Hannah Kosinovsky
STA 141
Section: 8 am Olson
11/30/15

Assignment 5

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.

Questions:

Question 1: How many actors and movies in data base:

I only want the cast info for people who were actors and actresses, so I specify the matches for role_type.role that I want.

SELECT DISTINCT cast_info.person_id FROM cast_info, role_type
WHERE cast_info.role_id=role_type.id AND role_type.role IN ('actor', 'actress')

This is the summary of the return:

3492018 Rows returned from: SELECT DISTINCT cast_info.person_id FROM cast_info, role_type
WHERE cast_info.role_id=role_type.id AND role_type.role IN ('actor', 'actress') (took 406833ms)

There are 3492018 actors/ actresses in this data base

In the second part of the question, I only want movies, not tv shows or other types of programs so I specify that I only want to match movie, tv movie, or video movie in the kind column for kind_type.

SELECT COUNT(*) from title, kind_type WHERE title.kind_id=kind_type.id AND kind_type.kind IN ("movie", "tv movie", "video movie")

Returned: 1145814

Question 2: What time period does database cover?

For this I can simply use the MIN and MAX functions on production year in the title table.

S*ELECT MIN(production_year), MAX(production_year) FROM title*

Returned: "1874" "2025"

Question 3: what proportion of actors are female/male?

SELECT DISTINCT cast_info.person_id FROM cast_info, role_type
WHERE cast_info.role_id=role_type.id AND role_type.role = 'actress'

1235135 Rows returned from: SELECT DISTINCT cast_info.person_id FROM cast_info, role_type
WHERE cast_info.role_id=role_type.id AND role_type.role = 'actress' (took 504331ms)

The number of rows in the result of this query will be the number of actresses, since I only looked for the cast info where role is actress.
From question 1, the total of actors and actresses is 3492018, so the proportion of actresses will be 1235135/3492018= .354 and the
proportion of male actors will be 1-.354=.646.

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

First, I want to look at the available types of movies.

*SELECT * FROM kind_type*

"1" "movie"
"2" "tv series"
"3" "tv movie"
"4" "video movie"
"5" "tv mini series"
"6" "video game"
"7" "episode"

Then I want the sum of each of these types and the total sum so I can calculate the proportion.

SELECT COUNT(*)AS total,
SUM (case when kind_id=1 then 1 end) AS num_movie,
SUM(case WHEN kind_id=2 THEN 1 END) AS num_tv_series,
SUM (case WHEN kind_id=3 THEN 1 END) AS num_tv_movie,
SUM (case WHEN kind_id=4 THEN 1 END) AS num_video_movie,
SUM (case WHEN kind_id=5 THEN 1 END) AS num_tv_mini_series,
SUM (case WHEN kind_id=6 THEN 1 END) AS num_video_game,
SUM (case WHEN kind_id=7 THEN 1 END) AS num_episode from title

Returns: "3527732" "878800" "124435" "120388" "146626" "NULL" "15314" "2242169"

So there are
878800/3527732= .25 movies,
124435/3527732=.035 tv series,
120388/3527732= .034 tv movies,
146626/ 3527732= .0145 video movies,
no tv mini series,
15314/3527732=.00434 video games,
and 2242169/3527732= .6356 episodes.

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

Here I needed to find a pathway to get just the names of the genres in the database. The column info in info_type has information that includes genres, so I needed to match the corresponding ids for the info that had 'genres' to the info type id that was in movie info so I could get the actual genres.
I then printed the names of the genres.

SELECT DISTINCT movie_info.info
FROM movie_info , info_type
WHERE movie_info.info_type_id = info_type.id AND info_type.info= 'genres'

Per return summary there are 32 genres:
32 Rows returned from: SELECT DISTINCT movie_info.info
FROM movie_info , info_type
WHERE movie_info.info_type_id = info_type.id AND info_type.info= 'genres' (took 36280ms)

The genres are listed in detaled query results:

"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"
"Erotica"

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

I expanded the previous query to include the count of movies for each genre, then I ordered the returned data by the count of movies in each genre in descending order and only printed the top 10 of these.

```
SELECT movie_info.info, COUNT(movie_info.id) as count
        FROM movie_info , info_type
        WHERE movie_info.info_type_id = info_type.id AND info_type.info= 'genres'
        GROUP BY movie_info.info ORDER BY count DESC LIMIT 10
```

Returned:

"Short"	"522672"
"Drama"	"330171"
"Comedy"	"243486"
"Documentary"	"215477"
"Adult"	"71983"
"Action"	"62830"
"Romance"	"62756"
"Thriller"	"61789"
"Animation"	"54606"
"Family"	"52197"

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?

Here I needed to connect titles to keywords and limit the search to just those with the keyword “space” so I found a pathway through the tables that relates the keyword table to the title table.

```
SELECT title.title, title.production_year
FROM keyword, movie_keyword, title
  WHERE keyword.id= movie_keyword.keyword_id
        AND movie_keyword.movie_id = title.id
        AND keyword.keyword = 'space'
```

Returns:
932 Rows returned from: SELECT title.title, title.production_year
FROM keyword, movie_keyword, title
  WHERE keyword.id= movie_keyword.keyword_id
        AND movie_keyword.movie_id = title.id
        AND keyword.keyword = 'space' (took 1080ms)

Since there were 932 Rows returned, there are 932 movies with the keyword “space”. Here is a sample of some of my outputs:

"Commander Toad in Space"	"1993"
"ALF"	"1986"
"Alien Hunter"	"2001"
"Race to the Moon"	"2005"
"Around Space"	"2001"
"Ascension"	"2014"
"Babylon 5"	"1994"
"Barbarella"	"NULL"
"Battlestar Galactica"	"1978"
"Battlestar Galactica"	"2004"
"Beast Wars Second: Chô seimeitai Transformer"	"1998"
"Blakes 7"	"1978"
"Blindpassasjer"	"1978"
"Bravest Warriors"	"2012"
"Brødrene Dal og spektralsteinene"	"1982"
"Buzz Lightyear of Star Command"	"2000"
"Caprica"	"2009"
"Captain Scarlet"	"2005"
"The Star Smugglers"	"1954"
"Challenge of the GoBots"	"1984"
"Chronic Misadventures of Slackers in Space"	"2014"

"Chô semeitai Transformer: Beast Wars Neo" "1999"

The next part of the question is to get the top 5 actors for each of these movies but not all of them have 5 leading actors/actresses, so the next full query will return info only for those that do.

In order to speed up the running of the query, instead of tying to role type table, I pre ran a query on the role type and found the ids that correspond to actors and actresses are 1 and 2 respectively. I then restricted each of the cast info pointers to role id being 1 or 2. In order to get 5 actors in each query result, I created 5 pointers each to the name table and the cast info table. I tied all of them to the movies that had the keyword “space” as in the preceding query and I tied each of the 5 actors to a specifically defined role id (cast_info.nr_order 1-5). Then I group by title because otherwise I was getting multiple identical results probably because of multiple pointers to cast_info.

