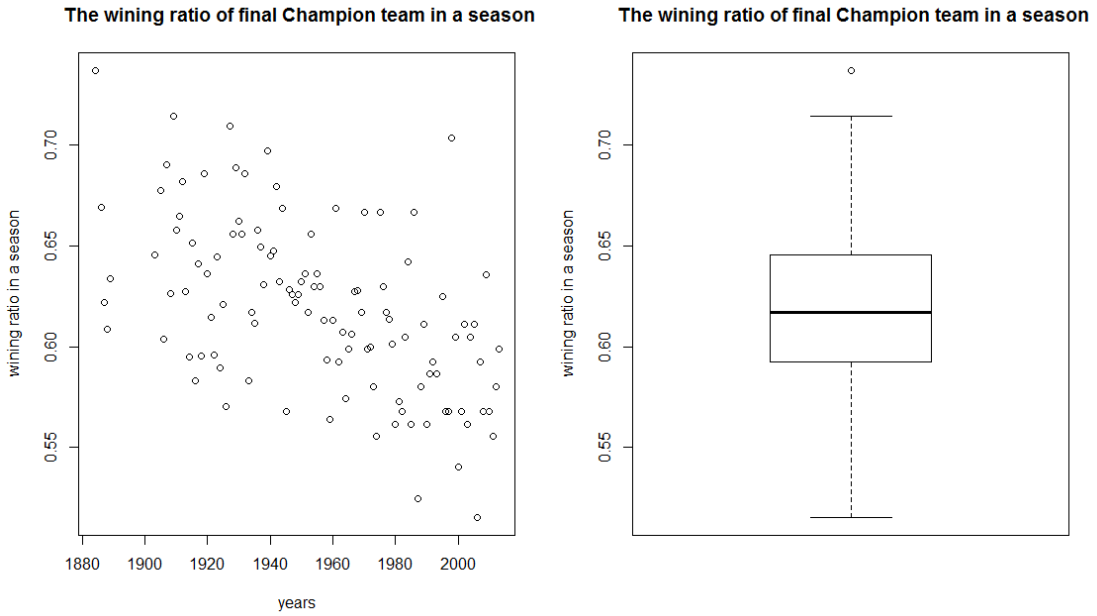
Answer 2.5:

From the scatter plot below, we can find there is a negative linear regression relationship between the final Champion team's wining ratio in a season and the years.

It indicates that with years increasing, there is a growing competition among each

teams. The gap between the strong teams and weak teams is in decreasing.

From the boxplot below, it indicates that a Champion team should win half of games in a season at least. Most of the Champion team's wining ratio is between 0.6 and 0.65.



Code 2.5:

```
team_WSwin = dbGetQuery(db, 'SELECT yearID, W, G
                             FROM    Teams
                             WHERE    WSWin = "Y" ')

# draw a scatter plot and a boxplot
par(mfrow=c(1,2))
plot(team_WSwin$yearID, team_WSwin$W/team_WSwin$G, ylab="wining ratio in a
    season",xlab="years", main=" The wining ratio of final Champion team in a
    season")
boxplot(team_WSwin$W/team_WSwin$G, main=" The wining ratio of final
        Champion team in a season", ylab="wining ratio in a season")
```

Question 6: In 2003, what were the three highest salaries?

Answer 2.6:

The three highest salaries are 22000000, 20000000, 18700000 in 2003.

Code 2.6:

```
playersalary = dbGetQuery(db, 'SELECT DISTINCT salary
                                FROM      Salaries
                                WHERE    yearID = 2003 ')
sort(playersalary$salary, decreasing= TRUE)[1:3]
```

Question 7: For 1999, 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.

Answer 2.7:

The results of total payroll of each of the different teams are below.

	Sumsalary	teamID
1	55388166	ANA
2	68703999	ARI
3	73140000	ATL
4	80605863	BAL
5	63497500	BOS
6	25620000	CHA
7	62343000	CHN
8	33962761	CIN
9	72978462	CLE
10	61935837	COL
11	36489666	DET
12	21085000	FLO
13	54914000	HOU
14	26225000	KCA
15	80862453	LAN
16	43377395	MIL
17	21257500	MIN
18	17903000	MON
19	86734359	NYA
20	65092092	NYN
21	24431833	OAK
22	31692500	PHI
23	24697666	PIT
24	49768179	SDN
25	54125003	SEA
26	46595057	SFN
27	49778195	SLN
28	38870000	TBA
29	76709931	TEX
30	45444333	TOR

