***Project 4***

***IMDb Mining***

**Di Jin (305026178)**

**Yangyang Mao (504945243)**

**Tao Wu (504946672)**

**Yufeng Huang (704944399)**

**6/5/2018**

**Question 1.**

First we performed the preprocessing on ‘actor\_movies’ and ‘actress\_movies’. We made a dictionary to store the actors and their movies. By the meanwhile we open a new txt file to store all the actors/actress and their movies into one file called ‘merged’.

In order to get an easy analysis in graph, we use the actor id to replace the actors name here. To get the number of the actors and the movies, we used the lists to store the all the actors and the actress and get the length of the list. But in the movie lists, there are some repeat movies when we add the movie in to the lists. Hence we used the set() function to count the number of the unique movies. For its a quick function to count the unique movies in a list.

What’s more, there may be two same movies with different version so we create a ‘clear function’ on the movie’s name to get the movie’s name. We only keep the information of the name and year. And then we can get our results from the lists we create.

The result is shown following:

The total number of actors is 74598

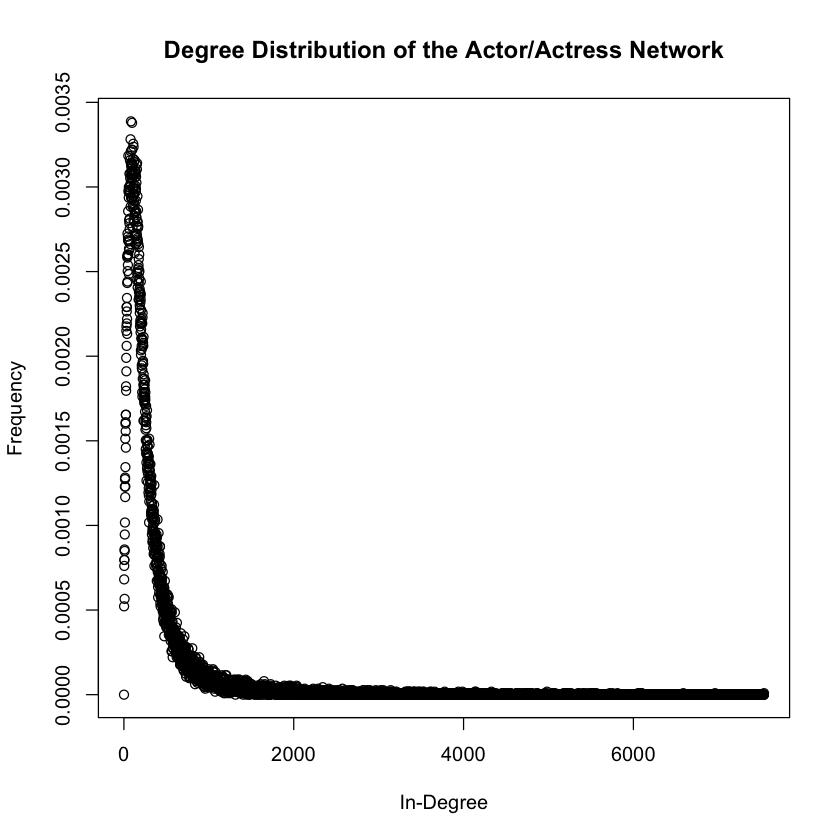
The total number of actresses is 38534

The total number of actors and actresses is 113132

The total number of unique movies is 468198

**Question 2.**

In order to make the network we first create a network txt file and calculate the weight of two nodes. We read the data from previous ‘merged.txt’ file and use the dic we created from previous part. Here we use len(set(actor1) & set(actor2)) to calculate the same movie between two actors. Then we open a new txt file to store the the nodes and weight between two nodes. After we create the network we get a degree distribution which is shown below:



We can find the the shape of the distribution looks like a 1/x function which means the nodes with small degree has a large frequency.

**Question 3.**

In this problem we want to find the the actors with whom the input actor prefers to work the most. In our algorithm, we first find the actor id for chosen actors and then in our weight dictionary we want to find the actor id with the highest weight corresponding with the chosen actor. And following is the result.

|  |  |
| --- | --- |
| ***Actor*** | ***Whom the actor prefers to work the most*** |
| George Clooney | Matt Damon |
| Tom Cruise | Nichole Kidman |
| Johnny Depp | Helena Bonham Cater |
| Leonardo DiCaprio | Martin Scorsese |
| Tom Hanks | Tim Allen(I) |
| Dwayne Johnson (I) | Steve Austin, Mark Calaway, Paul Levesque |
| Brad Pitt | George Clooney |
| Will Smith (I) | Darrell Foster |
| Meryl Streep | Robert De Niro, Kevin Kline(I) |
| Emma Watson (II) | Daniel Radcliffe |

**Question 4.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Top 10 Actors** | **Actor ID** | **Page rank** | **# of movies** | **in-degree** |
| Flowers, Bess | 85734 | 0.0002351278 | 828 | 7537 |
| Tatasciore, Fred | 65947 | 0.0001989547 | 353 | 3954 |
| Harris, Sam (II) | 27643 | 0.0001971991 | 600 | 6960 |
| Blum, Steve (IX) | 6539 | 0.0001954943 | 373 | 3316 |
| Miller, Harold (I) | 45415 | 0.0001727122 | 561 | 6587 |
| Jeremy, Ron | 32130 | 0.0001585656 | 637 | 2905 |
| Phelps, Lee (I) | 52784 | 0.0001573124 | 647 | 5563 |
| Lowenthal, Yuri | 40351 | 0.0001567347 | 317 | 2262 |
| Downes, Robin Atkin | 18112 | 0.0001517775 | 267 | 2953 |
| O'Connor, Frank (I) | 49651 | 0.0001469482 | 623 | 5502 |

We can easily find that all of actors in this top 10 list are not in the original top 10 list. This may be caused by the number of edges. These old actors/actresses have already finished a large amount of movies. Besides, the number of in degree edges is incredibly large which indicated a large relationship network. A large relationship network may indicate how popular this actor/actress. See the actors/actresses in the given top 10 list. We find that most of them are young. They are popular in recent years, but compare to others in the whole movie history, they may have a lower page rank score with less number of movies and in-degree.