SELECT title, production_year, n1.name, n2.name, n3.name , n4.name , n5.name FROM name n1, name n2, name n3, name n4, name n5, title t, cast_info ci1, cast_info ci2, cast_info ci3, cast_info ci4, cast_info ci5
WHERE ci1.person_id=n1.id AND ci2.person_id=n2.id AND ci3.person_id=n3.id AND ci4.person_id=n4.id AND ci5.person_id=n5.id
AND ci1.role_id in (1,2) AND ci2.role_id in (1,2) AND ci3.role_id in (1,2) AND ci4.role_id in (1,2) AND ci5.role_id in (1,2) AND t.id= ci1.movie_id AND t.id= ci2.movie_id AND t.id= ci3.movie_id AND t.id= ci4.movie_id AND t.id= ci5.movie_id
AND ci1.nr_order=1 AND ci2.nr_order=2 AND ci3.nr_order=3 AND ci4.nr_order=4 AND ci5.nr_order=5
AND ci1.movie_id IN (select t.id FROM title t, movie_keyword mk, keyword k, kind_type kt WHERE mk.movie_id=t.id AND mk.keyword_id=k.id AND keyword="space" AND t.kind_id=kt.id AND kt.kind in ("movie", "tv movie", "video movie")) group by title

Returns:

237 Rows returned from: SELECT title, production_year, n1.name, n2.name, n3.name , n4.name , n5.name FROM name n1, name n2, name n3, name n4, name n5, title t, cast_info ci1, cast_info ci2, cast_info ci3, cast_info ci4, cast_info ci5
WHERE ci1.person_id=n1.id AND ci2.person_id=n2.id AND ci3.person_id=n3.id AND ci4.person_id=n4.id AND ci5.person_id=n5.id
AND ci1.role_id in (1,2) AND ci2.role_id in (1,2) AND ci3.role_id in (1,2) AND ci4.role_id in (1,2) AND ci5.role_id in (1,2) AND t.id= ci1.movie_id AND t.id= ci2.movie_id AND t.id= ci3.movie_id AND t.id= ci4.movie_id AND t.id= ci5.movie_id AND ci1.nr_order=1 AND ci2.nr_order=2 AND ci3.nr_order=3 AND ci4.nr_order=4 AND ci5.nr_order=5
AND ci1.movie_id IN (select t.id FROM title t, movie_keyword mk, keyword k, kind_type kt WHERE mk.movie_id=t.id AND mk.keyword_id=k.id AND

keyword="space" AND t.kind_id=kt.id AND kt.kind in ("movie", "tv movie", "video movie")) group by title (took 22481ms)

There are 237 movies which have 5 top billed actors/actresses. Here is a sample of my results:

"002 operazione Luna" "1965" "Franchi, Franco" "Ingrassia, Ciccio" "Randall, MÛnica" "Sini, Linda" "Silva, MarÌa"

"1 Star" "2009" "Burns, Brendon" "Hadland, Sarah" "McHugh, Greg" "Macleod, Lewis" "Mitchell, Gavin"

"12 to the Moon" "1960" "Clark, Ken" "Kobi, Michi" "Conway, Tom""Dexter, Anthony" "Wengraf, John"

"20 Million Miles to Earth" "1957" "Hopper, William" "Taylor, Joan""Puglia, Frank" "Zaremba, John" "Henry, Thomas Browne"

"2001: A Space Odyssey" "1968" "Dullea, Keir" "Lockwood, Gary""Sylvester, William" "Richter, Daniel" "Rossiter, Leonard"

"2010" "1984" "Scheider, Roy" "Lithgow, John" "Mirren, Helen" "Balaban, Bob" "Dullea, Keir"

"4: Rise of the Silver Surfer" "2007" "Gruffudd, Ioan" "Alba, Jessica""Evans, Chris" "Chiklis, Michael" "McMahon, Julian"

"51 Degrees North" "2015" "von Zeddelmann, Moritz" "Osterloh, Dolly-Ann" "Cree, Steven" "Nallon, Steve" "Doyle, Jamie"

"A House Undivided""2015" "Antonopoulos, Erik" "Cox, Elliot" "Zdrazil, Nicholas" "McClure, Dawson" "Laipeneks, Matt"

"Abbott and Costello Go to Mars" "1953" "Abbott, Bud" "Costello, Lou" "Blanchard, Mari" "Paige, Robert" "McMahon, Horace"

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.

I ended up graphing this in R, so the R code is in the attached appendix, but, my original query in SQL was:

SELECT movie_info.info as info, title.production_year as year, count(movie_info.id) as count

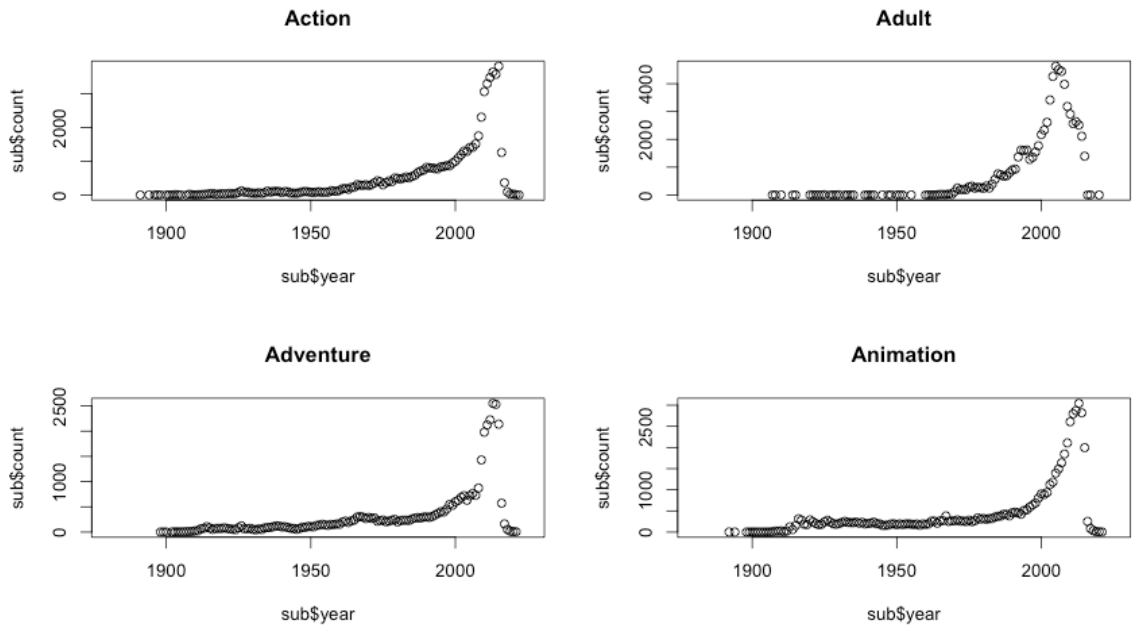
FROM movie_info, info_type, title WHERE movie_info.movie_id=title.id AND movie_info.info_type_id = info_type.id
AND info_type.info = 'genres' AND title.production_year IS NOT NULL GROUP BY title.production_year, movie_info.info ORDER BY movie_info.info

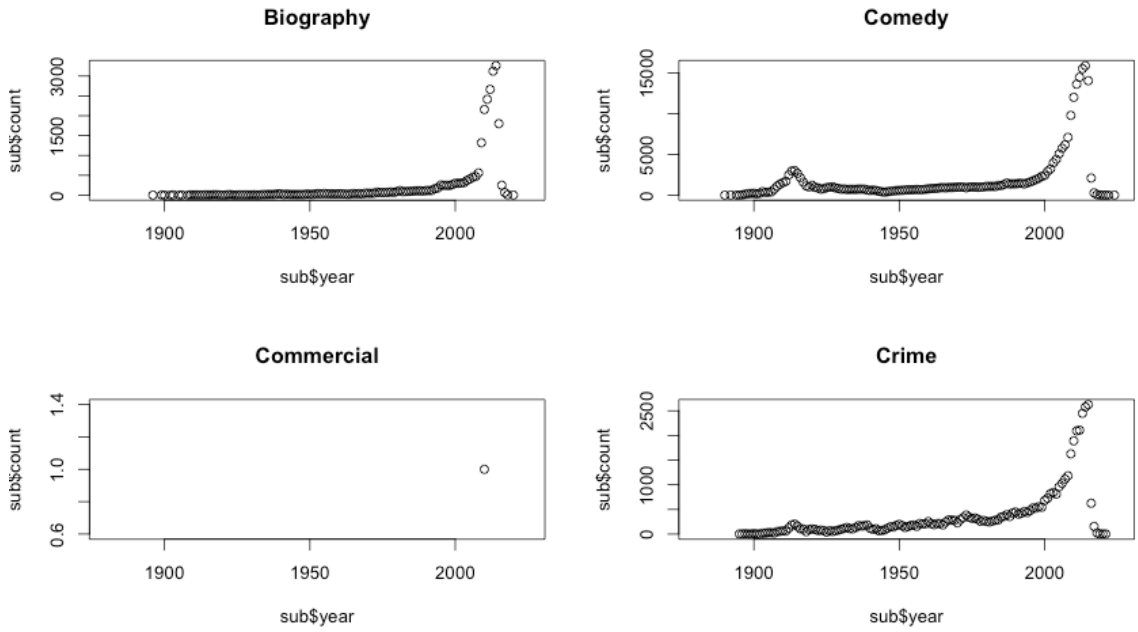
Grouping by year and genre is the key to this question.

