Katherine Olson

STA 141 Assignment 5

Report

I did this assignment by myself and developed and wrote the code for each part by myself, drawing only from class, section, Piazza posts and the Web. I did not use code from a fellow student or a tutor or any other individual.

Based on what it sounded like Duncan wants, for questions 1 - 4 I am using all types of movies/shows/programs and for questions 5 and on I am using just movies, not including TV and video movies. I have all of my code in an appendix at the end and included some queries in the report as Michael requested. I ended up only including them in the report for the first six questions, the rest had multiple parts or were too complicated.

Question 1: How many actors are there in the database? How many movies?

3,492,018 distinct actors and 3,527,732 distinct movies/shows/programs. It is interesting how there are about the same actors as there are movies/shows/programs.

Actor query:

```
dbGetQuery(imdb, "SELECT COUNT(DISTINCT person_id)
        FROM cast_info
        WHERE role_id = 2
 OR role_id = 1;")
```

Movie query:

```
dbGetQuery(imdb, "SELECT COUNT(DISTINCT id)
        FROM title;")
```

Question 2: What time period does the database cover?

The database has years from 1874 to 2025. The hardest part of this question for me was figuring out where the years were stored and deciding not to make the condition limit the range of the years it found. The years 1874 to 2025 do not make a lot of sense, but they are the years in the database, as the question asked.

Query:

```
dbGetQuery(imdb, "SELECT MIN(production_year), MAX(production_year)
            FROM title;")
```

Question 3: What proportion of the actors are female? male?

I found that 35.27% are female and 64.61% are male. This leaves 0.02% that are null. It is interesting, but not surprising that there are almost twice as many male actors as female actors. The hardest part of this question for me was trying to get the division to work in SQL, but not succeeding.

Total query:

```
total = dbGetQuery(imdb, "SELECT COUNT(DISTINCT person_id)
                          FROM cast_info
                          WHERE role_id = 2
                          OR role_id = 1;")
```

Female query:

```
female = dbGetQuery(imdb, "SELECT COUNT(DISTINCT person_id)
                           FROM cast_info, name
                           WHERE role_id IN (1, 2)
                           AND gender = 'f'
                           AND cast_info.person_id = name.id;")
```

Male query:

```
male = dbGetQuery(imdb, "SELECT COUNT(DISTINCT person_id)
                         FROM cast_info, name
                         WHERE role_id IN (1, 2)
                         AND gender = 'm'
                         AND cast_info.person_id = name.id;")
```

I did the division in R (see code appendix).

Question 4: What proportion of the entries in the movies table are actual movies and what proportion are television series, etc.?

Kind	Movie	TV Series	TV Movie	Video Movie	TV Mini Series	Video Game	Episode
Proportion	0.2491	0.0353	0.0341	0.0416	0.0000	0.0043	0.6356

The table tells us that most entries are movies and episodes, with the majority being episodes. The hardest part of this question for me was trying to the division part in SQL, but not being able to figure it out and having to do it in R.

Total query:

```
totalTitles = dbGetQuery(imdb, "SELECT COUNT(DISTINCT id)
                                FROM title;")
```

Movie query:

```
movie = dbGetQuery(imdb, "SELECT COUNT(DISTINCT id)
                          FROM title
                          WHERE kind_id = 1;")
```

... Repeated for the other 7 and did the division in R(See code appendix).

Question 5: How many genres are there? What are their names/descriptions?

There are 32 genres. They are Documentary, Reality-TV, Horror, Drama, Comedy, Musical, Talk-Show, Mystery, News, Sport, Sci-Fi, Romance, Family, Short, Biography, Music, Game-Show, Adventure, Crime, War, Fantasy, Thriller, Animation, Action, History, Adult, Western, Lifestyle, Film-Noir, Experimental, Commercial, and Erotica. The hardest part of this question was finding where genre was stored.

Query:

```
dbGetQuery(imdb, "SELECT DISTINCT(info)
                  FROM movie_info
                  WHERE info_type_id = 3;")
```

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

	info	COUNT(*)
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

My results can be seen in the table on the left. The hardest part of this question for me was figuring out that I needed to use GROUP BY and ORDER BY, which I found on posts from piazza. The results of the most common genres are not surprising, they are all genres that I would expect to be common.