**Question 5.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Actors** | **Actor ID** | **Page rank** | **# of movies** | **in-degree** |
| George Clooney | 12812 | 0.00004003069 | 67 | 1573 |
| Tom Cruise | 14503 | 0.00003974764 | 63 | 1651 |
| Johnny Depp | 16878 | 0.00005381414 | 98 | 2144 |
| Leonardo DiCaprio | 17285 | 0.00003168077 | 49 | 1301 |
| Tom Hanks | 27258 | 0.00005105135 | 80 | 2064 |
| Dwayne Johnson (I) | 32389 | 0.00004201959 | 78 | 1357 |
| Brad Pitt | 53248 | 0.00004297697 | 71 | 1739 |
| Will Smith (I) | 62774 | 0.00003201717 | 49 | 1319 |
| Meryl Streep | 107832 | 0.00003961195 | 97 | 1594 |
| Emma Watson (II) | 111298 | 0.00001748575 | 25 | 453 |

**Question 6.**

In this question we use the similar method in question 2 and the only difference here is the weight calculation. Here we created a new txt file called ‘moive\_network.txt’ to store the nodes and their weights. According to the equation from the statement, here we used following code to calculate the weight and write the network txt file.

def cal\_weight(movie1, movie2):

actor1 = set(dic[movie1])

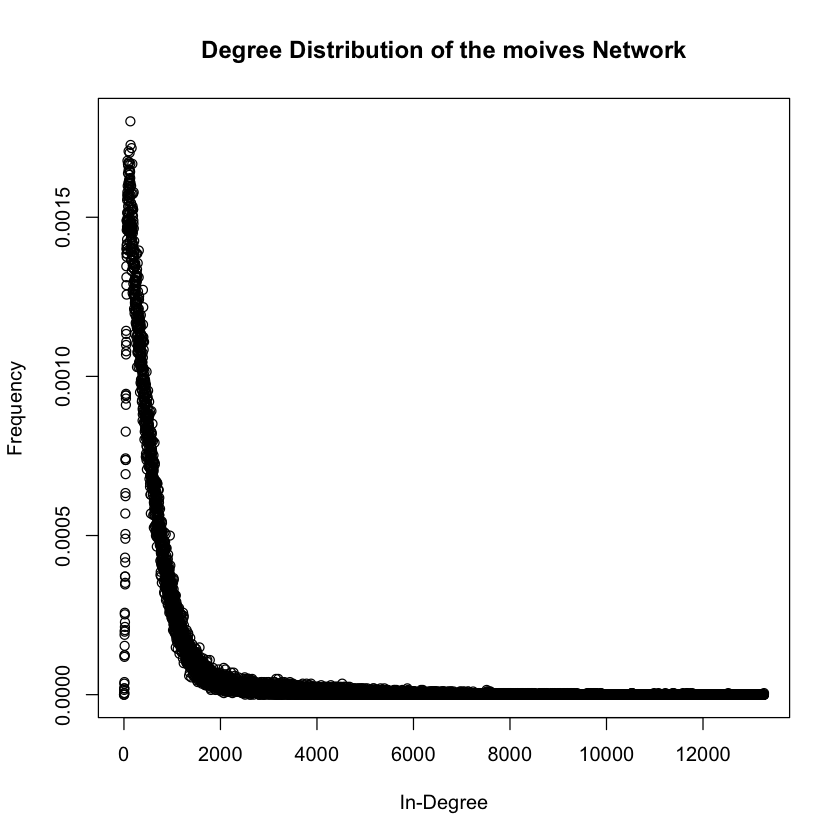
actor2 = set(dic[movie2])

common = len(actor1 & actor2)

total = len(actor1.union(actor2))

return common/total

The distribution is shown below:



From above figure we can find that the distribution is similar with question 2 which means the nodes with small degree has a large frequency. And the distribution has following features:

Mean Degree: 653

Edge number: 65972612

**Question 7.**

In this part, we are required to use the Fast Greedy community detection algorithm to find the communities in the movie network.Then we will plot the distribution of genres of the movies in 10 communities. we randomly choose 10 communities which are 1, 3, 8 , 11, 21, 23, 24, 26, 27, 29. The following is the distribution of the genres of the movies.

