Everytime use title and name table, if the question ask for movie or actors, I will limit the type of movie and role of person first. Becuase in the title and name table, they contain all the information. Like name table contains directors, writers, but the question asks for actors, the title table contains episode, movies, tv movies. I post it on Piazz, Ducan said "For questions 5 onwards, movie means movie, i.e. just 1" on post number 1354. so I add "JOIN kind\_type ON ON title.kind\_id = kind\_type.id AND　WHERE kind = 'movie'" for every questions who is asking for "movie".

From some questions I just pasted what I got at the first time, because some questions take a lot time if I use R Markdown to run it again, so I used R Markdown to use some questions that easy to get, and pasted the code and output for some questions that hard to run.

Set up working space, load data and load packages

setwd("C:/Users/liguoxin93/Downloads")  
library(RSQLite)

## Loading required package: DBI

library(igraph)

##   
## Attaching package: 'igraph'  
##   
## The following objects are masked from 'package:stats':  
##   
## decompose, spectrum  
##   
## The following object is masked from 'package:base':  
##   
## union

imbd = dbConnect(SQLite(), "lean\_imdbpy.db")  
  
# look tables in the data  
dbListTables(imbd)

## [1] "aka\_name" "aka\_title" "cast\_info"   
## [4] "info\_type" "keyword" "kind\_type"   
## [7] "movie\_id" "movie\_info" "movie\_info\_idx"   
## [10] "movie\_keyword" "name" "person\_info"   
## [13] "role\_type" "second\_name" "sqlite\_sequence"  
## [16] "title"

**1.How many actors are there in the database? How many movies?**

Since the id is unique for each actor's name, so the total rows in the name table is the total actors in the database. For the movies, since in the database it not only contains movies, but also TV shows and something eles. So first I join the title and kind\_type, then I only count the the number of titles that kind is movie.

# rows in name table  
dbGetQuery(imbd, "SELECT COUNT(\*) AS NumberofActors From name")

## NumberofActors  
## 1 5375509

# rows in title table, limit kind = "movie"  
dbGetQuery(imbd, "SELECT COUNT(\*) AS Numberofmovies  
 From title  
 JOIN kind\_type   
 ON title.kind\_id = kind\_type.id  
 WHERE kind = 'movie'")

## Numberofmovies  
## 1 878800

I got 5375509 actors in the database and 878800 movies in the database

**2.What time period does the database cover?** I looked at the schema, only title and aka\_title have year information, but in the title table, it has production year and series years.

# look at title table (won't show the output in the homwwork)  
#dbGetQuery(imbd, "SELECT \* From title LIMIT 5")

By looking at the title table, I think I need to use production year to decide time period, so I did MAX and MIN function to get the time period

# get max year  
dbGetQuery(imbd, "SELECT MAX(production\_year) From title")

## MAX(production\_year)  
## 1 2025

# get MIN year  
dbGetQuery(imbd, "SELECT MIN(production\_year) From title")

## MIN(production\_year)  
## 1 1874

The time period is from 1874 to 2025 in this database.

**3.What proportion of the actors are female? male?** For this question in order to do it in one query, I use the subquery to get the total number of observations. I first group the data by gender, count how many observations in different gender, devide the number of observations for different gender by the total number of observations.

# counts by gender and devide by the total count in name  
dbGetQuery(imbd, "SELECT gender, count(\*)\*100 / (SELECT COUNT(\*)   
 FROM name)   
 AS proportion  
 From name  
 GROUP BY gender")

## gender proportion  
## 1 <NA> 34  
## 2 f 23  
## 3 m 42

There are 34% NA, 23% Femal and 42% Male

**4. What proportion of the entries in the movies table are actual movies and what proportion are television series, etc.?** This one is similar like question 3, so I use the same idea to do it, just replace gender by kind and use the combine of title and kind\_type table

dbGetQuery(imbd, "SELECT kind, COUNT(\*)\*100 / (SELECT COUNT(\*)   
 FROM title)   
 AS proportion  
 From title  
 JOIN kind\_type   
 ON title.kind\_id = kind\_type.id  
 GROUP BY kind")

## kind proportion  
## 1 episode 63  
## 2 movie 24  
## 3 tv movie 3  
## 4 tv series 3  
## 5 video game 0  
## 6 video movie 4

There are 63% episode, 24% movie, 3% tv movie, 3% tv series, 0% video game and 4% video movie.

