ECE 232E Project 4

IMDb Mining

Jui Chang

Wenyang Zhu

Xiaohan Wang

Yang Tang

1 Actor/Actress network

Question 1.

In this part, we mainly use the data from the following text files:

- actor_movies.txt
- actress_movies.txt

In order to create the network in a consistent manner, we first do data preprocessing from two aspects:

1) Merge the two text files and then removing the actor/actress with less than 10 movies. Then we create a set of actor/actress names to deduplicate and count the total number of actors and actresses;

2) Clean the merged text file: from the given files, we find that there may be a year or one role in a pair of parenthesis after the movie. Since different actors/actresses may act as different roles in the same movie, we eliminate the role and the year, and then count the total number of unique movies.

After doing these two preprocessing parts, we get the following results:

- Total number of actors and actresses (with ≥ 10 movies): 113120
- Total number of unique movies these actors and actresses have acted in: 468190

1.1 Directed actor/actress network creation

Question 2.

In this part, we use the processed text file in question 1 to create a weighted directed actor/actress network. There are three steps:

1) Create a dictionary node_movies with the actor/actress as the key and their movies as the value;
2) Create an index_name_list.txt. We assign each actor/actress with an index for quick search and network generation. In this process, we create two lists name_list and movie_list;
3) Create edge_weight_list.txt for network generation. We traverse each pair of actor and actress, and calculate the edge weight between them using the following equation:

$$w_{i \to j} = \frac{|S_i \cap S_j|}{|S_i|}$$

Where S_i is the set of movies in which actor/actress v_i has acted in and S_j is the set of movies in which actor/actress v_j has acted in.

4) Create idx_ori2gra.txt for mapping. Since read.graph function will re-index the node according to their order of appearance, we need to generate a mapping from the original index to graph index.

Below are the in-degree distribution of the actor/actress network:

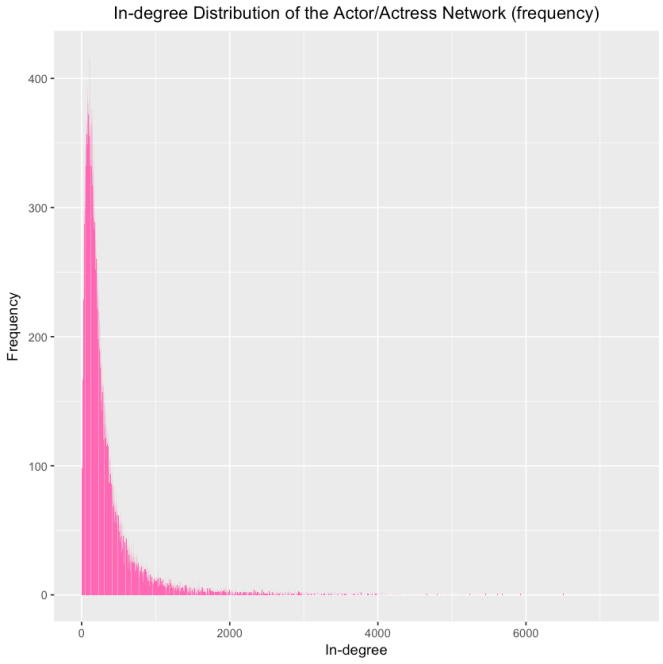


Figure 1. In-degree distribution of the actor/actress network (frequency)

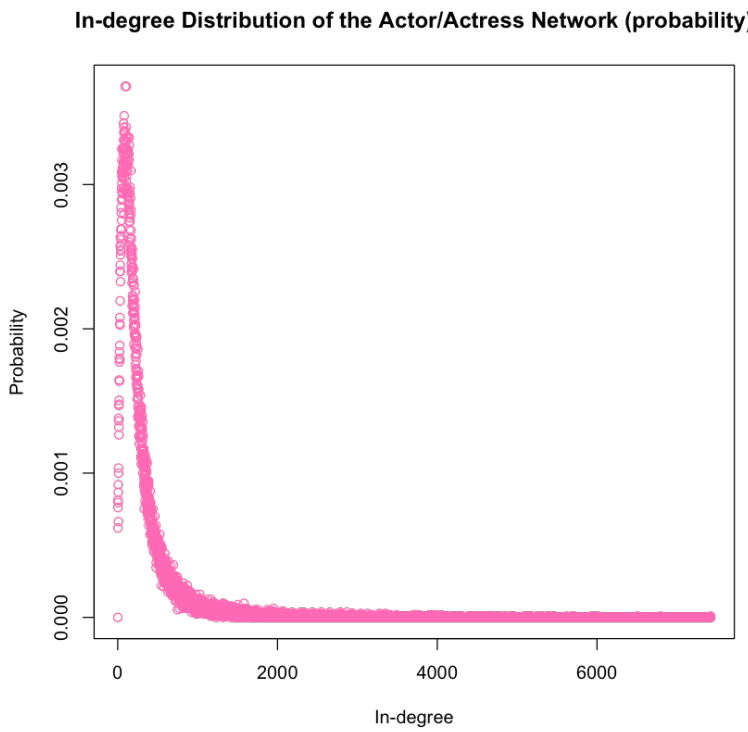


Figure 2. In-degree distribution of the actor/actress network (probability)

From two figures above, we can find that the in-degree distribution is disperse. The in-degree ranged from 0 to more than 7000, while the highest frequency is only 410. In general, the in-degrees mainly centralized from 0 to 1000. In real case, it indicates that among all those actors/actresses who have acted in no less than 10 movies, most people have cooperated with 0 to 1000 colleagues. There are about 410 actors/actresses who have acted in the same movies with about 250 other actors/actresses. Besides, there are also a few actors/actresses who have very wide social network with up to 7000 actors/actresses. They either acted in many movies or in some "big" movies with large number of actors/actresses.

1.2 Actor pairings

Question 3.

In this part, we design an algorithm to find 10 actors' pairings whom they prefer to work the most. Based on our network in the last question, we decide to choose the actor/actress connected with the target actor/actress through edge with max out weight. To be specific, take Tom Cruise as an example. We traverse all the out edges $e_{Tom \rightarrow neighbors}$ and find the edge with max weight as the pairing edge. The neighbor on the other side of this edge is Tom Cruise's pairing. The process steps are in figure 3.

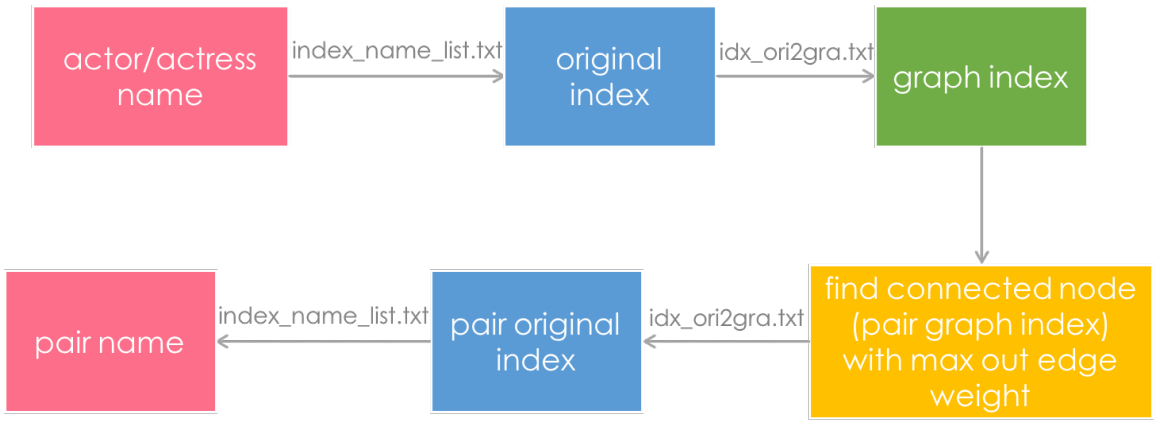


Figure 3. Data process of actor pairings

Below are the pairing results of those 10 actors:

Table 1. Actor pairings results

	Input Actors	Actor Pairings	Pairing Edge Weight	Relationship from Website
1	Tom Cruise	Kidman, Nicole	0.174603174603175	Ex-wife
2	Emma Watson (II)	Radcliffe, Daniel	0.52	Good friend
3	George Clooney	Damon, Matt	0.119402985074627	Good friend
4	Tom Hanks	Allen, Tim (I)	0.10126582278481	Friend
5	Dwayne Johnson (I)	Calaway, Mark	0.205128205128205	Friend
6	Johnny Depp	Bonham Carter, Helena	0.0816326530612245	Friend
7	Will Smith (I)	Foster, Darrell	0.122448979591837	Friend
8	Meryl Streep	De Niro, Robert	0.0618556701030928	Great impact
9	Leonardo DiCaprio	Scorsese, Martin	0.102040816326531	Great impact
10	Brad Pitt	Clooney, George	0.0985915492957746	Good friend

From the table above, in general, the results of actor pairing make sense except the first pair. For the other nine pairs, two actors/actresses are good friends and they have many collaborations. So they are familiar with the acting styles of each other and tend to work together again. Besides, we note that some pairs are tied with great impacts. For example, there is a special Wikipedia page about Leonardo DiCaprio and Martin Scorsese. It writes that *"The pair's relationship is one of the most successful collaborations in film industry, bringing a total of $1.3 billion earnings from their five feature films. DiCaprio called his collaboration with Martin as 'accidental' and considered Taxi Driver and Mean Streets as his inspiration of Martin's work.", "Describing his first collaboration with Scorsese on Gangs of New York, he said, it was an incredible undertaking" (https://en.wikipedia.org/wiki/Martin_Scorsese_and_Leonardo_DiCaprio).* It indicates

that Martin Scorsese has great influence on Leonardo's film career. Thus, it's reasonable that Leonardo wants to work with Martin most.

However, we also note one special relationship between the first pair. Nicole Kidman is Tom Cruise's ex-wife, though they are also partners collaborating in many movies. For some people in same situation, they may choose not to work together after divorcing even though they are very familiar with each other. Therefore, to improve the algorithm, we can add more factors into consideration, e.g. the relationship, characteristics, preference, etc.

1.3 Actor rankings

Question 4.