Code 2.7:

```
# compute the total payroll of each of the different teams
teamsalary = dbGetQuery(db, 'SELECT SUM(salary) AS Sumsalary, teamID
                             FROM Salaries
                             WHERE yearID = 1999
                             GROUP BY teamID ')
```

```
# get total payrolls for each team over years
teamsalary2 = dbGetQuery(db, 'SELECT yearID, teamID, SUM(salary)
                                FROM Salaries
                                GROUP BY yearID, teamID;')

# use ggplot to draw the plot
install.packages("ggplot2")
library(ggplot2)
names(teamsalary2) = c('Year', 'TeamID', 'TotalPayrolls')
ggplot(teamsalary2, aes(Year, TotalPayrolls)) + geom_point(aes(color = TeamID)) +
labs(title="TotalPayrolls over years")
```

Question 8: Study the change in salary over time. Have salaries kept up with inflation, fallen behind, or grown faster?

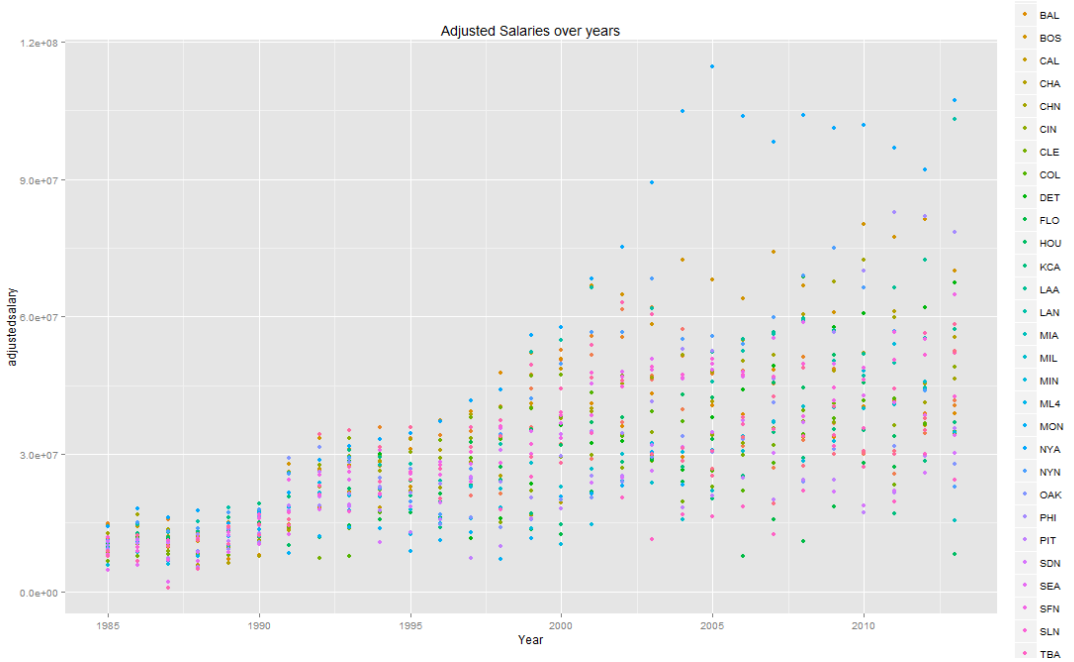
Answer 2.8:

We first adjust the total payroll of each teams based on inflation rates through 1986 to 2013. These adjusted salary will reflect the salary against the inflation rate.

Then, we draw a plot which includes all the teams' total adjusted payroll over years. In the plot, we can see that each team's adjusted salary has a risen trend, even sometimes it will decreases.

We could also observe that each team's adjusted salary is scarcely less than the salary level in 1985 which indicates that the average growth rate of salary is higher than the rate of inflation for all teams.

Especially, some teams such as have incredible growth rate of salary relative to other teams.



Code 2.8:

```
# input inflation rate and change it to the rate based on 1985
mydata=read.table("C:\\Users\\Administrator\\Desktop\\141 HW6\\Inflation rate.txt",
                  header=FALSE)
inflation=mydata$V2/100

a=rep(1,29)

for(i in 2:29){
   # calculate the inflation rate based on 1985
   a[i]=a[i-1]*(1+inflation[i-1])
}

baseinflation=a      # the inflation rate vector

# update the dataframe teamsalary2 and add the adjusted salary into it.
year = matrix(c(1985:2013))
inflationrate = data.frame(year,baseinflation)
teamsalary3 = merge(teamsalary2, inflationrate, by.x="Year", by.y="year")
teamsalary3$adjustedsalary = teamsalary3$TotalPayrolls/teamsalary3$baseinflation

# plot the adjusted salary for each teams over years
ggplot(teamsalary3, aes(Year, adjustedsalary)) + geom_point(aes(color = TeamID)) +
labs(title="Adjusted Salaries over years")
```

Question 9: Compare payrolls for the teams that are in the same leagues, and then in the same divisions. Are there any interesting characteristics? Have certain teams always had top payrolls over the years? Is there a connection between payroll and performance?