**5.How many genres are there? What are their names/descriptions?** The new data set doesn't like the old data set which has genres' table, I even didn't find any column names genres in schema. So the first I need to find the genres. I look at the info\_type, I think it should be related to the type of the movie.

# look at info\_type table (won't show it in the homework)  
#dbGetQuery(imbd, "SELECT \* From info\_type LIMIT 5")

I saw gnres in there, and id is 3. The movie info table is connected with info type table, so I looked at what in the movies info only for info type id = 3 which is genres. The info column from movie info table contains the genre information, so I did distinct to see how many genres in there.

# distinct genres type, get genres' names  
genre = dbGetQuery(imbd, "SELECT DISTINCT info   
 From movie\_info   
 WHERE info\_type\_id = 3 ")  
genre[,1][1:10]

## [1] "Documentary" "Reality-TV" "Horror" "Drama" "Comedy"   
## [6] "Musical" "Talk-Show" "Mystery" "News" "Sport"

length(genre[,1])

## 32

There are 32 genres.

**6.List the 10 most common genres of movies, showing the number of movies in each of these genres.**

First I only select the rows that info\_type\_id = 3, which represent genres. Then I group the rows by genres, and counts how many rows for each genres. I let the return counts in decreasing order, so the first 10 rows is the 10 monst common genres, and the frequncy means how many movies in this genre.

dbGetQuery(imbd, "SELECT movie\_info.info, count(\*) AS Frequncy   
 From title  
 JOIN kind\_type   
 ON title.kind\_id = kind\_type.id  
 JOIN movie\_info   
 ON title.id = movie\_info.movie\_id  
 WHERE info\_type\_id = 3   
 AND kind\_type.kind = 'movie'  
 GROUP BY movie\_info.info  
 ORDER BY Frequncy DESC   
 LIMIT 10")

## info Frequncy  
## 1 Short 470488  
## 2 Drama 269898  
## 3 Comedy 180315  
## 4 Documentary 145018  
## 5 Romance 52324  
## 6 Thriller 51961  
## 7 Action 45077  
## 8 Horror 38620  
## 9 Animation 38461  
## 10 Crime 33010

**7.Find all movies with the keyword 'space'. How many are there? What are the years these were released? and who were the top 5 actors in each of these movies?** I first look at schema, I saw that I need to connect title, movie\_keyword and keyword together to count how many movies have "space". The title has information about movies, the movie keyword connect the title table with keyword table, and the keyword table contains keyword, so I set the keyword column in keyword table equal to "space", then I count how many rows in there to get how many movies have "space".

# count the number of movies where keyword = space  
dbGetQuery(imbd, "SELECT count(\*) AS MovieswithSpace   
 FROM title   
 JOIN kind\_type   
 ON title.kind\_id = kind\_type.id  
 JOIN movie\_keyword   
 ON movie\_keyword .movie\_id = title.id  
 JOIN keyword  
 ON keyword.id = movie\_keyword.keyword\_id  
 WHERE keyword.keyword = 'space'   
 AND kind = 'movie'")

## MovieswithSpace  
## 1 401

For the years of these movie, I use the same way as above, I only change the return value to be production\_year.

year = dbGetQuery(imbd, "SELECT production\_year   
 FROM title   
 JOIN kind\_type   
 ON title.kind\_id = kind\_type.id  
 JOIN movie\_keyword   
 ON movie\_keyword .movie\_id = title.id  
 JOIN keyword  
 ON keyword.id = movie\_keyword.keyword\_id  
 WHERE keyword.keyword = 'space'  
 AND kind = 'movie' ")  
year[,1][1:10]