In this part, we use Google's PageRank algorithm (with damping = 0.85) to find the top 10 actor/actress in the network. Below are our results:

Table 2. Top 10 actor/actress

	Actor/Actress	Number of movies	In-degree	Comments from website
1	Flowers, Bess	828	7421	1898-1984. Known as "The Queen of the Hollywood Extras".
2	Tatasciore, Fred	353	3816	A voice actor/animator known for his work in film, television and games.
3	Harris, Sam (II)	600	6796	1877-1969. Appeared in five Best Picture Academy Award winners.
4	Blum, Steve (IX)	373	3193	A voice actor of animation and video games known for his distinctive deep voice.
5	Miller, Harold (I)	561	6504	1894-1972.
6	Jeremy, Ron	637	2841	Number one adult film star in the U.S during 1980s and 1990s.
7	Lowenthal, Yuri	317	2629	Well-known voice actor in video games and animation.
8	Phelps, Lee (I)	647	5466	1893-1953. Near the top of Hollywood's most prolific

				but virtually unknown supporting players.
9	Downes, Robin Atkin	267	2888	An English screen and voice actor in live action, animation and video games.
10	O'Connor, Frank (I)	623	5392	An actor in supporting roles and most of his work was uncredited.

From the above table, we find that the top 10 list doesn't have any actor/actress listed in the previous section. And we almost don't know those 10 actors/actress. Here are the possible explanations for this phenomenon. First, these top 10 actor/actress are almost in the last century. We weren't born while they were in their prime. So it's reasonable that we didn't know them. Second, according to PageRank algorithm, those who acted in more movies and collaborated with more significant people tend to have higher PageRank scores. Many of these top 10 actor/actress are voice actors or supporting actors, which means that they prefer or have more opportunities to act in many kinds of movies and collaborate with more colleagues, though some of their work was uncredited. From the discussion above, we can conclude that the degree of famous or reputation of an actor/actress is not fully correlated with PageRank scores.

Question 5.

In this part, we run PageRank algorithm on the 10 actor/actress listed in the previous section. Below are our results:

Table 3. PageRank on famous actor/actress

	Listed actor/actress	PageRank scores	Number of movies	In-degree
1	Tom Cruise	3.97411189209544e-05	63	172
2	Emma Watson (II)	1.81106352721262e-05	25	17
3	George Clooney	4.03756231784476e-05	67	98
4	Tom Hanks	5.25320784419174e-05	79	1982
5	Dwayne Johnson (I)	4.23668498123318e-05	78	143
6	Johnny Depp	5.52646971233162e-05	98	285
7	Will Smith (I)	3.2202644209924e-05	49	113
8	Meryl Streep	4.07521762646319e-05	97	72
9	Leonardo DiCaprio	3.21126536720177e-05	49	327
10	Brad Pitt	4.39545211449912e-05	71	161

From the table above, we note that the PageRank scores of the famous actor/actress are significantly less than the top 10's. Besides, those famous actors have much less number of movies and collaborated with much less other actors than those in the top 10 list. This is because the famous actors acted more as the leading roles instead of acting as the supporting actors in many movies as extras. This justifies their lower PageRank scores.

2 Movie network

2.1 Undirected movie network creation

Question 6.

In this part, we use the processed text files (trim.txt in which each actor/actress has no less than 10 movies) to create the movie network.

1) Create a dictionary with the movies as keys and a set of actor/actress as values.
2) Remove the movies with less than 5 actor/actress acted in.
3) Calculate the weights of the edges:

$$w_{i\to j} = \frac{|A_i \cap A_j|}{|A_i \cup A_j|}$$

   Where A_i is the set of actors in movies v_i and A_j is the set of actors in movie v_j.

4) Create xxx.txt for mapping. Since read.graph function will re-index the node according to their order of appearance, we need to generate a mapping from the original index to graph index.

Below are the degree distributions of the movie network:

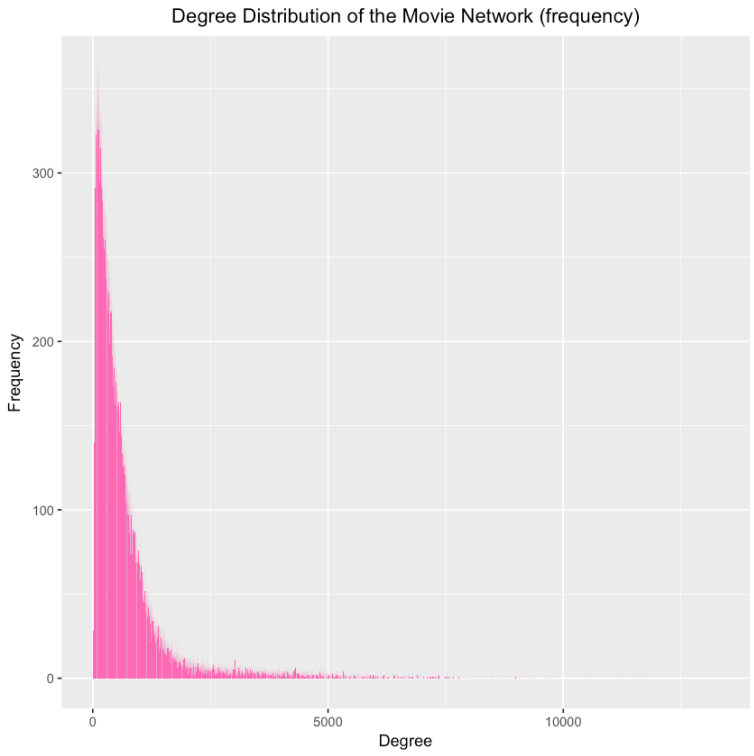


Figure 4. Degree distribution of the movie network (frequency)

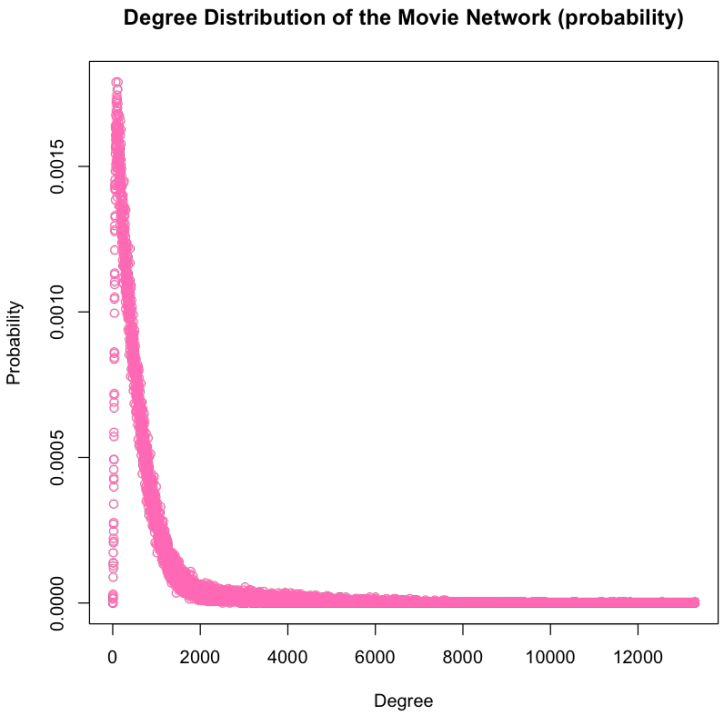


Figure 5. Degree distribution of the movie network (probability)

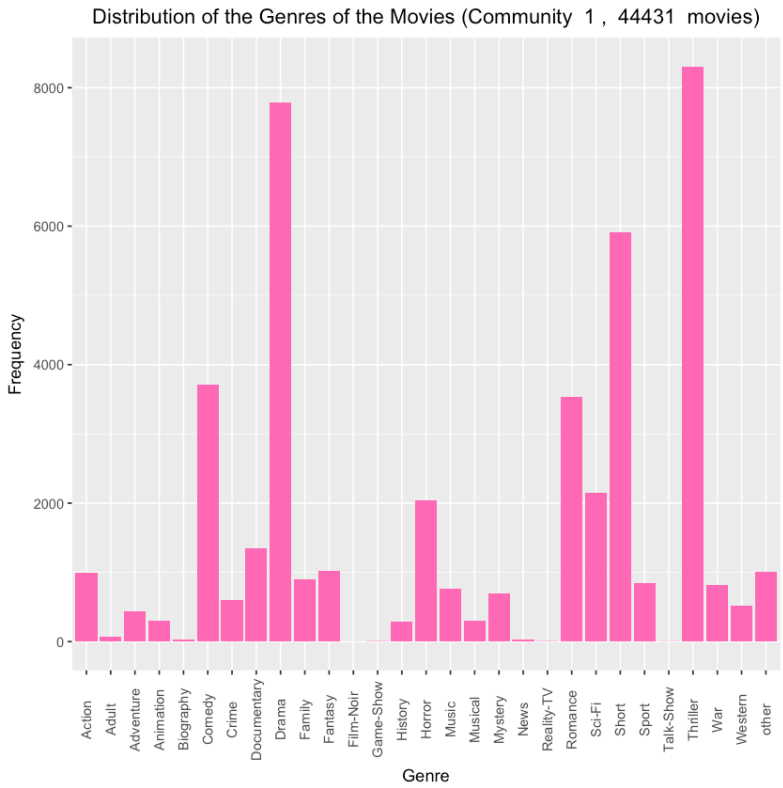
From two figures above, we can find that the degree distribution is disperse. The degree ranged from 0 to more than 13000, while the highest frequency is only 350. In general, the degrees mainly centralized from 0 to 1500. In real case, it indicates that among all movies with at least 5 actors/actresses, most movies have common actor/actress with 0 to 1500 other movies. Besides, there are also a few movies have common actor/actress with up to 13000 other movies. These movies may have many "top 10" like actors/actresses who either are voice or screen actors or supporting actors majoring in many movies as extras.

