Central European University
Statistical Methods for Network Science
Midterm Assignment

Marvel Characters Network Analysis

by

Chyngyz Davletov

Professor: Rosario Nunzio Mantegna
Teaching Assistant: Luca Marotta

October 2015

Table of Contents

List of Tables

List of Figures

1 General Analysis of the Marvel Characters Projected Network

The projection of the initial bipartite Marvel network on the set of characters has 16,827 links between 6,486 characters, resulting in 51.89 average degree."Captain America" has the largest degree of 1,933, whereas"Berserker I" has degree 0. The distribution of the degrees is presented in Figure 1. It contains two plots: the left is a log-frequency against degree, and the right is a log-log plot. It shows that degree is distributed according to a power law, which proves the existence of giant components. If we look, for example at the log-log plot, we observe that there is a small number of nodes on the left, increasing side of the distribution, and the majority of nodes are on the right, decreasing side, of the plot which follow the power law. Hence, we expect the model to exhibit preferential attachment.

Apart from degree centrality, there is another metric always discussed in a general description of a graph – betweenness centrality, which is the number of shortest paths passing through a particular vertex. It turns out that the highest value of betweenness centrality measure is the one of Havok/Alex Summers character, who stands on the way of 889,404 shortest paths; the character named Abominatrix has the smallest number for this metric – 0.

Metrics that were mentioned so far were about vertex centrality. Since there are not only vertices in graphs, but also edges, it is natural to discuss edge betweenness centrality. In Marvel projection graph the edge with the largest betweenness value is Devourer – Hawk which has 49,684 passing shortest paths, about twenty times less than its vertex analogue.

To summarize the relative frequency of a graph, we estimate clustering coefficient metric. In Marvel projection on characters, clustering coefficient is 0.19 meaning that only about one-fifth of the connected triples (subgraph of three vertices connected by two edges) close to form a triangle.

The metrics that describe connectivity of the graph conclude the general overview of the Marvel's character network. The network density is measured by the metrics of the same name, that is, density of the network is 0.008, i.e. out of possible 21,030,855 edges the graph has only 21,030,855 $\times$ 0.008$=$ $168,267$ edges, which is not that dense. The graph is, obviously not connected, however, as pointed above, it has connected components; finally, the average path length is 2.64, which is quite small, therefore the graph has some features of a small-world.

2 Analysis of Fantastic Four, X Men, and Amazing Spider Man Networks

After providing a general analysis of the Marvel Network projected on the set of characters, we go more in details and switch to three arguably most successful Marvel's comics and analyse the properties of Fantastic Four, X Men, and Amazing Spider-Man networks.

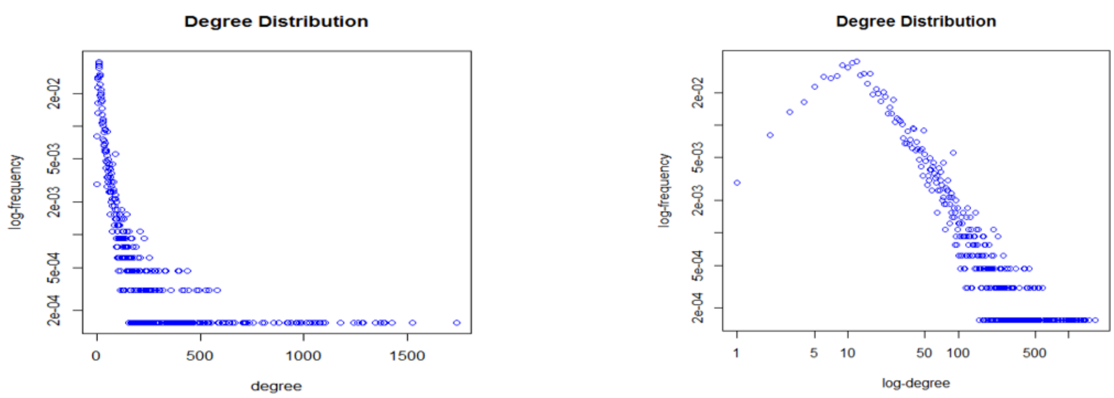


Figure 1: Degree Distribution for Marvel Network Projected on a Set of Characters

2.1 Obtaining the Networks

First, we needed to obtain the networks themselves. This is done by getting the subgraphs of the original complete bipartite Marvel network that are related to Fantastic Four (FF) issues, X men (UX) issues, and Amazing Spider-Man (ASM) issues. The function induced.subgraph was used for this purpose. See appendix for the full code.

2.2 Selecting the Subgraphs Based on the Number of Issues

After obtaining the separate networks for FF, UX, and ASM, we then choose 6 subgraphs from these three networks. This will be important when we analyse the dynamics of the networks as the number of issues increase, that is, how the characteristics of the FF, UX, and ASM networks changed over time with the appearance of more and more new issues. For this purpose, in each of the three sets obtained in section 2.1, 6 subgraphs related to a different number of issues are chosen: the first issue, first 3 issues, first 10 issues, first 30, first 100, and 300.[1]

2.3 Projections of the FF, UX, and ASM on the Set of Characters

2.3.1 Obtaining Projections of 6 subgraphs on the Characters Set of Nodes

From each of the 6 subgraphs related to different number of issues, obtained in section 2.2, we get the networks projected on the set of characters for further analysis. The function used for this purpose is bipartite.projection.

2.3.2 Average Degree

This section discusses the behaviour of the three networks over time in the context of the average degree of a network. Average degree measures the degree to which the graph is connected, thus we want to see how the overall connection in the three networks has been evolving as new issues of comics came out.

We can analyse this dynamics with the help of the function mean(degree()) in R. Table 1 summarizes the values of average degree of 6 sub-graphs related to a different number of issues obtained from the initial FF, UX, and ASM networks. FF network has an average degree of 4, when it is related only to the first issues , it has a value of 4.67 when it is comprised of first three issues, 8 – when the network is obtained from 10 issues, and so on until it reaches 24.28 when obtained from the network with 300 issues. It is clear that the value of the average degree is growing as there are more and more issues. The same pattern is observed for networks UX and ASM. Therefore, we can state that the

[1] Both in sections 2.1 and 2.2 there were technical difficulties with programming the extractions of the subgraphs using induced.subgraph function in R. A sufficient amount of hours were spent and different programming manipulations were implemented before eventually obtaining the wanted results.