## [1] 1965 1960 1957 1968 1984 2007 2008 2015 2004 2006

For the top 5 actors, if I want to get the infomation for keyword, movies, billing position and name, I need to connect 5 tables, so I create a temporary table which only contains movies'id and movies' name for movies who have "space" in the keyword. I named this table as space\_movie. Now I only need to connect space movie table to the cast info and name to get the name and billing postion information. For the top 5 actors, I decide the billing position in 1 to 5 is top 5 actors. So I only select the actors with billing position in 1 to 5 for each movices with space

# creat keyword and movie\_id temporary table  
dbGetQuery(imbd, "CREATE TEMPORARY TABLE space\_movie AS   
 SELECT title.id, title.title  
 FROM title   
 JOIN kind\_type   
 ON title.kind\_id = kind\_type.id  
 JOIN movie\_keyword   
 ON movie\_keyword.movie\_id = title.id  
 JOIN keyword  
 ON keyword.id = movie\_keyword.keyword\_id  
 WHERE keyword.keyword = 'space'  
 AND kind = 'movie'")  
  
# get the top five actors name, billing position in 1 to 5 for each movies.   
dbGetQuery(imbd, "SELECT name.name, cast\_info.nr\_order, space\_movie.title  
 FROM space\_movie  
 JOIN title  
 ON title.id = space\_movie.id  
 JOIN cast\_info  
 ON space\_movie.id = cast\_info.movie\_id  
 JOIN name  
 ON cast\_info.person\_id = name.id  
 WHERE cast\_info.nr\_order IN (1,2,3,4,5)   
 LIMIT 5 ")

## name nr\_order title  
## 1 Aaron, Aidan 3 The Astronaut  
## 2 Abart, Boleslaw 2 Test pilota Pirxa  
## 3 Abbott, Bud 1 Abbott and Costello Go to Mars  
## 4 Abdiani, Murwarid 1 The Meaning of Life  
## 5 Adams, Nick 1 Kaij没 daisens么

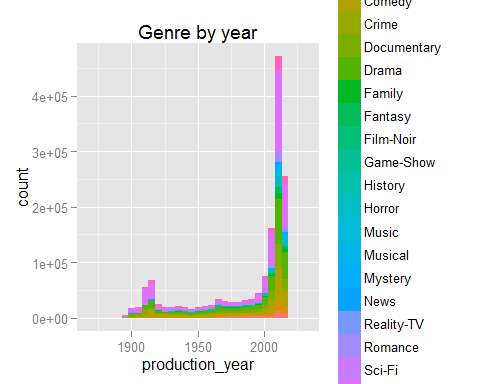
\*\* 8.Has the number of movies in each genre changed over time? Plot the overall number of movies in each year over time, and for each genre.\*\*

For this question, I first limit the info type id = 3, because only this type is genre, then I get the production year and info for all the rows, I save them into num movie. After get the year and genres information, I use ggplot to draw the plot.

num\_movie = dbGetQuery(imbd, "SELECT production\_year, info  
 From movie\_info   
 JOIN title  
 ON title.id = movie\_info.movie\_id  
 JOIN kind\_type   
 ON title.kind\_id = kind\_type.id  
 WHERE info\_type\_id = 3   
 AND kind = 'movie'  
 AND production\_year IS NOT NULL  
 ORDER BY production\_year")  
  
# use ggplot to draw the number of movies in each year  
library(ggplot2)  
ggplot(num\_movie, aes(x=production\_year , fill=info )) + geom\_bar() + ggtitle("Genre by year")

## stat\_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.

## Warning: position\_stack requires constant width: output may be incorrect



The x axis is produciton year, y axis is counts, and the bar is fill by different genres. We can see after 2000, a lot of movies were producted. because the ggplot automedictly group the years, so the second right bar is very high. From the plot I can see that "adult", "drama", and "short", those three type appear most in most years

**#9 Who are the actors that have been in the most movies? List the top 20.** For get the top 20 actors have been in the most, I think I need to get the all the actors names for each movies, then count the those actors' names, I will select the 20 actors whoes name appear most. In order to do that, I connect the title table with name table through cast info, now I have the information about movies and actors, I also connect the role type table to limit the role only to "actor" and "actress". I group the data by name and count the rows, order the counts by decreassing order. The first 20 rows is the top 20 actors who have been in the most movies.