Answers 2.9:

Since there two leagues which are NL and AL and each league has three divisions such as W, E, C. I draw eight scatter plots. The first two plots (figure 9.1 and figure 9.2) represent the payrolls for the teams that are in the same leagues.

From these plot, I find that NYA's payrolls is obvious greater than other teams in AL league and it almost had top payrolls over the years.

From the rest plots (figure 9.3, 9.4, 9.5, 9.6, 9.7, 9.8), the teams that are in the same leagues and then in the same divisions have similar pattern which is that each division in different league has one or two team with high payrolls and one or two team with low payrolls. The gap between high payrolls and low payrolls become bigger with years

growing.

Finally, we draw the wining ratio representing the performance against payroll. The scatter plot shows that there is no obvious relationship between payroll and performance and the correlation is 0.2198609.

Payrolls over years for each teams in NL

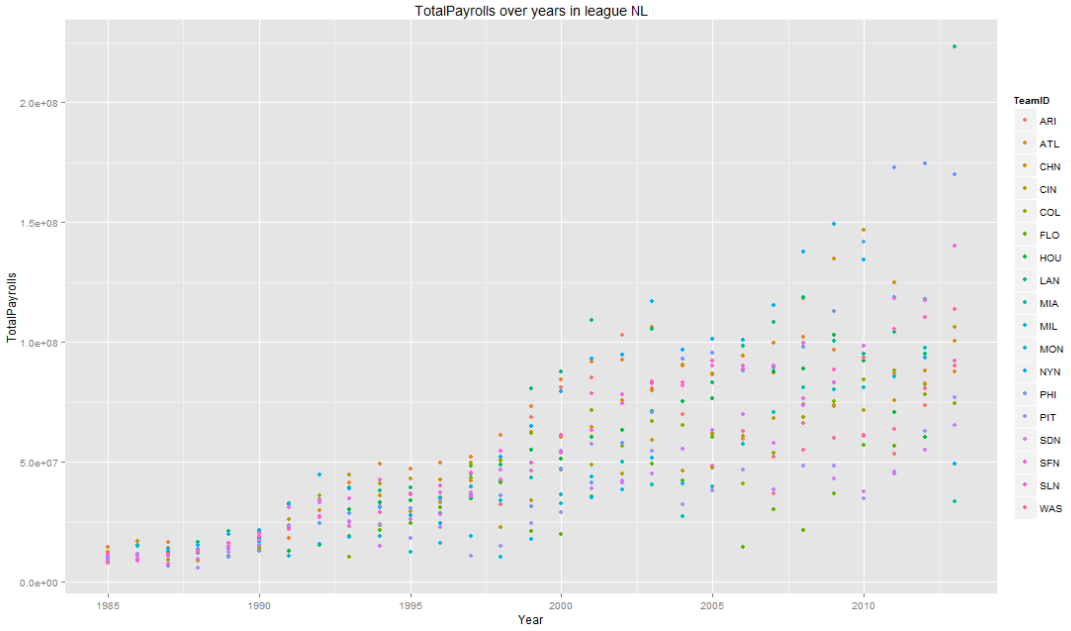


Figure 9.1

Payrolls over years for each teams in AL

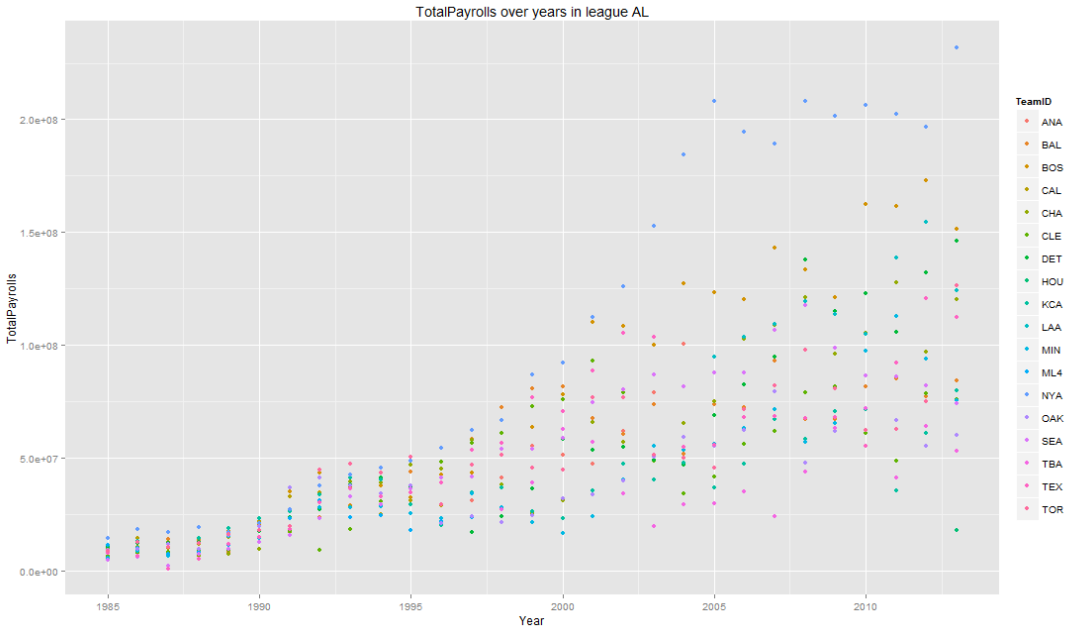


Figure 9.2

Payrolls over years for each teams in AL and division W