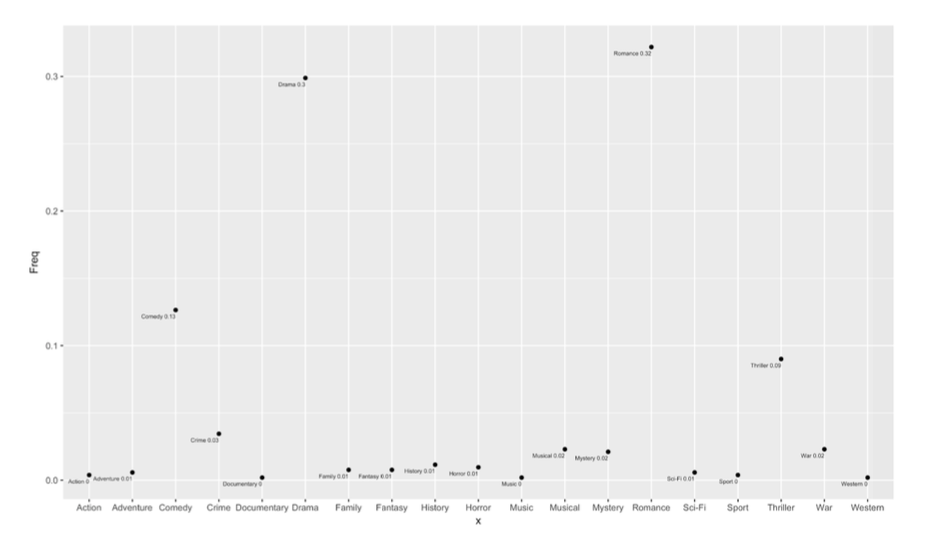


Figure: Community 1

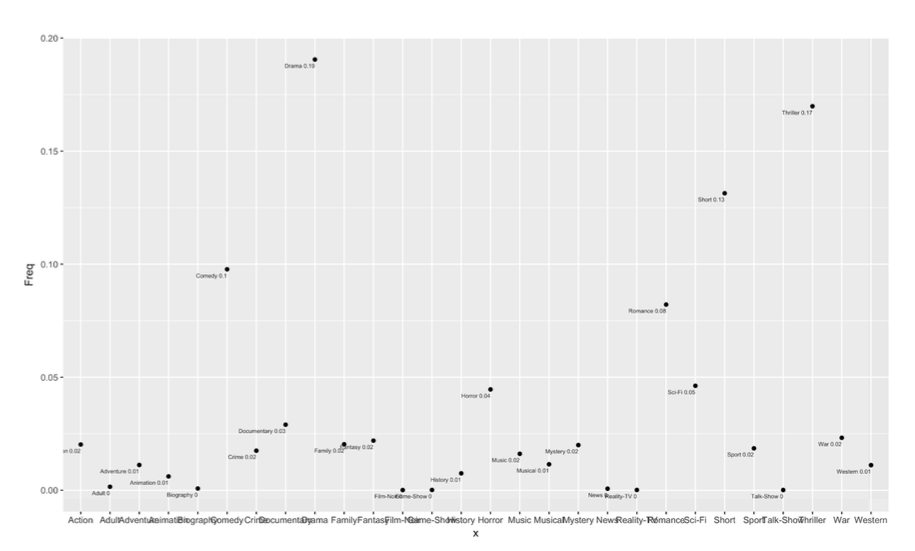


Figure: Community 3

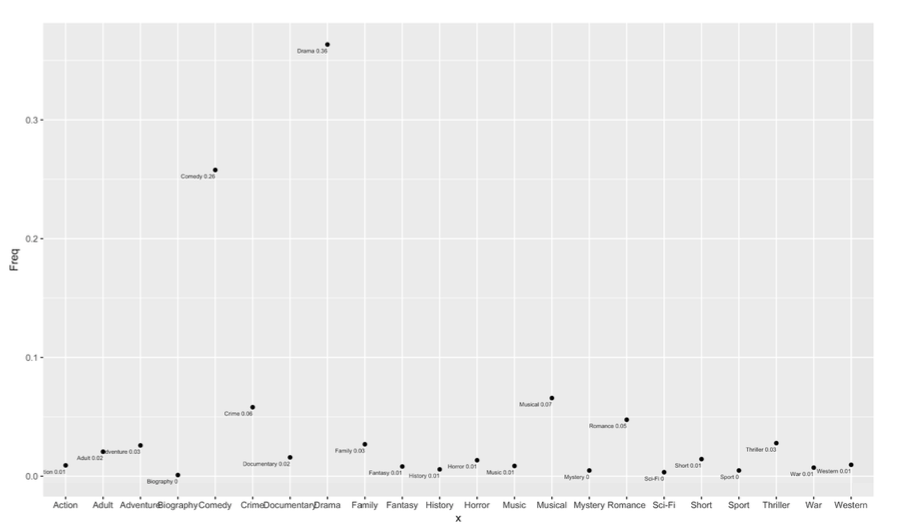


Figure: Community 8

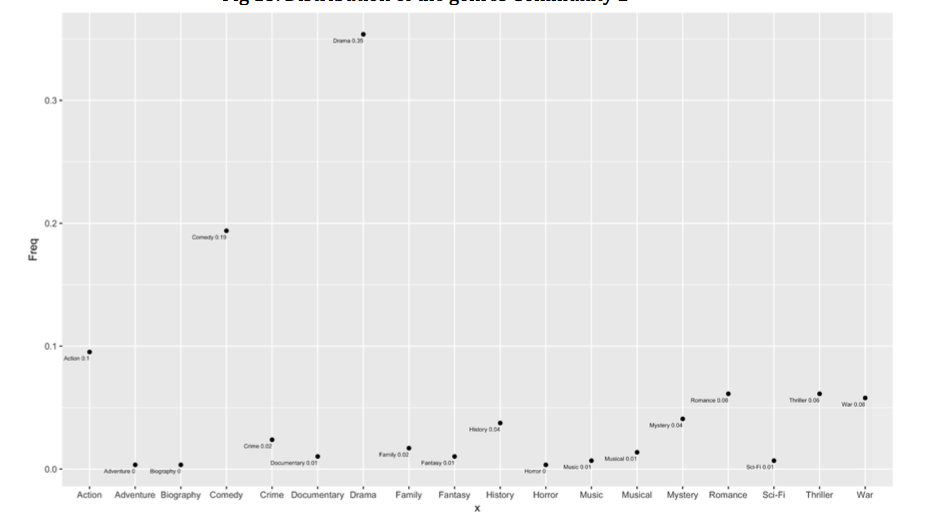


Figure: Community 11

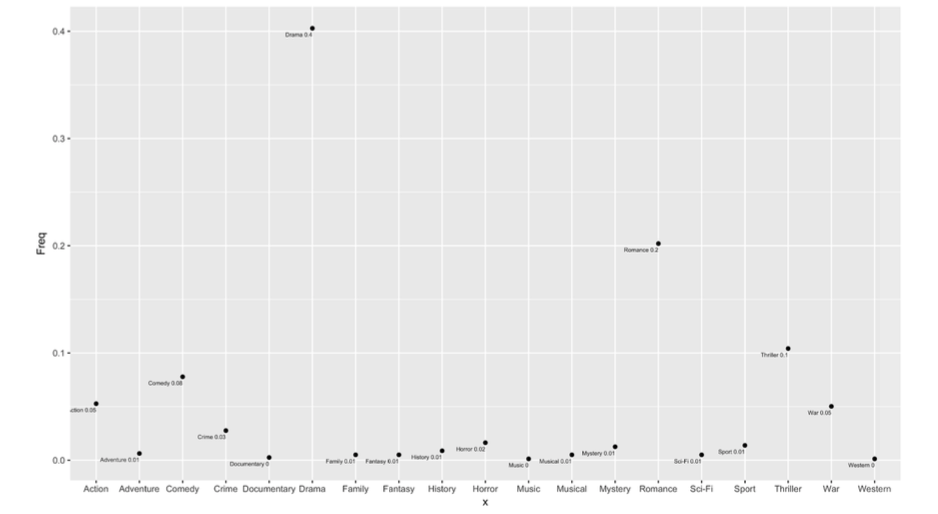


Figure: Community 21

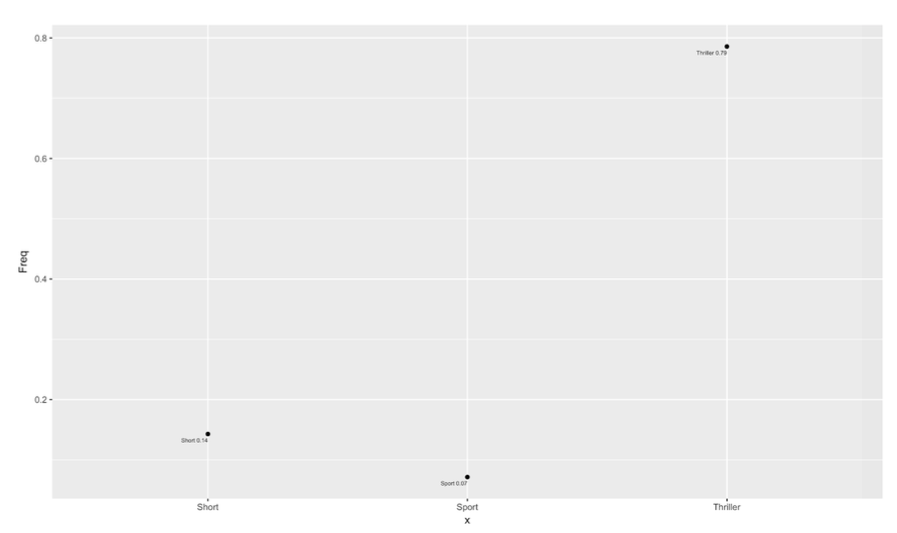


Figure: Community 23

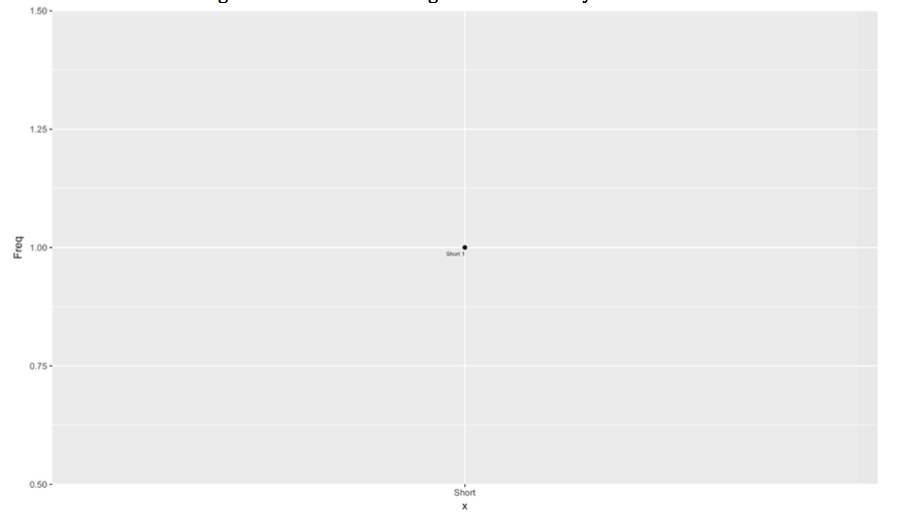