Table 1: Average Degrees of FF, UX, and ASM Networks Corresponding to the Number of Issues

Issues	Fantastic Four	X Men	Amazing Spider-Man
1	4	6	5
1:3	4.67	8	6.75
1:10	8	14.97	10.84
1:30	12.64	17.73	19.48
1:100	19.33	23.97	22.97
1:300	24.28	47.03	25.51

Table 2: Main Hubs and Their Degrees

Fantastic Four					
Character	"Human Torch" Johnny S	"Mr.Fantastic" Reed R	"Invisible Woman" Sue	"Thing" Benjamin J	Gr Richards Franklin B
Degree	237	236	231	224	131
X Men					
Character	"Cyclops" Scott Summer	"Storm" Ororo Munroe	"Wolverine" Logan	"Colossus" Peter Ra	"Professor X" Charles
Degree	361	358	355	352	330
Amazing Spider-Man					
Character	"Spider-Man" Peter Parker	Jameson J.Jonah	May Parker	Joe Robertson	Betty Brant Leeds
Degree	265	197	167	167	163

average degree grows with the number of issues. This result is not a surprise, because the more there are issues, the more characters are added into the network, and these new characters almost always have to be connected with more than one character from previous issues, thus, an increase in the number of additional connections (i.e. edges) is greater than an increase in the number of characters (nodes) with each new character from a new issue. As a result, the average degree grows in each of FF, UX, and ASM networks as new issues appear.

Figure 2 presents the dynamics of the average degree over time. Similar pattern of the average degree growth is observed in FF and ASM networks, where the shape of the polylines can be approximated by the logarithmic curve. The pattern differs to some extent in UX network, where we can apply such approximation only for the first 50 issues. Following 50th issue the shape of the average degree growth curve reflects linear relationship.

2.3.3 Main Hubs

In this section we want to figure out the behaviour of the main hubs over time. First, the hubs are defined as 5 top characters in terms of the value of a degree in a network that is related to the issues from 1 to 300. So, let us see who these hubs are in the Table 2. The top character by means of degree in a network that is related to first 300 issues in, for example, FF network, is Johnny, Human Torch,

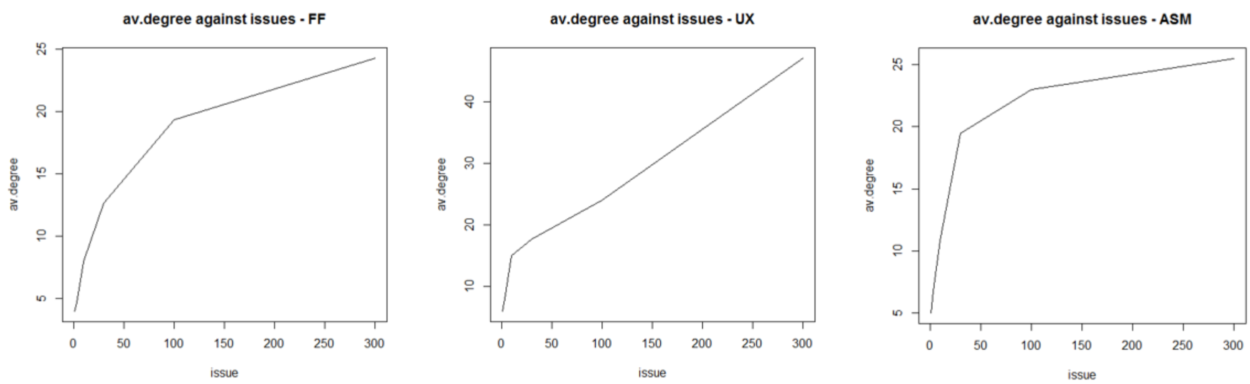


Figure 2: Dynamics of the Average Degree with respect to the Number of Issues

Table 3: Dynamics of Fantastic Four Network Hubs' Degrees

Issues	"Human Torch" Johnny S	"Mr.Fantastic" Reed R	"Invisible Woman" Sue	"Thing" Benjamin J	Gr Richards Franklin B
1	4	4	4	4	–
1:3	5	5	5	5	–
1:10	13	13	13	13	–
1:30	49	49	49	49	–
1:100	109	109	108	109	25
1:300	237	236	231	224	131

with a degree value of 237, followed by Mr.Fantastic with degree of 236, then by Invisible Woman (231), Thing (224), and Gr Richards, Franklin B (131).

Now, let us see how the hubs' degree, that is, the importance of them in the network, has been evolving from their first appearance in the issue up to the last within first 300 issues. To do this we used again the `degree()` function, but now for specific nodes. In order to make it easily done, the new attribute `name` was added to each node in the initial graph with all issues (though the final solution was indeed easy, the way to it was rather a long and winding road). Table 3 summarizes the dynamics for FF network. Four main super-heroes, that is, Fantastic Four itself, have been evolving in a very similar fashion in terms of their degree as Marvel introduced new issues of the comics. Therefore, there is no much interest in their degree dynamics. On the contrary, it is much more interesting to look at how the degree of Franklin B Richards grew. There are no any degree entries for this character up to 30th issue in the table above. The first appearance of Franklin B is observed in one of the subsequent issues and his degree is already very high and increases almost 5 times in the following 100 issues. It would be hard to explain this fact that Franklin B became so important a hero in this comics if we do not know that this character is actually the son of Mr. Fantastic and Invisible Woman. Being so much important to the main hubs of FF network, Franklin B obviously became one of the main hubs of the whole network as well.

Figure 3 shows the degree dynamics of Johnny S and Franklin B. Since the growing pattern of first four main hubs in this network are very similar, only the evolution of the degree for Johnny S is shown to compare it with that of Franklin B. The growth of Johnny's degree is fastest in the early issues (before 50th), and then the speed of growth starts to decrease, nevertheless being quite high. On the contrary, Franklin's degree grows with the highest speed after 100th issue, and it is worth noting that it is higher than the corresponding speed of Johnny.

