**STAT6289 Network HW4**

**Chenrui Xu**

**I. Background**

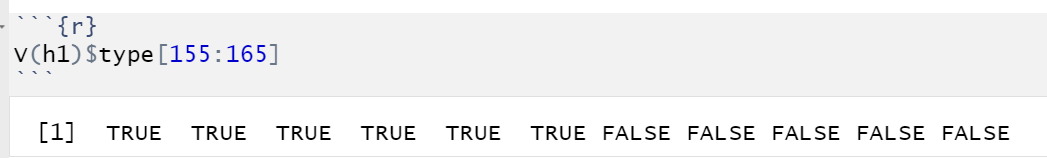
The data is about the Hollywood movies and actors’ data. There are 1365 nodes (160 movies and 1205 actors) and 1600 edges in the network data. As the data has two types of nodes, one is actor node and the other one is group nodes, we will use affiliation network knowledge to analysis it.

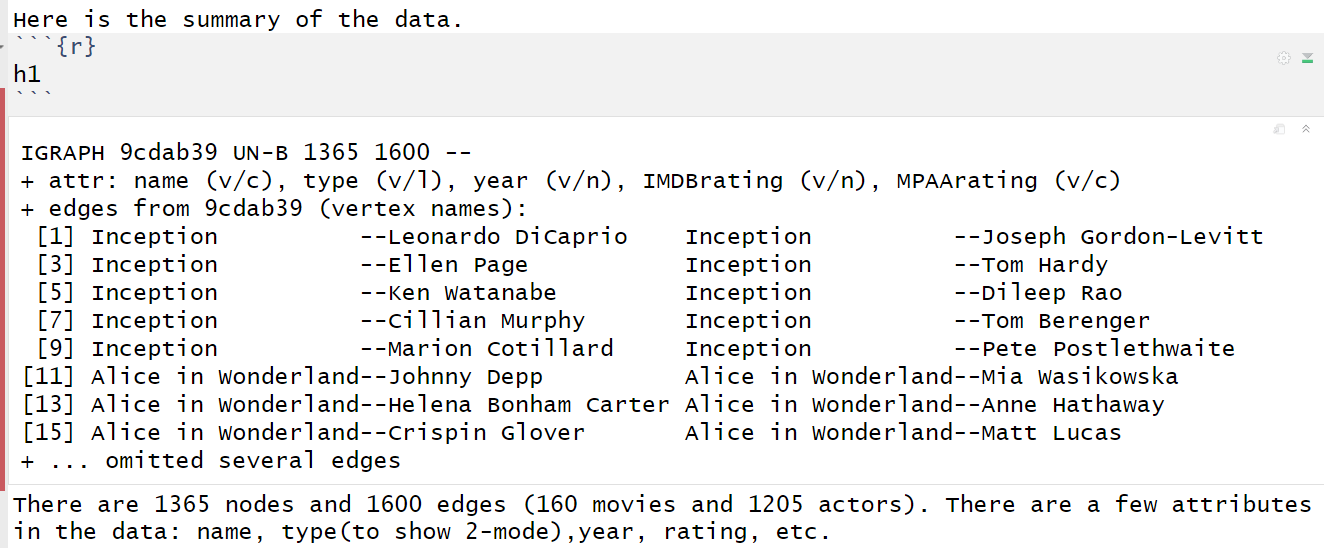
Affiliation network: It is a network where members are affiliated with one another based on co-membership in a group, or co-participation in some type of event. In this homework, we will use two-mode network and Bipartite graphs to analysis network.

Also, in all these three parts, we will use kinds of methods to extract information from the network to show plots via different angles (movie and actor subgraph).

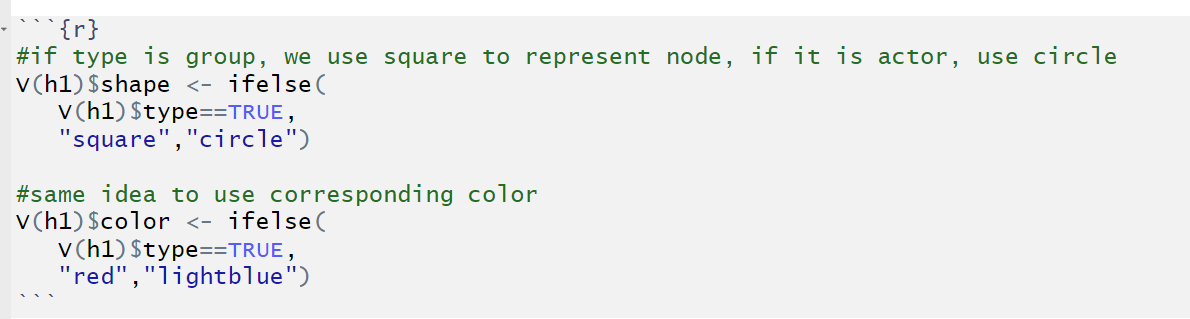
**II. Part I**

The vertex has an attribution called name, so we can use it to see the name of movies and actors. Also, we can check the type of the nodes as the network is a two-mode model. If the node belongs to actor, the type should be false and the group corresponding type is true. What we can see from the analysis is that the first 160 nodes have the true type and the rest 1205 have false type.