Figure: Community 24

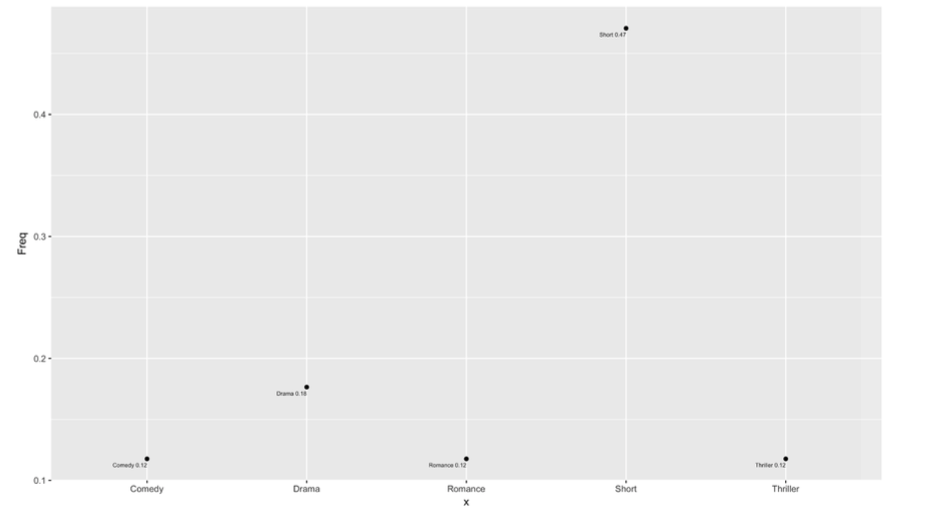


Figure: Community 26

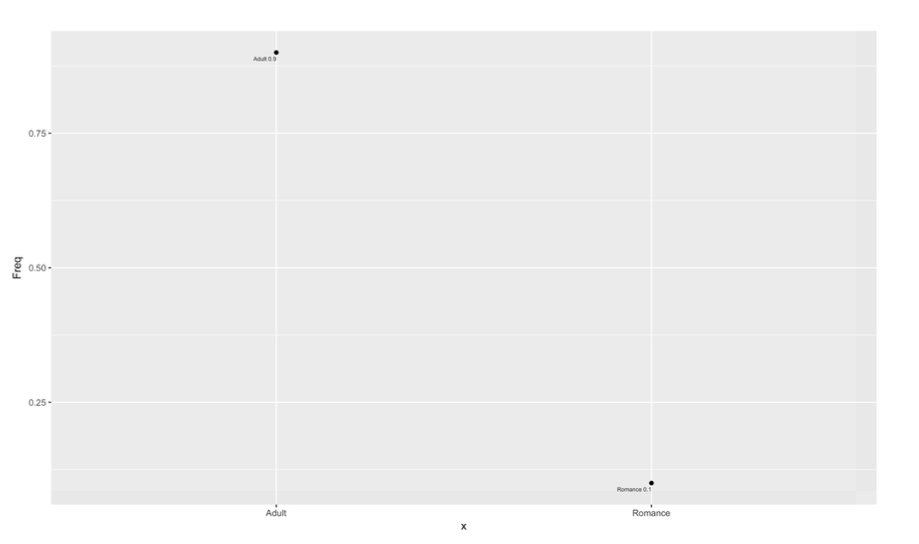


Figure: Community 27

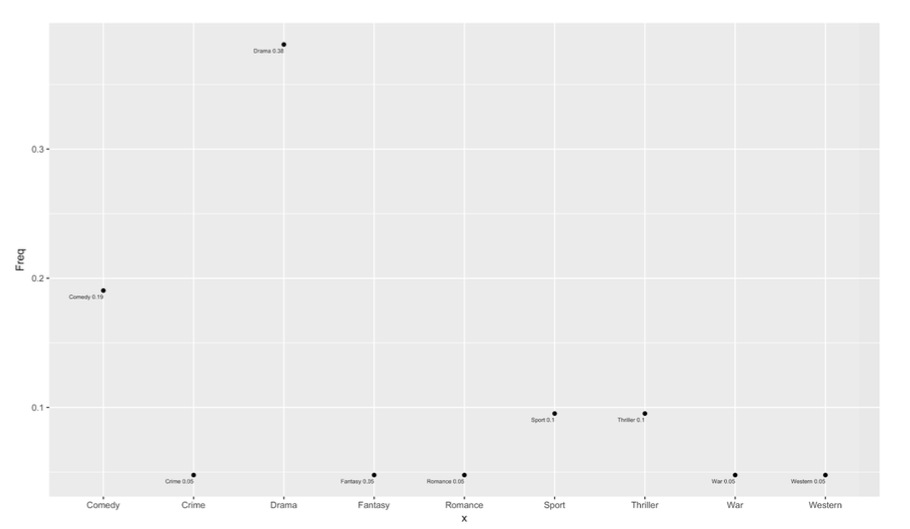


Figure: Community 28

**Question 8.**

a). From the previous section, we can get the most dominant genre in each community based on counting the frequency. The following is our counting result:

|  |  |  |
| --- | --- | --- |
| **Community** | **Genre** | **Frequency** |
| **1** | **Romance** | **166** |
| **2** | **Drama** | **8800** |
| **3** | **Drama** | **3478** |
| **4** | **Drama** | **8900** |
| **5** | **Drama** | **660** |
| **6** | **Drama** | **304** |
| **7** | **Short** | **1889** |
| **8** | **Drama** | **1598** |
| **9** | **Drama** | **688** |
| **10** | **Short** | **1772** |
| **11** | **Short** | **9322** |
| **12** | **Adult** | **1670** |
| **13** | **Drama** | **1544** |
| **14** | **Drama** | **2417** |
| **15** | **Drama** | **122** |
| **16** | **Drama** | **1066** |
| **17** | **Drama** | **757** |
| **18** | **Drama** | **183** |
| **19** | **Drama** | **664** |
| **20** | **Drama** | **2803** |
| **21** | **Short** | **334** |
| **22** | **Drama** | **1487** |
| **23** | **Thriller** | **15** |
| **24** | **Short** | **19** |
| **25** | **Adult** | **12** |
| **26** | **Short** | **8** |
| **27** | **Drama** | **67** |
| **28** | **Family** | **18** |

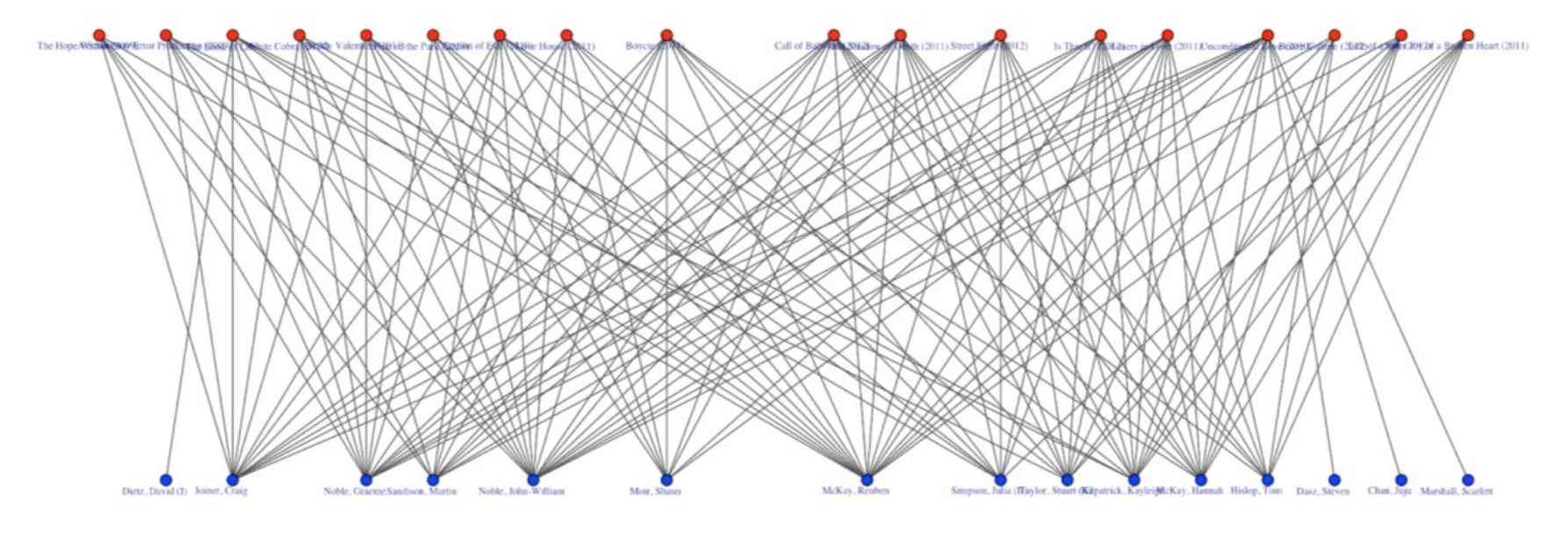
From the above table, we can find the most dominant genre is **Drama**, and the second one is short. According to this result, we can guess that most of the movies are drama because drama is funny. People prefer to watch drama to relax themselves compare with other genre movies.

b). For each communtiy, we use the equation ln(c(i)) \* p(i)/q(i) to calculate a score and assign the score to each genre. Then we can find the new most dominant genre in each community.

|  |  |  |
| --- | --- | --- |
| **Community** | **Genre** | **Frequency** |
| **1** | **Romance** | **53.64** |
| **2** | **Sci-Fi** | **32.56** |
| **3** | **War** | **36.65** |
| **4** | **Comedy** | **37,93** |
| **5** | **Comedy** | **41.07** |
| **6** | **History** | **28.24** |
| **7** | **Western** | **44.62** |
| **8** | **Family** | **41.94** |
| **9** | **Musical** | **61.22** |
| **10** | **Adventure** | **57.47** |
| **11** | **Film-Noir** | **162.74** |
| **12** | **Adult** | **74.65** |
| **13** | **Action** | **87.3** |
| **14** | **Drama** | **24.15** |
| **15** | **Action** | **25,98** |
| **16** | **Adventure** | **78.92** |
| **17** | **Crime** | **42.87** |
| **18** | **Action** | **25.34** |
| **19** | **War** | **67.87** |
| **20** | **Romance** | **53.43** |
| **21** | **Short** | **33.19** |
| **22** | **Drama** | **13.93** |
| **23** | **Thriller** | **35.93** |
| **24** | **Romance** | **36.34** |
| **25** | **Adult** | **31.34** |
| **26** | **Romance** | **2.63** |
| **27** | **Short** | **9.86** |
| **28** | **Family** | **195.87** |

From the above table, we can see that there are lots of different genres. It is totally different compare with the 8(a). In 8(a), there are only 5 different genres, but in 8(b), there are like 20 kinds of genres. Also, it the genre which appears the most dominant in 8(a) maybe will also dominant in 8(b).

c). In this part, we will find a community whose movie has the size between 10 and 20. Then determine all the actors who acted in these movies and plot the bipartite graph. we choose the community 24 which contains 16 movies. The all actors are Craig, Reuben, Sandison, Taylor, Moir, Shaun, Julia, Chan, Mckay, Tom, Kayleigh, Hannah, Noble, Graeme, John.The following is our graph.