Now let us switch to another popular comics, namely X Men. Table 4 presents X Men hubs' degree dynamics. It can be inferred from the table that two characters exhibit a similar pattern in terms of the

Table 4: Dynamics of X Men Network Hubs' Degrees

Issues	"'Cyclops" Scott Summer	"Storm" Ororo Munroe	"Wolverine" Logan	"Colossus" Peter Ra	"Professor X" Charles
1	6	–	–	–	6
1:3	9	–	–	–	9
1:10	28	–	–	–	28
1:30	66	–	–	–	66
1:100	146	51	51	51	121
1:300	361	358	355	352	330

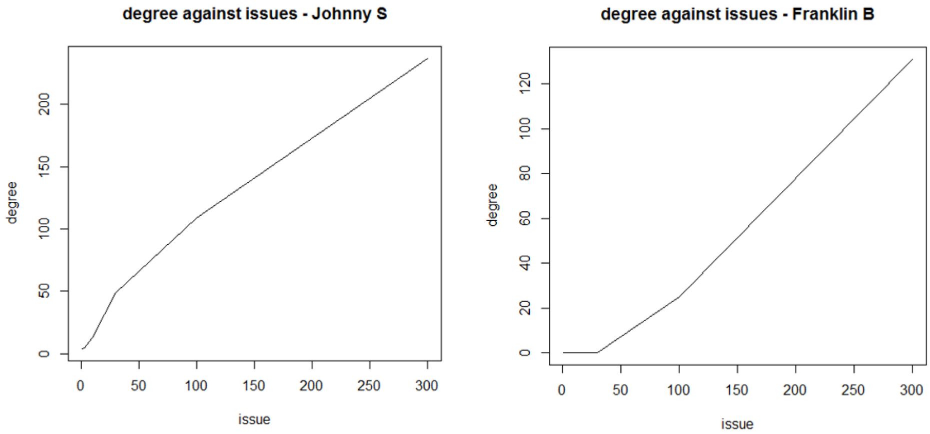


Figure 3: Degree Dynamics of Johnny S and Franklin B as a function of the Number of Issues

growth of the degree with an increase in the number of issues: Cyclops and Professor Xavier. These are the two heroes who participated in almost all (if not all) actions in the world of X men from the beginning of the story. Unsurprisingly, they are among the 5 main hubs in the network that contained issues up to 300th (with Cyclops being the biggest hub with 361 degree value, and Professor X – 5th (330)). These two characters do not seem show a behaviour that could be named an extreme one – their degree grew very much proportion-wise with the increase in the number of issues. Therefore, we leave the discussion of these heroes and switch to the hubs who are ranked 2nd, 3rd, and 4th by means of their degree in the network with up to 300 issues. We observe the similar pattern of growth in the degrees of Storm, Wolverine, and Colossus: each of them appear for the first time in the one of the issues that come after the 30th issues and their importance grows extremely fast. The degree of each of them increased seven times in the last network that contained issues up to 300. Their extreme growth of importance to the network can be explained by the luckily chosen abilities that were given them by their authors and which met a strong success in the reading audience's minds. As a result, Storm, Wolverine, and Colossus participated in almost all actions in the subsequent issues, and their degrees increased accordingly.

Comparison of the growing patterns in the degrees of Wolverine and Cyclops is depicted in Figure 4. Wolverine's degree grew faster than that of Cyclops in the last group of issues. Cyclops' degree has the highest speed of growth before 50th issues, and then starts to slow down just like the degree of

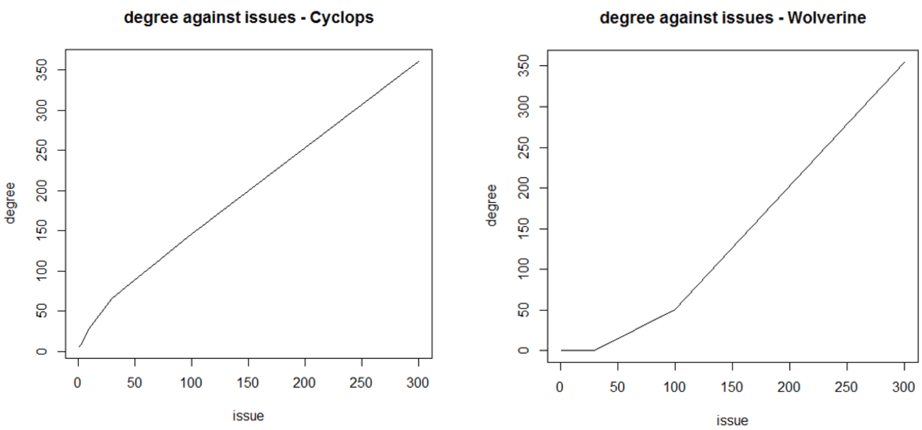


Figure 4: Degree Dynamics of Cyclops and Wolverine as a function of the Number of Issues

Johnny S in FF network. Wolverine's pattern is very similar to that of Franklin B. There is no need to plot the growing patterns of Professor Xavier, since it will be very much like that of Cyclops' shape. For the same reason, Storm's and Colossus' degree dynamics are not plotted.

The degree dynamics of Amazing Spider-Man network hubs' is summarized in Table 5. First three hubs, namely, Spider-Man, J. Jonah Jameson, and May Parker, demonstrate a moderate growth, associated with at most proportionate growth of a degree with an increase in the number of issues[2]. The hub ranked 5th, Betty Brant, also does not exhibit extreme behaviour in terms of the degree growth, the value increases slightly more than 2 times between 5th subgraph (issues from 1 to 100) and 6th (issues up to 300). The most interesting character by means of degree growth is Joe Robertson. His behaviour can be called an extreme one, because he "came out of nowhere" in the ASM 51st issue and by the 6th subgraph his degree rose three times, i.e. with the same speed as the growth of the Spider-Man, reaching the level of May Parker, Spider-Man's aunt.