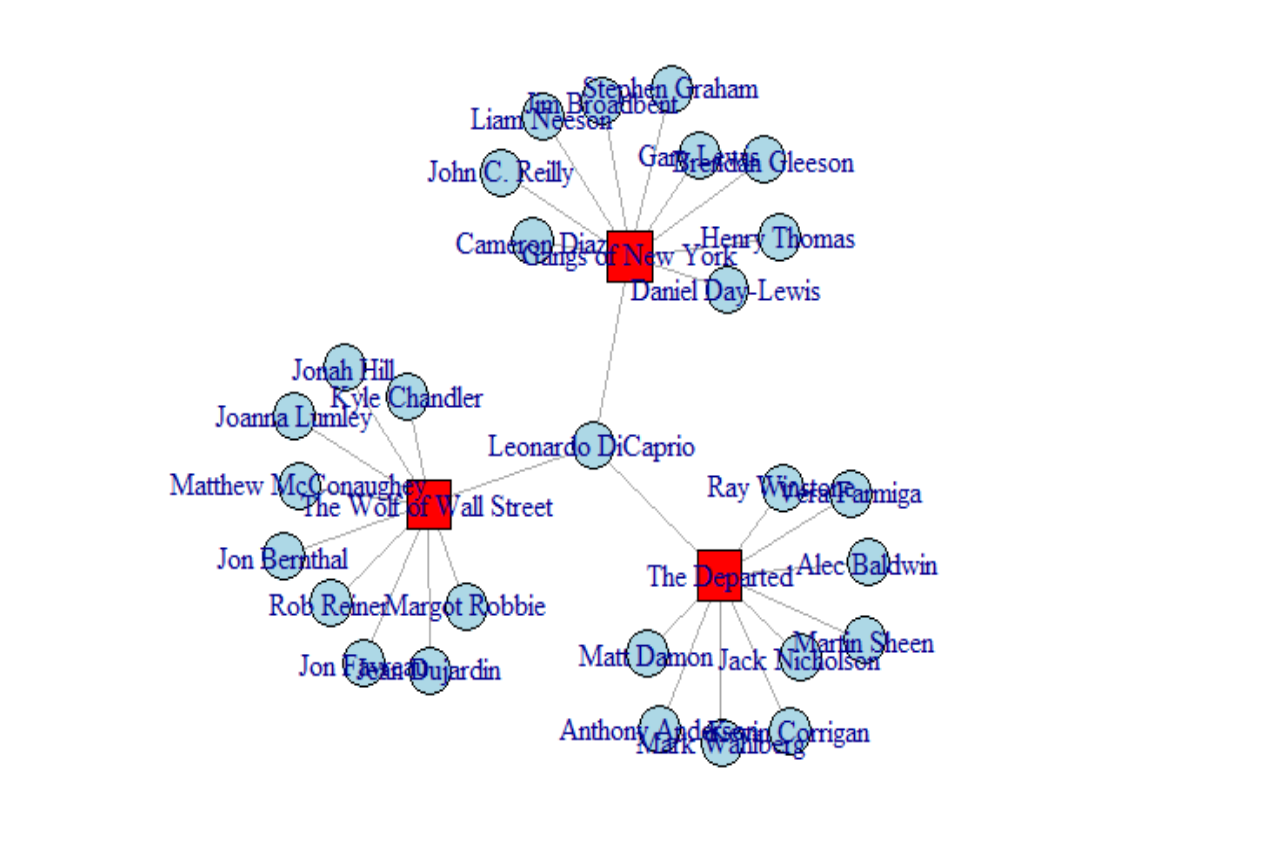


According to the summary, we can know that the name is the vertex of character, type is true or false, the year is numeric, two ratings one is numeric and the other one is character (R/PG-13).

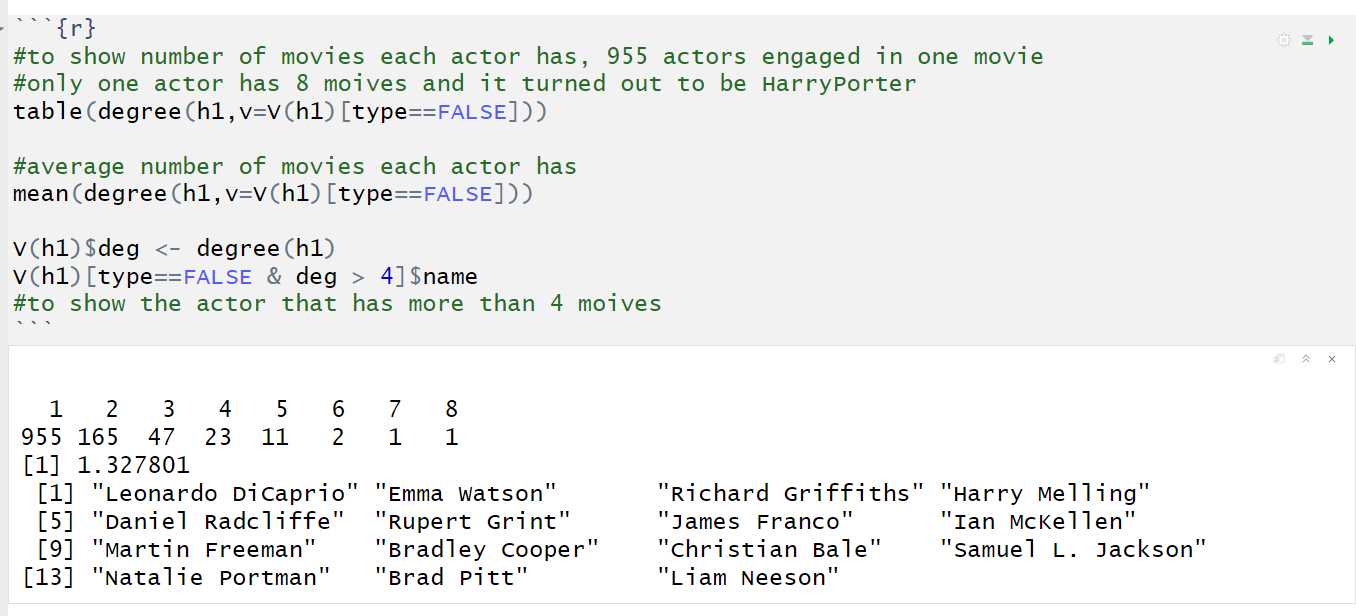


We can use *if-else* function to attribute shapes and colors to nodes according to the type of nodes.