The sample return (from 3213 rows) is as follows:

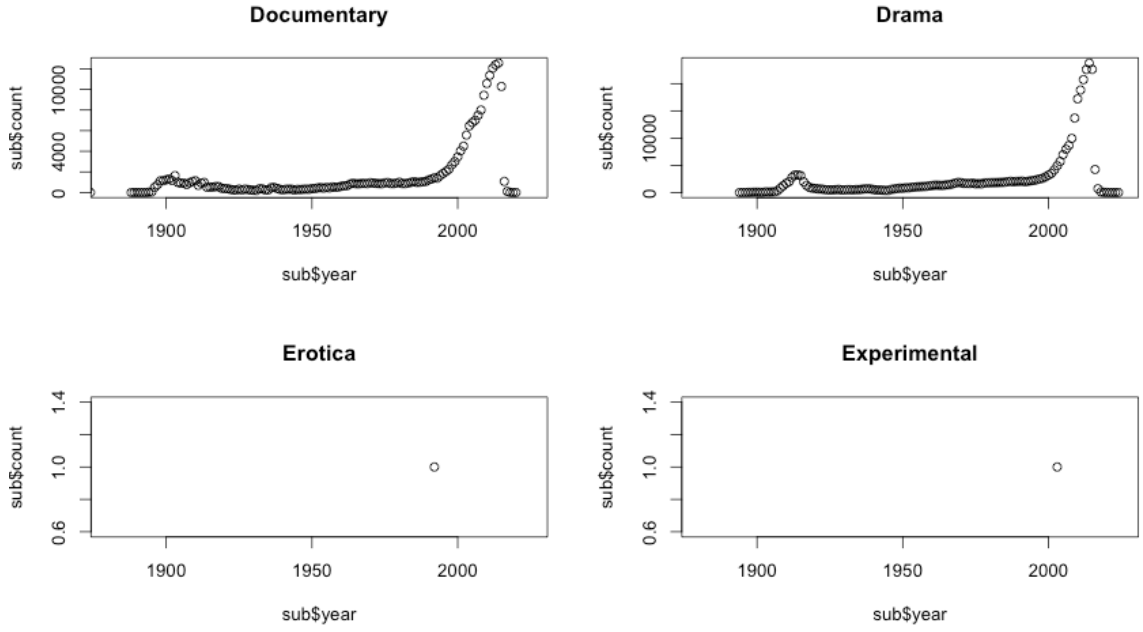
"Action"	"1891"	"1"
"Action"	"1894"	"3"
"Action"	"1896"	"1"
…		
"Action"	"2020"	"26"
"Action"	"2021"	"6"
"Action"	"2022"	"3"
"Adult"	"1907"	"1"
"Adult"	"1908"	"1"

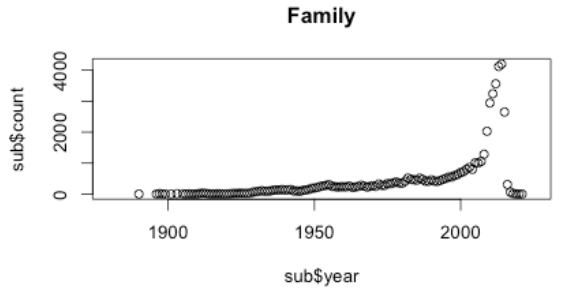
The graphs I get from my R code are:

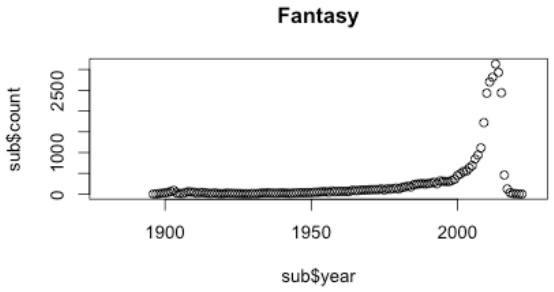


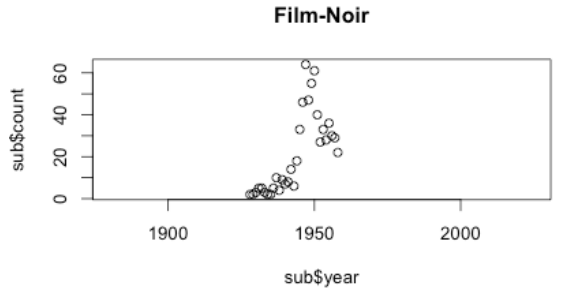


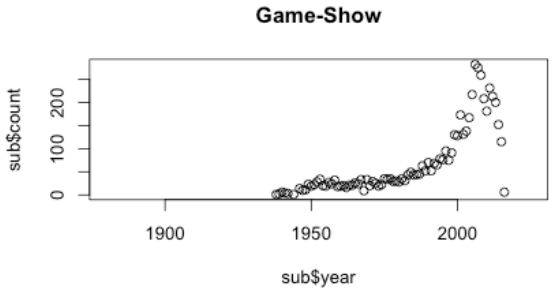
Most genre offerings are greatly increasing in recent years (since about 2000). There was a comedy spike in the 1920s. And there were also some spikes in crime movies over the years especially around 1920, 1940, and 1970.



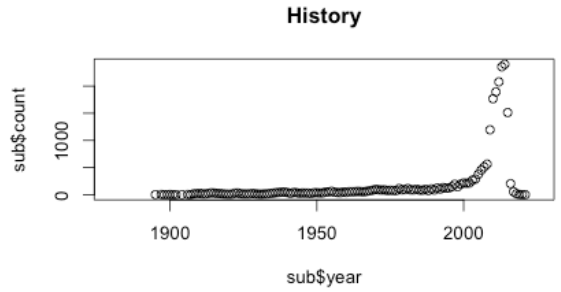


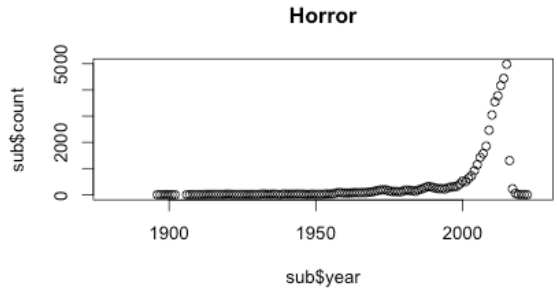


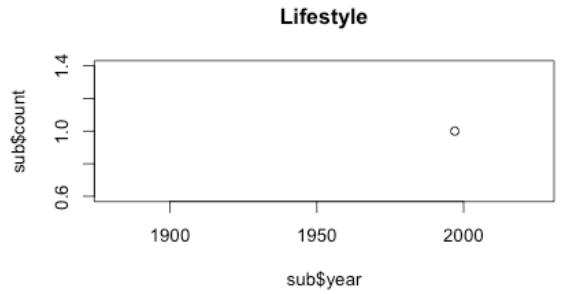


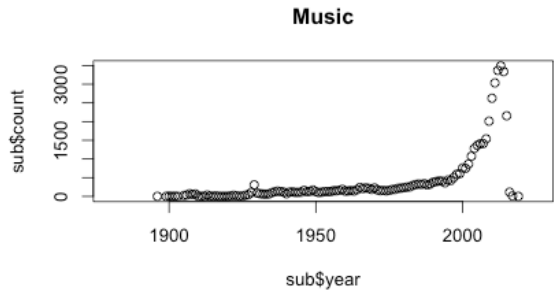


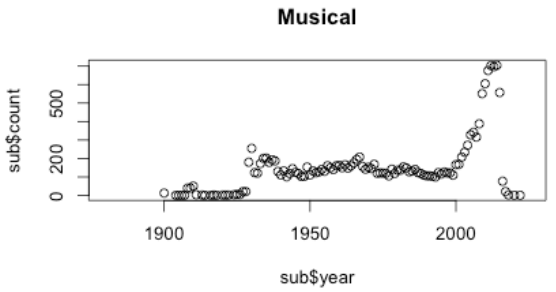
Film-Noir was only made between 1930 and 1960 with a spike at 1950. Game Shows only started in 1940s.