dbGetQuery(imbd, "SELECT name, COUNT(\*) AS appear   
 FROM title   
 JOIN kind\_type   
 ON title.kind\_id = kind\_type.id  
 JOIN cast\_info   
 ON cast\_info.movie\_id = title.id  
 JOIN name  
 ON name.id = cast\_info.person\_id  
 WHERE kind = 'movie'  
 GROUP BY name  
 ORDER BY appear DESC  
 LIMIT 20")

## name appear  
## 1 Lubin, Siegmund 3291  
## 2 Blanc, Mel 2904  
## 3 Selig, William Nicholas 2498  
## 4 Sennett, Mack 1976  
## 5 Roach, Hal 1761  
## 6 Terry, Paul 1729  
## 7 MacLeod, Kevin 1702  
## 8 Laemmle, Carl 1662  
## 9 Christie, Al 1422  
## 10 Ambrosio, Arturo 1386  
## 11 Anderson, Gilbert M. 'Broncho Billy' 1310  
## 12 Bitzer, G.W. 1270  
## 13 Bray, John Randolph 1090  
## 14 Disney, Walt 1069  
## 15 Ince, Thomas H. 1049  
## 16 Brahmanandam 995  
## 17 Rivers, Scott 993  
## 18 Fox, William 988  
## 19 Powers, Pat 987  
## 20 Lantz, Walter 970

**10 Who are the actors that have had the most number of movies with "top billing", i.e., billed as 1, 2 or 3? For each actor, also show the years these movies spanned?**

I am confused with this question. so read some posts on Piazza, so what I did for this question is I find the top 5 actors with billing position 1, 5 for billing position 2 and 5 for billing position 3. I use sapply to loop the billing position from 1 to 3 to get exactly 5 actors for different billing position. So I connect title, cast\_info and name table together, cast info table has the information about billing position, I limit nr order which is billing position in 1, 2 and 3. Then I get the actors nanme for each movies, count how many time of those name appear, and select the first 5 actors who appear most.

```{r}

actor10 = sapply(1:3, function(x) {

y = paste0("SELECT name, nr\_order, cast\_info.movie\_id, COUNT(\*) AS appear

FROM title

JOIN kind\_type

ON title.kind\_id = kind\_type.id

JOIN cast\_info

ON title.id = cast\_info.movie\_id

JOIN name

ON cast\_info.person\_id = name.id

WHERE cast\_info.nr\_order = ",

x,

"

AND kind = 'movie'

GROUP BY name

ORDER BY appear DESC

LIMIT 5")

dbGetQuery(imbd,y)

})

actor10[[1]]

```

[1] "Blanc, Mel" "Kerrigan, J. Warren" "Anderson, Gilbert M. 'Broncho Billy'" "Lyons, Eddie"

[5] "Shin, Sung-il"

Here I only show the name of people who in the billing position 1.

For the years of these movies spanned, I use the same code as above, but I add MAX(production year) and MIN(production year) to get the year span.

```{r}

year = sapply(1:3, function(x) {

y = paste0("SELECT name, nr\_order, cast\_info.movie\_id, MAX(production\_year), MIN (production\_year), COUNT(\*) AS appear

FROM title

JOIN kind\_type

ON title.kind\_id = kind\_type.id

JOIN cast\_info

ON title.id = cast\_info.movie\_id

JOIN name

ON cast\_info.person\_id = name.id

WHERE cast\_info.nr\_order = ",

x,

"

AND kind = 'movie'

GROUP BY name

ORDER BY appear DESC

LIMIT 5")

dbGetQuery(imbd,y)

})

year[[1]]

```

[1] "Blanc, Mel" "Kerrigan, J. Warren" "Anderson, Gilbert M. 'Broncho Billy'" "Lyons, Eddie"

[5] "Shin, Sung-il"