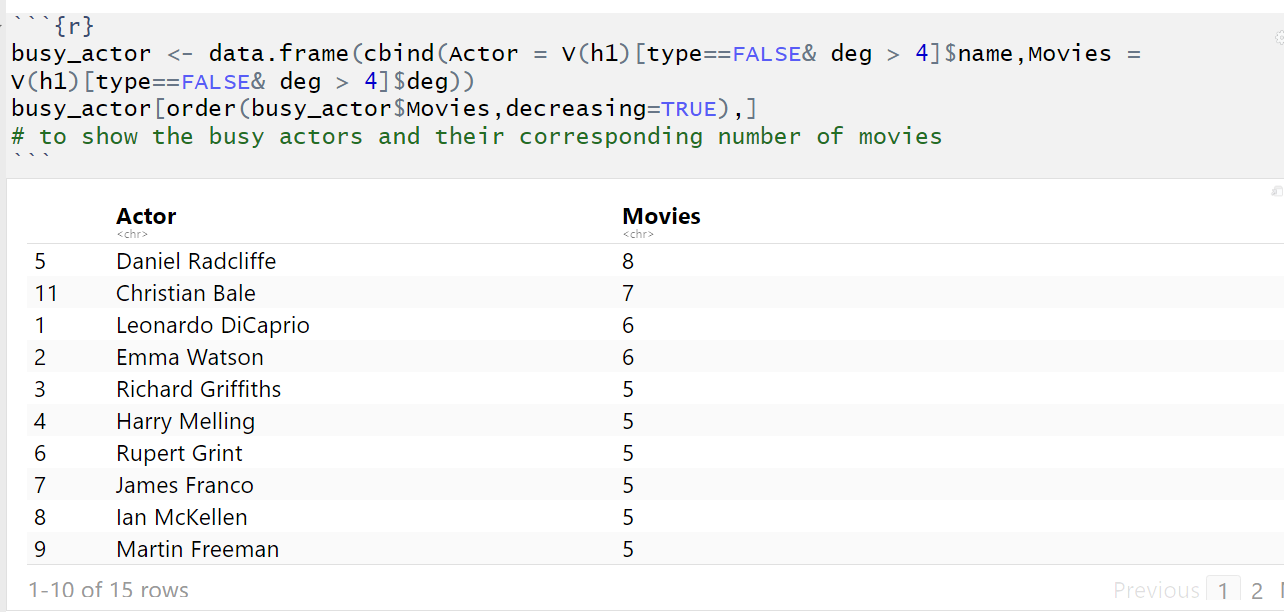
The next step, we can draw some specific network graph. By using filtering character, we can find three movies’ nodes and their actors nodes.



The next step is to figure out how many actors in each movie.

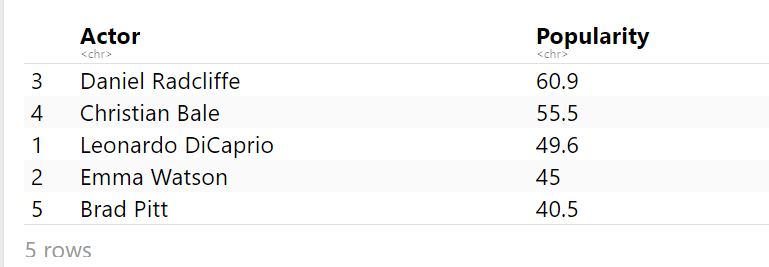


It turns out that 955 actors only act one movie and the max number of movies one single actor has is 8. The average number is 1.328. And we can filter those actors whose degrees are more than four. Those actors are familiar with us.