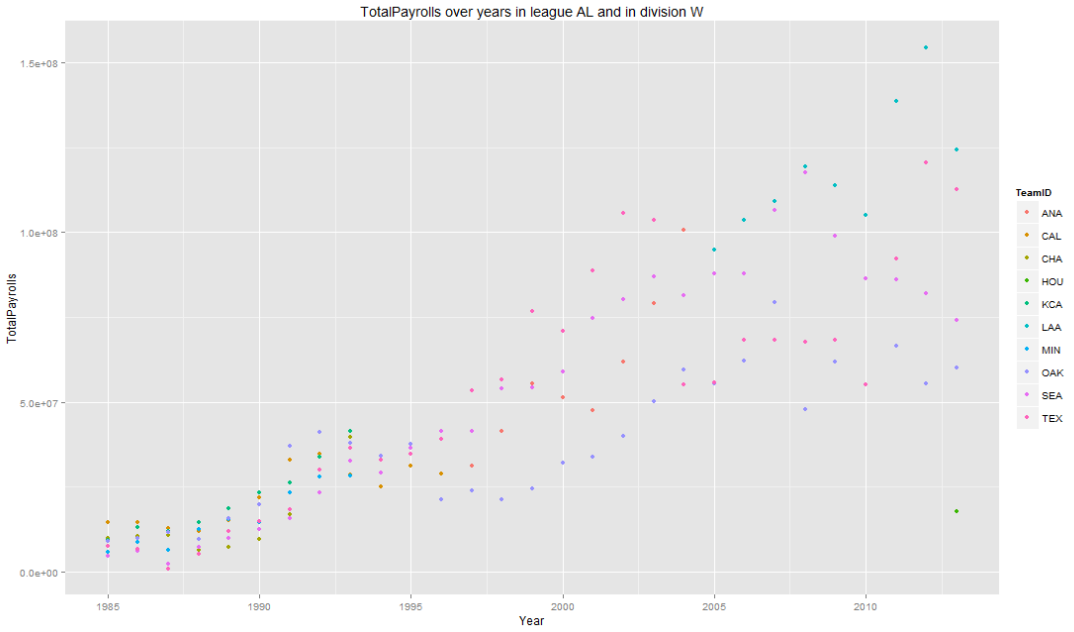


Figure 9.3

Payrolls over years for each teams in AL and division E

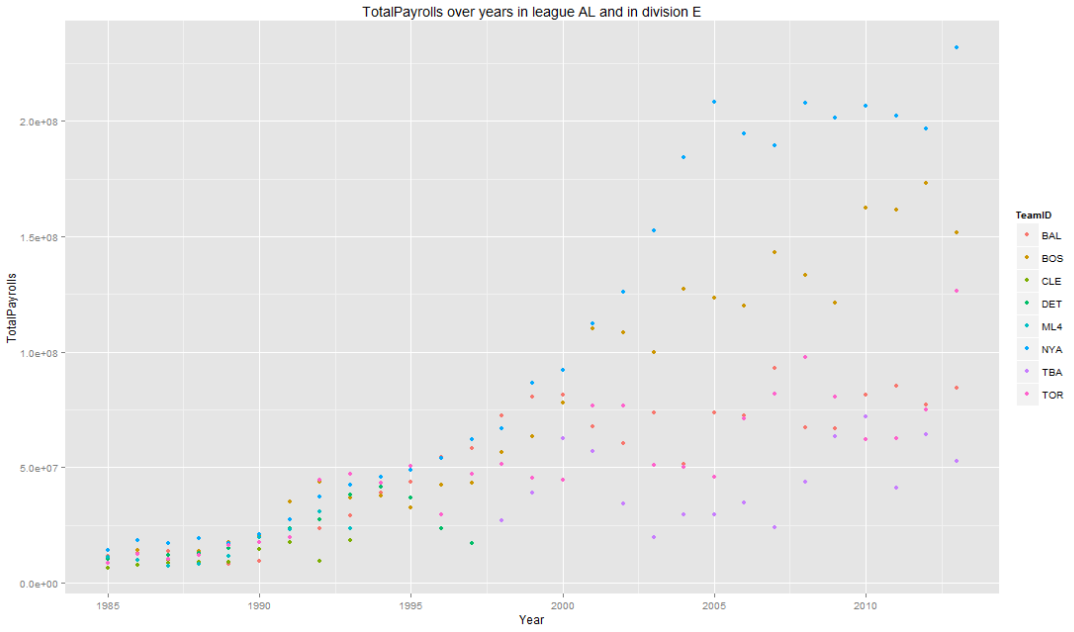


Figure 9.4

Payrolls over years for each teams in AL and division C

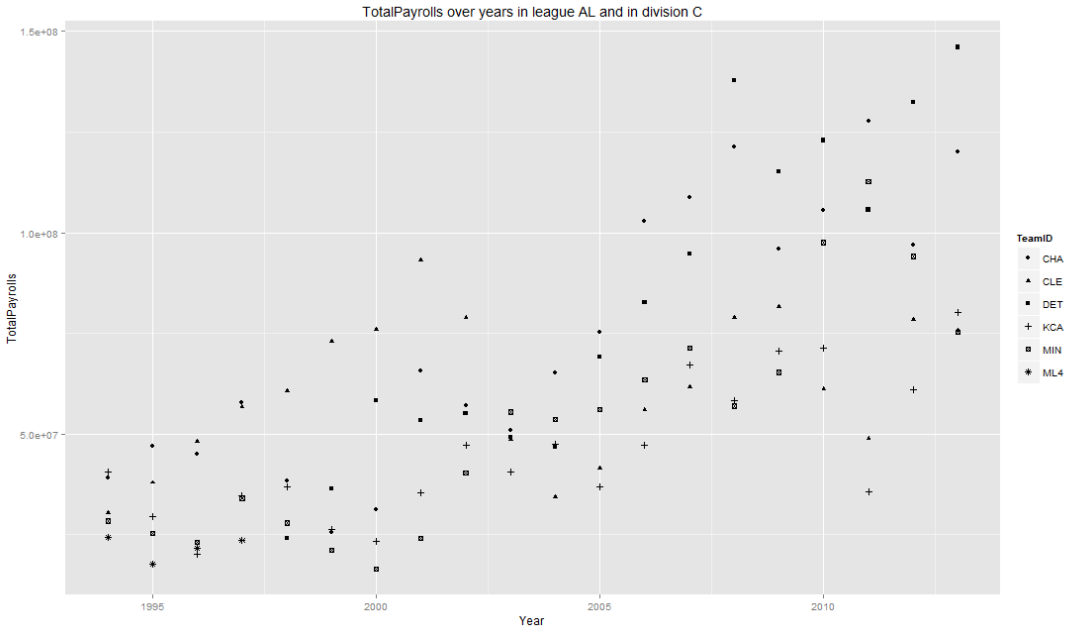


Figure 9.5

Payrolls over years for each teams in NL and division W

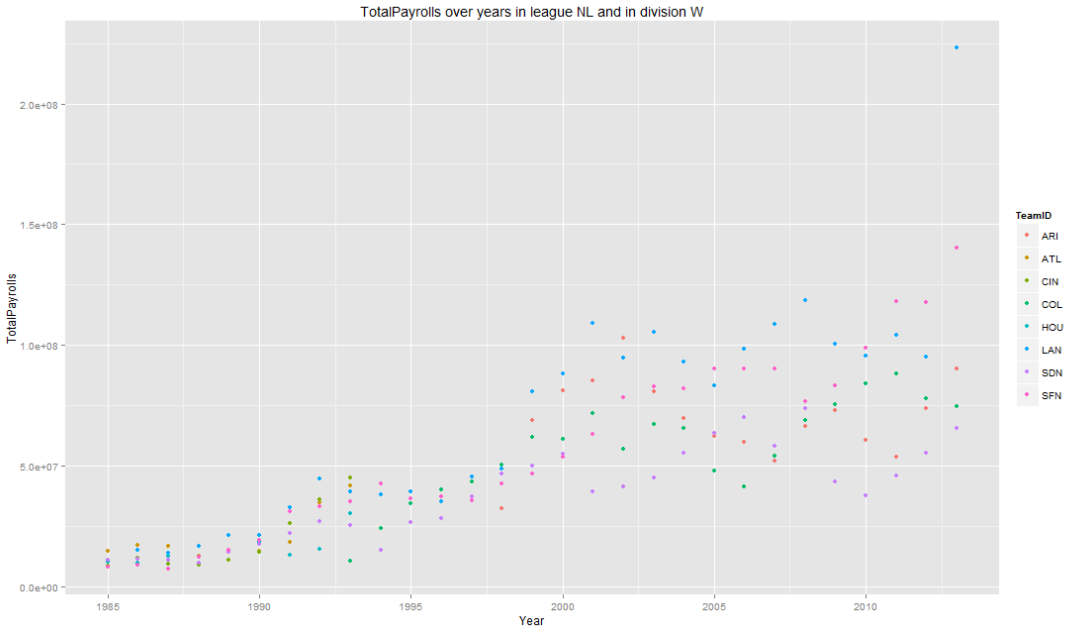


Figure 9.6

Payrolls over years for each teams in NL and division E

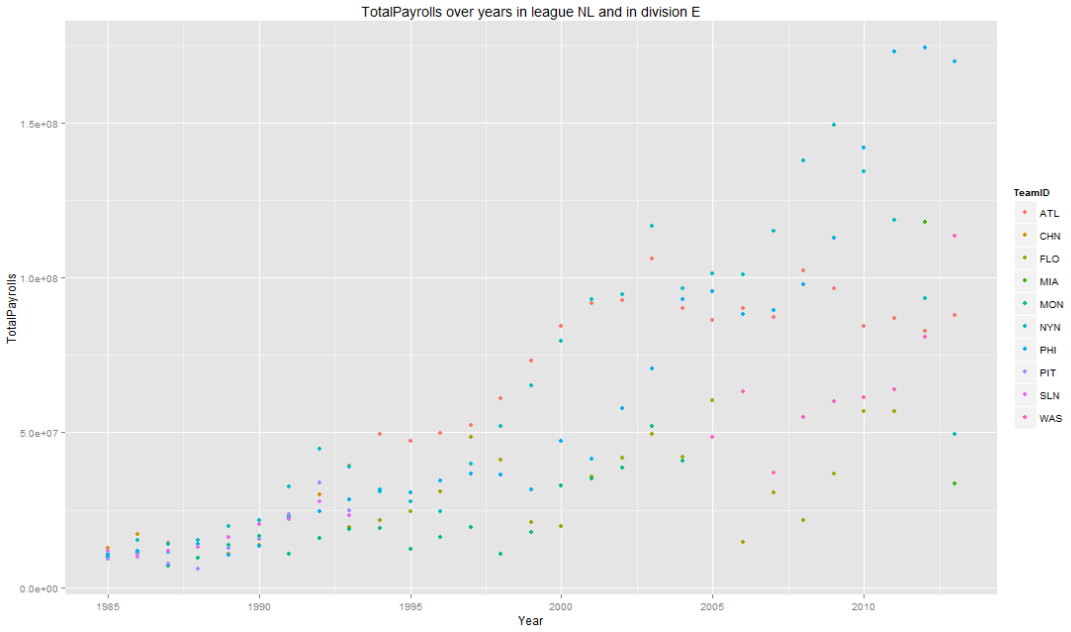


Figure 9.7

Payrolls over years for each teams in NL and division C

Figure 9.8

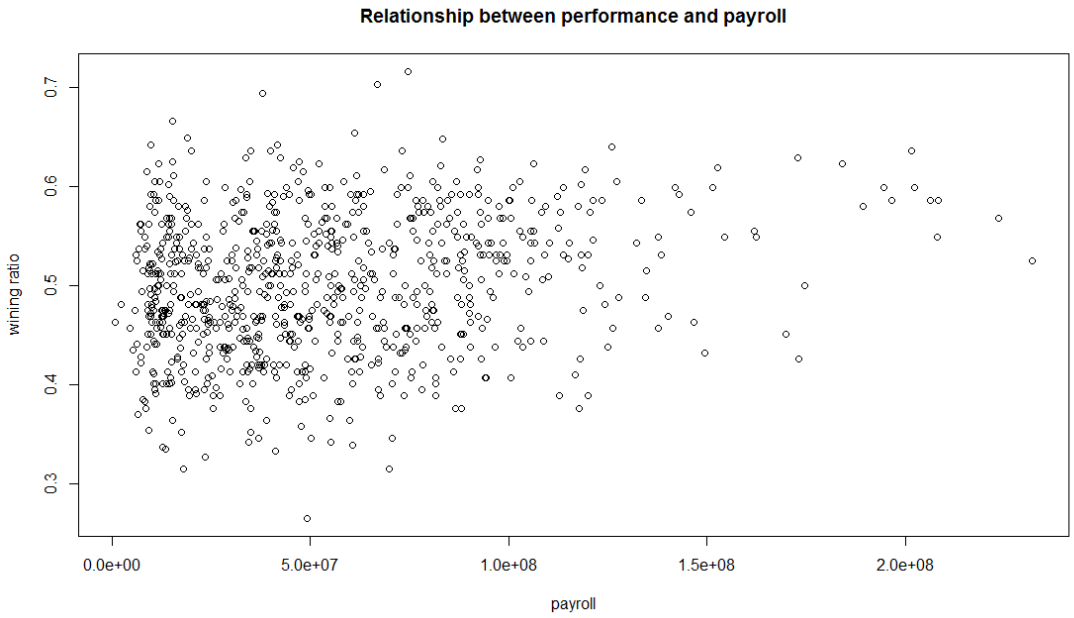


Figure 9.9

Code 2.9:

```
# compare payrolls for the teams that are in the NL leagues
teamsalary4NL = dbGetQuery(db, 'SELECT yearID, teamID, SUM(salary), lgID
                                FROM Salaries
                               WHERE lgID = "NL"
                                GROUP BY yearID, teamID;')

names(teamsalary4NL) = c('Year', 'TeamID', 'TotalPayrolls')
ggplot(teamsalary4NL, aes(Year, TotalPayrolls)) + geom_point(aes(color = TeamID))
        + labs(title="TotalPayrolls over years in league NL")