[1] 2011 1924 1922 1924 1990

Here I only show the year that they start to act in movie

```{r}

dbGetQuery(imbd,"SELECT name, nr\_order, cast\_info.movie\_id, MAX(production\_year), COUNT(\*) AS appear

FROM title

JOIN cast\_info

ON title.id = cast\_info.movie\_id

JOIN name

ON cast\_info.person\_id = name.id

WHERE cast\_info.nr\_order = 1

GROUP BY name

ORDER BY appear DESC

LIMIT 5")

```

**11 Who are the 10 actors that performed in the most movies within any given year? What are their names, the year they starred in these movies and the names of the movies?**

For this question, I join the title, cast\_info and name tables together, I group the data by year, and names. I count the movies that the actor performed in that year. I order the data by decreasing order of the number of the movies. So the first 10 actors are the 10 actors that performed in the

dbGetQuery(imbd, "CREATE TABLE question11 AS

SELECT title.production\_year, name.name AS name, name.id AS id, COUNT(\*) AS appear

FROM title

JOIN kind\_type

ON title.kind\_id = kind\_type.id

JOIN cast\_info

ON title.id = cast\_info.movie\_id

JOIN name

ON cast\_info.person\_id = name.id

WHERE kind = 'movie'

GROUP BY title.production\_year, name.name

ORDER BY appear DESC

LIMIT 10")

dbGetQuery(imbd, "SELECT \* FROM question11 LIMIT 10")

#title.production\_year name id appear

#1 1903 Lubin, Siegmund 1212081 484

#2 1913 Lubin, Siegmund 1212081 373

#3 1914 Selig, William Nicholas 3899165 368

#4 1914 Lubin, Siegmund 1212081 358

#5 1915 Selig, William Nicholas 3899165 342

Here I only shows five of them, from this output, I see that Lubin, Siegmund acted in most movies for some years. I google him, he did a lots contribution to the motion picture industry.

dbGetQuery(imbd, "SELECT MIN(title.production\_year), name

FROM title

JOIN kind\_type

ON title.kind\_id = kind\_type.id

JOIN cast\_info

ON title.id = cast\_info.movie\_id

JOIN question11

ON cast\_info.person\_id = question11.id

WHERE kind = 'movie'

GROUP BY question11.name")

# MIN(title.production\_year) name

#1 2013 Edward, Noah

#2 1897 Lubin, Siegmund

#3 1987 MacLeod, Kevin

#4 1896 Selig, William Nicholas

Here is the year that they start in the movie. I google it, it correct for them.

**12 Who are the 10 actors that have the most aliases**

For this question, I group the actor's name and count how many aloases he or she has in the aka\_name table, since in the aka name table has person id represent actor's name, so I group by the person id, after that I connect the aka name table to the name table to get the real name for the actors.

dbGetQuery(imbd, "SELECT name.name, count(\*) AS appear  
 FROM aka\_name  
 JOIN name  
 ON aka\_name.person\_id = name.id  
 GROUP BY aka\_name.person\_id  
 ORDER BY appear DESC  
 LIMIT 10")

## name appear  
## 1 Franco, Jes煤s 78  
## 2 D'Amato, Joe 71  
## 3 Digard, Uschi 63  
## 4 Savage, Herschel 53  
## 5 Ho, Godfrey 50  
## 6 Silvera, Joey 42  
## 7 Albert, Kimson 40  
## 8 Le贸n, Nathanael 39  
## 9 Clark, Christoph 38  
## 10 Presova, Zuzana 38

I get some characters for some actors, name, so I can’t check all of them, so I only check some of them. I google D’ Amato, Joe, I find interesting thing is one of his aliases is Michael Wotruba, even not related to his name.