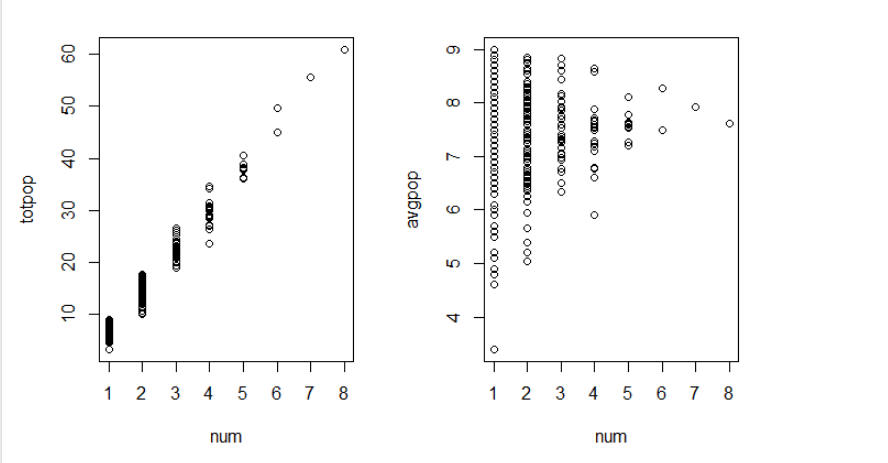


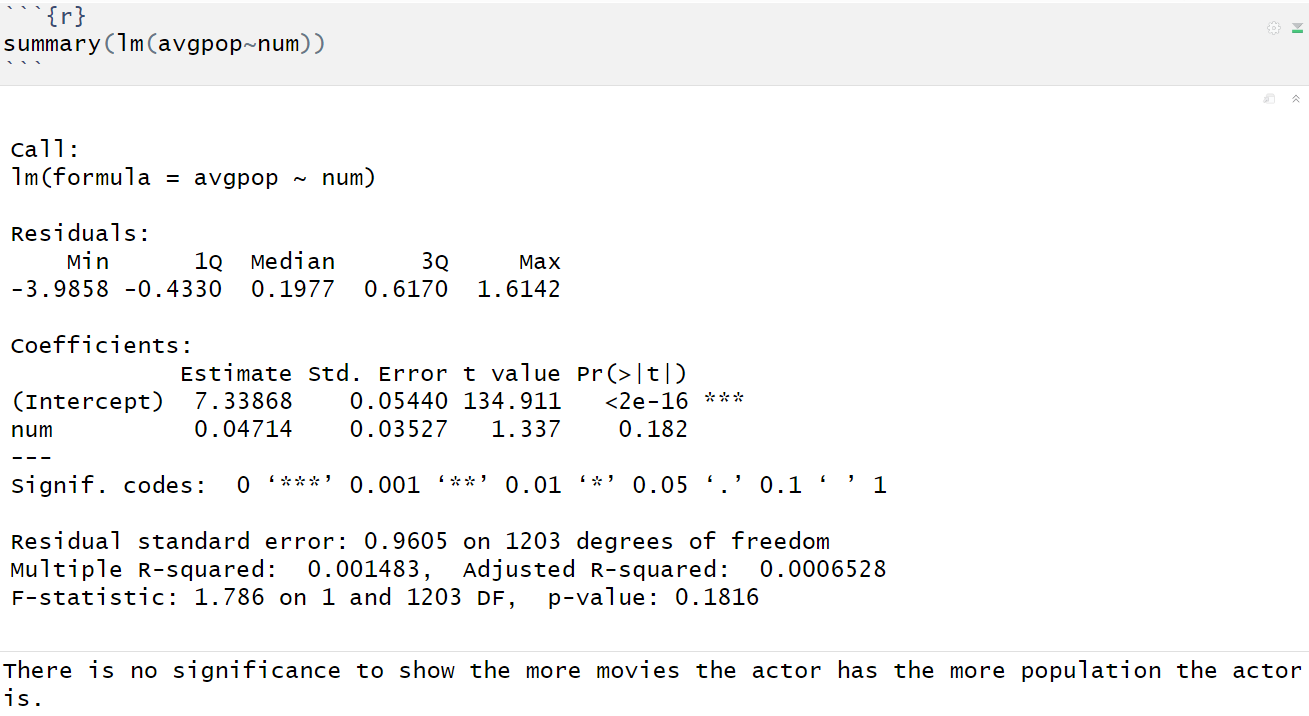
Here is the list that showed how many movies one single actor has. It showed that the HorryPoter series’ actor appeared mostly.

For the next step, we can analysis how popular the actors are and the relationship between busyness and popular level.

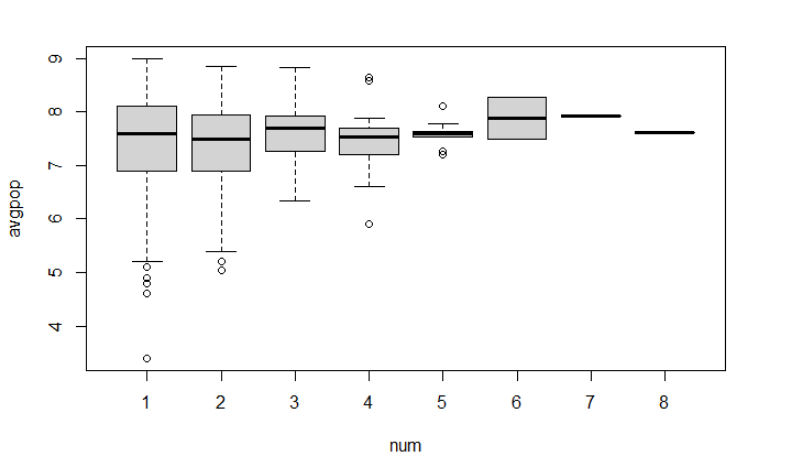


The table showed the total score is similar which the last table.