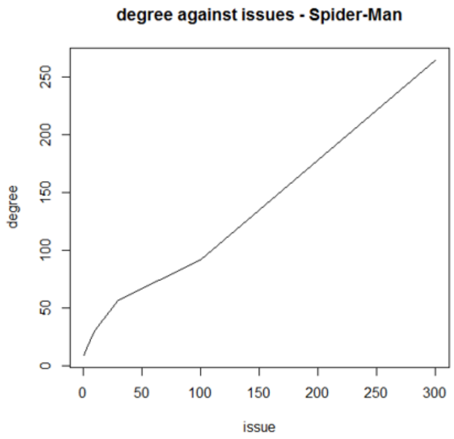
The degree growth patterns of Spider-Man and Joe Robinson are shown in Figure 5. What is inferred in this figure is not that much different from what we was observed in figures 2 and 3 except one thing: the growth of the old hub – Spider-Man – is quickest in the beginning, just as the old hubs' in the previous figures, but the speed is not decreasing steadily: after some slowing down somewhere between 30 and 100 issues, Spider-Man's degree starts to grow with higher speed, though not as high as in the very beginning, before 30th issues. Thus, the processes behind this fact might be different in ASM network in comparison with UX and FF. It seems that new characters from new issues increased effectively not only the average degree of the network, but were tied closely with the hubs from the previous issues.

To summarize, in each network there are main hubs that appeared in the very first issues of the comics and exhibit for some time the pattern of degree growth that is similar to logarithmic. And in some middle and late issues there come new characters that "come out of nowhere" and quickly become very important in the network showing the increasing growth (i.e. the speed of growth in degree increases with time) and in some cases reaching the speed of growth of the largest hubs (for example, Joe Robertson in ASM network), and even outrunning in the others (e.g. Wolverine in UX). However, if we think about where did they come from, it is clear that actually they came out not from nowhere, but from their authors minds, which means that the purpose of their appearance in the comics was to get into all the main actions of the story lines of the comics, thus there is nothing surprising in their speed of increase in the degree value. So, their extreme behaviour was planned by the authors.

[2] Spider-Man's degree increased almost three times with a three times increase in the number of issues (from 100 to 300).

Table 5: Dynamics of Fantastic Four Network Hubs' Degrees

Issues	"Spider-Man" Peter Parker	Jameson J.Jonah	May Parker	Joe Robertson	Betty Brant Leeds
1	9	5	5	–	–
1:3	15	12	8	–	–
1:10	30	28	25	–	21
1:30	57	56	55	–	52
1:100	92	90	75	47	72
1:300	265	197	167	167	163



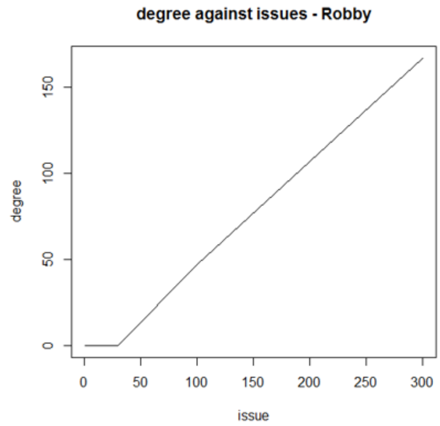


Figure 5: Degree Dynamics of Spider-Man and Joe Robinson as a function of the Number of Issues

2.3.4 Diameters

In this section we look at the evolution of connectedness of our networks by means of another measure –– diameter of a network. To measure the longest distance in a graph, we use the function diameter() in R. Table 6 contains the entries for the values of the diameter of each of the 6 subgraphs for networks FF, UX, and ASM. It can be concluded from the table that diameters grew steadily with the number of issues. This is an expected outcome, because when new issues come out, the network becomes more complex, and, thus, there is less freedom for a particular vertex to be connected with another directly (in other words, the least number of other vertices that must be traversed in order to reach that vertex increases), therefore, we expected the diameter to increase with the number of issues. However, the speed of the growth of the diameters is very low (though its increase is steady anyway). There were a lot of experimental studies of the dependence of the diameter on the number of nodes (for example, see Ricard (2006)). It is possible to implement the results of those studies in this analysis, because increase in the number of issues means increase in the number of nodes. The result of experiments testing the hypothesis whether or not the diameter of a graph depends on the number of nodes suggests that if nodes are added or deleted randomly into or from the graph, its diameter does not change significantly. However, same those results say that if a change in a number of nodes of a graph is done non-randomly, than diameter tends to be significantly elastic to those changes. Moreover, if the nodes that were taken out of the network had been important, i.e. highly connected (simply, hubs), the diameter of a graph increased. And this is where our analysis coincides with the experimental research: 1) because in new issues the nodes were added probably non-randomly, but rather based on the history of previous issues and possibly many other reasons related to a change in story line of the comics, perhaps marketing and profit-maximization and so on,

Table 6: Diameters of FF, UX, and ASM Networks Corresponding to the Number of Issues

Issues	Fantastic Four	X Men	Amazing Spider-Man
1	1	1	2
1:3	2	2	3
1:10	3	3	3
1:30	5	5	3
1:100	5	5	4
1:300	6	5	4

the increase in the nodes was not random; and 2) as it will become apparent in the next sections, the characters, that were added in the early-to-middle groups of issues, we supportive and not always highly-connected. Second point is very similar to the practice of taking out important nodes, thus, network diameter tends to increase, which is actually observed in our networks.

2.3.5 Coreness

The definition (Kolaczyk and Csárdi, 2014) says: a k-core of a graph G is a subgraph of G for which all vertex degrees are at least k, and such that no other subgraph obeying the same condition contains it (i.e., it is maximal in this property). In Marvel projection on characters set of nodes the values of k for FF, UX, and ASM networks in 6 subgraphs from section 2.3.1 are presented in Table 7. In each network degree of the maximal k-core is increasing with the number of issues. The speed of growth is moderate in the early stages when the number of issues is not so big (up to 10), and starting with the subsequent issues the growth in the value of k in maximal k-core boosts. Thus, k increases roughly 2 times from issues 1:10 to issues 1:30 in FF and about two and a half times in ASM network. The similar pattern is observed during the increase in the number of issues from 30 to 100 in same FF network, and in UX network, where k increases from 14 to 24 in k-core maximal. However, both FF and ASM do not show a growth of k when number of issues increases from 100 to 300 (which is not different in ratio-increase from an increase from 10 to 30, or from 1 to 3, but is much bigger in absolute number of issues, therefore a very important change); on the contrary, k increases by more than a half during the same period in UX network (from 24 to 40).