\*\*13Networks: Pick a (lead) actor who has been in at least 20 movies. Find all of the other actors that have appeared in a movie with that person. For each of these, find all the people they have appeared in a move with it. Use this to create a network/graph of who has appeared with who. Use the igraph or statnet packages to display this network. \*\*

From Piazza posts, I know this question will have a lot of actors. so in order to limit the actors I have, so I the actor that I pick is who has been in only 21 movies. Then I use the idactors to get all the movies that one person acted in, use those movies'id to get the people who also acted movies with this person, use the same way to get the actors who acted with the actors who acted with the first person. For saving time and eaier to do it for several times, I created tables to save the in the database.

```{r}

# find actots who acted exactly 21 movies

dbGetQuery(imbd, "SELECT actors.idactors, lname,fname,COUNT(\*) AS appear

FROM movies

JOIN acted\_in

ON movies.idmovies = acted\_in.idmovies

JOIN actors

ON actors.idactors = acted\_in.idactors

GROUP BY actors.idactors

HAVING appear = 21

LIMIT 10")

# random select one actor and find all the movies he or she acted in

dbGetQuery(imbd, "CREATE TABLE movie1 AS

SELECT DISTINCT movies.idmovies

FROM movies

JOIN acted\_in

ON movies.idmovies = acted\_in.idmovies

JOIN actors

ON actors.idactors = acted\_in.idactors

WHERE actors.idactors = 806

")

# find all the other actors that also acted in the movies with the person I selected

dbGetQuery(imbd, " CREATE TABLE first\_name AS

SELECT actors.idactors, lname,fname, billing\_position

FROM movies

JOIN movie1

ON movie1.idmovies = movies.idmovies

JOIN acted\_in

ON movies.idmovies = acted\_in.idmovies

JOIN actors

ON actors.idactors = acted\_in.idactors

WHERE movies.idmovies IN (2203, 2204, 2205, 2206)")

# get the find actors who acted with

dbGetQuery(imbd, " CREATE TABLE second\_name AS

SELECT actors.idactors, actors.lname, actors.fname, acted\_in.billing\_position, movies.idmovies

FROM movies

JOIN acted\_in

ON movies.idmovies = acted\_in.idmovies

JOIN actors

ON actors.idactors = acted\_in.idactors

JOIN first\_name

on first\_name.idactors = actors.idactors

")

```

Now I get the actor names to R to plot the igraph. I look at how to do the igraph on google. For ploting an igraph, I need to build a relation matirx, so my relation matrix has two columns, the first column is "from", the second column is "to", and the data in the matrix are all actors name. The meaning of this matrix is, for example, if one row has "Tom" in "from" column, and "Jack" in "to" column, when I draw the igraph, there will be a line connect this two person's name. Since I have a lot of actors in the data, if I plot all of them, then it will be very mess, so when I plot, I only plot the first 100 data to show what I get.

```{r}

# get the name

second\_name = dbReadTable(imbd, "second\_name")

# remove duplications

second\_name1 = second\_name[!duplicated(second\_name,incomparables = FALSE),]

# remove NAs

second\_name1 = na.omit(second\_name1)

#select actors in high billing position

top\_name = second\_name1[second\_name1$billing\_position %in% c(1,2,3),]

# start to build data frame for ploting igraph

#split data by idmovies

data = split(top\_name, top\_name$idmovies)

# build connect between actors who are in the same movie

to = lapply(1:length(data), function(x) rep(data[[x]]$lname[1], nrow(data[[x]])))

# combine the data together

to1 = data.frame(unlist(to))

data = do.call(rbind, data)

data = cbind(data, to)

# remove columns that won't be used

connection = data[,-c(1,3,4,5)]

names(connection) = c('from','to')

# get first 100 data

plot\_data = connection[1:100,]

# remove the connect that to himself or herself

for (i in 1:100)

{

if(plot\_data[i,1]==plot\_data[i,2])

plot\_data[i,2] = NA

}

# remove duplication

plot\_data = plot\_data[!duplicated(plot\_data),]

# remove NA

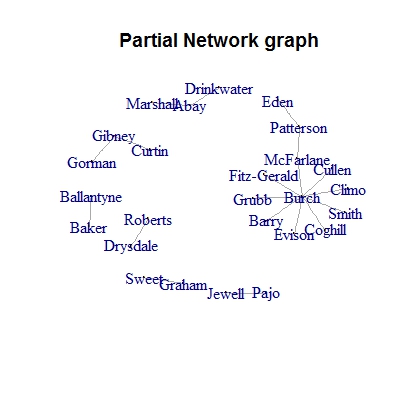
plot\_data = na.omit(plot\_data)

# plot igraph

plot\_data <- graph.data.frame(plot\_data, directed = T)

plot(plot\_data,layout=layout.kamada.kawai,vertex.shape='circle',vertex.label=V(g4)$name, vertex.size=1,edge.arrow.size=0,asp=F)

```



From this graph, because I only select the billing position in 1 to 3, so there are some names only connect to 1 names or two names.