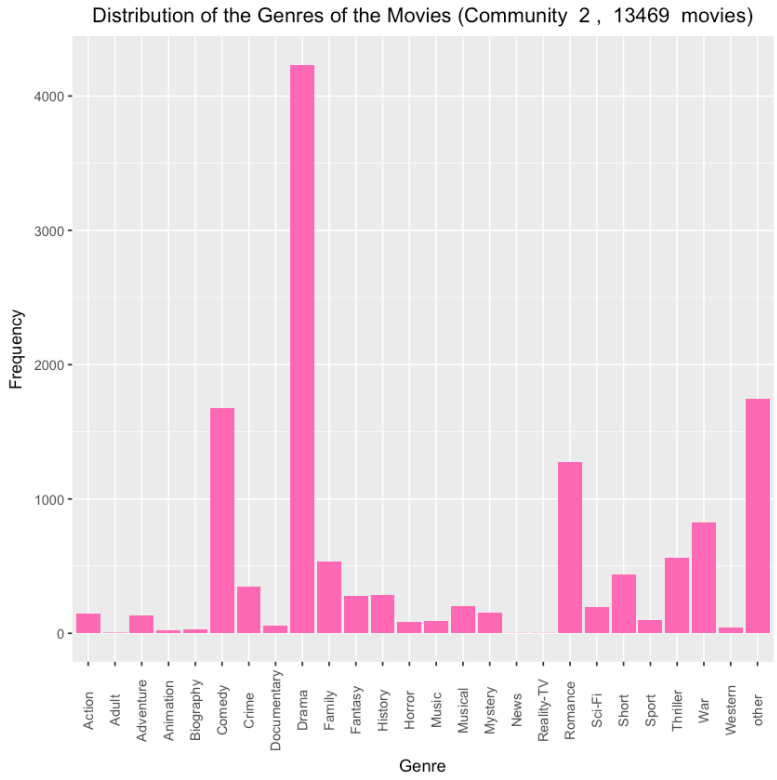
2.2 Communities in the movie network

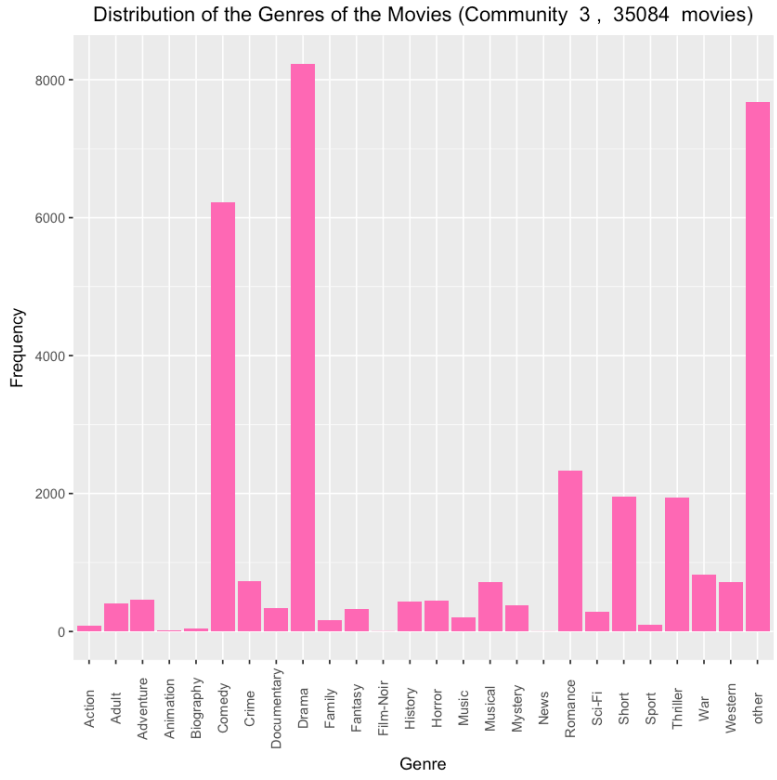
Question 7.

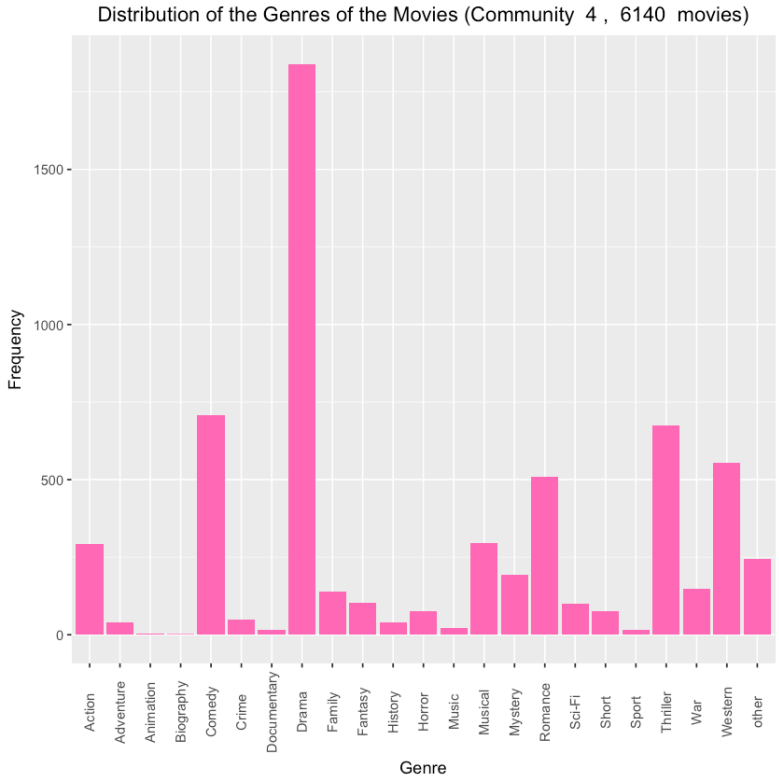
In this part, we use Fast Greedy community detection algorithm to find the communities in the movie network. Then, we pick the first 10 communities. For each community, we create a dictionary with genres as key and number of each genre as value. Then, we traverse each movie in this community and check its genre in movie_genre.txt, and add 1 to the corresponding value. Last, we use the data in dictionary to plot the genre distribution of the movies in each community.

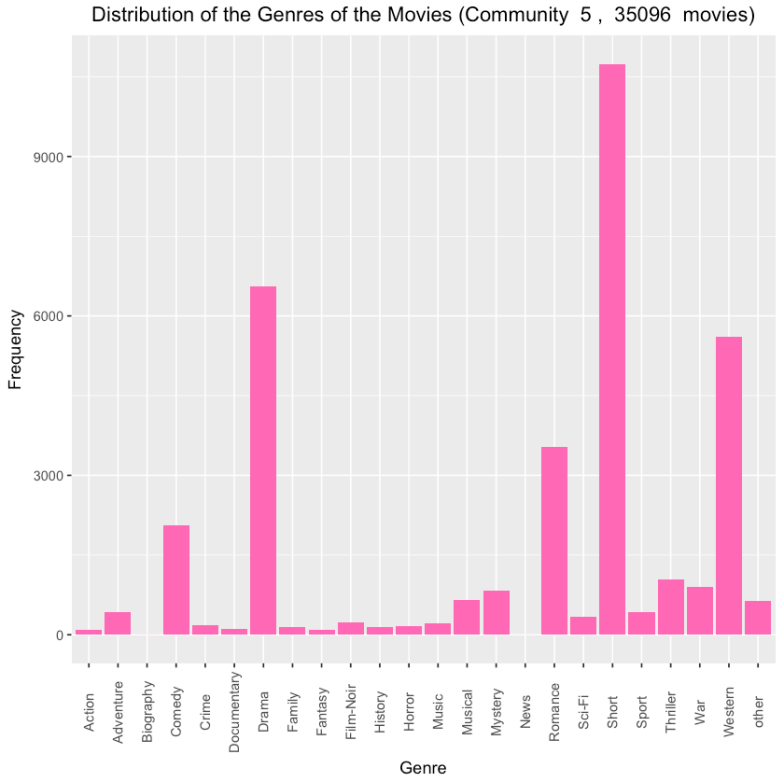
After clustering, we get 29 groups, and the modularity is 0.8. We pick the first 10 groups and plot the genre distribution of the movies in each community. Below are our results:

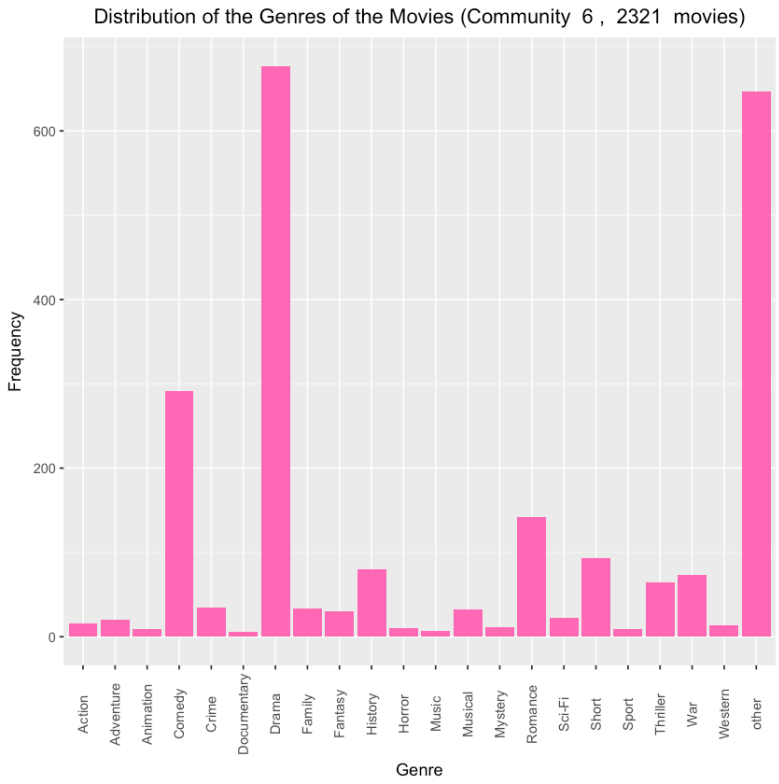


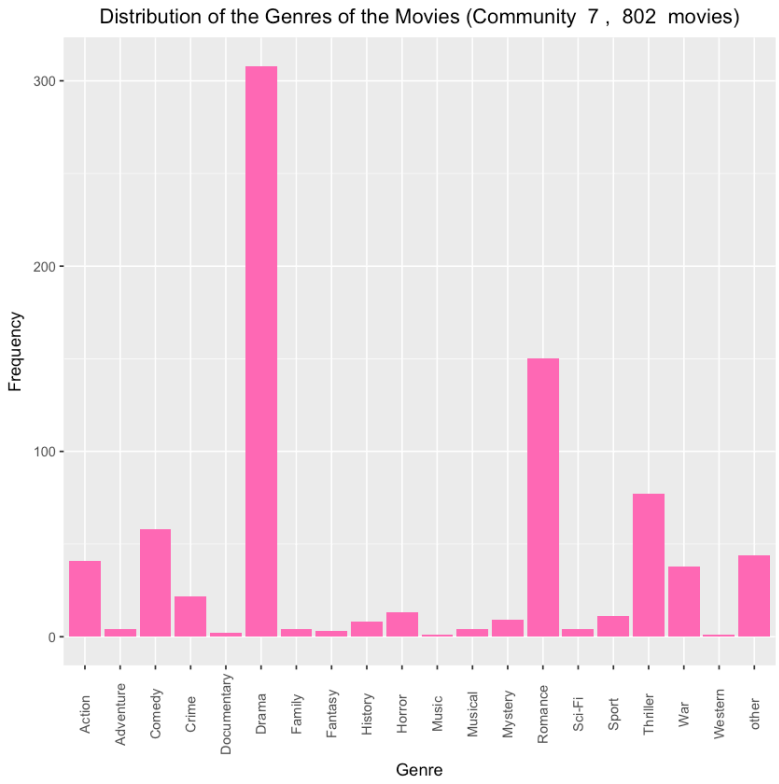

Distribution of the Genres of the Movies (Community 2 , 13469 movies)
Frequency
0
1000
2000
3000
4000
Action
Adult
Adventure
Animation
Biography
Comedy
Crime
Documentary
Drama
Family
Fantasy
History
Horror
Music
Musical
Mystery
News
Reality-TV
Romance
Sci-Fi
Short
Sport
Thriller
War
Western
other
Genre

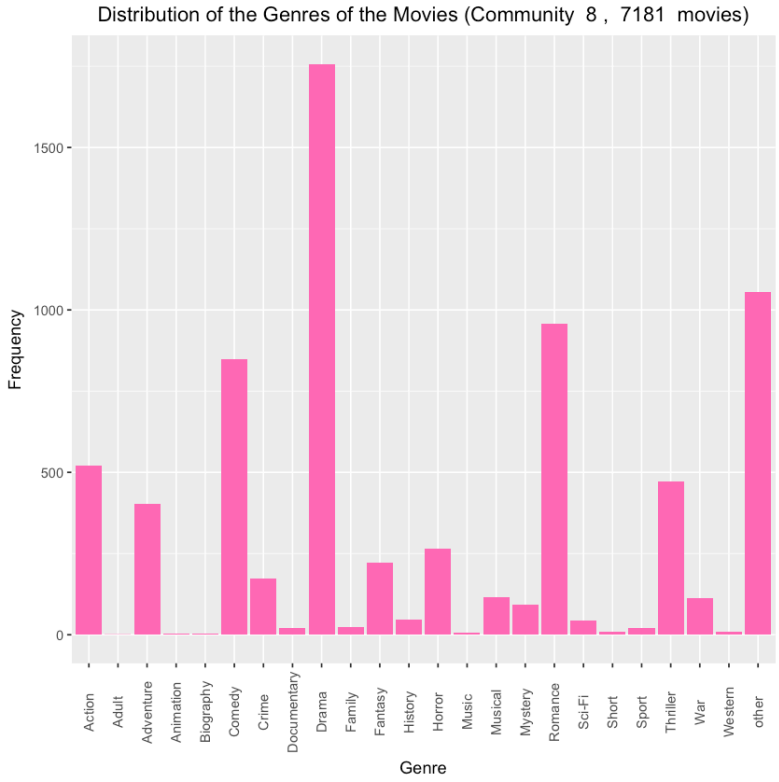

Distribution of the Genres of the Movies (Community 3 , 35084 movies)
Frequency
0
2000
4000
6000
8000
Action
Adult
Adventure
Animation
Biography
Comedy
Crime
Documentary
Drama
Family
Fantasy
Film-Noir
History
Horror
Music
Musical
Mystery
News
Romance
Sci-Fi
Short
Sport
Thriller
War
Western
other
Genre

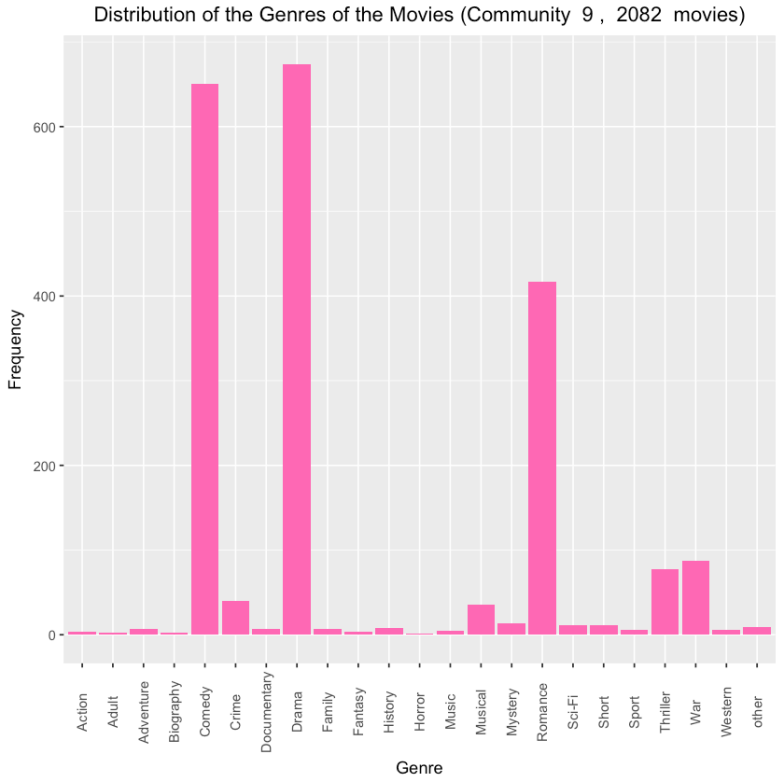

Distribution of the Genres of the Movies (Community 4 , 6140 movies)
Frequency
1500
1000
500
0
Action
Adventure
Animation
Biography
Comedy
Crime
Documentary
Drama
Family
Fantasy
History
Horror
Music
Musical
Mystery
Romance
Sci-Fi
Short
Sport
Thriller
War
Western
other
Genre


Distribution of the Genres of the Movies (Community 5 , 35096 movies)
Frequency
9000
6000
3000
0
Action
Adventure
Biography
Comedy
Crime
Documentary
Drama
Family
Fantasy
Film-Noir
History
Horror
Music
Musical
Mystery
News
Romance
Sci-Fi
Short
Sport
Thriller
War
Western
other
Genre


Distribution of the Genres of the Movies (Community 6 , 2321 movies)
Frequency
600
400
200
0
Action
Adventure
Animation
Comedy
Crime
Documentary
Drama
Family
Fantasy
History
Horror
Music
Musical
Mystery
Romance
Sci-Fi
Short
Sport
Thriller
War
Western
other
Genre


Distribution of the Genres of the Movies (Community 7 , 802 movies)
Frequency
300
200
100
0
Action
Adventure
Comedy
Crime
Documentary
Drama
Family
Fantasy
History
Horror
Music
Musical
Mystery
Romance
Sci-Fi
Sport
Thriller
War
Western
other
Genre


Distribution of the Genres of the Movies (Community 8 , 7181 movies)
Frequency
1500
1000
500
0
Action
Adult
Adventure
Animation
Biography
Comedy
Crime
Documentary
Drama
Family
Fantasy
History
Horror
Music
Musical
Mystery
Romance
Sci-Fi
Short
Sport
Thriller
War
Western
other
Genre


Distribution of the Genres of the Movies (Community 9 , 2082 movies)
Frequency
600
400
200
0
Action
Adult
Adventure
Biography
Comedy
Crime
Documentary
Drama
Family
Fantasy
History
Horror
Music
Musical
Mystery
Romance
Sci-Fi
Short
Sport
Thriller
War
Western
other
Genre

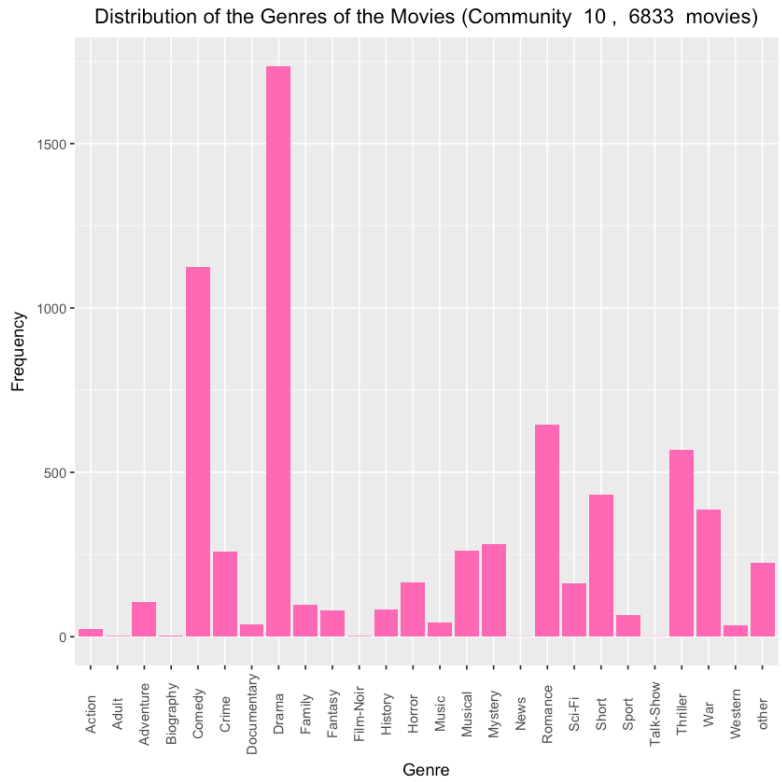


Figure 6. Distribution of genre of the movies (first 10 communities)

Question 8.