How can this be explained? It should be the case that the new characters that came with new issues, when the number of issues rose from 10 to 30, were supporting characters in FF and ASM networks, i.e. their main purpose was not to interact with main actors in these comics, but rather to bring more connections among the secondary characters, thus increasing their degrees in the network. Intuitively, number of issues 10 means that the comics is quite popular, but in order to keep the audience and sales, the network of the comics should be widened by means of new characters, and this characters should be introduced into the network in a smart and smooth way, without damaging the positions of the existing hubs and bringing more connectivity, thus making the network more complex[3]. Why did not k increase during this period in UX network? If we look at the value of k when number of issues is 1:10, then we see that k is 13, i.e. sufficiently higher than in FF and ASM, even when issues increase up to 30, value of k=12 in FF network still does not reach the value of k in UX with 10 issues (k=13). In other words, it might be the case that it was simply unnecessary to introduce new characters in UX network in the issues following the 10$^{\text{th}}$ issue. If we go further, we see that k increased by more than a half in the issues following 100$^{\text{th}}$ (i.e. issues 1:300) in UX network. According to the argument introduced in this paragraph, this makes sense.

[3] This is partly (if not fully) an economic intuition, because the author of this text has been trained to think as an economist, and sometimes this economic thinking can be introduced in other areas as well.

Table 7: Degrees of the maximal k-core in FF, UX, and ASM Networks

Issues	Fantastic Four	X Men	Amazing Spider-Man
1	4	6	5
1:3	4	7	6
1:10	7	13	9
1:30	12	14	23
1:100	21	24	23
1:300	23	40	24

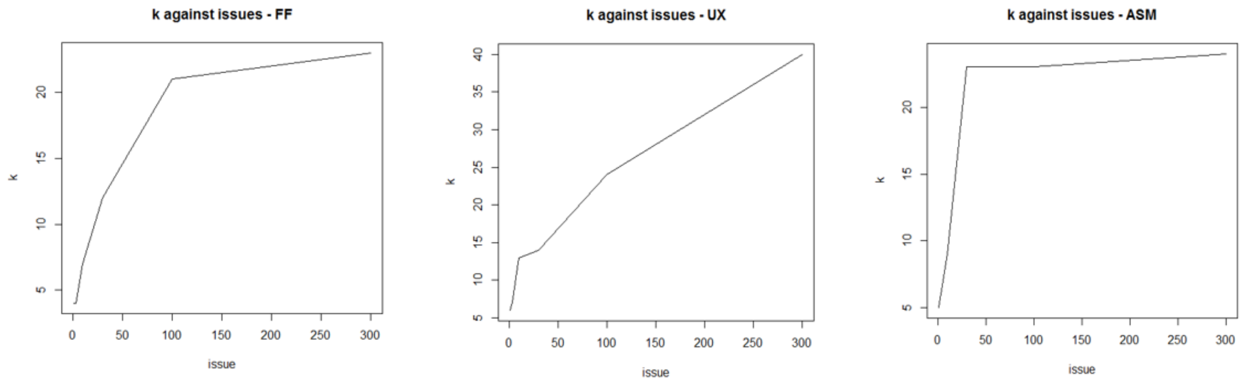


Figure 6: Dynamics of k with respect to the Number of Issues

Figure 6 shows the growth of the k in maximal k-core as a function of the number of issues. The slopes and growing patterns repeat the discussion of the preceding paragraph, that is, in UX network k grew with a highest speed between the 5^{th} and 6^{th} group of issues (between 1:100 and 1:300 issue groups), in ASM network k basically did not grow starting from the 40-50 issues, and in FF network k grew very fast up to first 100 issues and then slowed down substantially. Generally, in all networks the speed of growth of k decreases with time. This reminds us of logarithmic nature, hence we can characterize the growth of k as having roughly logarithmic pattern.

2.3.6 Composition and Size of Maximal k-Cores

Now let us discuss the size of the maximal k-cores in FF, UX, and ASM networks in each 6 subgraphs from section 2.3.1. Table 8 presents the sizes of k-cores in three networks. Table shows that the number of vertices increased steadily with the number of issues. This fact is of no surprise and could have been proposed in section 2.3.3 where we analysed the dynamics of the degrees in three networks across the time. Indeed, since new characters added more connections than an increase in the number of characters themselves with each new issue, the more and more nodes got their degrees increased. If we think about how maximal k-core is obtained, everything becomes even more obvious: the more there are nodes with higher degree values the higher is the value of k in the maximal k-core.

It would be cumbersome to put the composition of the maximal k-cores in each three networks for each 6 subgraph here, because UX network, for example, has 118 names in the last group of issues, thus listing all of them requires time and space, and is probably useless. The names can be easily reached in the R-studio if there is such a purpose. Other than that, it is worth noting that up to issues 30, FF maximal k-cores were composed mainly from the five main hubs discussed in section 2.3.3, and starting from the subsequent issues some other hubs joined the maximal k-cores (Cyclops from UX,

Table 8: Number of Vertices in the Maximal k-Core in FF, UX, and ASM Networks

Issues	Fantastic Four	X Men	Amazing Spider-Man
1	5	7	6
1:3	6	8	7
1:10	8	14	15
1:30	13	15	24
1:100	22	42	24
1:300	35	118	54