9, 10, 11, 12 doing by R

**9 Who are the actors that have been in the most movies? List the top 20.**

For get the top 20 actors have been in the most, I think I need to get the all the actors names for each movies, then count the those actors' names, I will select the 20 actors whoes name appear most. In order to do that, I get the title table, name table, cast info, now I have the information about movies and actors, I use kind type table to get movies. in order to let it run faster in my laptop, I remove the column that I won't use. I table the person id in cast info table, order the counts by decreassing order. The first 20 rows is the top 20 actors who have been in the most movies.

```{r}

# get all the tables that I need

title = dbReadTable(imbd, "title")

cast\_info = dbReadTable(imbd, "cast\_info")

kind\_type = dbReadTable(imbd, "kind\_type")

name = dbReadTable(imbd, "name")

# get only movies

title = title[title$kind\_id == kind\_type[kind\_type$kind == "movie",1],]

# remove the columns that won't be used

title = title[,c("id", "title", "kind\_id", "production\_year")]

cast\_info = cast\_info[,-5]

#get only movies for cast\_info

cast\_info = cast\_info[cast\_info$movie\_id %in% title$id,]

#table person\_id

person\_id9 = as.data.frame(table(cast\_info$person\_id))

person\_id9 = person\_id9[order(person\_id9[,2], decreasing = T),]

#get the name for top 20 person\_id

name9 = lapply(1:20, function(x) name[name$id == person\_id9[x,1], 2])

unlist(name9)[1:5]

```

[1] “Lubin, Siegmund” “Blanc, Mel” “Selig, William Nicholas”

[4] “Sennett, Mack” “Roach, Hal”

Only show the first 5 names, it matches what I got from SQL

**10 Who are the actors that have had the most number of movies with "top billing", i.e., billed as 1, 2 or 3? For each actor, also show the years these movies spanned?**

I use lapply to split the data into billing position from 1 to 3. I table the person id for three different billing position, order the counts by decreassing order. The first 10 rows is the top 10 actors who have been in the most movies.

```{r}

# split data by billing position

name10 = lapply(1:3, function(x) na.omit(cast\_info[cast\_info$nr\_order == x,]))

# table the person\_id

name\_top = lapply(1:3, function(x) as.data.frame(table(name10[[x]]$person\_id), decreasing = T))

# get the first 10 in each list

name\_top\_order = lapply(1:3, function(x) name\_top[[x]][order(name\_top[[x]][,2], decreasing = T),][1:10,])

# mathc the person id to the name table to get name

name\_top1 = lapply(1:10, function(x) name[name$id %in% name\_top\_order[[x]], 2])

name\_top1[[1]]

```

[1] "Blanc, Mel" "Kerrigan, J. Warren" "Anderson, Gilbert M. 'Broncho Billy'" "Lyons, Eddie"

[5] "Shin, Sung-il"

For the years of these movies spanned, I first use cast info get the movies id that those atcors acted, then I use movies id to get the movie year in title table, after that I use max() and min() to get the year span for each actors.

```{r}

# get movie\_id for each actors

movie\_id = lapply(1:10, function(x) cast\_info[cast\_info$person\_id %in% [[x]], "movie\_id"])

#get the year for each movies

year = lapply(1:length(movie\_id), function(x) title[title$id %in% movie\_id[[x]], "production\_year"])

# find the max and min year

max\_year = lapply(1:length(year), function(x) max(year[[x]]))

min\_year = lapply(1:length(year), function(x) min(year[[x]]))

unlist(max\_year)[1:5]

```

[1] 2011 1924 1922 1924 1990

It matches what I got from SQL.