# compare payrolls for the teams that are in the AL leagues
teamsalary4AL = dbGetQuery(db, 'SELECT yearID, teamID, SUM(salary), lgID
                                FROM Salaries
                                WHERE lgID = "AL"
                                GROUP BY yearID, teamID;')

names(teamsalary4AL) = c('Year', 'TeamID', 'TotalPayrolls')
ggplot(teamsalary4AL, aes(Year, TotalPayrolls)) + geom_point(aes(color = TeamID))
+ labs(title="TotalPayrolls over years in league AL")

# compare payrolls for the teams that are in the same leagues and in the same division
teamsalary4AL = dbGetQuery(db, 'SELECT yearID, teamID, SUM(salary), lgID
                                FROM Salaries
                                WHERE lgID = "AL"
```

```
GROUP BY yearID, teamID;')

frame=dbGetQuery(db, 'SELECT teamID, divID, yearID from Teams')

# Join two dataframe
teamsalary4ALdiv=merge(teamsalary4AL, frame, by=c("teamID","yearID"))

# the team in AL league and in W, E, C divisions.
teamsalary4ALdivw=teamsalary4ALdiv[which(teamsalary4ALdiv$divID=="W"),]
names(teamsalary4ALdivw) = c('TeamID','Year','TotalPayrolls')
ggplot(teamsalary4ALdivw, aes(Year, TotalPayrolls)) + geom_point(aes(color =
TeamID)) + labs(title="TotalPayrolls over years in league AL and in division W")