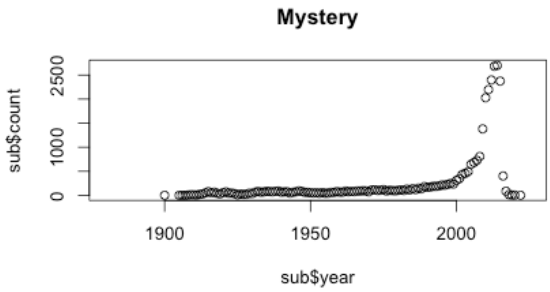


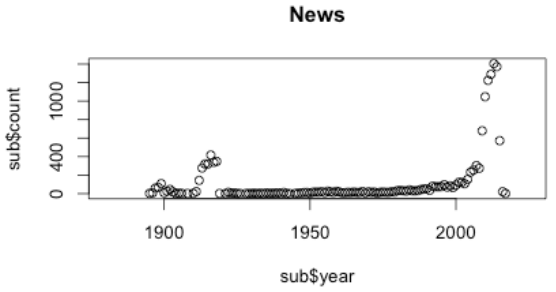


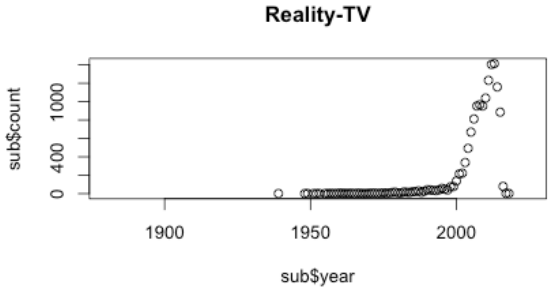




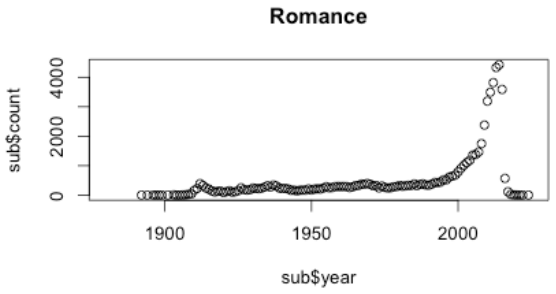


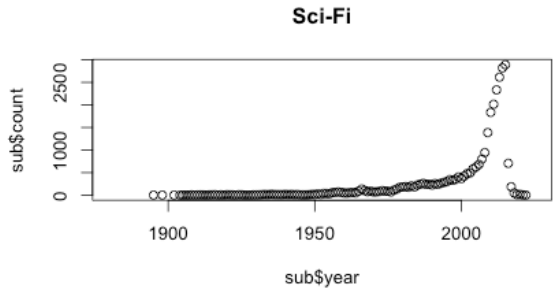


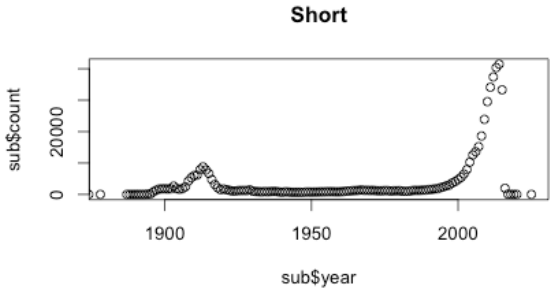


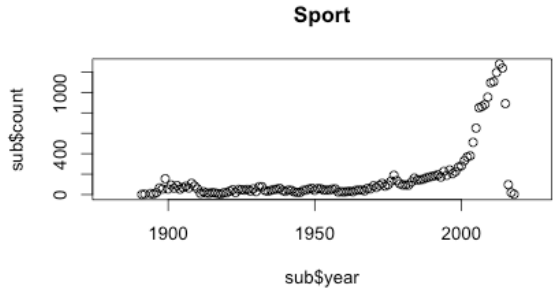


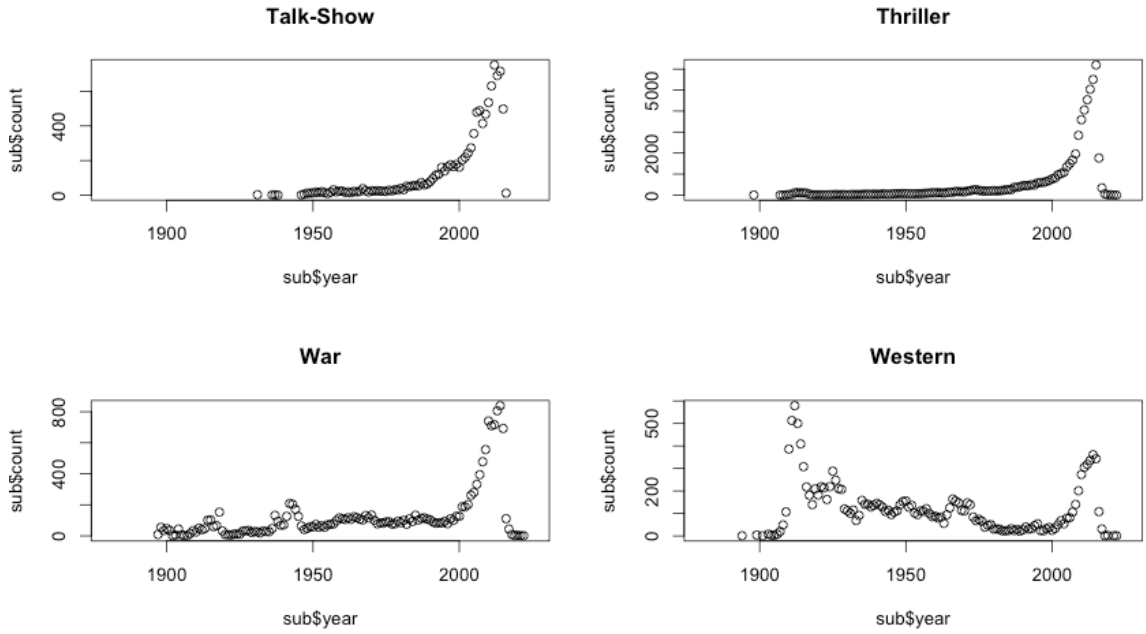
Musicals had a sudden and sustained increase in 1930s and then increased again with other genres after 2000s. News had a spike in 1920 and then declined until late 1990s. Reality TV did not begin until 1960.











Romance and shorts both had spikes in the 1920s. Talk Shows didn't begin until 1940. War had spikes in 1930 and 1940 (which makes sense with the world wars). Westerns had a huge spike at 1920 and have since then been declining, even with the increase all genres have seen in production Westerns have never been made at the same rate they were in 1920.

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

In my first try I did not restrict the types of "movies" as defined in kind_type:

SELECT count(ci.movie_id) as count, n.name FROM cast_info ci, role_type rt, name n WHERE ci.role_id=rt.id AND n.id=ci.person_id AND rt.role in ('actor','actress') GROUP BY ci.person_id ORDER BY count desc limit 20

"7885"	"Welker, Frank"
"7262"	"Trebek, Alex"
"7234"	"Gilbert, Johnny"
"6899"	"Barker, Bob"
"6279"	"Sajak, Pat"
"6245"	"Boyd, Jim"
"6220"	"White, Vanna"
"6061"	"Freeman, Morgan"
"5824"	"¡lvarez, FÈlix"
"5741"	"Vorderman, Carol"
"5540"	"Pennington, Janice"
"5530"	"Baker, Dee Bradley"

"5393"	"Leno, Jay"
"5201"	"Shaffer, Paul"
"4974"	"Olson, Johnny"
"4955"	"Kenny, Tom"
"4843"	"Bennett, Jeff"
"4755"	"Letterman, David"
"4702"	"Hinnant, Skip"
"4649"	"Castellaneta, Dan"

This returns lots of talk show people, so I narrowed it to movies, tv movies, and video movies:

SELECT count(ci.movie_id) as count, n.name FROM cast_info ci, role_type rt, name n, title t, kind_type kt
WHERE ci.role_id=rt.id AND n.id=ci.person_id AND rt.role in ('actor','actress') AND t.id=ci.movie_id AND t.kind_id=kt.id AND kt.kind in ("movie", "tv movie", "video movie")
GROUP BY ci.person_id ORDER BY count desc limit 20

20 Rows returned from: SELECT count(ci.movie_id) as count, n.name FROM cast_info ci, role_type rt, name n, title t, kind_type kt
WHERE ci.role_id=rt.id AND n.id=ci.person_id AND rt.role in ('actor','actress') AND t.id=ci.movie_id AND t.kind_id=kt.id AND kt.kind in ("movie", "tv movie", "video movie")
GROUP BY ci.person_id ORDER BY count desc limit 20 (took 1483651ms)

"3245"	"Blanc, Mel"
"1936"	"Byron, Tom"
"1839"	"North, Peter"
"1744"	"Jeremy, Ron"
"1741"	"Wood, Mark"
"1727"	"Pete, Mr."
"1468"	"Blue, Mick"
"1466"	"Deen, James"
"1449"	"Everhard, Erik"
"1443"	"Stone, Lee"
"1375"	"Stone, Evan"
"1363"	"Strong, John"
"1360"	"Ferrara, Manuel"
"1359"	"Davis, Mark"
"1354"	"Sanders, Alex"
"1250"	"Spears, Randy"
"1212"	"Banderas, Marco"
"1211"	"Wallice, Marc"
"1189"	"Boy, T.T."

"1159" "Silvera, Joey"

This (with obvious exception of Mel Blanc) returned mostly porn stars, which admittedly are actors.

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 wanted to print the count of movies they've been in as top billing, their name, and the range of years for their performances. I specified that they must be actors, in a movie, and have top billing in kind type, role type, and nr_order. I ordered the actors by the count of movie ids and printed the top 20 by specifying to print in descending order.

SELECT count(ci.movie_id) as count, n.name, min(production_year), max(production_year) FROM cast_info ci, role_type rt, name n, title t, kind_type kt WHERE ci.role_id=rt.id AND n.id=ci.person_id AND rt.role in ('actor','actress') AND t.id=ci.movie_id AND
t.kind_id=kt.id AND kt.kind in ("movie", "tv movie", "video movie") AND ci.nr_order IN (1,2,3)
GROUP BY ci.person_id ORDER BY count DESC limit 20

"1691"	"Blanc, Mel"	"1944"	"2011"
"394"	"Shin, Sung-il"	"1960"	"1992"
"374"	"Kerrigan, J. Warren"	"1910"	"1934"
"368"	"Moran, Lee"	"1912"	"1933"
"355"	"Lyons, Eddie"	"1911"	"1924"
"327"	"Hardy, Oliver"	"1914"	"2011"
"323"	"Anderson, Gilbert M. 'Broncho Billy'"	"1904"	"1922"
"302"	"Pollard, 'Snub'"	"1915"	"1933"
"297"	"Mohanlal"	"1980"	"2015"
"297"	"Richardson, Jack"	"1911"	"1929"
"292"	"Garcia, Eddie"	"1953"	"2013"
"266"	"Ford, Francis"	"1909"	"1930"
"266"	"Prince, Charles"	"1906"	"1929"
"263"	"Polidor"	"1910"	"1957"
"242"	"Mammootty"	"1981"	"2015"
"239"	"Hamilton, Lloyd"	"1913"	"1933"
"239"	"Bush, Pauline"	"1910"	"1917"
"238"	"Mix, Tom"	"1909"	"1940"
"235"	"Franey, Billy"	"1914"	"1941"
"233"	"Myers, Harry"	"1909"	"1935"

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?

From Question 2 we have the min and max production year: "1874" "2025"
But I ended up using the lean database (lean_imdbpy_2010_idx.db) because it ended up taking to much time to run my queries.
So my new range was 2010 to 2025.

I first wrote a query that finds the actors in the most movies for 2010. I did this by ordering the actors by the count of movie ids and specifying to print them in a descending order.

SELECT count(ci.movie_id) as count, n.name FROM cast_info ci, role_type rt, name n, title t, kind_type kt
WHERE ci.role_id=rt.id AND n.id=ci.person_id AND rt.role in ('actor','actress') AND t.id=ci.movie_id AND
t.kind_id=kt.id AND kt.kind in ("movie", "tv movie", "video movie") AND ci.nr_order IN (1,2,3) AND production_year=2010
GROUP BY ci.person_id ORDER BY count desc limit 10

10 Rows returned from: SELECT count(ci.movie_id) as count, n.name FROM cast_info ci, role_type rt, name n, title t, kind_type kt
WHERE ci.role_id=rt.id AND n.id=ci.person_id AND rt.role in ('actor','actress') AND t.id=ci.movie_id AND
t.kind_id=kt.id AND kt.kind in ("movie", "tv movie", "video movie") AND production_year=2000
GROUP BY ci.person_id ORDER BY count desc limit 10 (took 612900ms)

"115" "Steele, Lexington"
"104" "Davis, Mark"
"89" "Hardman, Dave"
"87" "Everhard, Erik"
"84" "Bune, Tyce"
"82" "Marcus, Mr."
"82" "Stone, Evan"
"80" "Savage, Herschel"
"77" "Cannon, Chris"
"76" "Kerkove, Bridgette"

Then I wrote a function in R called npy (can be seen in attached appendix) that:

1. Iterated over the range of years

2. For each year it found the count of movies for each actor and gave the top 10 counts using an SQL query much like the one above (in each iteration query needed to be constructed with the corresponding year)
3. then for each of those names it printed the movie titles.

A sample of the results I got from this function(including all would take too much space in my report):

```
[1] 2010
[1] "Brahmanandam"
 [1] "9th Class C/o Eleshwaram"     "Adhurs"
 [3] "Baava"                        "Bhavani IPS"
 [5] "Bindaas"                      "Brindaavanam"
 [7] "Buridi"                       "Chalaki"
 [9] "Comedy Express"               "Desadrohi"
[11] "Don Seenu"                    "Emaindhi Naalo"
[13] "Gudu Gudu Gunjam"             "Holidays"
[15] "Jhummandi Nadam"              "Kalyan Ram Kathi"
[17] "Kedi"                         "Khaleja"
[19] "Kotha Bandham"                "Maa Nanna Chiranjeevi"
[21] "Mondi Mogullu Penki Pellalu"  "Mukkanti"
[23] "Nagavalli"                    "Namo Venkatesha"
[25] "Ninnu Choosina Kshanana"      "Oka Tupaki Moodu Pittalu"
[27] "Orange"                       "Panchakshari"
[29] "Police Police"                "Prema Pilustondi"
[31] "Ragada"                       "Rama Rama Krishna Krishna"
[33] "Rs 999 Matrame"               "Sare Nee Istam"
[35] "Seetharamula Kalyanam Lankalo" "Simha"
[37] "Taj Mahal"                    "Varudu"
[39] "Vedam"                        "Visu"
[41] "Yagam"
[1] "Obama, Barack"
 [1] "8: The Mormon Proposition"
 [2] "A Conversation with President Obama"
 [3] "Beyoncé's I Am... World Tour"
 [4] "Cesky mir"
 [5] "Christmas in Washington"
 [6] "Climate Refugees"
 [7] "Closer to the Dream"
 [8] "Cool It"
 [9] "Fair Game?"
[10] "God Bless You Barack Obama?"
[11] "How Obama Won the West"
[12] "I Am"
[13] "I'm Still Here"
[14] "Inside Job"
```

[15] "Investing in America: A CNBC Town Hall Event with President Obama"
[16] "Jean Genet, le contre-exemplaire"
[17] "Marijuana USA"
[18] "Motherland"
[19] "Nuclear Tipping Point"
[20] "Obama in NC: The Path to History"
[21] "Spirit of America"
[22] "Stand Up to Cancer"
[23] "The Big Fat Quiz of the Year"
[24] "The Furious Force of Rhymes"
[25] "The Library of Congress Gershwin Prize for Popular Song: In Performance at the White House - Paul McCartney"
[26] "The National Christmas Tree Lighting"
[27] "The President's Photographer: Fifty Years Inside the Oval Office"
[28] "The War You Don't See"
[29] "Tony & Janina's American Wedding"
[30] "Under the Boardwalk: The Monopoly Story"
[31] "United States of Obama"
[32] "Variations on a High School Romance"
[33] "WWE Tribute to the Troops"
[34] "What is the Electric Car?"