Query:

```
dbGetQuery(imdb, "SELECT info, COUNT(*)
        FROM movie_info, title
        WHERE info_type_id = 3
        AND kind_id = 1
        AND title.id = movie_id
        GROUP BY info
        ORDER BY COUNT(*) DESC
        LIMIT 10;")
```

Question 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 found 401 movies with the keyword space, using just movies (kind_id = 1). The years these were released are 1911, 1918, 1922, 1925, 1930, 1946, 1947, 1950, 1951, 1953-1975, 1977-

1994, and 1996-2018. Showing all top 5 actors for each movie would be way too long, so I am showing the results for the first 3:

[[1]]

	person_id	name	nr_order	title
1	661113	Franchi, Franco	1	002 operazione Luna
2	935665	Ingrassia, Ciccio	2	002 operazione Luna
3	3172528	Randall, MÃ³nica	3	002 operazione Luna
4	3291555	Sini, Linda	4	002 operazione Luna
5	3286328	Silva, MarÃa	5	002 operazione Luna

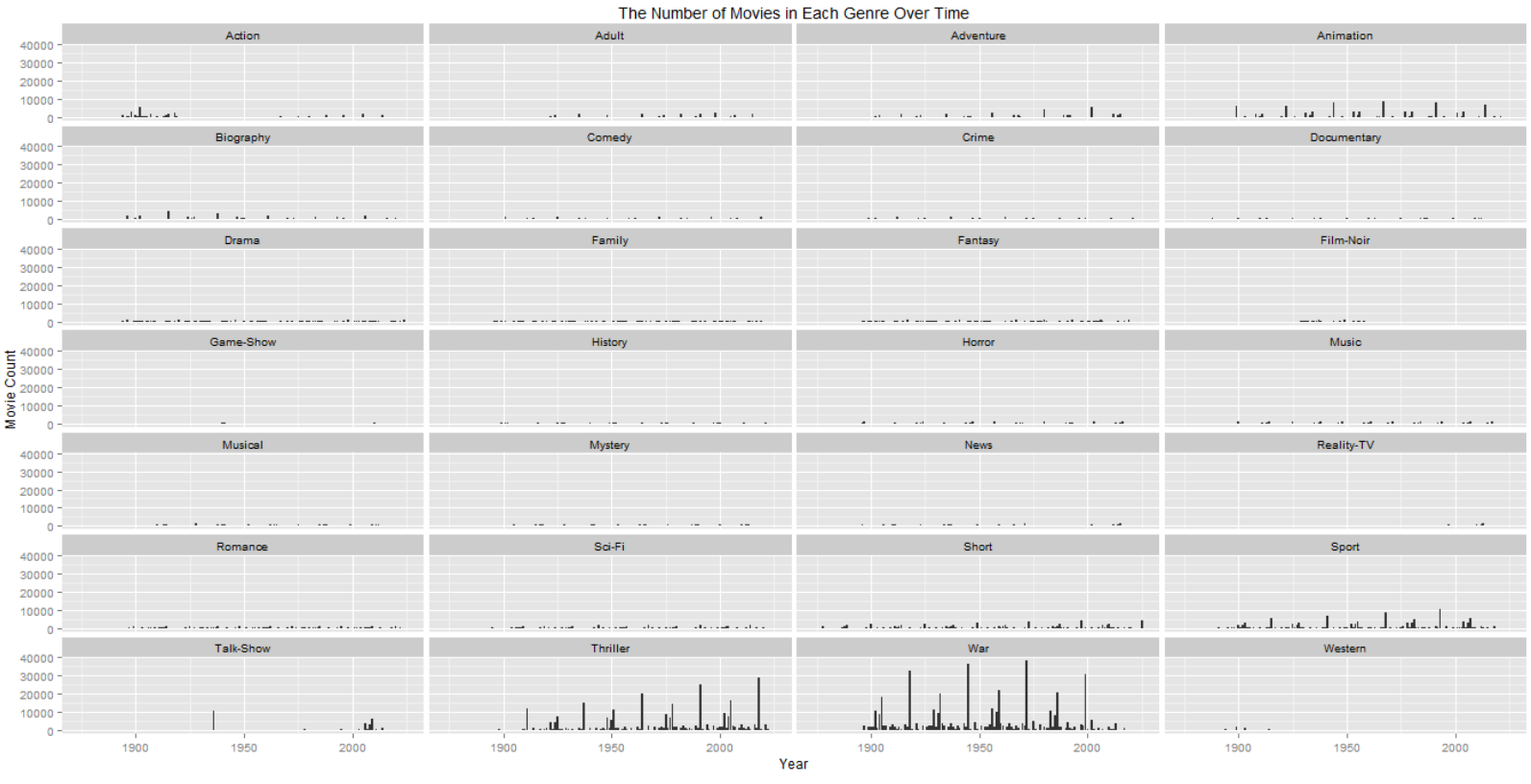
[[2]]

	person_id	name	nr_order	title
1	374630	Clark, Ken	1	12 to the Moon
2	2845023	Kobi, Michi	2	12 to the Moon
3	402504	Conway, Tom	3	12 to the Moon
4	506884	Dexter, Anthony	4	12 to the Moon
5	2164166	Wengraf, John	5	12 to the Moon