teamsalary4ALdivE=teamsalary4ALdiv[which(teamsalary4ALdiv$divID=="E"),]
names(teamsalary4ALdivE) = c('TeamID','Year','TotalPayrolls')
ggplot(teamsalary4ALdivE, aes(Year, TotalPayrolls)) + geom_point(aes(color =
TeamID)) + labs(title="TotalPayrolls over years in league AL and in division E")

teamsalary4ALdivC=teamsalary4ALdiv[which(teamsalary4ALdiv$divID=="C"),]
names(teamsalary4ALdivC) = c('TeamID','Year','TotalPayrolls')
ggplot(teamsalary4ALdivC, aes(Year, TotalPayrolls)) + geom_point(aes(pch =
TeamID)) + labs(title="TotalPayrolls over years in league AL and in division C")

# Join two talbes
teamsalary4NLdiv=merge(teamsalary4NL, frame, by=c("teamID","yearID"))

# the team in AL league and in W, E, C divisions.
teamsalary4NLdivw=teamsalary4NLdiv[which(teamsalary4NLdiv$divID=="W"),]
names(teamsalary4NLdivw) = c('TeamID','Year','TotalPayrolls')
ggplot(teamsalary4NLdivw, aes(Year, TotalPayrolls)) + geom_point(aes(color =
TeamID)) + labs(title="TotalPayrolls over years in league NL and in division W")

teamsalary4NLdivE=teamsalary4NLdiv[which(teamsalary4NLdiv$divID=="E"),]
names(teamsalary4NLdivE) = c('TeamID','Year','TotalPayrolls')