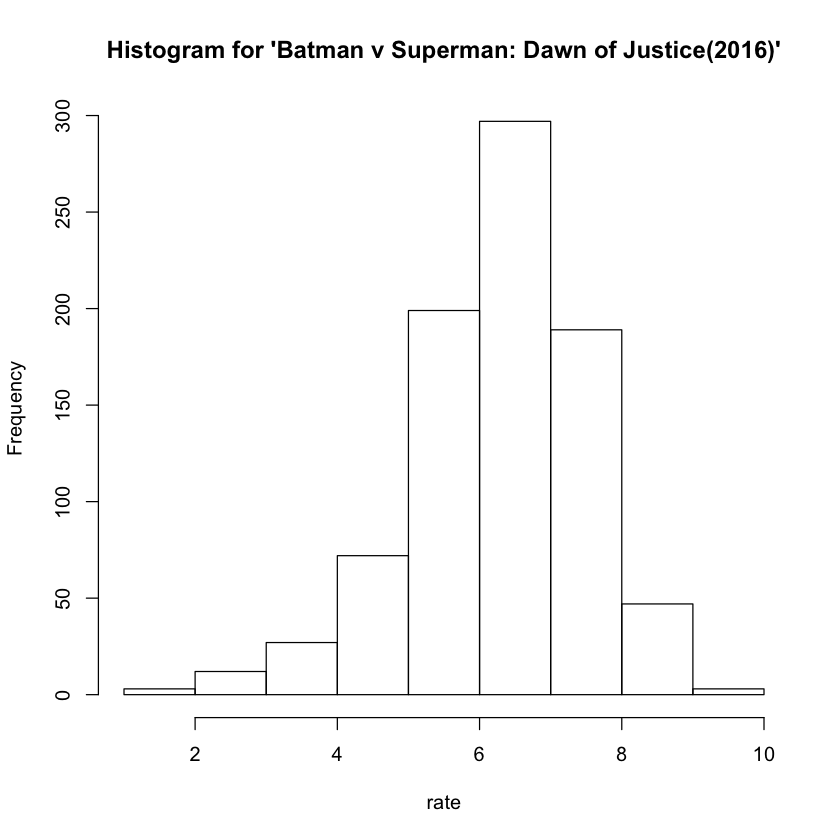
From the above graph, we can find the three most important actors are Reuben with degree 16, and Noble with degree 15 and Graeme with degree 14. Since these three actors almost act all the movies in the community, they play an important role in this community. They make this community more consistent and tight the movie together. In 8(a), the most dominant genre is short. In 8(b), the most dominant genre is romance. We can see the most dominant actors in this community have ever acted short and romance movies.

**Question 9.**

In this part,we are required to extract each movie’s neighbors and plot the distribution of the available ratings of the movies in the neighborhood. The average rating of movies in the neighborhood and its corresponding distribution are shown as following:

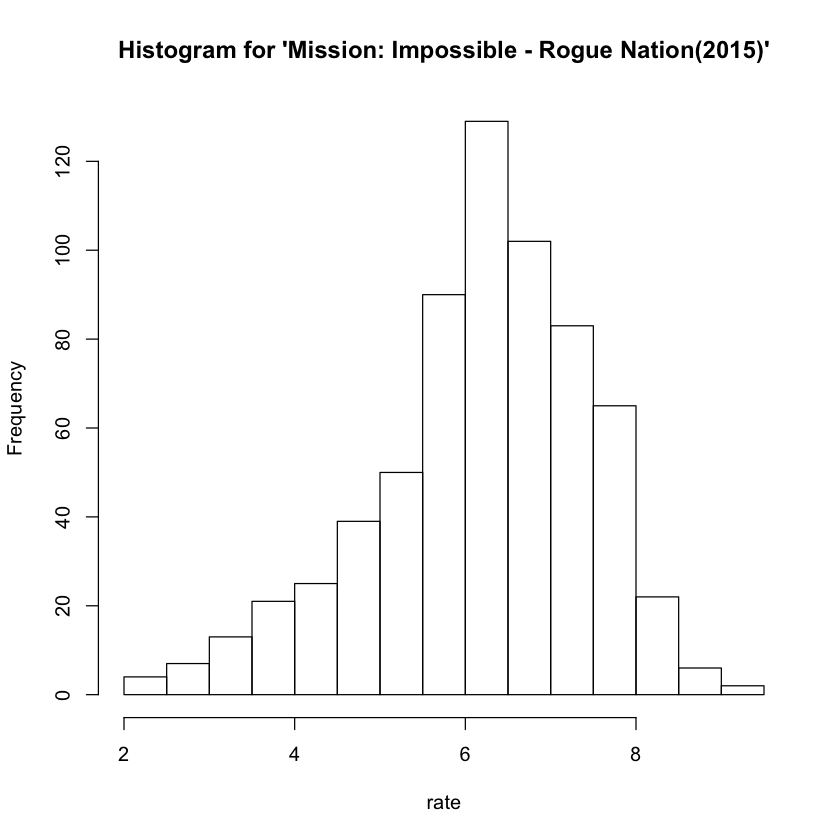
* Batman v Superman: Dawn of Justice (2016); Rating: 6.6