(a) There are total 29 communities. The most frequent dominant genres across communities are shown in table below:

Table 4. Most frequent dominant genres across communities

1	Thriller	11	Adult	21	Thriller
2	Drama	12	Short	22	Action
3	Drama	13	Drama	23	Romance
4	Drama	14	Drama	24	Short
5	Short	15	Drama	25	Adult
6	Drama	16	Drama	26	Short
7	Drama	17	Drama	27	Musical
8	Romance	18	Romance	28	Short
9	Drama	19	Romance	29	Drama
10	Drama	20	Drama		

In each community, the most frequent dominant genre means that the number of movies with such genre is the largest one among all genres in this community. There are so many "Drama" dominant genre because a large number of movies belongs to genre "Drama".

(b) The most dominant genre in each community based on the modifies scores are as below:

Table 5. Most frequent dominant genres across communities (modified)

1	Sci-Fi	11	Adult	21	Thriller
2	War	12	Short	22	Action
3	Crime	13	Romance	23	Action
4	Western	14	Musical	24	Short
5	Film-Noir	15	War	25	Adult
6	History	16	Family	26	Romance
7	Romance	17	Drama	27	Musical
8	Adventure	18	Adventure	28	Short
9	Comedy	19	Romance	29	Drama
10	Mystery	20	Comedy		

The genres of most communities changed due to the modified score, and most of the "Drama" genres change to other genres. It is because that the modified score is based on fraction instead of the number of genres. Though there are many movies with "Drama" genre in each community, the fraction between the number of "Drama" movies in a community and the one in the whole dataset is low.

(c) Community with index 26 has size between 10 and 20 (actually there are 18 movies). There are 17 actors who acted in these 18 movies, and here is the bipartite graph.

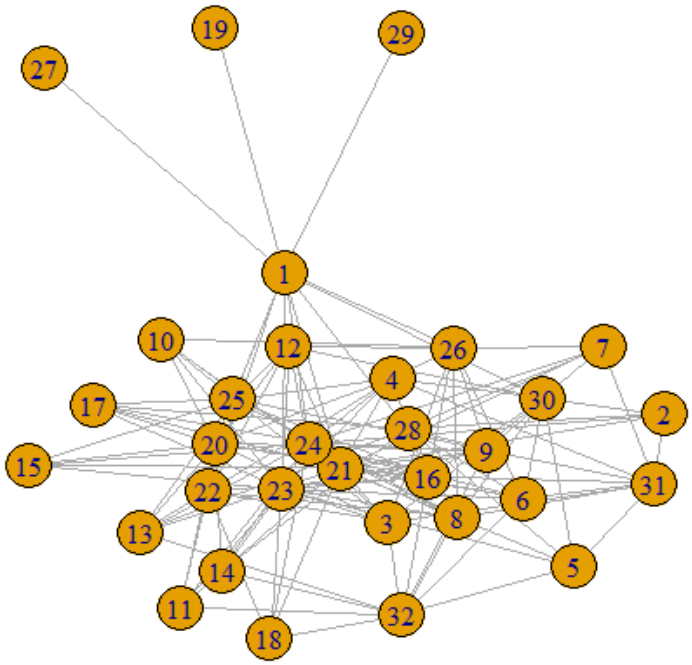


Figure 7. bipartite graph

And here is the map between the index in bipartite graph and the names of movies and actors.

Table 6. Movies in bipartite graph

1	Unconditional Love (2010)	10	A Schoolboy Error Production (2009)
2	Booze Culture (2012)	11	Be My Valentine (2011)
3	Boycie (2011)	12	Legion of Evil (2010)
4	Call of Babylon (2012)	13	Love House (2011)
5	Inner Joy of a Broken Heart (2011)	14	The Book of Life (2013/I)
6	Is This It? (2012)	15	The Hope Within (2009)
7	Life of a Spy (2012)	16	The Shadow of Death (2011)
8	Losers in Love (2011)	17	White Cobra (2014)
9	Street Fight (2012)	18	Fear of the Park (2010)

Table 7. Actors in bipartite graph

19	Dasz, Steven	26	Taylor, Stuart (X)
20	Joiner, Craig	27	Chan, Juju
21	McKay, Reuben	28	Kilpatrick, Kayleigh
22	Moir, Shaun	29	Marshall, Scarlett
23	Noble, Graeme	30	McKay, Hannah
24	Noble, John-William	31	Hislop, Tom
25	Sandison, Martin	32	Simpson, Julia (II)

Table 8. Three most important actor

Index	Actor	Num. of movies acted in this community
1	McKay, Reuben	18
2	Noble, John-William	17
3	Noble, Graeme	15

These three actors acted the largest number of movies in this community among all 14 actors. For community 26, the genre in 8(a) is “Short” and the one in 8(b) is “Romance”. There is a correlation between these actors and the dominant genres, since a large part of the movies these actors acted belong to “Romance” and “Short”.

2.3 Neighborhood analysis of movies

Question 9.

The average rating of the movies in the neighborhood of "Batman v Superman: Dawn of Justice (2016)" is 6.3388, which is similar to the rating 6.6.

The distribution of the available ratings of the movie in the neighborhood is:

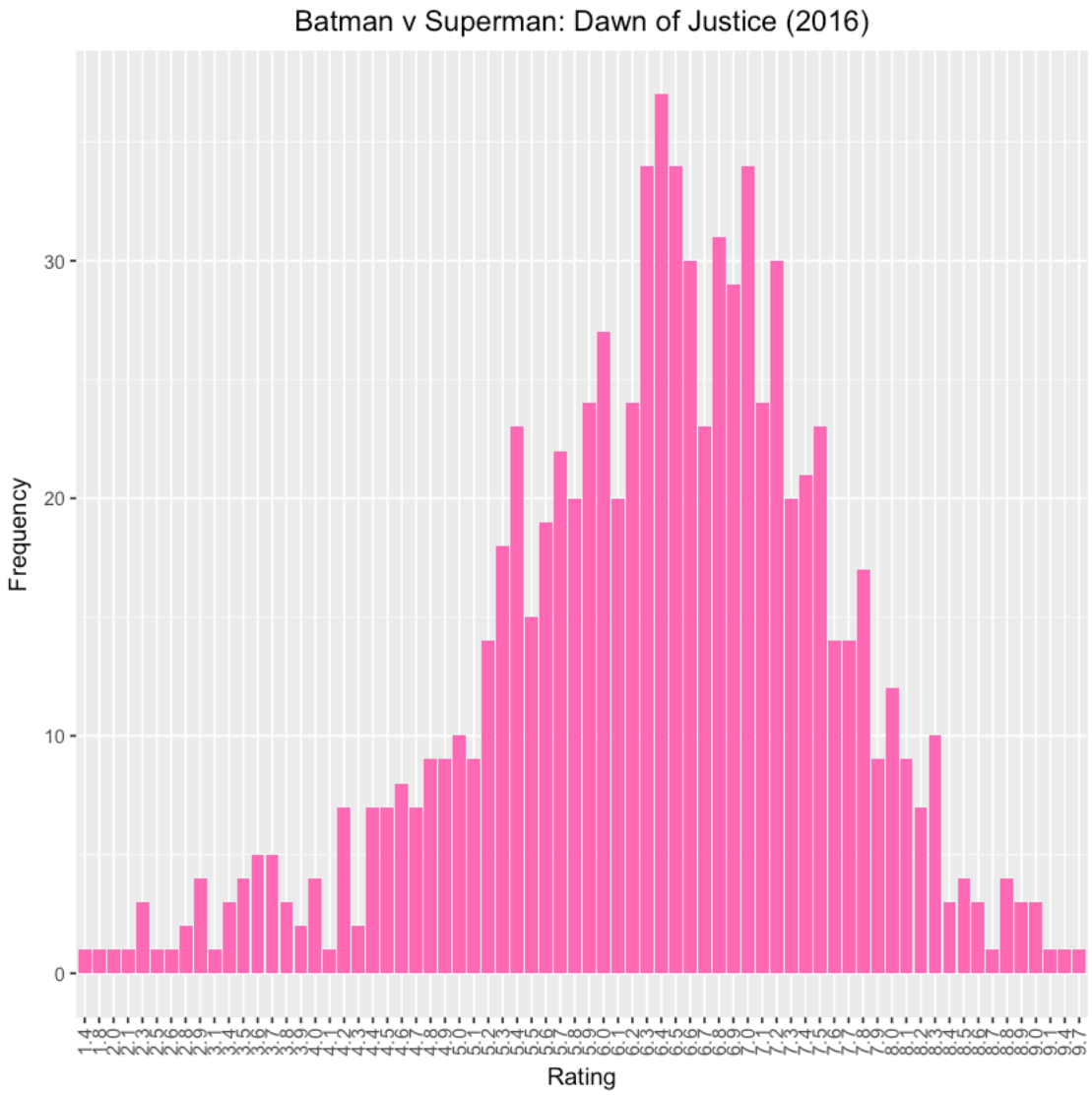


Figure 8. Rating distribution of Batman v Superman: Dawn of Justice (2016)

The average rating of the movies in the neighborhood of “Mission: Impossible - Rogue Nation (2015)” is 6.2284, which is a lower than the rating 7.4.