**11 Who are the 10 actors that performed in the most movies within any given year? What are their names, the year they starred in these movies and the names of the movies?**

For this question, I first merge the title and cast\_info table into table11, because title table has year and cast info has peson id. Then I table the table11 by year and person id, but unfortunatly, my laptop gave me an error said that no memory was free to do it, I try to remove all the informations that I didn't need from the table11, my laptop memory still not big enought to do it. Because the size of the title table or cast info table is around 2 to 3 GB, it is hard for my laptop to run it. I only wrote the code, but didn't run it.

For this question, I join the title, cast\_info and name tables together, I group the data by year, and names. I count the movies that the actor performed in that year. I order the data by decreasing order of the number of the movies. So the first 10 actors are the 10 actors that performed in the most movies with in any given yaer. I also create a table as to save those information, because I will be eaier for me to do the rest of this question.

```{r}

# rename the title table

name(title) = c("movie\_id", "title", "kind\_id", "production\_year")

#merge title with cast\_info

table11 = merge(title, cast\_info, by = "movie\_id")

# table () table11 by person\_id and peroduction\_year

person\_id11 = as.data.frame(table(table11$person\_id, table11$peroduction\_year))

# get the first 10 person\_id

person\_id11 = person\_id11[order(person\_id11[,2], decreasing = T),][1:10,]

# mathc the person id to the name table to get name

name11 = name[name$id %in% person\_id11, 2]

name11

```

In order to get the started year. I use person ids that I got from above to find all the movie ids that associate person ids, which is find all the movies that the person acted in. Then use the movie ids to get the movies' year. For getting the start year for each actors, I split the data by person\_id and use lapply to find the minimum year value.

```{r}

# find the movie\_id

movie\_id11 = cast\_info[cast\_info %in% person\_id11,c("person\_id","movie\_id")]

#use movie\_id to find the peroduction\_yeare

year = title[title %in% movie\_id11$movie\_id,"peroduction\_year"]

# combine year with person\_id and movie\_id

year = cbind(year, movie\_id11)

# split year by person\_id

year = split(year, year$person\_id )

# find minimum year for each person\_id

year = lapply(1:length(year), function(x) min(year[[x]]$peroduction\_yeare))

year = do.call(rbind, year)

# use person\_id to find the name

name11 = name[name$id %in% year$person\_id, 2]

year = cbind(year, name11)

```

Since I already have movie ids inforamtion, it is eaier to do this one, I just use movie ids to find the movie titles from title table.

```{r}

# use movie\_id that I got from above to find the title

title11 = title[title %in% movie\_id11$movie\_id,"title"]

```

\*\*12 Who are the 10 actors that have the most aliases\*\*

For this question, I take aka name table out from SQL named it aka name. In order to make it run faster, I only get the person id out from the aka name.Then I use table() function to get the counts for each person id. make the order of the counts in decreasing, so the first 10 actors are the 10 actors that have the most aliases.

```{r}

# get aka\_name table

aka\_name = dbReadTable(imbd, "aka\_name")

# get "person\_id" column names aka\_name1

person\_id12 = aka\_name[,"person\_id"]

# table person\_id and make into data.frame

name12 = as.data.frame(table(person\_id12))

# order the table

name12 = name12[order(name12[,2], decreasing = T),]

# get the first 10 person\_id, and match to the name table to get the name.

ture\_name12 = lapply(1:10, function(x) name[name$id == name12[x,1], 2])

unlist(ture\_name12)

```

[1] "Franco, Jes煤s" "D'Amato, Joe" "Digard, Uschi" "Savage, Herschel" "Ho, Godfrey" "Silvera, Joey"

[7] "Albert, Kimson" "Le贸n, Nathanael" "Clark, Christoph" "Presova, Zuzana" "Pietila, Bonita" "Kronos, Donald Arthur"

[13] "Redgrave, Joana" "Martin, Jon" "Sarno, Joseph W." "Kalerman, Sandra" "Mrazkova, Jana" "Caffarel, Jos茅 Mar铆a"

[19] "Hwang, Jang Lee" "Rafi, Mohammad"

It matches what I get from SQL