[[3]]

	person_id	name	nr_order	title
1	899083	Hopper, William	1	20 Million Miles to Earth
2	3357735	Taylor, Joan	2	20 Million Miles to Earth
3	1633148	Puglia, Frank	3	20 Million Miles to Earth
4	2243618	Zaremba, John	4	20 Million Miles to Earth
5	860958	Henry, Thomas Browne	5	20 Million Miles to Earth

Question 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.
The plot can be seen on the previous page. Since some genres have way more movies than other genres, the pattern is harder to see for less popular genres. I was surprised by how many genres have ups and downs. This makes sense for war, but is more surprising in thriller, animation, and sport.

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

	COUNT(DISTINCT movie_id)	name
1	2747	Cavaco, Manuel
2	2661	Cerdeira, AntÃ³nio Pedro
3	2117	Davidson, Doug
4	2049	Costa, Orlando
5	2008	Castelo, VirgÃlio
6	1839	CatarrÃ©, JoÃ£o
7	1837	Corrula, Jorge
8	1807	Colbert, Stephen
9	1745	Cummings, Jim
10	1670	Dattilo, Bryan
11	1564	Clark, Dick
12	1546	Daly, Carson
13	1399	CÃ´rte-Real, Francisco
14	1369	Davis, Mark
15	1226	Cooper, Anderson
16	1201	CÃ©sar, JÃºlio
17	1192	Castellaneta, Dan
18	1160	Cosby, Bill
19	1156	Chittell, Chris
20	1150	Christian, Shawn

My results can be seen on the left. I struggled the most with figuring out how to connect three tables and how to go about getting the information I needed for each actor through each title.

Question 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 started my getting the following table:

	name	COUNT(DISTINCT movie_id)	production_year	person_id
1	Blanc, Mel	473	1965	195959
2	Shin, Sung-il	394	1972	1856461
3	Kerrigan, J. Warren	370	1913	1042765
4	Moran, Lee	368	1912	1397573
5	Lyons, Eddie	354	1919	1223506
6	Anderson, Gilbert M. 'Broncho Billy'	320	1910	54832

7	Hardy, Oliver	311	1951	825587
8	Pollard, 'Snub'	301	1919	1608412
9	Richardson, Jack	294	1918	1693188
10	Garcia, Eddie	292	1966	695721

I then applied over each 10 person_ids to get the range:
1. Blanc, Mel: 1944 to 2011
2. Shin, Sung-il: 1960 to 1992
3. Kerrigan, J. Warren: 1910 to 1934
4. Moran, Lee: 1912 to 1933
5. Lyons, Eddie: 1911 to 1924
6. Anderson, Gilbert M. 'Broncho Billy': 1904 to 1922
7. Hardy, Oliver: 1914 to 1982
8. Pollard, 'Snub': 1915 to 1933
9. Richardson, Jack: 1911 to 1929
10. Garcia, Eddie: 1953 to 2013

It is interesting how there is a combination of actors who were in movies for most of their lives and actors who acted for 20 or 30 years.

Question 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?

	name	production_year	COUNT(DISTINCT movie_id)	title	person_id
1	Sennett, Mack	1909	125	With Her Card	1833458
2	Johnson, Arthur V.	1909	116	With Her Card	977755
3	Barnett, Chester	1913	105	With Her Rival's Help	127463
4	White, Pearl	1913	104	With Her Rival's Help	3452431
5	Moore, Owen	1909	102	With Her Card	1394456
6	Kerrigan, J. Warren	1912	99	White Treachery	1042765
7	Onoe, Matsunosuke	1918	92	Ã›wabami no ocho	1509290
8	Kerrigan, J. Warren	1911	86	When East Comes West	1042765
9	Onoe, Matsunosuke	1915	86	Ã”okubo hikozaemon kiso manyuki	1509290
10	Onoe, Matsunosuke	1914	84	Ã”shio HeihachirÃ´	1509290

I started this problem by getting the above table. Then I applied over each 10 to get the names of the movies. Since the results have about 100 movies per actor, which would be a lot to print, I am only showing around the first 6 and last 6 movies for each actor. My results are below:
1. Sennett, Mack 1909:

```
  [1] "A Baby's Shoe"                  "A Burglar's Mistake"
  [3] "A Change of Heart"              "A Convict's Sacrifice"
  [5] "A Corner in Wheat"              "A Drunkard's Reformation"
   ...
[113] "Those Awful Hats"               "Through the Breakers"
```