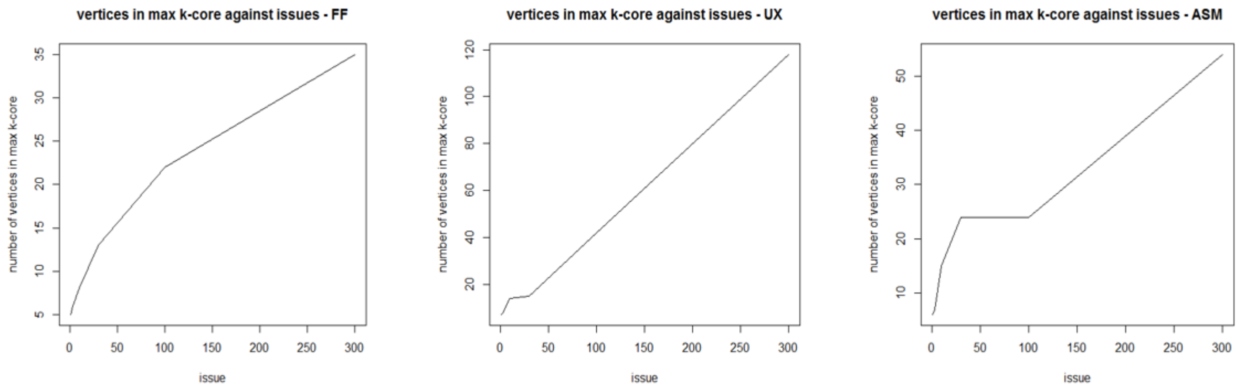


Figure 7: Evolution of the Size of Maximal *k*-Core over Time

Captain America, Spider-Man from ASM, and many others). The similar situation went in UX and ASM networks. So, it can be concluded that in late issues the world of Marvel started to become more and more connected. This also satisfies the "economic" argument in the previous section. Figure 7 shows an illustration of what has happened between issues 1 and 300. The growth of the number of nodes in the maximal *k*-cores in FF network has decreasing returns to scale (it is concave); up to 100^{th} issues it increases with a more than one-to-one rate, and falls a bit lower than one-to-one ever since. On the contrary, UX and ASM are more stable in the sense of growing in the number of nodes in the maximal *k*-cores. UX has around one-to-one growth rate of the number of nodes all the way starting from the 30^{th} or 40^{th} issues, while ASM – from 100^{th}. One explanation to this difference in the growth patterns of the number of nodes with respect to the number of issues is probably because newly appeared characters starting from 100^{th} issues in UX and ASM networks increased the degrees of relatively low-degreed nodes. This helped to increase the degrees of those nodes who were not in the maximal *k*-core before, thus, after an issue with such supporting characters is introduced, those low-degreed nodes get into the maximal *k*-core. As a result, UX and ASM networks maintained the same speed of growth in the number of nodes in the maximal *k*-cores, whereas FF did not exhibit the same behaviour.

3 Conclusion

Different measures and network statistics were used in this work in the analysis of the evolution of the Marvel network over time. First, the general overview has shown that the projection of the whole Marvel network on its character set of nodes is relatively not a dense one; clustering coefficient is low meaning that there are relatively small number of triangles in the network, and overall the connectivity level is low on average. However, the graph has small-world features that are shown by a quite low average path length and presence of giant components, and very big in terms of degree hubs, suggesting the high level of connection in many small parts of which the whole network is comprised. Also, as we saw in the analysis of the evolution of the main hubs, we can conclude that the network has the preferential attachment property. Altogether it makes the analysis of the Marvel projection on the characters set fairly interesting.

More in-depth analysis of particular comics' families revealed the dependence of the main network characteristics on the time. In general, for the Marvel network, measures of connectivity are elastic with respect to time. This was examined by the behaviour of maximal *k*-cores and diameters. *K*-cores, in turn, were highly correlated with degree measures, as we observed it by the examples of some of the main hubs in Fantastic Four, X Men, and Amazing Spider-Man networks for different number of issues, in relation to which these networks were split. To sum up, as the time goes on, centrality, density, connectivity measures tend to grow. Though the bottom line here is a bit too general, because being more specific requires more experience, time (and time again), and purpose, which can be all implemented in future works, namely final project.

References

Kolaczyk, E. and Csárdi, G. (2014). *Statistical Analysis of Network Data with R*. Use R! Springer New York.

Ricard, J. (2006). *Emergent Collective Properties, Networks and Information in Biology*. Elsevier.