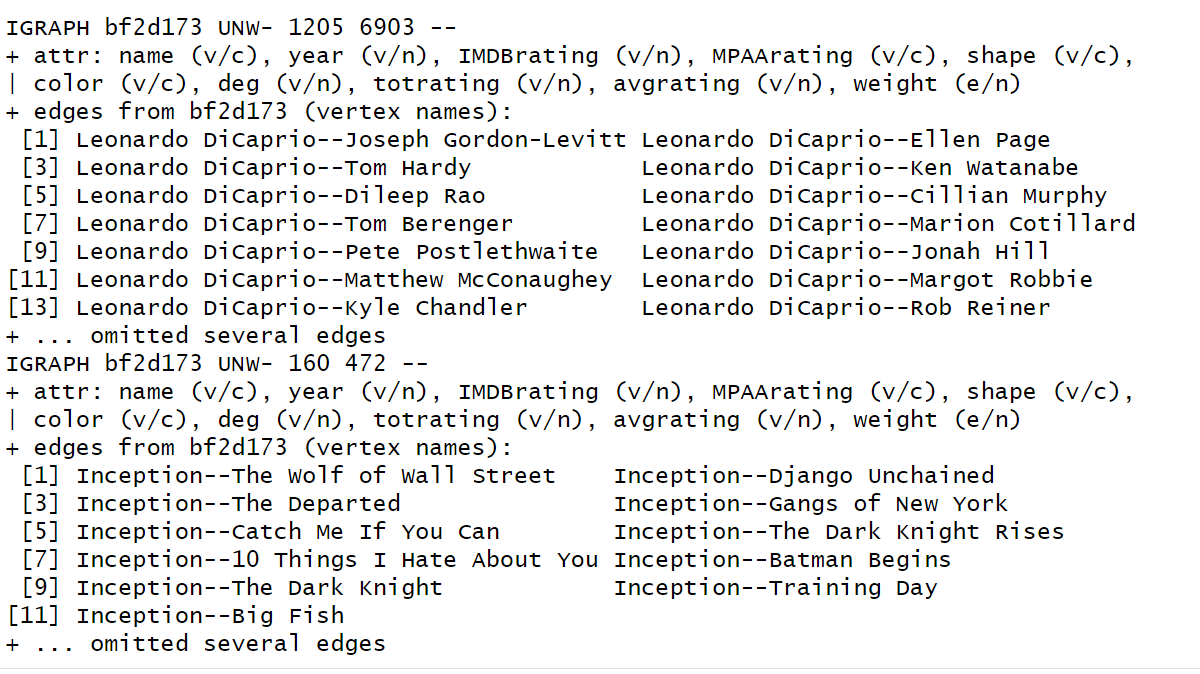


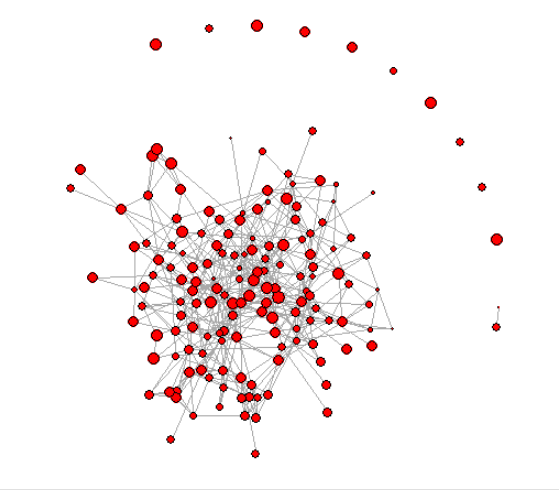
From the results above, we can know that the parameter of number is not significant. So we can get the result the number of movies has no relationship with popular.



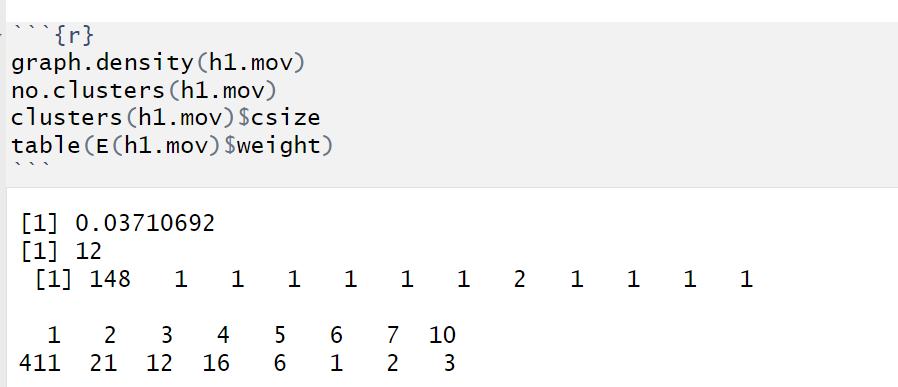
And from the boxplot, we can see that whatever the number is, the rate of popular is always around 7.5.

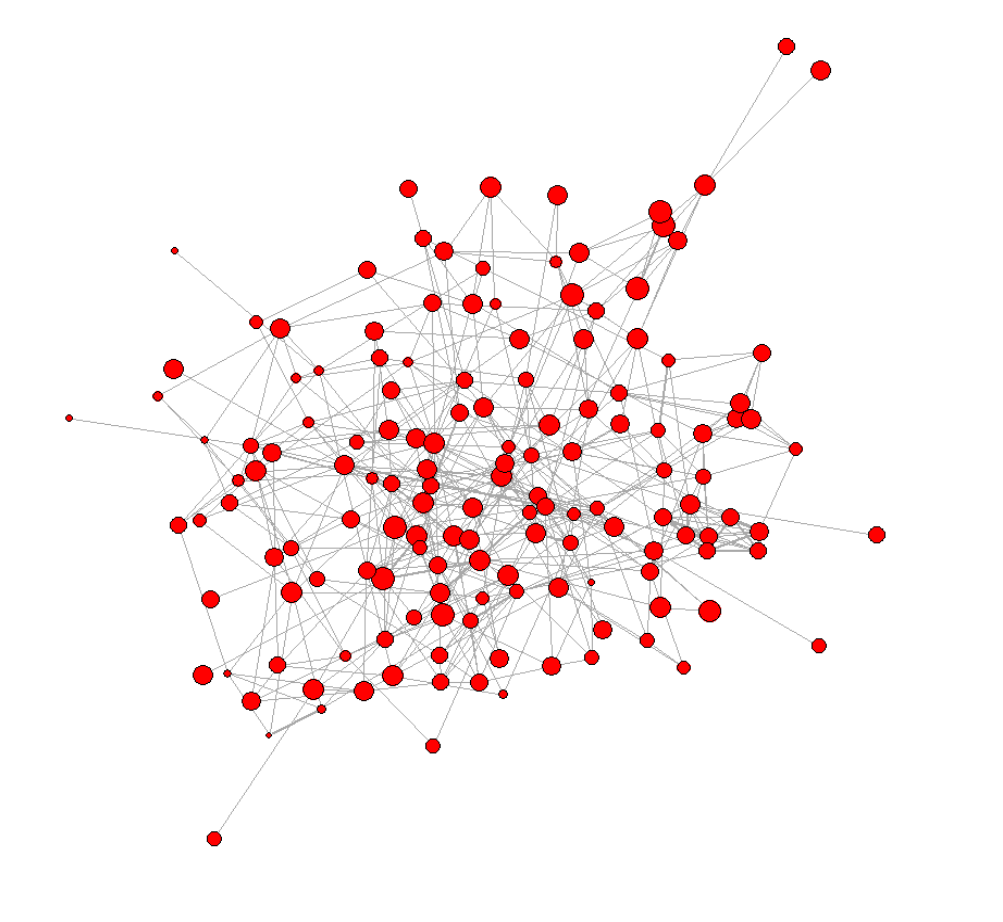
We will do the projection. We will divide the network into actor network and movie network. Here are the summaries.