The distribution of the available ratings of the movie in the neighborhood is:

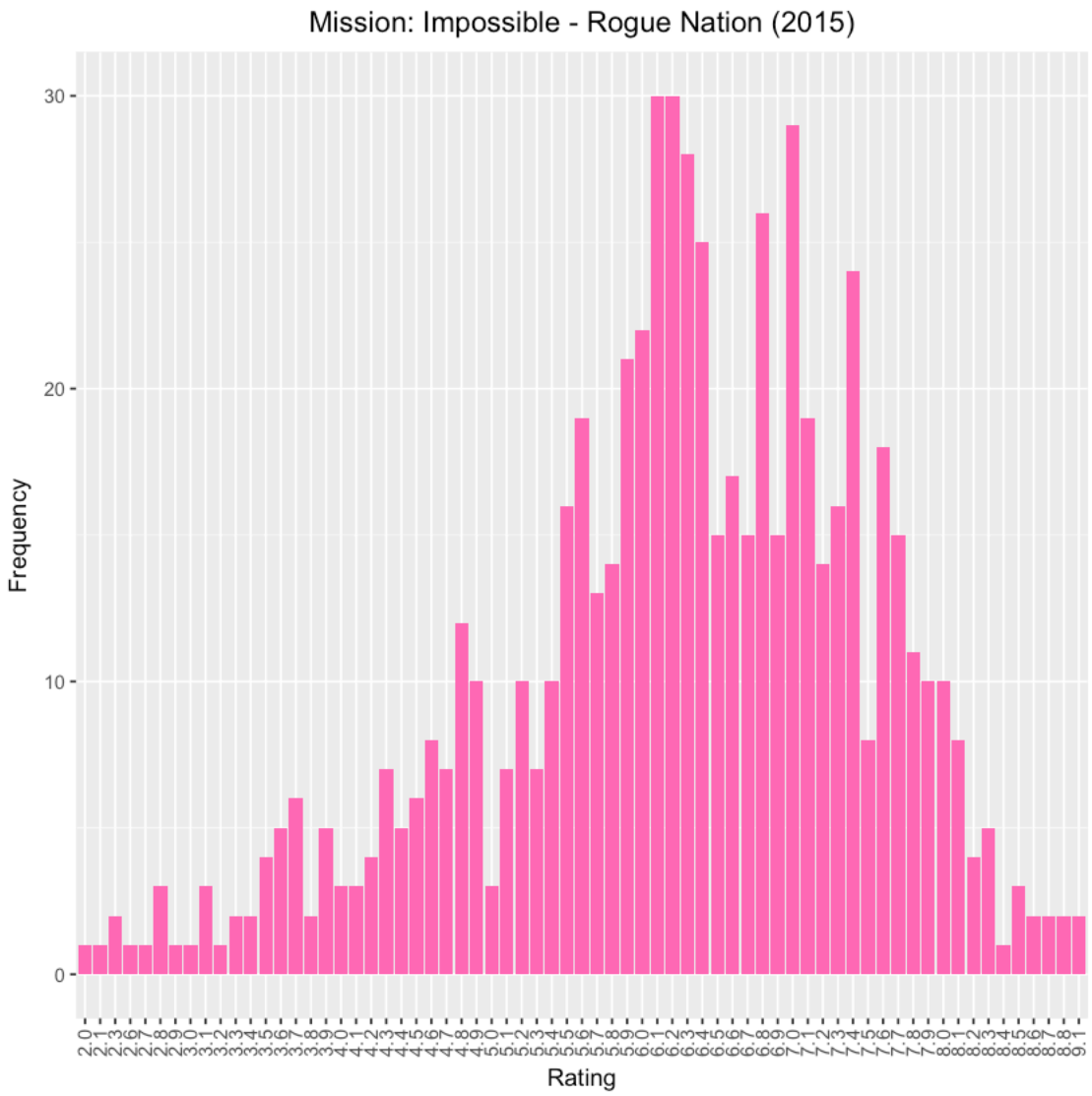


Figure 9. Rating distribution of Mission: Impossible - Rogue Nation (2015)

The average rating of the movies in the neighborhood of “Minions (2015)” is 6.8283, which is a little higher than the rating 6.4.

The distribution of the available ratings of the movie in the neighborhood is:

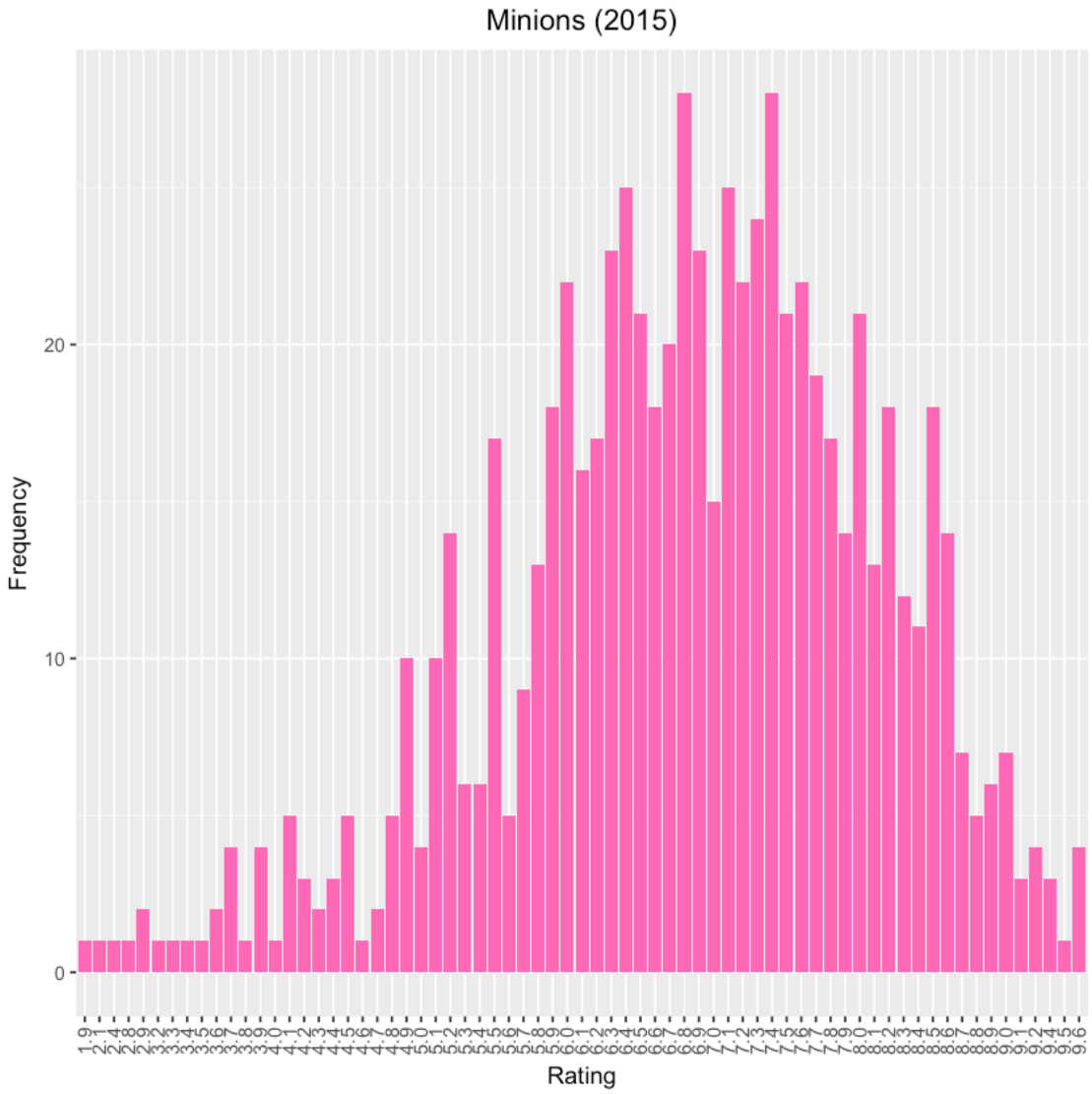


Figure 10. Rating distribution of Minions (2015)

Question 10.

The average rating of the movies in the restricted neighborhood of "Batman v Superman: Dawn of Justice (2016)" is 6.3325, which is similar to the rating 6.6.

The distribution of the available ratings of the movie in the neighborhood is:

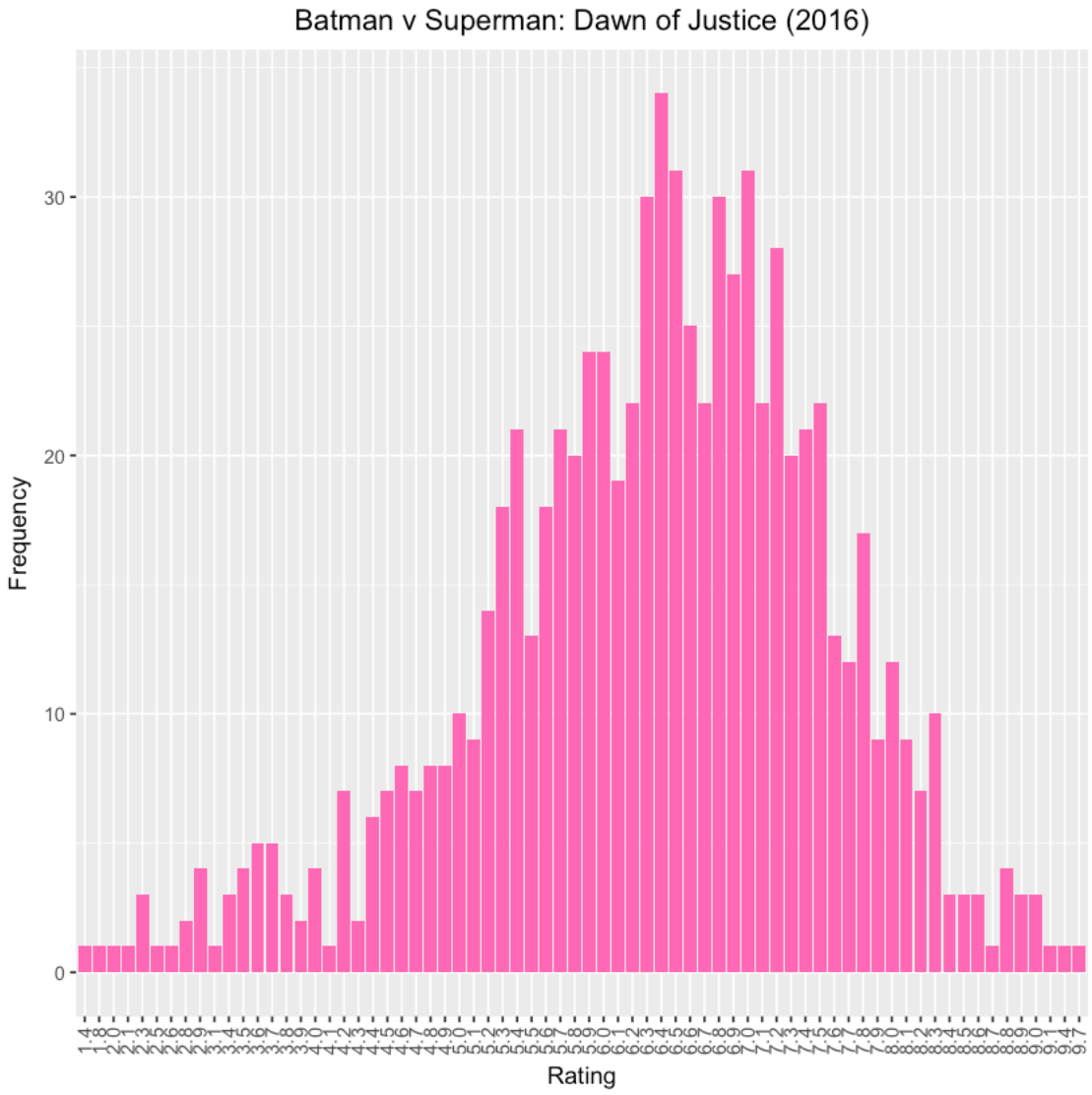


Figure 11. Rating distribution of Batman v Superman: Dawn of Justice (2016) (restricted neighborhood)

The average rating of the movies in the restricted neighborhood of "Mission: Impossible - Rogue Nation (2015)" is 6.2635, which is a little lower than the rating 7.4.