[115] "Tis an Ill Wind That Blows No Good" "To Save Her Soul"
[117] "Tragic Love" "Trying to Get Arrested"

2. Johnson, Arthur V. 1909:
[1] "A Baby's Shoe" "A Burglar's Mistake"
[3] "A Change of Heart" "A Convict's Sacrifice"
[5] "A Corner in Wheat" "A Drunkard's Reformation"
...
[111] "Twin Brothers" "Two Memories"
[113] "Two Women and a Man" "Was Justice Served?"
[115] "What Drink Did" "With Her Card"

3. Barnett, Chester 1913:
[1] "A Bachelor's Finish" "A Call from Home" "A Child's Influence" "A Dip Into Society"
[5] "A Hidden Love" "A Joke on the Sheriff" "A News Item" "A Night at the Club"
...
[101] "Who Is in the Box?" "Who Is the Goat?" "Will Power" "Willie's Great Scheme"
[105] "With Her Rival's Help"

4. White, Pearl 1913:
[1] "A Call from Home" "A Child's Influence" "A Dip Into Society" "A Hidden Love"
[5] "A Joke on the Sheriff" "A News Item" "A Night at the Club" "A Night in Town"
...
[97] "When Duty Calls" "When Love Is Young" "Where Charity Begins" "Who Is in the Box?"
[101] "Who Is the Goat?" "Will Power" "Willie's Great Scheme" "With Her Rival's Help"

5. Moore, Owen 1909:
[1] "A Baby's Shoe" "A Burglar's Mistake"
[3] "A Change of Heart" "A Convict's Sacrifice"
[5] "A Corner in Wheat" "A Drunkard's Reformation"
...
[97] "Twin Brothers" "Two Memories"
[99] "Two Women and a Man" "Was Justice Served?"
[101] "What Drink Did" "With Her Card"

6. Kerrigan, J. Warren 1912:
[1] "A Bad Investment" "A Life for a Kiss" "After School"
[4] "An Assisted Elopement" "An Idyl of Hawaii" "An Innocent Grafter"
...
[94] "The Wordless Message" "The Would-Be Heir" "Their Hero Son"

[97] "Under False Pretenses" "Where Broadway Meets the Mountains" "White Treachery"

7. Onoe, Matsunosuk 1918:
[1] "Akakabe myÃ´jin kaibyÃ´ kidan" "Amakusa ShirÃ´"
[3] "Araki mataemon" "AsagaÃ´ nikki"
[5] "BanshÃ» sarayashiki" "BanzÃ»in ChÃ´bei"
...
[87] "YotsÃ»ya kaidan jitsÃ»hi kanetani goro" "YÃ»me no hankÃ»ro asakusa oadaÃ»chi"
[89] "ÃŽzumo kaidan" "Ã”okubo hikozaemon"
[91] "Ã›shiwaka kÃ´kingÃ´ to Tamagawa Oyoshi" "Ã›wabami no ocho"

8. Kerrigan, J. Warren 1911:
[1] "$5000 Reward, Dead or Alive" "A California Love Story" "A Cowboy's Sacrifice"
[4] "A Daughter of Liberty" "A Pittsburgh Millionaire" "A Trooper's Heart"
...
[82] "The Witch of the Range" "The Yiddisher Cowboy" "Three Daughters of the West"
[85] "Three Million Dollars" "When East Comes West"

9. Onoe, Matsunosuke 1915:
[1] "Adachihara ubagaike yÃ»rai" "Akagaki Genzo" "Akegarasu juyushi"
[4] "Amagasa rokuro" "Araki mataemon" "Asakusa kanzenon rishÃ´ki"
...´"
[82] "Uwajima sÃ´dÃ´" "Uzura gonbei" "Yuten kichimatsu"
[85] "YÃ»dachi monzÃ´" "Ã”okubo hikozaemon kiso manyuki"

10. Onoe, Matsunosuke 1914:
[1] "Akechi Samanosuke" "Akogi no Heiji"
[3] "Arao Hidemaru" "Arima Gennosuke"
[5] "Asamagatake" "Bancho sarayashiki"
...
[79] "Tsukahara Bokuden" "Umagashira MatagorÃ´"
[81] "Yasuda sakubei" "Yoshiwara kaidan teburi bÃ´zu"
[83] "Yumiharizuki" "Ã”shio HeihachirÃ´"

The hardest part about this question for me was figuring out what the question wanted, and then figuring out I could not do it all in one query.