ggplot(teamsalary4NLdivE, aes(Year, TotalPayrolls)) + geom_point(aes(color =
TeamID)) + labs(title="TotalPayrolls over years in league NL and in division E")

teamsalary4NLdivC=teamsalary4NLdiv[which(teamsalary4NLdiv$divID=="C"),]
names(teamsalary4NLdivC) = c('TeamID','Year','TotalPayrolls')
ggplot(teamsalary4NLdivC, aes(Year, TotalPayrolls)) + geom_point(aes(pch =
TeamID)) + labs(title="TotalPayrolls over years in league NL and in division C")


# draw a plot to show a connection between payroll and performance
```

```
names(teamsalary2)
names(dataframe)
dataframe=dbGetQuery(db, 'SELECT yearID, teamID, AVG(W) AS avgW, AVG(G)
                          AS avgG
                          FROM Teams
                          Where yearID > 1984
                          GROUP BY teamID, yearID')
# join two tables
performance_pay= merge(teamsalary2, dataframe, by=c("yearID", "teamID"))
performance_pay$winratio=performance_pay$avgW/performance_pay$avgG

# show the relationship between payoff and team performance
plot(performance_pay[,3],performance_pay$winratio,  xlab="payroll",  ylab="wining
    ratio", main="Relationship between performance and payroll")
cor(performance_pay[,3],performance_pay$winratio)
```