The distribution of the available ratings of the movie in the neighborhood is:

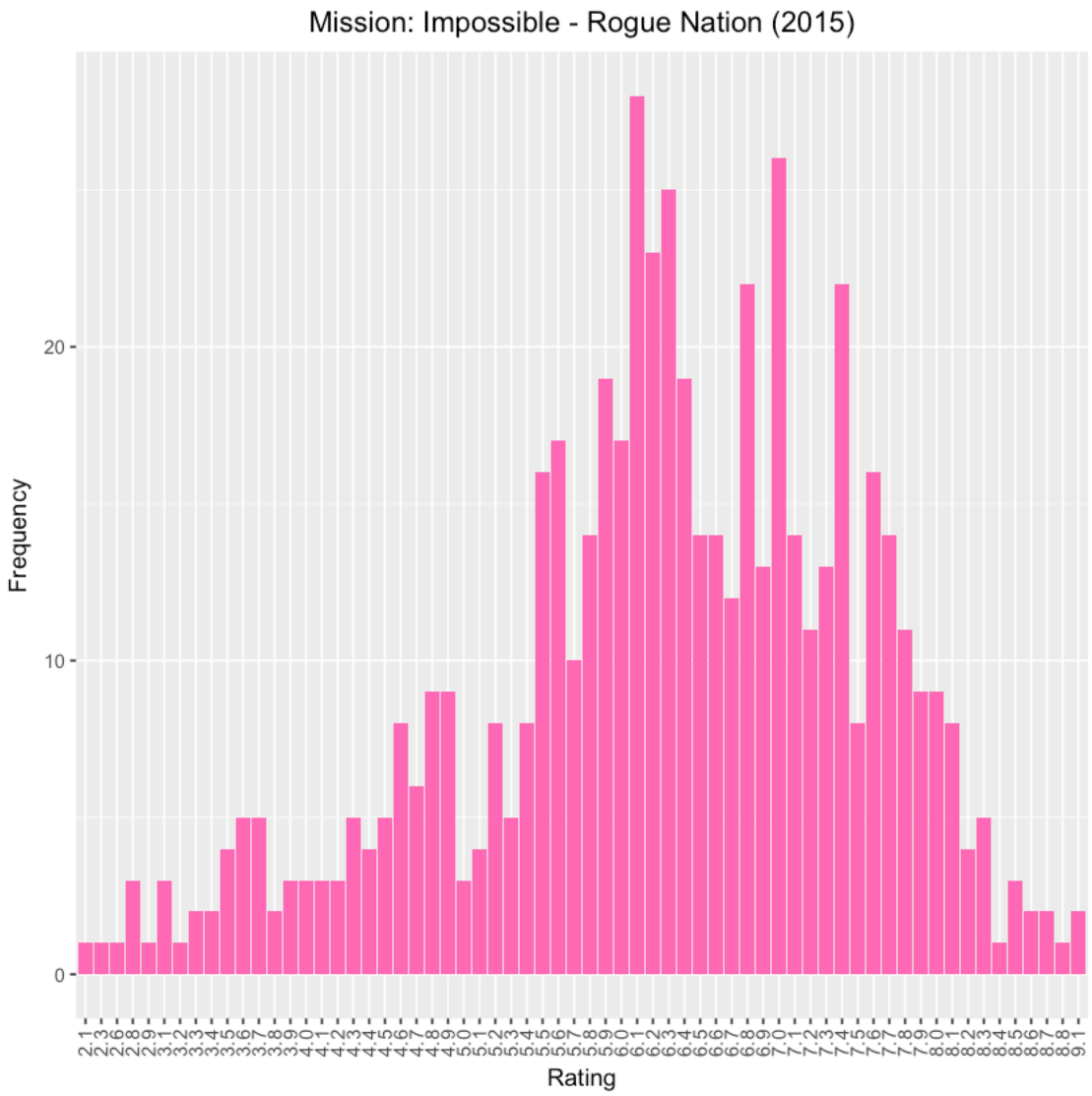


Figure 12. Rating distribution of Mission: Impossible - Rogue Nation (2015) (restricted neighborhood)

The average rating of the movies in the restricted neighborhood of “Minions (2015)” is 6.8388, which is similar to the rating 6.4.

The distribution of the available ratings of the movie in the neighborhood is:

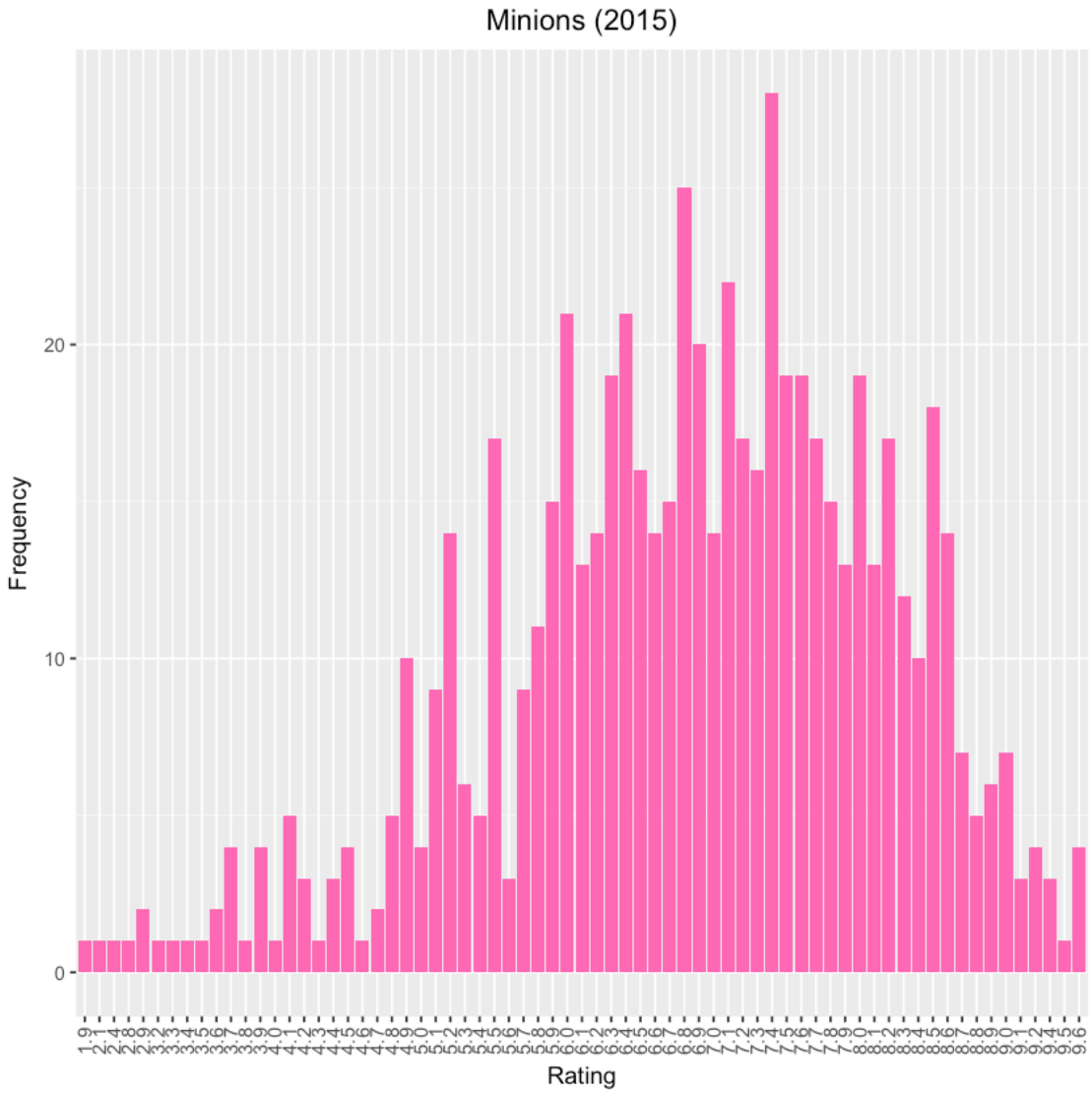


Figure 13. Rating distribution of Minions (2015) (restricted neighborhood)

Below is the summary of the average movie ratings:

Table 9. Compare average movie ratings

Central movies	Ratings of the central movies	Average ratings of the neighborhood	Average rating of the restricted neighborhood
Batman v Superman: Dawn of Justice (2016)	6.6	6.3388	6.3325
Mission: Impossible - Rogue Nation (2015)	7.4	6.2284	6.2635
Minions (2015)	6.4	6.8283	6.8388

From the table above, we find that the average ratings of the neighborhood and the restricted neighborhood are generally similar to the ratings of the central movies. Thus, we may deduce that the actor/actress's professional skills are generally stable that the movies they acted in tend to have similar ratings. Besides, we didn't get a better match in question 10, in other words, the average ratings of the restricted neighborhood didn't improve much than that without restriction. So we can conclude that the community didn't correlate to the movie ratings.

Question11.

From previous question, we can get neighbor indexes of the three target movies, but the indexes are from graph, so we have to change the graph index to movie index to do further steps. For each target movie, we have to use its neighbor index to search in top 5 edge weight neighbor and find the community number of the top 5 neighbors. The results are as follows.