Average rating of the movies in the neighborhood: 6.326737



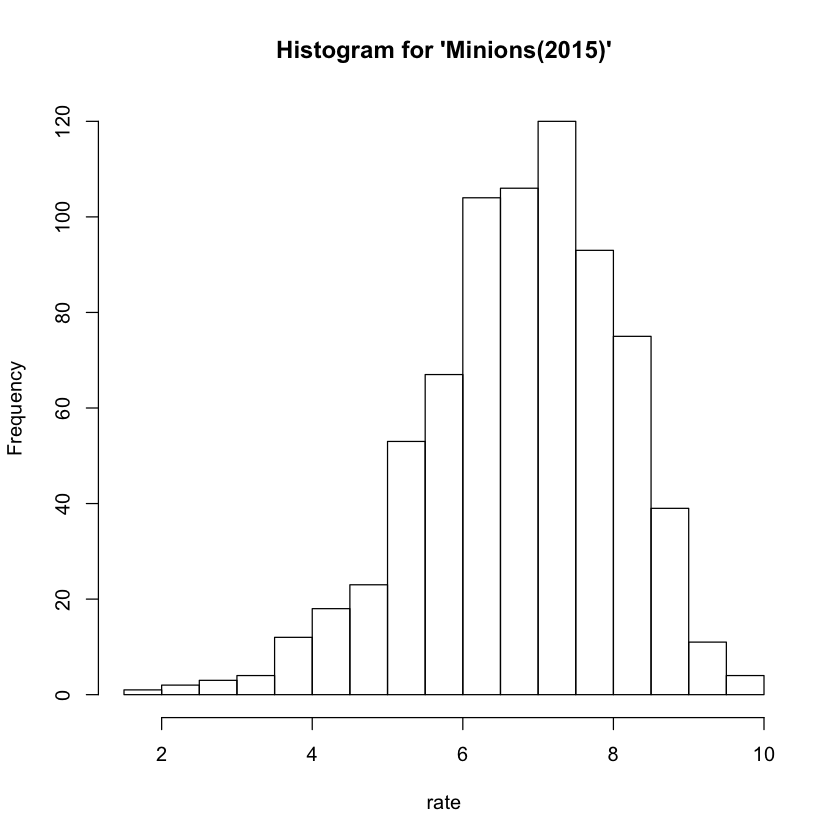
* Mission: Impossible - Rogue Nation (2015); Rating: 7.4

Average rating of the movies in the neighborhood: 6.234195



* Minions (2015); Rating: 6.4

Average rating of the movies in the neighborhood: 6.82966



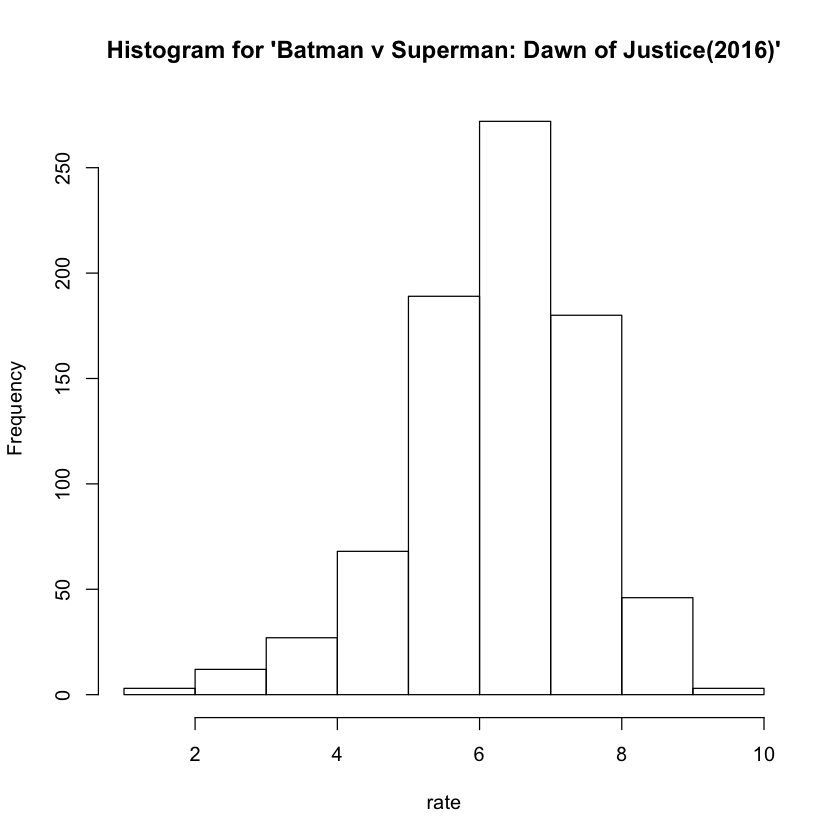
From the result shown above, for the movie ‘Batman v Superman: Dawn of Justice (2016)’, we can see that average rating of the movie in the neighborhood are similar to its rating, they have only 0.274 in variance; for the movie ‘Mission: Impossible - Rogue Nation (2015)’, we can see that the average rating of the movie in the neighbor are much smaller to its rating, they have 1.17 in variance; for the movie ‘Minions (2015)’, we can see that the average rating of the movie in the neighbor are similar to its rating, they have 0.429 in variance.

**Question 10.**

In this part, we are required to repeat question 10, but now restrict the neighborhood to consist of movies from the same community, the results are shown as following:

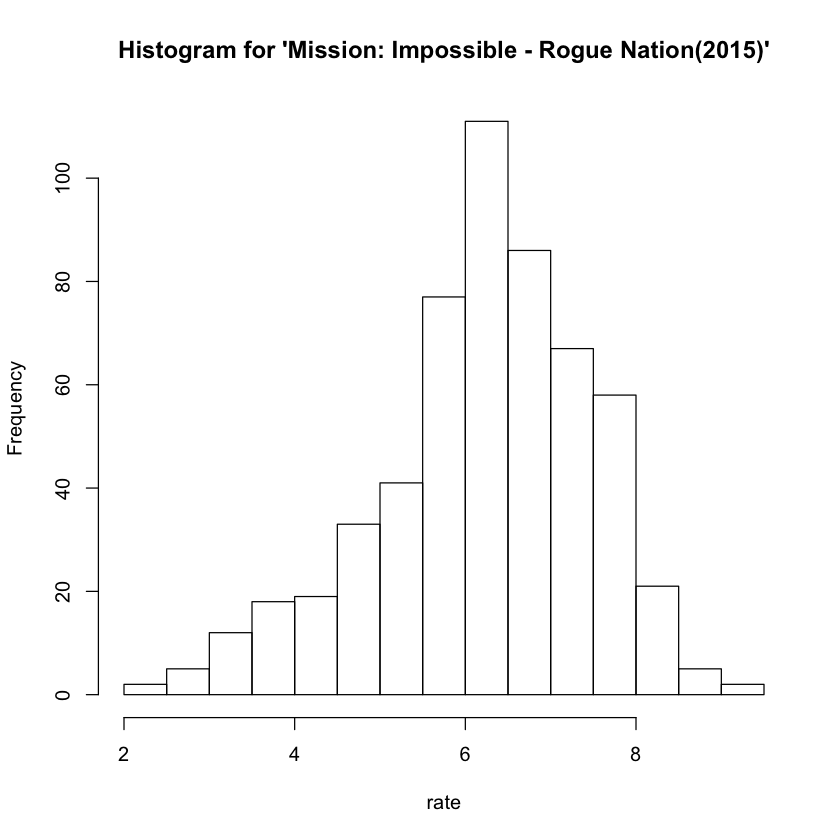
* Batman v Superman: Dawn of Justice (2016); Rating: 6.6