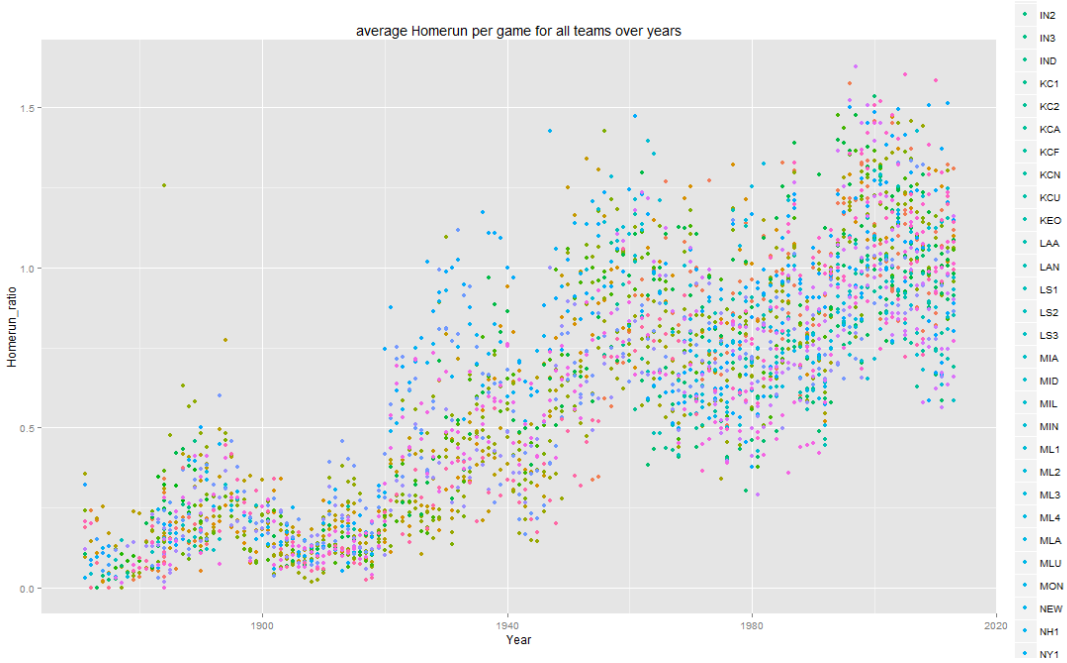
Question 10: Has the distribution of home runs for players increased over the years? When answering the questions, try to summarize the results in convenient and informative form that illustrate the key features.

Answer 2.10:

I draw a scatter plot which describe the relationship between each team's average Home run per game and year.

In the plot, it indicates that the home run's ratio increases with the year growing. Besides, after 1930, the dispersion of home run's ratio start to increase largely. This may be related to increase in team members.

After 1930, the dispersion of home run's ratio seem equal, however the mean of all teams' home run's ratio had been increasing.



Code 2.10:

```
# first get a dataframe in which one observation includes year teamID whole
# year's Homerun for the team in the year and whole game number of the team

Homerun = dbGetQuery(db, 'SELECT yearID, teamID, SUM(HR) AS sumHR,
                          SUM(G) AS sumG
                         FROM    Teams
                         GROUP BY yearID, teamID ')

Homerun$ratio = Homerun$sumHR/ Homerun$sumG

# draw the average Homerun per game for all teams over years
names(Homerun)=c("Year","Team","Homerun_number","game_number","Homerun_
                 ratio")
ggplot(Homerun, aes(Year, Homerun_ratio)) + geom_point(aes(color = Team)) +
      labs(title="average Homerun per game for all teams over years")
```