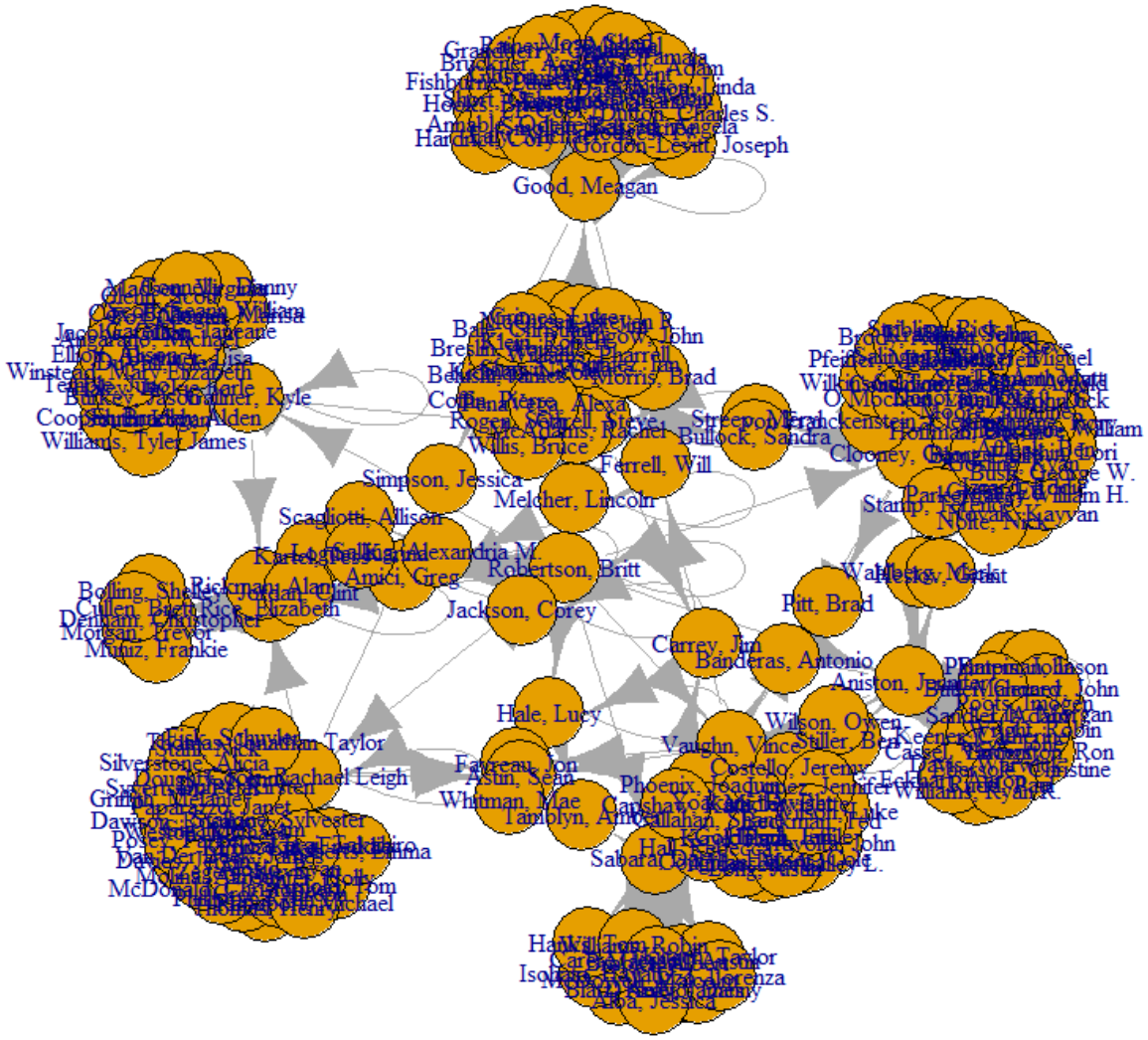
Question 12: Who are the 10 actors that have the most aliases (i.e., see the aka_names table).

	name	COUNT(aka_name.name)
1	Castellaneta, Dan	125523

My results can be seen on the left. Once I figured out how to connect

2	Azaria, Hank	91329	three tables in the earlier questions,
3	Shearer, Harry	69853	this question did not take me too
4	MacNeille, Tress	64832	long to figure out. It is crazy how
5	Jeremy, Ron	59179	many aliases some of the actors
6	Savage, Herschel	58353	have.
7	Welker, Frank	55195	
8	O'Brien, Conan	52663	
9	Silvera, Joey	48762	
10	Cartwright, Nancy	44935	

Question 13: Networks: 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.