Table 10. The top 5 neighbors and their community for three target movies

(a) Batman v Superman: Dawn of Justice (2016)

Batman v Superman: Dawn of Justice (2016), community = 1		
top	Movie name	Community number
#1	Eloise (2015)	1
#2	The Justice League Part One (2017)	1
#3	Into the Storm (2014)	1
#4	Love and Honor (2013)	1
#5	Man of Steel (2013)	1

(b) Mission Impossible – Rogue Nation (2015)

Mission Impossible – Rogue Nation (2015), community = 1		
top	Movie name	Community number
#1	Fan (2015)	13

#2	Phantom (2015)	13
#3	Breaking the Bank (2014)	1
#4	Suffragette (2015)	1
#5	Now You See Me: The Second Act (2016)	1

(c) Minions (2015)

Minions (2015), community = 1		
top	Movie name	Community number
#1	The Lorax (2012)	1
#2	Inside Out (2015)	1
#3	Up (2009)	1
#4	Surf's Up (2007)	1
#5	Despicable Me 2 (2013)	1

From the result in above table, we can observe that for a target movie, the top 5 neighbors would usually be in the same community. Besides, the neighbor movies share some common point with the target movie. For example, for top neighbors of Minions, they are all cartoons.

2.4 Predicting ratings of movies

Question 12.

From question 6, we can get the weights of edge in movie network. In this part, we would use the weight to train a regression model to predict the rating of the target three movies: Batman v Superman: Dawn of Justice (2016), Mission Impossible – Rogue Nation (2015), Minions (2015).

First, we load the edge weight data of the movie networks. Because of time and space constraint, we only take the first 10,000 movies + 3 target movies vs first 10,000 movies rating to analyze. We create a table named 'rating_mat', which is movie index vs movie index, the value is the edge weight of two movies. Then, we load movie_rating.txt, and remove the movie in table if the movie has no rating, so there are 7950 movies + 3 target movies to analyze.

To extract our feature, for each movie, we take top 5 edge weight movies and take their rating as features.

Based on assumption that the two movies with higher edge weight influence more on each other's rating, so we take top 5 edge weight movies and take their rating as features.

After finding the 5 feature, we take 7950 movies as train set, and 3 target movies as test set. We train and predict the rating by linear regression from sklearn.linear_model, and we get the following result.

Table 11. The predicted rating of three target movies and RMSE of training set

	Real Rating	**Predict Rating**
Batman v Superman: Dawn of Justice (2016)	6.6	6.575
Mission Impossible – Rogue Nation (2015)	7.4	5.661
Minions (2015)	6.4	6.947
Training RMSE = 1.0613		

From the table above, we can discover that the predict rating of 'Batman v Superman: Dawn of Justice (2016)' and 'Minions (2015)' are good, but the predict rating of Mission Impossible – Rogue Nation (2015) are far lower than the real rating, which might because the top 5 related movies received low rating and affect the prediction.

Question 13.

First, we load the trimmed text from question1 (movie to actors) and remain the movies that has at least 5 actors/actress as in question 6. Then, we turn the movies and actors/actress to into index. The following are two parts of question.

1. Create bipartite graph: To prevent confusion in the index of movies and actors/actress, I add 500,000 to the index of actors/actress. Because the largest movie index is 202874, so the index below 500,000 represents movie index and above 500,000 are actor/actress index. Then, we load the modified movie to actor text file in r and create an undirected graph, and the following figures are the degree distribution of the graph:

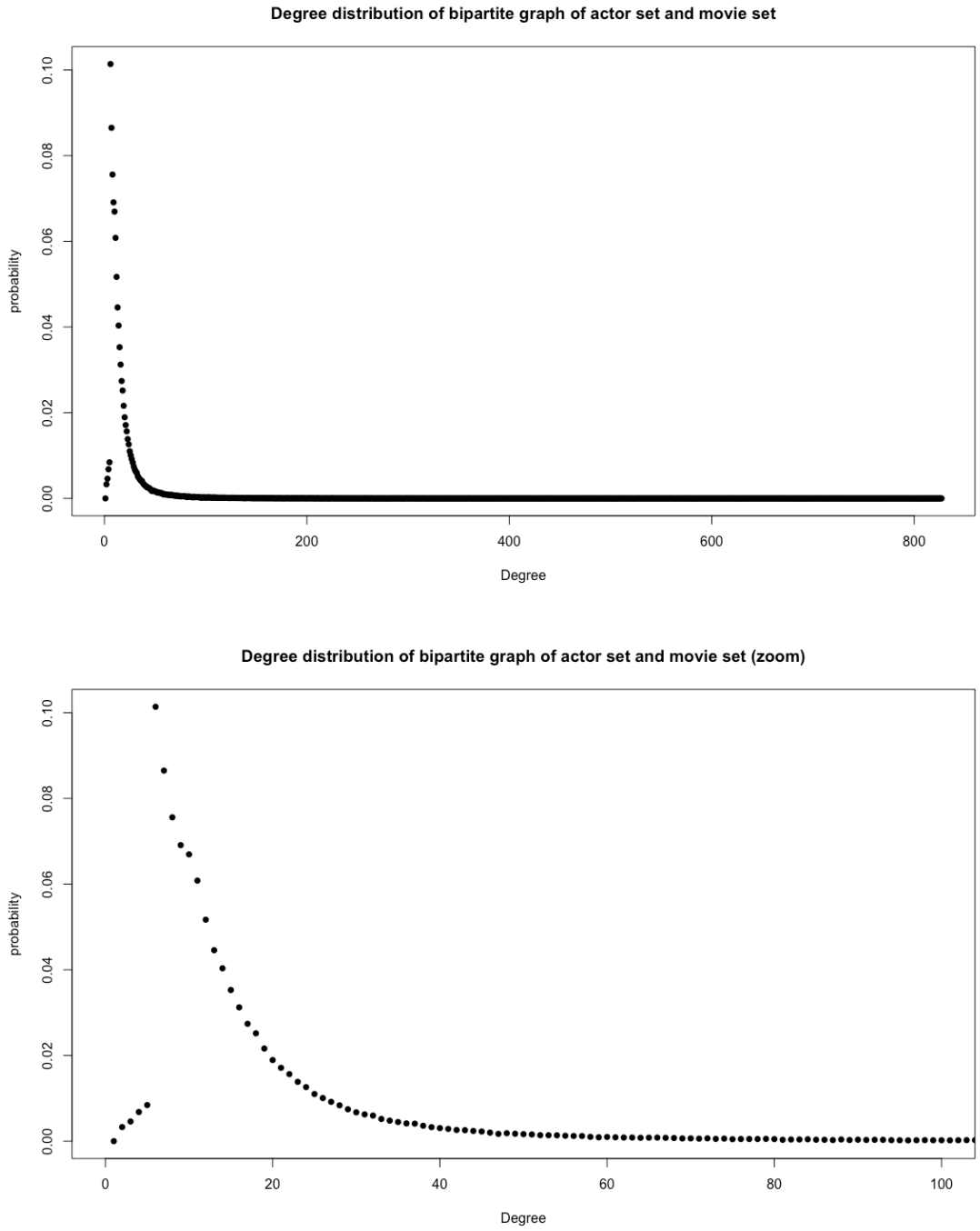


Figure 14. In-degree Distribution of Bipartite Graph of Actor/Actress Set and Movie Set

In the degree distribution figure above, we can observe that the degree of 1 to 4 is low because in the beginning, we remove the movies with less than 5 actors/actress. The degree of 6 to 20 has the highest probability, which means the movie acted by 6 to 20 actors/actress or an actor/actress acts in 6 to 20 movies.

2. Regression:

First, we create a table named 'mov2act', which is movie vs actor index. If an actor/actress acts in the movie, the value is 1. For example, if actor#2 and actor#4 act in movie#1, (2,1) and (4,1) in table would be 1. Like question 12, because of time and space constraint, we only take the first 10,000 movies + 3 target movies to analyze. Then, we load the movie_rating.txt and remove the movie in our set if the movie has no rating, so there are 7950 movies + 3 target movies to analyze. (table 'mov2act_new')

To find the weight of each actor, we sum the number of movie he/she act in.

To find the average rating of an actor, in training set, I take the average movie rating for the actors they act and use the array as 'actor rating'. Then, I plug the actor rating into 'mov2act_new' to create a raw feature table. In raw feature table, for each movie(row), if an actor acts in the movie, the value of that position would be the actor's rating.

To reduce dimension and make better performance, I would reduce dimension into 5 new features:

1. weight average
2. percentile 75
3. percentile 50
4. percentile 25
5. variance

For feature 1, weight average. For each movie(row), we take the weight average of actor's rating (weight = actor weight).

For feature 2, 3, and 4, for each movie(row), we take the percentile 75, 50, and 25 of actor's rating.

For feature 5, variance. For each movie(row), we take the variance of actor's rating.

Based on the assumption that the casts affect the movie rate, so we take the actor's rate into account. Because the weight of actor might affect the rating of movie, we take actor's rating weight average as a feature. However, there might be some outlier actor's rating affect a lot to the feature. Therefore, for a movie, we want to add the actor's rating distribution by take the percentile 25, 50, and 75, and variance of actor's rating as feature to prevent the bias from the weight average.

After finding the 5 new feature, we take 7950 movies as train set, and 3 target movies as test set. We train and predict the rating by linear regression from sklearn.linear_model, and we get the following result.

Table 11. The predicted rating of three target movies and RMSE of training set

	Real Rating	**Predict Rating**
Batman v Superman: Dawn of Justice (2016)	6.6	7.718
Mission Impossible – Rogue Nation (2015)	7.4	7.385
Minions (2015)	6.4	6.750
Training RMSE = 0.558		

From the above table, we can observe that the predicted rating of this mechanism is better than the one in question 12. The predict ratings are much more close to the real rating of the movies, and also the training RMSE is 0.558 is much lower than the RMSE in question 12, which is 1.06. This might because in this part, we directly take actor into account, rather than using movie to judge movie by the relation of the casts. For example, if a protagonist plays in two movies which has low actor relation, the two movies will not have high edge weight, and in question 12, the feature will not be selected out, but protagonist plays import role in a movie and affect the rating a lot. However, in question 13, the protagonist will have effect on the feature because we take weight average of actors rating and percentile as features. For the feature in this question, the actor rating directly affects the movie prediction, and in the real world, the cast usually plays important role on the rating of the movie, so it is reasonable that the performance of rating mechanism in this question is better than previous one.