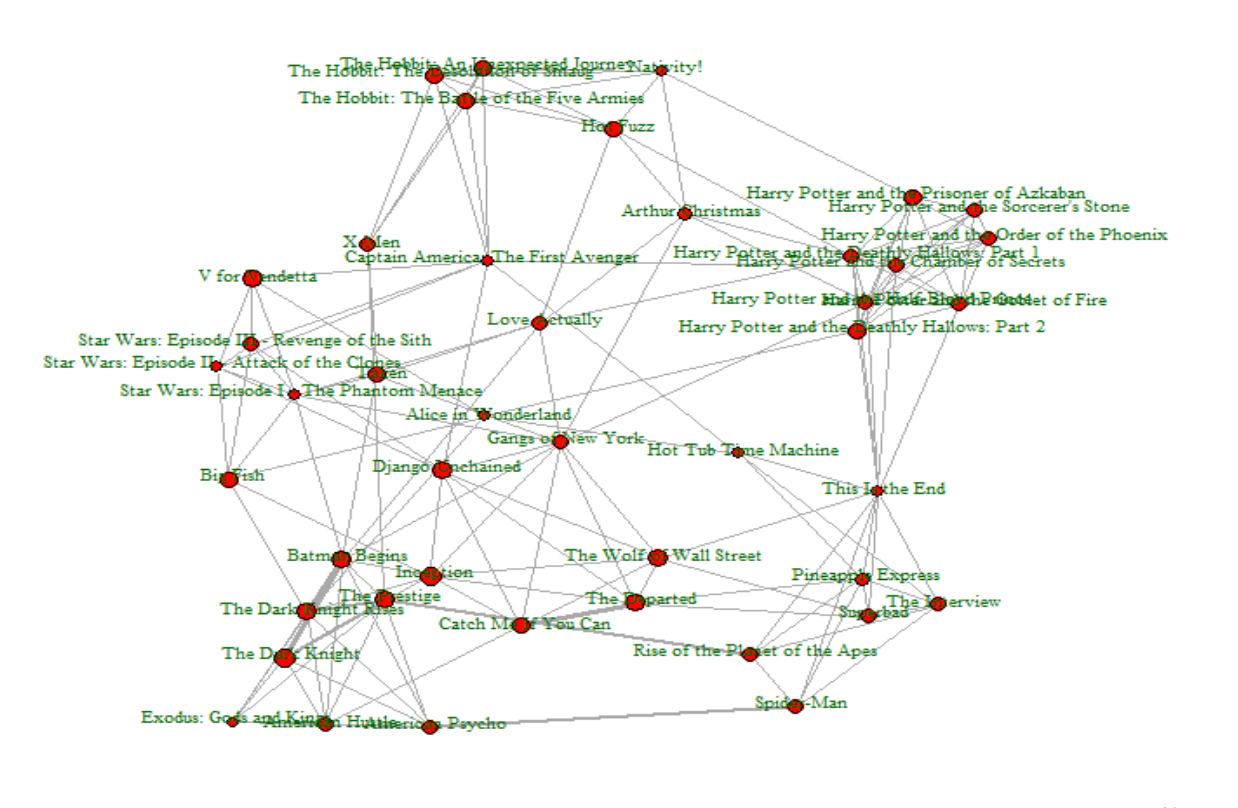


When analyzing the movie network, there are some results shown below. There are 12 clusters in the network. The largest membership size is 148 and the rest of clusters are 1 or 2.





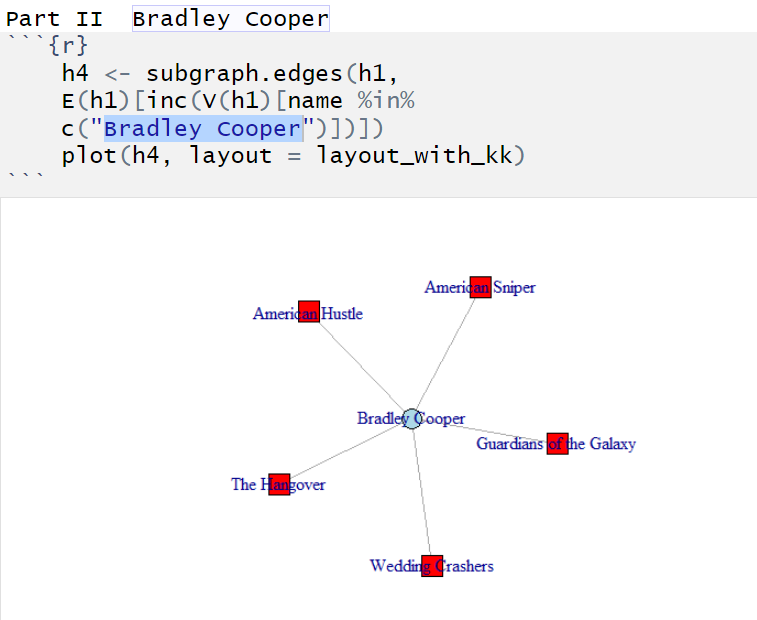
Above is the largest cluster for the movie network. Also, we calculated the coreness of the network. We visualized the plot to show the nodes that coreness larger than four.