… then it would start with Lloyd Kaufman and list his movies… etc until the year was 2011:

1] 2011
[1] "Rea, Kyle"
[1] "Digimon Fusion Battles"
[2] "Fist of the North Star: The Kaioh Saga"
[3] "Fist of the North Star: The Raul Saga"
[4] "Fist of the North Star: The Ray Saga"
[5] "Fist of the North Star: The Souther Saga"
[6] "Fist of the North Star: The Toki Saga"
[7] "Gaiking I"
[8] "Gaiking II"
[9] "Gaiking III"
[10] "Galaxy West"
[11] "Karma"
[12] "Kitaro's Graveyard Gang 2"
[13] "Pretty Cure 5"
[14] "Rift"
[15] "Starzinger"
[16] "Starzinger II"
[17] "Starzinger III"
[18] "The Adventures of Nadja"
[19] "The Adventures of Nadja II"

[20] "The Standards: Espinosa"
[1] "Lorente, Txema"
 [1] "Alienación"
 [2] "Arcano"
 [3] "Cafetería perdida"
 [4] "Cafetería perdida 2"
 [5] "D.E.P."
 [6] "Dies Irae"
 [7] "Discusión abstracta"
 [8] "Doña María"
 [9] "Dulces de Halloween"
[10] "El aburrido de Don Álvaro"
[11] "El contrato"
[12] "El convidat"
[13] "El maldito tesoro de los hermanos Crown"
[14] "El marrano"
[15] "Esperando"
[16] "Expiación"
[17] "Gisëlle&Malice"
[18] "Guía de supervivencia estudiantil"
[19] "I Have a Story"
[20] "Imago"
[21] "Indigestión de muerte"
[22] "Komboloi"
[23] "La Grabadora"
[24] "La brutícia"
[25] "La ventana"
[26] "Lenguaje silencioso"
[27] "Les veus de l'Arnau"
[28] "Luchador Records"
[29] "Moribundo"
[30] "Narco"
[31] "Obsesión"
[32] "Pornosotros"
[33] "Quant costa la vida d'un català?"
[34] "Rere la màscara"
[35] "Secuencia de terror"
[36] "Sol, piruletas y Arcoiris"
[37] "Sola en el mundo"
[38] "Tetas para matar"
[39] "Trinchera"
[40] "Un encuentro"
[41] "Una partida cualquiera"
[42] "Vivo sin vivir en mí"
[43] "When All Hope Is Gone"
[44] "Y si cae aquí?"

[1] "Billany, Martin"
[1] "Bump"
[2] "Eiga Keion!"
[3] "None Piece"
[4] "Spoof Movie No Jutsu!"
[5] "Yu-Gi-Oh! 3D: Bonds Beyond Time Abridged"
[6] "Yu-Gi-Oh! 5D's One-Shot Parody Special"
[1] "Thingvall, Joel"
[1] "After the Beginning"
[2] "Allergic to Love"
[3] "Alright Then"
[4] "Amy Della Dobble and the Fairy Messy Bedroom"
[5] "Big Highway"
[6] "Bloodshed Love"
[7] "Captain America: The First Avenger"
[8] "Chris"
[9] "Curitol"
[10] "DREAMer"
[11] "Directions"
[12] "Go"
[13] "Gods' Green Earth"
[14] "Invincible Force"
[15] "Jeremy Messersmith's Violet"
[16] "Kinetic"
[17] "Lambent Fuse"
[18] "Land of Sky Blue Water"
[19] "Le Fling"
[20] "Leichenwasser"
[21] "Memorial Day"
[22] "My Milton"
[23] "Out of Character"
[24] "Paul"
[25] "Prodigal"
[26] "SomeFangs"
[27] "Stuck Between Stations"
[28] "The Convincer"
[29] "The Fancy Men"
[30] "The Last Night"
[31] "The Love, the Leather & the L337"
[32] "The Sixties Film"
[33] "The U-Brew"
[34] "Them!"
[35] "Topher (The Life and Depressing Times of Christopher Peck)"
[36] "Transformers: Dark of the Moon"
[37] "Voice Inside"
[38] "Wonder Woman"

[39] "Year 2035"
[40] "Young Adult"

And so forth until 2025 where there is only one movie title - StreetDance4 and 4 actors set to play in it.

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

This query outputs the count of aliases for a given actor by connections through the name table and aka_name tables.

SELECT COUNT(*) AS count, name.name FROM aka_name, name WHERE name.id=aka_name.person_id
GROUP BY person_id ORDER BY count DESC LIMIT 10

"78" "Franco, Jesús"
"71" "D'Amato, Joe"
"63" "Digard, Uschi"
"53" "Savage, Herschel"
"50" "Ho, Godfrey"
"42" "Silvera, Joey"
"40" "Albert, Kimson"
"39" "León, Nathanael"
"38" "Clark, Christoph"
"38" "Presova, Zuzana"

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.

I wanted to find an actor with top billing that has been in 20 movies, so I ordered the count of movies and chose from the middle by offsetting by 40.

SELECT count(ci.movie_id) as count, n.name, n.id FROM cast_info2 ci, role_type rt, name2 n, title2 t, kind_type kt
WHERE ci.role_id=rt.id AND n.id=ci.person_id AND rt.role in ('actor','actress') AND t.id=ci.movie_id AND t.kind_id=kt.id AND
kt.kind in ("movie", "tv movie", "video movie") AND ci.nr_order IN (1,2,3)
GROUP BY ci.person_id ORDER BY count desc limit 20 offset 40

The “magic” numbers of 20 and 40 above came about because I didn’t want a lot of returns so I made several queries until I got the one that gave me a range of actors with 20 movies in their credits

"22" "Estevez, Joe" "596651"
"22" "Landis, Nick" "1126619"
"22" "Aggarwal, Kajal" "2269985"
"21" "Ferch, Heino" "621319"
"21" "Jones, Vinnie" "991805"
"21" "Klanfer, Derek" "1065870"
"21" "Van Dien, Casper" "2084434"
"20" "Assante, Armand" "86634"
"20" "Bauer, Steven" "141030"
"20" "Idle, Eric" "929707"
"20" "Karl, Fritz" "1018968"
"20" "King, Evan" "1057438"
"20" "Le Coq, Bernard" "1145837"
"20" "Murray, Martin Glyn" "1426749"
"20" "Santhanam" "1786692"
"20" "Bhatia, Tamannaah" "2360756"
"20" "Demoustier, AnaÔs" "2533117"
"20" "Kikukawa, Rei" "2832160"
"20" "Yamaguchi, Mari" "3477693"
"19" "Collins, Keith" "392860"

I then chose Yamaguchi, Mari from these results. Then I wanted to find all the actors that had worked with Yamaguchi. I looked at all the titles that Yamaguchi was in and then I made a distinct selection of all other actors in those titles except for Yamaguchi himself (using Yamagichi's id value of 392860).