Appendix

```
setwd("C:/Users/User/Desktop/Чина=CEU/2nd Year/1.Fall2015/Statistical Methods/midterm")

#### 1 ####

library(igraph)
m1 <- read.graph("marvel.net", format="pajek") #complete initial bipartite graph
V(m1)$name=V(m1)$id # adding name attribute
m2 <- bipartite.projection(m1, multiplicity = TRUE) # to keep weights normal in projected graphs
m3 <- m2$proj1 # projection of network on the character set of nodes
vcount(m3)
ecount(m3)

### vertex centrality
degree(m3)
mean(degree(m3)) # average degree = 51.88622
min(degree(m3)) # 0
which.min(degree(m3)) # Berserker II
max(degree(m3)) # 1933
which.max(degree(m3)) # Captain America
betweenness(m3, v=V(m3), directed=F)
max(betweenness(m3)) # 889404.7
which.max(betweenness(m3)) # Havok/Alex Summers
min(betweenness(m3)) # 0
which.min(betweenness(m3)) # Abominatrix

### degree distribution
plot(degree.distribution(m3), xlab="degree", ylab="frequency", main = "Degree Distribution")
hist(degree.distribution(m3), col="blue",
     xlab = "Degree", ylab = "Frequency",
     main = "Degree Distribution")

dd <- degree.distribution(m3)
d <- 1:(max(degree(m3))-1)
ind <- (dd!= 0)
plot(d[ind], dd[ind],log = "xy",col="blue",
     xlab = c("log-degree"), ylab = c("log-frequency"),
     main = "Degree Distribution")

### clustering coefficient
transitivity(m3)

### edge centrality
eb <- edge.betweenness(m3, e=E(m3), directed = F)
E(m3)[order(eb, decreasing=T)[1:3]]
max(eb) # 49684.43
which.max(eb)

### density
n <- vcount(m3)
l <- ecount(m3)
density <- l/(n*(n-1)/2)
graph.density(m3) # ~0.008

### components
is.connected(m3)
cl.m3 <- components(m3)
cl.m3
average.path.length(m3)

#### 2 ####

ff.graph <- induced.subgraph(m1, vids=unlist(neighborhood(m1,order=1,nodes=(grep("^FF [0-9]|^FF [0-9]+/", V(m1)$id)))))
ux.graph <- induced.subgraph(m1, vids=unlist(neighborhood(m1,order=1,nodes=(grep("^UX [0-9]|^UX [0-9]+/", V(m1)$id)))))
asm.graph <- induced.subgraph(m1, vids=unlist(neighborhood(m1,order=1,nodes=(grep("^ASM [0-9]|^ASM [0-9]+/", V(m1)$id)))))

#### 3 ####

ff.1 <- induced.subgraph(graph=ff.graph,vids=unlist(neighborhood(ff.graph,order=1,nodes=grep("^FF 1$|^FF 1/[1-9]$",V(ff.graph)$id))))
ff.3 <- induced.subgraph(graph=ff.graph,vids=unlist(neighborhood(ff.graph,order=1,nodes=grep("^FF [1-3]$|^FF [1-3]/[1-9]$",V(ff.graph)$id))))
ff.10 <-induced.subgraph(graph=ff.graph,vids=unlist(neighborhood(ff.graph,order=1,nodes=grep("^FF [1-9]$|^FF [1-9]/[1-9]$|^FF 10$|^FF 10/[1-
9]$",V(ff.graph)$id))))
ff.30 <- induced.subgraph(graph=ff.graph,vids=unlist(neighborhood(ff.graph,order=1,nodes=grep("^FF [1-9]$|^FF [1-9]/[1-9]$|^FF [1-2][0-9]$|^FF [1-
2][0-9]/[1-9]$|^FF 30$|^FF 30/[1-9]$",V(ff.graph)$id))))
ff.100 <- induced.subgraph(graph=ff.graph,vids=unlist(neighborhood(ff.graph,order=1,nodes=grep("^FF [1-9]$|^FF [1-9]/[1-9]$|^FF [1-9][0-9]$|^FF [1-
9][0-9]/[1-9]$|^FF 100$|^FF 100/[1-9]$",V(ff.graph)$id))))
ff.300 <- induced.subgraph(graph=ff.graph,vids=unlist(neighborhood(ff.graph,order=1,nodes=grep("^FF [1-9]$|^FF [1-9][0-9]$|^FF [1-2][0-9][0-9]$|^FF
300$|^FF [1-9]/[1-9]$|^FF [1-9][0-9]/[1-9]$|^FF [1-2][0-9][0-9]/[1-9]$|^FF 300/[1-9]$",V(ff.graph)$id))))

ux.1 <- induced.subgraph(graph=ux.graph,vids=unlist(neighborhood(ux.graph,order=1,nodes=grep("^UX 1$|^UX 1/[1-9]$",V(ux.graph)$id))))
ux.3 <- induced.subgraph(graph=ux.graph,vids=unlist(neighborhood(ux.graph,order=1,nodes=grep("^UX [1-3]$|^UX [1-3]/[1-9]$",V(ux.graph)$id))))
ux.10 <-induced.subgraph(graph=ux.graph,vids=unlist(neighborhood(ux.graph,order=1,nodes=grep("^UX [1-9]$|^UX [1-9]/[1-9]$|^UX 10$|^UX 10/[1-
9]$",V(ux.graph)$id))))
ux.30 <- induced.subgraph(graph=ux.graph,vids=unlist(neighborhood(ux.graph,order=1,nodes=grep("^UX [1-9]$|^UX [1-9]/[1-9]$|^UX [1-2][0-9]$|^UX [1-
2][0-9]/[1-9]$|^UX 30$|^UX 30/[1-9]$",V(ux.graph)$id))))
ux.100 <- induced.subgraph(graph=ux.graph,vids=unlist(neighborhood(ux.graph,order=1,nodes=grep("^UX [1-9]$|^UX [1-9]/[1-9]$|^UX [1-9][0-9]$|^UX [1-
9][0-9]/[1-9]$|^UX 100$|^UX 100/[1-9]$",V(ux.graph)$id))))
ux.300 <- induced.subgraph(graph=ux.graph,vids=unlist(neighborhood(ux.graph,order=1,nodes=grep("^UX [1-9]$|^UX [1-9][0-9]$|^UX [1-2][0-9][0-9]$|^UX
300$|^UX [1-9]/[1-9]$|^UX [1-9][0-9]/[1-9]$|^UX [1-2][0-9][0-9]/[1-9]$|^UX 300/[1-9]$",V(ux.graph)$id))))

asm.1 <- induced.subgraph(graph=asm.graph,vids=unlist(neighborhood(asm.graph,order=1,nodes=grep("^ASM 1$|^ASM 1/[1-9]$",V(asm.graph)$id))))
asm.3 <- induced.subgraph(graph=asm.graph,vids=unlist(neighborhood(asm.graph,order=1,nodes=grep("^ASM [1-3]$|^ASM [1-3]/[1-9]$",V(asm.graph)$id))))
asm.10 <-induced.subgraph(graph=asm.graph,vids=unlist(neighborhood(asm.graph,order=1,nodes=grep("^ASM [1-9]$|^ASM [1-9]/[1-9]$|^ASM 10$|^ASM 10/[1-
9]$",V(asm.graph)$id))))
asm.30 <- induced.subgraph(graph=asm.graph,vids=unlist(neighborhood(asm.graph,order=1,nodes=grep("^ASM [1-9]$|^ASM [1-9]/[1-9]$|^ASM [1-2][0-
9]$|^ASM [1-2][0-9]/[1-9]$|^ASM 30$|^ASM 30/[1-9]$",V(asm.graph)$id))))