Average rating of the movies in the neighborhood: 6.32075



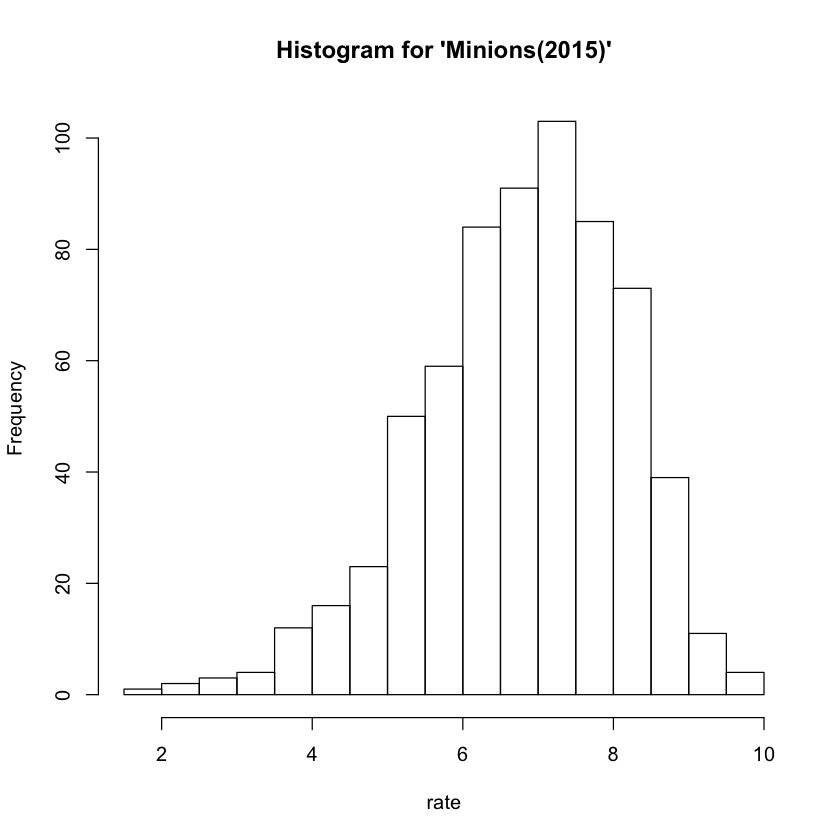
* Mission: Impossible - Rogue Nation (2015); Rating: 7.4

Average rating of the movies in the neighborhood: 6.267504



* Minions (2015); Rating: 6.4

Average rating of the movies in the neighborhood: 6.846515



From the result shown above, we can see that for both of the movies ‘Minions (2015) ‘ and ‘Batman v Superman: Dawn of Justice (2016)’, the average rating of the movies in the neighborhood are similar to the result we derived in question 9. For the movie ‘ Mission: Impossible - Rogue Nation (2015)’, the average rating of the movie in the neighborhood are better match than the result we derived in question 9, it's more close to the movie’s rating.

**Question 11.**

In this part, we are required to extract each movie’s top 5 neighbors based on the edge weight and also report the community membership of the top 5 neighbors. The results are shown as following:

* Batman v Superman: Dawn of Justice (2016); Rating: 6.6

Top 5 neighbors and corresponding community membership:

Eloise (2015) 1

The Justice League Part One (2017) 1

Into the Storm (2014) 1

Love and Honor (2013) 1

Man of Steel (2013) 1

* Mission: Impossible - Rogue Nation (2015); Rating: 7.4

Top 5 neighbors and corresponding community membership:

Fan (2015) 5

Phantom (2015) 5

Breaking the Bank (2014) 1

Suffragette (2015) 1

Now You See Me: The Second Act (2016) 1

* Minions (2015); Rating: 6.4

Top 5 neighbors and corresponding community membership:

The Lorax (2012) 1

Inside Out (2015) 1

Despicable Me 2 (2013) 1

Up (2009) 1

Surf's Up (2007) 1

**Question 12.**

Here I use linear regression to predict ratings of three movies through the top 5 page ranks of the actors in each movie. The RMSE is computed and equal to 1.243792. Then, based on the regression I got, I predict the rating of those three movies:

Batman v Superman: Dawn of Justice (2016)

real rating: NA

Predicted rating: 6.303812

Mission: Impossible - Rogue Nation (2015)

real rating: NA

Predicted rating: 6.124734

Minions (2015)

real rating: NA

Predicted rating: 6.081247

We can find that these three movies do not have a rating score at beginning. After prediction, Batman v Superman is rated at 6.303812 which is very close to 6.6 (what we got from the statement). Mission: Impossible is rated at 6.124734. This is a little away from 7.4. Finally, Minions is rated at 6.081247 which is close to 6.4

**Question 13**

In this problem, we create a bipartite graph following the procedure described in the statement. Then give the weight for each actor. I made the actor weight to be the average of top 5 ratings of the actor’s movie. I chose top 10 actor weights to be features of each movie and still use linear regression to train the model. I got RMSE equal to 1.186899 this time. The predicted rating is shown below:

Batman v Superman: Dawn of Justice (2016)

real rating: 6.6

Predicted rating: 6.529988

Mission: Impossible - Rogue Nation (2015)

real rating: 7.4

Predicted rating: 6.498286

Minions (2015)

real rating: 6.4

Predicted rating: 7.010742

This time I add real rating of these three movies. We can see that the rating of Batman v Superman is 6.529988 which is really close to 6.6. For Mission: Impossible, it is still not that correct, compare to the real rating, but the difference between real rating and predicted rating is closer than before. However, we see the accuracy of the third movie rating which is Minions is decrease. In conclusion, I think the prediction accuracy is better than before in general. We find that RMSE is decreased. Besides, although the error still exists, its accuracy is better than what we got in question 12.