I Started this question by picking Britt Robertson as my actress. I wanted someone who had been more than 20 movies and someone I have seen in a lot of movies that would be fun to use. I did not want to pick someone who had been in a crazy amount of movies that would make my graph too huge. I tried a few different actors and when I found out that Britt was in 24 movies (defining movie as kind_id = 1), I decided to go with her. After attempting to find every single actor that was in a movie with every actor she had ever worked with, it was clear that I had way too many actors to create a readable graph. I kept limiting the result of each movie by highest paid and ended up just picking the highest paid actor that was connected to the actor (either Britt or a co-actor of Britt) for each movie. I ended up with 222 unique actors and a list of 257 pairs to graph. My graph can be seen above. There were still too many actors, so all of the names overlap, but it could be worse. A good amount of names are readable and the graph gives a good feel for the overall shape.

Resources

- http://stackoverflow.com/questions/10529764/sqlite-reverse-the-order-of-the-result-set
- http://stackoverflow.com/questions/3996779/how-to-divide-two-columns
- http://stackoverflow.com/questions/20349883/selecting-na-values-from-sql-file-in-r
- http://statistics.berkeley.edu/computing/r-vectors-matrices
- http://stackoverflow.com/questions/5663888/trying-to-remove-all-margins-so-that-plot-region-comprises-the-entire-graphic8
- http://www.cookbook-r.com/Graphs/Facets_(ggplot2)/
- Piazza
- Lecture, office hours, and discussion
- The class website

Code Appendix

```
# Getting the data:
install.packages("RSQLite")
library("RSQLite")
imdb = dbConnect(drv = SQLite(), dbname = "~/UC Davis/STA 141/HW 5/lean_imdbpy.db")
dbListTables(imdb)

# Question 1
# ========
# How many actors are there in the database? How many movies?
```

```
# Actors:
# Finding out that 1 = actor and 2 = actress
dbGetQuery(imdb, "SELECT DISTINCT *
        FROM role_type
        LIMIT 20")
# Getting the count:
dbGetQuery(imdb, "SELECT COUNT(DISTINCT person_id)
        FROM cast_info
        WHERE role_id = 2
        OR role_id = 1;")

# Movies:
# Getting count for just all movies/shows/programs:
dbGetQuery(imdb, "SELECT COUNT(DISTINCT id)
        FROM title;")
```

 # Question 2

```
# ========
# What time period does the database cover?
dbGetQuery(imdb, "SELECT MIN(production_year), MAX(production_year) FROM title;")
```