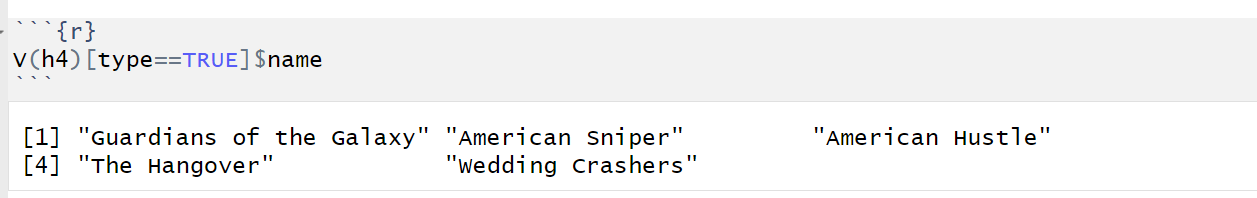
We can see that in this cluster, some movies are quite reasonable related. For example, there are Hobbit (The lord of the ring) series, Harry Potter series, Star Wars series and some movies shared the same famous celebrities.

**III. Part II**

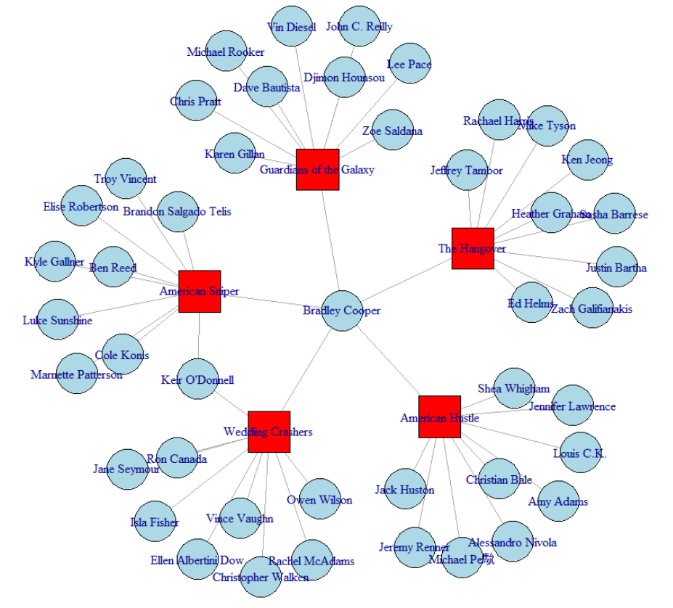
In this part, we need to extract the actor Bradley Cooper’s movies’ related actors.



The graph showed the actor’s related movies.

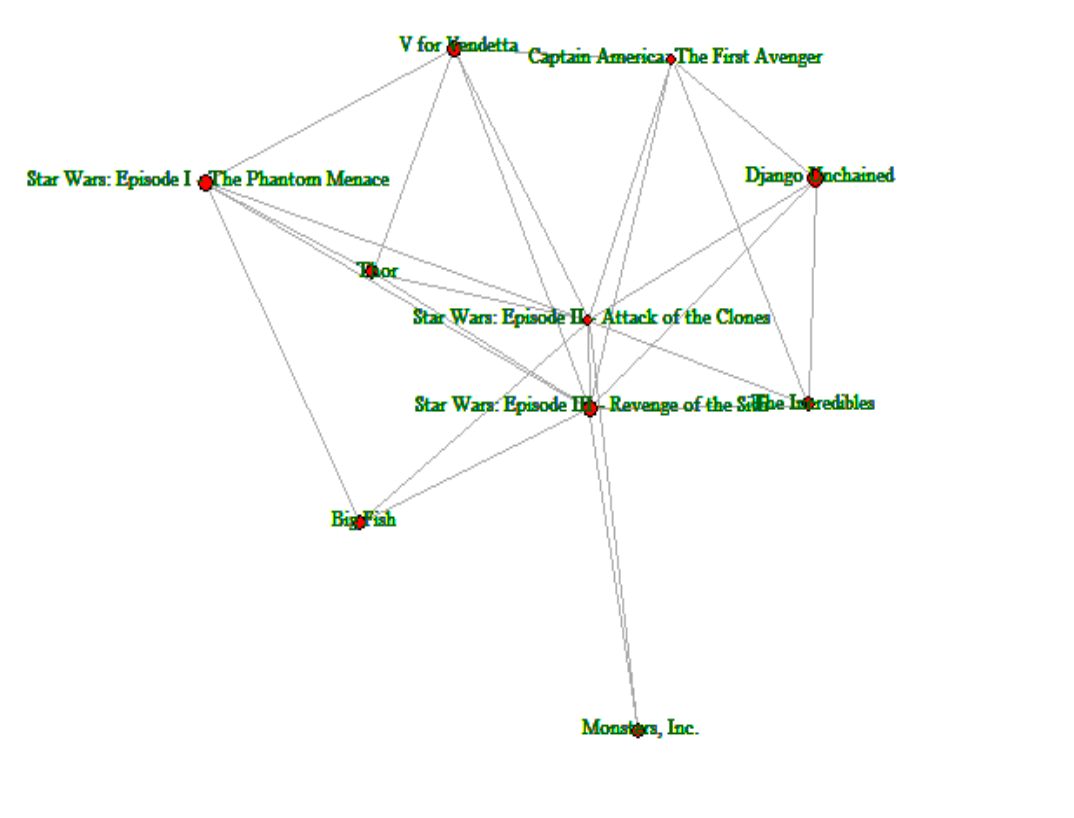


Here are the list of movies and we will apply the list to show the related actors.