SELECT DISTINCT n.name, n.id from name2 n, cast_info2 ci, title2 t, role_type rt, kind_type kt WHERE ci.role_id=rt.id AND t.kind_id=kt.id AND kt.kind in ("movie", "tv movie", "video movie") AND n.id != 3477693 AND rt.role in ('actor','actress') AND ci.person_id=n.id AND ci.movie_id=t.id AND t.id in (SELECT t.id FROM title2 t, cast_info2 ci WHERE ci.movie_id=t.id AND ci.person_id=3477693)

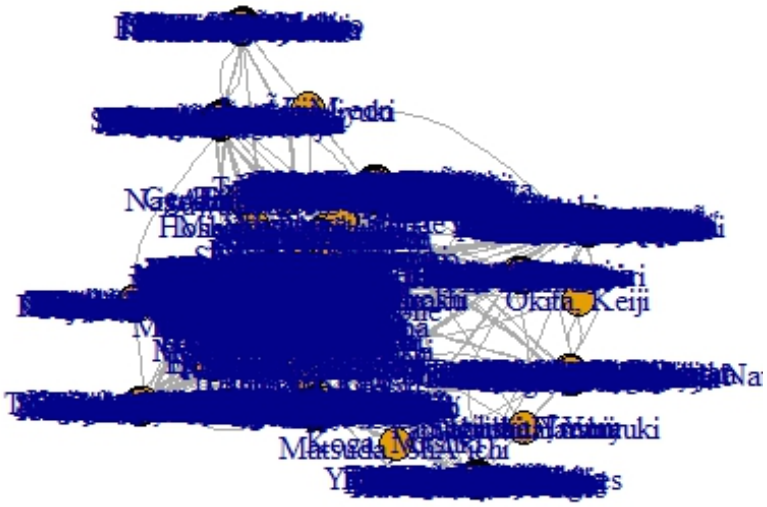
The result was 76 actors who collaborated with Yamagichi*:*
"Nakamitsu, Seiji" "1439449"
"Nomura, Takahiro" "1475060"
"Tsuda, Atsushi" "2055237"
"Itsuki, Karin" "2772746"
"Yokoyama, Mirei" "3482440"
"Haraguchi, Daisuke" "823355"
"Horimoto, Yoshinori" "899731"
"Ishii, Ryo" "940582"
"Shijimi" "3279422"
"Yokoyama, Miyuki" "3482444"
"Bishari, Toshitaka" "189064"
"Iwatani, Kenji" "944927"

"Okada, Tomohiro" "1500597"
"Sakeyama, Salmon" "1770036"
"Tsuda, Satoshi" "2055258"
"Kasumi, Kaho" "2816007"
"Kazuha, Mirei" "2820432"
"KÙda, Riri" "2865849"
"Kubota, Yasunari" "1101215"
"Shiraishi, Masahiko" "1857550"
"Yanagi, TÙshi" "2223110"
"Chiyumi, KochÙ" "2457914"
"Hatsuki, Nozomi" "2718269"
"Naha, Takashi" "1437990"
"Kan'no, Ichiha" "2810022"
"Kimino, Ayumi" "2834398"
"Araki, TarÙ" "70336"
"Azumi, Koi" "2316900"
"Misugi, Asuka" "3017093"
"‘ta, Hajime" "2260137"
"Tanaka, Hitomi" "3352597"
"Keisuke" "1032113"
"SaitÙ, Takashi" "1769182"
"Yamamoto, SÙsuke" "2222211"
"Asakura, Maria" "2307665"
"Hoshi, Y°no" "2751565"
"ItÙ, Rina" "2772930"
"Natsui, Ami" "3057483"
"Tsubomi" "3387467"
"Kawase, YÙta" "1027156"
"Nishioka, Hideki" "1470658"
"Asami" "2307702"
"Maika" "2944273"
"Maki, Azusa" "2945712"
"Makimura, KÙji" "1245714"
"Oka, Teruo" "1500448"
"Takemoto, Yasushi" "1984916"
"‘shiro, Kaede" "3498559"
"ItÙ, Takeshi" "943129"
"SatÙ, Yoshihiro" "1794951"
"Shinra, ManzÙ" "1857067"
"Hayama, Mei" "2720366"
"Hoshino, Yuzu" "2751662"
"Matsui, Riko" "2973724"
"Tomoda, Ayaka" "3376882"
"Aoyama, Minami" "2298958"
"Nishino, ShÙ" "3071823"
"Saito, Takashi" "1769050"

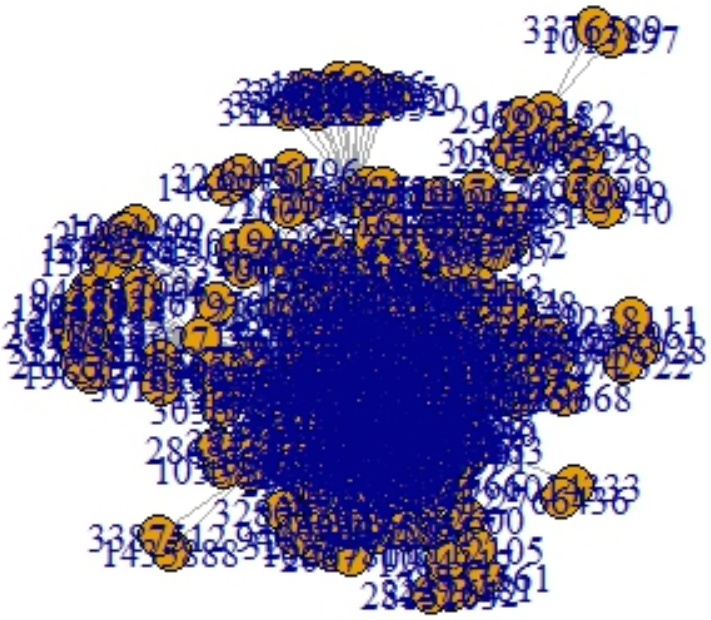
"Sakaki-kun" "1769668"
"Tsukamoto, Rei" "3387687"
"‘tsuki, Hibiki" "3498647"
"Okita, Keiji" "1501209"
"Sanada, Mikiya" "1779701"
"Ikejima, Yutaka" "931530"
"Kanno, Shizuka" "2811191"
"Sasaki, Motoko" "3246017"
"Horiken" "899707"
"Oda, Mako" "3086017"
"Kotaki, Miina" "2852454"
"Tia" "3370776"
"Amakawa, Masumi" "47276"
"Maino, Miya" "2944595"
"Mizusawa, Maki" "3019661"
"Moriyama, ShÙgo" "1403804"
"Hatano, Yui" "2717897"
"Ogura, Momo" "3087385"

I then used R to run the same queries on each of the 76 actors listed above and build up a data frame were the first column had the names of actors on which the search was done and the second column had each of the collaborators for that actor. Thus every row of the data frame represented a “connection” on the graph.
I used the “igraph” library to plot the graph, but it was too busy to follow. I then refined the search for collaborators to first just the 2 with top 2 billings, and then even down to 1. I also used people’s ids rather than names in the last 2 efforts to make it more readable. I would say that the third effort is slightly more readable than the first 2, but there is still too much information for a truly readable graph.

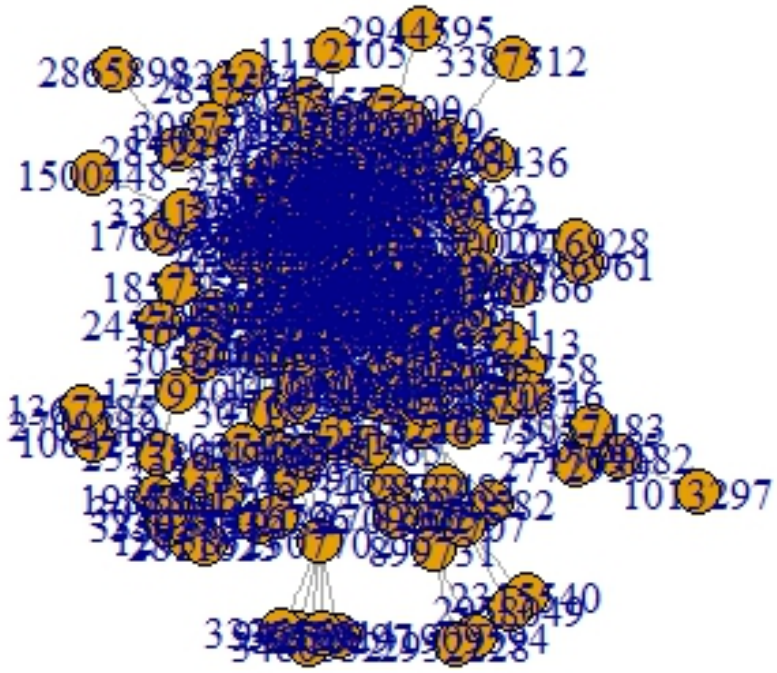
Original plot with all connections and names of actors:



Plot of just top two billing connections, with ids in nodes instead of names:



Just the top billed actor connections plot:



R Appendix:

```
#install package RSQLite in order to use sql queries in R
install.packages("RSQLite")
library(RSQLite)
imdb = dbConnect(SQLite(), "/Users/hannah/Desktop/imdbpy.db")

#question 8
genre_by_year = dbGetQuery(imdb, "SELECT movie_info.info as info,
title.production_year as year, count(movie_info.id) as count
FROM movie_info, info_type, title  WHERE movie_info.movie_id=title.id AND
movie_info.info_type_id = info_type.id
AND info_type.info = 'genres' AND title.production_year IS NOT NULL GROUP BY
title.production_year, movie_info.info ORDER BY movie_info.info")

#check if subset in plot work for just action
Action = subset(genre_by_year, genre_by_year$info == "Action")
plot(Action$count, Action$year)

#create a data frame of all the genres
```

```
genres = dbGetQuery(imdb, "select distinct(mi.info) from movie_info mi, info_type it
where mi.info_type_id = it.id AND it.info = 'genres' ORDER BY mi.info")

#create a function genreplot to make a plot of movies
#made per year for each genre
genreplot =function() {
  #iterates through all the genre
  for(i in 1:nrow(genres)) {
    #create name for each genre by going through each row in
    #column info
    genrename =genres$info[i]
    #create subset of genre_by_year for each genre
    sub = subset(genre_by_year, genre_by_year$info == genrename)
    #plot the number of films/ year for each genre with the
    #appropriate name as the title
    plot(sub$year, sub$count, main = genrename, xlim=c(1880,2025))
  }
}

#make connection to smaller data frame
imdb_lean = dbConnect(SQLite(),
"/Users/hannah/Downloads/lean_imdbpy_2010_idx.db")

#question 11
#find the minimum and maximum production years in lean database
min_max = dbGetQuery(imdb_lean, "SELECT min(production_year) as minY,
max(production_year) as maxY FROM title2 t")

#find top 10 actors with most movies for every year
top10_year = dbGetQuery(imdb_lean, "SELECT count(ci.movie_id) as count, n.name
as nname FROM cast_info2 ci, role_type rt, name2 n, title2 t, kind_type kt
WHERE ci.role_id=rt.id AND n.id=ci.person_id AND rt.role in ('actor','actress') AND
t.id=ci.movie_id AND
t.kind_id=kt.id AND kt.kind in ('movie', 'tv movie', 'video movie') AND
production_year=2010
GROUP BY ci.person_id ORDER BY count desc limit 10")

#to create this function, I had to use sprintf in order to build up
#an appropriate query for each question. use sprintf to format a query to be specific
to each iteration

#create function that goes through every year and finds the top 10 actors
#for that year and all of their movies
npy =function() {
  #iterate through all production years then print year
  for(i in min_max$minY:min_max$maxY) {
```

```
  print(i)
  #get information for every year
  da=dbGetQuery(imdb_lean, sprintf("SELECT count(ci.movie_id) as count, n.name
as nname, n.id as id
  FROM cast_info2 ci, role_type rt, name2 n, title2 t, kind_type kt
         WHERE ci.role_id=rt.id AND n.id=ci.person_id AND rt.role in ('actor','actress')
        AND t.id=ci.movie_id AND
         t.kind_id=kt.id AND kt.kind in ('movie', 'tv movie', 'video movie') AND
production_year=%d
         GROUP BY ci.person_id ORDER BY count desc limit 10",i))
  #if there is no information for the year, continue to next
  if(nrow(da)==0) {
    next
  }
  #iterate through every actor
  for(j in 1:nrow(da)) {
    #print actor's name
    print(da$nname[j])
    #find titles for the actor
    net= dbGetQuery(imdb_lean, sprintf("SELECT DISTINCT title from title2 t, name2
n, cast_info2 ci WHERE n.id=%d AND production_year = %d AND ci.person_id=n.id
AND ci.movie_id=t.id",da$id[j], i))
    #print title
    print(net$title)
  }
  }
}

# question 13

#Find all the actors connected to Mari Yamaguchi through a movie
connection1 = dbGetQuery(imdb_lean, "SELECT DISTINCT n.name, n.id from name2
n, cast_info2 ci, title2 t, role_type rt, kind_type kt WHERE ci.role_id=rt.id AND
t.kind_id=kt.id AND kt.kind in ('movie', 'tv movie', 'video movie')
AND n.id != 3477693 AND rt.role in ('actor','actress') AND ci.person_id=n.id AND
ci.movie_id=t.id AND t.id IN (SELECT title2.id FROM title2, cast_info2 WHERE
cast_info2.movie_id=title2.id AND cast_info2.person_id=3477693)")

#create an empty data frame
data <- data.frame(n1 = character(0), n2 = character(0))
#iterate through every actor connected to Yamaguchi and query all
#those connections in for loop
  for(i in 1:nrow(connection1)) {
    name1=connection1$name[i]
    id1=connection1$id[i]
    #print name and id of the connected actors
```

```
   print(id1)
   print(name1)
   #get name and id of each person using the ids of Yamaguchi's connections
   da = dbGetQuery(imdb_lean, sprintf("SELECT DISTINCT n.name, n.id from name2
n, cast_info2 ci,
   title2 t, role_type rt, kind_type kt WHERE ci.role_id=rt.id AND t.kind_id=kt.id AND
   kt.kind in ('movie', 'tv movie', 'video movie') AND n.id != %d
   AND rt.role in ('actor','actress') AND ci.person_id=n.id AND ci.movie_id=t.id AND
   t.id IN (SELECT title2.id FROM title2, cast_info2 WHERE
cast_info2.movie_id=title2.id AND cast_info2.person_id=%d)",id1,id1))
   #if there is no information, continue
   if(nrow(da)==0) {
    next
   }
   #iterating through the second order connection
   #column 1 will have first order and column 2 will have second order
   for(j in 1:nrow(da)) {
    nextRow=data.frame(name1,da$name[j])
    colnames(nextRow) <- c("n1","n2")
    data=rbind(data,nextRow)
   }
   print(data)
 }

#make data network from data frame
data.network<-graph.data.frame(data, directed=F)
#initialize vertices
V(data.network)
#initialize edges
E(data.network)
plot(data.network)

#names were too big, restrict to ids,
#frame of integers not strings
dataS <- data.frame(n1 = integer(0), n2 = integer(0))

#same loop, except only for top billed actor's connections.
for(i in 1:nrow(connection1)) {
 name1=connection1$name[i]
 id1=connection1$id[i]
 print(id1)
 print(name1)
 #specify nr_order to only top billing
 #originally graphed for ci.nr_order IN(1,2) but ended up chosing just top 1
 da = dbGetQuery(imdb_lean, sprintf("SELECT DISTINCT n.name, n.id from name2
n, cast_info2 ci,
```

```
                        title2 t, role_type rt, kind_type kt WHERE ci.role_id=rt.id AND
t.kind_id=kt.id AND
                        kt.kind in ('movie', 'tv movie', 'video movie') AND n.id != %d AND
ci.nr_order IN (1)
                        AND rt.role in ('actor','actress') AND ci.person_id=n.id AND
ci.movie_id=t.id AND
                        t.id IN (SELECT title2.id FROM title2, cast_info2 WHERE
cast_info2.movie_id=title2.id AND cast_info2.person_id=%d)",id1,id1))
 if(nrow(da)==0) {
  next
 }
 for(j in 1:nrow(da)) {
  nextRow=data.frame(id1,da$id[j])
  colnames(nextRow) <- c("n1","n2")
  dataS=rbind(dataS,nextRow)
 }
 print(dataS)
}
dataS.network<-graph.data.frame(dataS, directed=F)
V(dataS.network)
E(dataS.network)
plot(dataS.network)
```

Some online resources I used besides piazza and class notes/office hour notes:

http://www.w3schools.com/sql/
http://stackoverflow.com/questions/635836/how-can-i-get-both-the-count-of-a-subset-as-well-as-the-count-of-the-total-set-i
http://www.r-bloggers.com/network-visualization-in-r-with-the-igraph-package/http
http://beginner-sql-tutorial.com/sql-subquery.htm
http://stackoverflow.com/questions/7383753/is-it-possible-to-use-the-same-table-twice-in-a-select-queryting-queries