 # Question 3

```
# ========
# What proportion of the actors are female? male?
# Total:
total = dbGetQuery(imdb, "SELECT COUNT(DISTINCT person_id)
                FROM cast_info
                WHERE role_id = 2
                OR role_id = 1;")
# Female:
female = dbGetQuery(imdb, "SELECT COUNT(DISTINCT person_id)
                FROM cast_info, name
                WHERE role_id IN (1, 2)
                AND gender = 'f'
                AND cast_info.person_id = name.id;")
# Male:
male = dbGetQuery(imdb, "SELECT COUNT(DISTINCT person_id)
              FROM cast_info, name
              WHERE role_id IN (1, 2)
```

```
            AND gender = 'm'
            AND cast_info.person_id = name.id;")

female / total # 0.3537
male / total # 0.6461

# Question 4
# ========
# What proportion of the entries in the movies table are actual movies and what proportion
# are television series, etc.?
# 1 = movie    2,5,6,7 = not    3,4 = tv movie, video movie
dbGetQuery(imdb, "SELECT *
        FROM kind_type
        LIMIT 20;")
# Total
totalTitles = dbGetQuery(imdb, "SELECT COUNT(DISTINCT id)
                  FROM title;")
# Each of the seven:
movie = dbGetQuery(imdb, "SELECT COUNT(DISTINCT id)
               FROM title
               WHERE kind_id = 1;")
tvSeries = dbGetQuery(imdb, "SELECT COUNT(DISTINCT id)
                 FROM title
                 WHERE kind_id = 2;")
tvMovie = dbGetQuery(imdb, "SELECT COUNT(DISTINCT id)
                FROM title
                WHERE kind_id = 3;")
vidMovie = dbGetQuery(imdb, "SELECT COUNT(DISTINCT id)
                 FROM title
                 WHERE kind_id = 4;")
tvMini = dbGetQuery(imdb, "SELECT COUNT(DISTINCT id)
               FROM title
               WHERE kind_id = 5;")
videoGame = dbGetQuery(imdb, "SELECT COUNT(DISTINCT id)
                  FROM title
                  WHERE kind_id = 6;")
episode = dbGetQuery(imdb, "SELECT COUNT(DISTINCT id)
                FROM title
                WHERE kind_id = 7;")
# Getting percents:
```

```
all = c(movie[[1]], tvSeries[[1]], tvMovie[[1]], vidMovie[[1]], tvMini[[1]],
      videoGame[[1]], episode[[1]]) / totalTitles[[1]]

# Question 5
# ========
# How many genres are there? What are their names/descriptions?
# Finding out that we want info with info_type_id = 3 (genres):
dbGetQuery(imdb, "SELECT *
        FROM info_type")
# Getting the distinct genres:
dbGetQuery(imdb, "SELECT DISTINCT(info)
        FROM movie_info
        WHERE info_type_id = 3;")

# Question 6
# ========
# List the 10 most common genres of movies, showing the number of movies in each of
# these genres.
dbGetQuery(imdb, "SELECT info, COUNT(*)
         FROM movie_info, title
         WHERE info_type_id = 3
         AND kind_id = 1
         AND title.id = movie_id
         GROUP BY info
         ORDER BY COUNT(*) DESC
         LIMIT 10;")

# Question 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?
# First find what keyword_id = 'space':
dbGetQuery(imdb, "SELECT id
        FROM keyword
        WHERE keyword = 'space'") # 9680

# Then find how many:
dbGetQuery(imdb, "SELECT COUNT(*)
        FROM title, movie_keyword
        WHERE title.id = movie_id
```

```
        AND keyword_id = 9680
        AND kind_id = 1;") # 401

# Then find the years released:
dbGetQuery(imdb, "SELECT DISTINCT production_year
        FROM title, movie_keyword
        WHERE title.id = movie_id
        AND keyword_id = 9680
        AND kind_id = 1
        ORDER BY production_year;")

# Then find top 5 actors in each of these movies:

# Getting the IDs for the 401 movies
movIDs = dbGetQuery(imdb, "SELECT movie_id FROM title, movie_keyword WHERE title.id
= movie_id AND
         keyword_id = 9680 and kind_id = 1;")

# Within each movie, find top 5 billed actors
Top5Actors = function(movie_id, imdb){
  q7 = sprintf("SELECT DISTINCT person_id, name, nr_order, title
            FROM cast_info, name, title
            WHERE person_id = name.id
            AND title.id = movie_id
            AND role_id IN (1, 2)
            AND movie_id = %i
            AND nr_order IS NOT NULL
            ORDER BY nr_order
            LIMIT 5;", movie_id)
  result = dbGetQuery(imdb, q7)
  return(result)
}
top5 = lapply(movIDs$movie_id, function(ids) Top5Actors(ids, imdb))
# Question 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.
q8 = dbGetQuery(imdb, "SELECT production_year, COUNT(DISTINCT movie_id), info
        FROM title, movie_info
        WHERE kind_id = 1
```

```
        AND title.id = movie_id
        AND info_type_id = 3
        GROUP BY production_year, info;")

library("ggplot2")
plotQ8 = ggplot(q8, aes(x = production_year, y = q8[, 2])) + geom_histogram(stat ="identity")
plotQ8 + facet_wrap(~info, ncol = 4, drop = TRUE) + ggtitle("The Number of Movies in Each
        Genre Over Time") + xlab("Year") + ylab("Movie Count")

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

# Using SQL:
dbGetQuery(imdb, "SELECT COUNT(DISTINCT movie_id), name
        FROM cast_info, name, title
        WHERE person_id = name.id
        AND cast_info.id = title.id
        AND role_id IN (1, 2)
        AND kind_id = 1
        GROUP BY person_id
        ORDER BY COUNT(DISTINCT movie_id) DESC
        LIMIT 20;")

# Using R:
title = dbReadTable(imdb, "title")
cast_info = dbReadTable(imdb, "cast_info")
name = dbReadTable(imdb, "name")