**IV. Part III**

For this part, we need to draw the plot to show how many movies are related to the movie Star Wars III. According to the function induced.subgraph, the vids is the condition to select nodes. At first, we select the movies are neighbor to Star Wars III, but not include SW III itself. So, the network plot only showed nine nodes. Therefore, I made a combination of the single node SW III and neighbors. So, the plot is shown below:



**V. Appendix**

---

title: "Network\_HW4"

author: "Chenrui Xu"

date: "2021/2/28"

output: html\_document

---

```{r}

library(UserNetR)

library(statnet)

library(igraph)

```

```{r}

data(hwd)

h1=hwd

```

```{r}

V(h1)$name[1:10]

```

```{r}

V(h1)$name[155:165]

```

```{r}

V(h1)$type[1:10]

```

```{r}

V(h1)$type[155:165]

```

So we can get that the first 160 index are the movies and from 161, the rest of them are actors.

Here is the summary of the data.

```{r}

h1

```

There are 1365 nodes and 1600 edges (160 movies and 1205 actors). There are a few attributes in the data: name, type(to show 2-mode),year, rating, etc.

```{r}

#if type is group, we use square to represent node, if it is actor, use circle

V(h1)$shape <- ifelse(

V(h1)$type==TRUE,

"square","circle")

#same idea to use corresponding color

V(h1)$color <- ifelse(

V(h1)$type==TRUE,

"red","lightblue")

```

```{r}

#create a subgraph that filter three movies to analysis

h2 <- subgraph.edges(h1,

E(h1)[inc(V(h1)[name %in%

c("The Wolf of Wall Street",

"Gangs of New York",

"The Departed")])])

plot(h2, layout = layout\_with\_kk)

```

```{r}

#to show number of movies each actor has, 955 actors engaged in one movie

#only one actor has 8 moives and it turned out to be HarryPorter

table(degree(h1,v=V(h1)[type==FALSE]))

#average number of movies each actor has

mean(degree(h1,v=V(h1)[type==FALSE]))

V(h1)$deg <- degree(h1)

V(h1)[type==FALSE & deg > 4]$name

#to show the actor that has more than 4 moives

```

```{r}

busy\_actor <- data.frame(cbind(Actor = V(h1)[type==FALSE& deg > 4]$name,Movies = V(h1)[type==FALSE& deg > 4]$deg))

busy\_actor[order(busy\_actor$Movies,decreasing=TRUE),]

# to show the busy actors and their corresponding number of movies

```

```{r}

for (i in 161:1365) {

V(h1)[i]$totrating <- sum(V(h1)[nei(i)]$IMDBrating)

}

for (i in 161:1365) {V(h1)[i]$avgrating <- mean(V(h1)[nei(i)]$IMDBrating)

}

```

```{r}

V(h1)[161:171]$totrating

```

```{r}

V(h1)[161:171]$avgrating

```

```{r}

pop\_actor <- data.frame(cbind(Actor = V(h1)[type==FALSE & totrating > 40]$name,Popularity =V(h1)[type==FALSE &totrating > 40]$totrating))

pop\_actor[order(pop\_actor$Popularity,decreasing=TRUE),]

```

```{r}

num <- V(h1)[type==FALSE]$deg

avgpop <- V(h1)[type==FALSE]$avgrating

totpop <- V(h1)[type==FALSE]$totrating

op <- par(mfrow=c(1,2))

plot(num,totpop)

plot(num,avgpop)

par(op)

```

```{r}

summary(lm(avgpop~num))

```

There is no significance to show the more movies the actor has the more population the actor is.

The average rating of numbers of movies are around 7.5

```{r}

boxplot(avgpop~num)

```

Projection

```{r}

h1.pr <- bipartite.projection(h1)

h1.act <- h1.pr$proj1

h1.mov <- h1.pr$proj2

h1.act

h1.mov

```

```{r}

op <- par(mar = rep(0, 4))

plot(h1.mov,vertex.color="red",

vertex.shape="circle",

vertex.size=(V(h1.mov)$IMDBrating)-3,

vertex.label=NA)

par(op)

```

```{r}

graph.density(h1.mov)

no.clusters(h1.mov)

clusters(h1.mov)$csize

table(E(h1.mov)$weight)

```

```{r}

h2.mov <- induced.subgraph(h1.mov,vids=clusters(h1.mov)$membership==1)

plot(h2.mov,vertex.color="red",

edge.width=sqrt(E(h1.mov)$weight),

vertex.shape="circle",

vertex.size=(V(h2.mov)$IMDBrating)-3,

vertex.label=NA)

```

```{r}

table(graph.coreness(h2.mov))

h3.mov <- induced.subgraph(h2.mov,vids=graph.coreness(h2.mov)>4)

h3.mov

plot(h3.mov,vertex.color="red",

vertex.shape="circle",

edge.width=sqrt(E(h1.mov)$weight),

vertex.label.cex=0.7,

vertex.label.color="darkgreen",

vertex.label.dist=0.3,

vertex.size=

(V(h3.mov)$IMDBrating)-3)

```

Part II Bradley Cooper

```{r}

h4 <- subgraph.edges(h1,

E(h1)[inc(V(h1)[name %in%

c("Bradley Cooper")])])

plot(h4, layout = layout\_with\_kk)

```

```{r}

V(h4)[type==TRUE]$name

```

```{r}

h5 <- subgraph.edges(h1,

E(h1)[inc(V(h1)[name %in%c(V(h4)[type==TRUE]$name)

])])

plot(h5, layout = layout\_with\_kk)

```

Part III

```{r}

summary(h3.mov)

V(h3.mov)

```

```{r}

SW3 <- induced.subgraph(h1.mov,vids=c(V(h1.mov)[nei("Star Wars: Episode III - Revenge of the Sith")],V(h1.mov)[name%in%"Star Wars: Episode III - Revenge of the Sith"]))

```

```{r}

plot(SW3, layout = layout\_with\_kk,,vertex.color="red",

vertex.shape="circle",

edge.width=sqrt(E(h1.mov)$weight),

vertex.label.cex=0.7,

vertex.label.color="darkgreen",

vertex.label.dist=0.3,

vertex.size=(V(h1.mov)$IMDBrating)-3)

```