movies = title[which(title$kind_id == 1), ]
actors = cast_info[which((cast_info$role_id == 1) | (cast_info$role_id == 2)), ]
sameMovie = match(movies$id, actors$movie_id)
actors = actors[sameMovie, ]

samePerson = match(actors$person_id, name$id)
name8 = name[samePerson, ]

actSplit = split(actors, actors$person_id)
numMovs = lapply(1:333655, function(i) length(unique(actSplit[[i]]$movie_id)))

# Question 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?
q10 = dbGetQuery(imdb, "SELECT name, COUNT(DISTINCT movie_id), production_year,
          person_id
         FROM cast_info, name, title
         WHERE title.id = cast_info.movie_id
         AND cast_info.person_id = name.id
         AND kind_id = 1
         AND role_id IN (1, 2)
         AND nr_order IN (1, 2, 3)
         GROUP BY person_id
         ORDER BY COUNT(DISTINCT movie_id) DESC
         LIMIT 10;")
# Getting the min and max of production year for those 10:
GetYearRange = function(person_id, imdb){
  qminmax = sprintf("SELECT MIN(production_year), MAX(production_year)
              FROM title, cast_info
              WHERE title.id = cast_info.movie_id
              AND person_id = %i
              AND role_id IN (1, 2)
              AND nr_order IN (1, 2, 3)
              AND kind_id = 1;", person_id)
  return(dbGetQuery(imdb, qminmax))
}
# Mapply over both for all 10 to get range:
ranges = lapply(q10$person_id, function(id) GetYearRange(id, imdb))

# Question 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?
q11 = dbGetQuery(imdb, "SELECT name, production_year, COUNT(DISTINCT movie_id),
          title, person_id
         FROM title, cast_info, name
         WHERE kind_id = 1
         AND role_id IN (1, 2)
         AND title.id = cast_info.movie_id
         AND cast_info.person_id = name.id
         GROUP BY person_id, production_year
```

```
        ORDER BY COUNT(DISTINCT movie_id) DESC
        LIMIT 10")

GetTitles = function(person_id, production_year, imdb){
  quer = sprintf("SELECT DISTINCT movie_id, title
              FROM title, cast_info
              WHERE production_year = %i
              AND person_id = %i
              AND kind_id = 1
              AND role_id IN (1, 2)
              AND title.id = movie_id", production_year, person_id)
  return(dbGetQuery(imdb, quer)[2])
}

titles = mapply(function(id, year) GetTitles(id, year, imdb), q11$person_id,
         q11$production_year)

# Question 12
# =========
# Who are the 10 actors that have the most aliases (i.e., see the aka_names table).
dbGetQuery(imdb, "SELECT name.name, COUNT(aka_name.name)
        FROM name, aka_name, cast_info
        WHERE name.id = aka_name.person_id
        AND name.id = cast_info.person_id
        AND role_id IN (1, 2)
        GROUP BY name.id
        ORDER BY COUNT(aka_name.name) DESC
        LIMIT 10;")

# Question 13
# =========
# Returns the person_ids of all actors in the movie (including the actor we are using)
FindActors = function(movieID, imdb){
  # Limiting to top 4 billed actors:
  qr2 = sprintf("SELECT person_id
              FROM cast_info
              WHERE movie_id = %i
              AND role_id IN (1, 2)
              AND nr_order IS NOT NULL
              ORDER BY nr_order
```

```
            LIMIT 1", movieID)
  peeps = dbGetQuery(imdb, qr2)
  return(peeps$person_id)
}

# Function that takes in an actor ID and returns a list of person_ids of co-actors:
GetCoActors = function(actorID, imdb){
  qr = sprintf("SELECT title.id
            FROM title, cast_info
            WHERE title.id = movie_id
            AND person_id = %i
            AND kind_id = 1;", actorID[[1]])
  titleIDs = dbGetQuery(imdb, qr)

  # List where each element in list is a vector with $person_id for each co-actor:
  # (May have repeats of ids)
  allActors = lapply(titleIDs$id, FindActors, imdb)
  allActors = unlist(allActors)
  allActors = unique(allActors) # Removng repeats
  return(allActors)
}

# Takes in person_id, returns name
GetNames = function(person_id, imdb){
  qr = sprintf("SELECT name
            FROM name
            WHERE name.id = %i", person_id)
  return(dbGetQuery(imdb, qr)$name)
}

idBritt = dbGetQuery(imdb, "SELECT id
                FROM name
                WHERE name = 'Robertson, Britt';")
brittCoActors = GetCoActors(idBritt, imdb)
brittCANames = lapply(brittCoActors, GetNames, imdb)
brittCANames = unlist(brittCANames)
# Now need to loop over GetCoActors for all of Britt's co-actors to get all of her
# co-actors co-actors:
coActCoAct = lapply(brittCoActors, GetCoActors, imdb)
CACA = unlist(coActCoAct)
```

```
CACA = unique(CACA)
# CACA is a vector of 222 person_ids after limiting to top 1 paid actors in each movie
# Want in pair form, where pair = worked together

MakePairs = function(name, connection){
  x = rep(name, length(connection))
  return(x)
}
# X and Y for pairs:
pairsX = lapply(1:length(brittCANames), function(i) MakePairs(brittCANames[i],
coActCoAct[[i]]))
X = unlist(pairsX) # Have x in the (x, y) pairs we need for graph

# Going to get Y in IDs and then change to names
pairsY = unlist(coActCoAct)
Y = lapply(pairsY, GetNames, imdb)
Y = unlist(Y)

# Now have X and Y for the graphing:
install.packages("igraph")
library("igraph")
edges = data.frame(x = X, y = Y)
edges = as.matrix(edges)
g = graph.edgelist(edges)
V(g)
E(g)
par(mar = rep(0, 4))
plot(g)
```