asm.100 <- induced.subgraph(graph=asm.graph,vids=unlist(neighborhood(asm.graph,order=1,nodes=grep("^ASM [1-9]$|^ASM [1-9]/[1-9]$|^ASM [1-9][0-
9]$|^ASM [1-9][0-9]/[1-9]$|^ASM 100$|^ASM 100/[1-9]$",V(asm.graph)$id))))
asm.300 <- induced.subgraph(graph=asm.graph,vids=unlist(neighborhood(asm.graph,order=1,nodes=grep("^ASM [1-9]$|^ASM [1-9][0-9]$|^ASM [1-2][0-9][0-
9]$|^ASM 300$|^ASM [1-9]/[1-9]$|^ASM [1-9][0-9]/[1-9]$|^ASM [1-2][0-9][0-9]/[1-9]$|^ASM 300/[1-9]$",V(asm.graph)$id))))

#### 4 ####

ff.1.proj <- bipartite.projection(ff.1, multiplicity = TRUE)
ff.3.proj <- bipartite.projection(ff.3, multiplicity = TRUE)
ff.10.proj <- bipartite.projection(ff.10, multiplicity = TRUE)
ff.30.proj <- bipartite.projection(ff.30, multiplicity = TRUE)
```

```
ff.100.proj <- bipartite.projection(ff.100, multiplicity = TRUE)
ff.300.proj <- bipartite.projection(ff.300, multiplicity = TRUE)

ff.1.proj.char <- ff.1.proj$proj1
ff.3.proj.char <- ff.3.proj$proj1
ff.10.proj.char <- ff.10.proj$proj1
ff.30.proj.char <- ff.30.proj$proj1
ff.100.proj.char <- ff.100.proj$proj1
ff.300.proj.char <- ff.300.proj$proj1

ux.1.proj <- bipartite.projection(ux.1, multiplicity = TRUE)
ux.3.proj <- bipartite.projection(ux.3, multiplicity = TRUE)
ux.10.proj <- bipartite.projection(ux.10, multiplicity = TRUE)
ux.30.proj <- bipartite.projection(ux.30, multiplicity = TRUE)
ux.100.proj <- bipartite.projection(ux.100, multiplicity = TRUE)
ux.300.proj <- bipartite.projection(ux.300, multiplicity = TRUE)

ux.1.proj.char <- ux.1.proj$proj1
ux.3.proj.char <- ux.3.proj$proj1
ux.10.proj.char <- ux.10.proj$proj1
ux.30.proj.char <- ux.30.proj$proj1
ux.100.proj.char <- ux.100.proj$proj1
ux.300.proj.char <- ux.300.proj$proj1

asm.1.proj <- bipartite.projection(asm.1, multiplicity = TRUE)
asm.3.proj <- bipartite.projection(asm.3, multiplicity = TRUE)
asm.10.proj <- bipartite.projection(asm.10, multiplicity = TRUE)
asm.30.proj <- bipartite.projection(asm.30, multiplicity = TRUE)
asm.100.proj <- bipartite.projection(asm.100, multiplicity = TRUE)
asm.300.proj <- bipartite.projection(asm.300, multiplicity = TRUE)

asm.1.proj.char <- asm.1.proj$proj1
asm.3.proj.char <- asm.3.proj$proj1
asm.10.proj.char <- asm.10.proj$proj1
asm.30.proj.char <- asm.30.proj$proj1
asm.100.proj.char <- asm.100.proj$proj1
asm.300.proj.char <- asm.300.proj$proj1

mean(degree(ff.1.proj.char)) # 4
mean(degree(ff.3.proj.char)) # 4.(6)
mean(degree(ff.10.proj.char)) # 8
mean(degree(ff.30.proj.char)) # 12.64
mean(degree(ff.100.proj.char)) # 19.32727
mean(degree(ff.300.proj.char)) # 24.27615

mean(degree(ux.1.proj.char)) # 6
mean(degree(ux.3.proj.char)) # 8
mean(degree(ux.10.proj.char)) # 14.96552
mean(degree(ux.30.proj.char)) # 17.73134
mean(degree(ux.100.proj.char)) # 23.98649
mean(degree(ux.300.proj.char)) # 47.02845

mean(degree(asm.1.proj.char)) # 5
mean(degree(asm.3.proj.char)) # 6.75
mean(degree(asm.10.proj.char)) # 10.83871
mean(degree(asm.30.proj.char)) # 19.48276
mean(degree(asm.100.proj.char)) # 22.96774
mean(degree(asm.300.proj.char)) # 25.51128

degree(ff.300.proj.char)[order(degree(ff.300.proj.char), decreasing=TRUE)[1:5]]
##HUMAN TORCH/JOHNNY S MR. FANTASTIC/REED R INVISIBLE WOMAN/SUE  THING/BENJAMIN J. GR RICHARDS, FRANKLIN B
##       237                  236                  231                  224                  131


degree(ff.1.proj.char,"HUMAN TORCH/JOHNNY S") # 4
degree(ff.3.proj.char,"HUMAN TORCH/JOHNNY S") # 5
degree(ff.10.proj.char,"HUMAN TORCH/JOHNNY S") # 13
degree(ff.30.proj.char,"HUMAN TORCH/JOHNNY S") # 49
degree(ff.100.proj.char,"HUMAN TORCH/JOHNNY S") # 109
degree(ff.300.proj.char,"HUMAN TORCH/JOHNNY S") # 237

degree(ff.1.proj.char,"MR. FANTASTIC/REED R") # 4
degree(ff.3.proj.char,"MR. FANTASTIC/REED R") # 5
degree(ff.10.proj.char,"MR. FANTASTIC/REED R") # 13
degree(ff.30.proj.char,"MR. FANTASTIC/REED R") # 49
degree(ff.100.proj.char,"MR. FANTASTIC/REED R") # 109
degree(ff.300.proj.char,"MR. FANTASTIC/REED R") # 236

degree(ff.1.proj.char,"INVISIBLE WOMAN/SUE ") # 4
degree(ff.3.proj.char,"INVISIBLE WOMAN/SUE ") # 5
degree(ff.10.proj.char,"INVISIBLE WOMAN/SUE ") # 13
degree(ff.30.proj.char,"INVISIBLE WOMAN/SUE ") # 49
degree(ff.100.proj.char,"INVISIBLE WOMAN/SUE ") # 108
degree(ff.300.proj.char,"INVISIBLE WOMAN/SUE ") # 231

degree(ff.1.proj.char,"THING/BENJAMIN J. GR") # 4
degree(ff.3.proj.char,"THING/BENJAMIN J. GR") # 5
degree(ff.10.proj.char,"THING/BENJAMIN J. GR") # 13
degree(ff.30.proj.char,"THING/BENJAMIN J. GR") # 49
degree(ff.100.proj.char,"THING/BENJAMIN J. GR") # 109
degree(ff.300.proj.char,"THING/BENJAMIN J. GR") # 224

degree(ff.1.proj.char,"RICHARDS, FRANKLIN B") # is not in first issues
degree(ff.3.proj.char,"RICHARDS, FRANKLIN B") # is not in these issues
degree(ff.10.proj.char,"RICHARDS, FRANKLIN B") # is not in these issues
degree(ff.30.proj.char,"RICHARDS, FRANKLIN B") # is not in these issues
degree(ff.100.proj.char,"RICHARDS, FRANKLIN B") # 25
degree(ff.300.proj.char,"RICHARDS, FRANKLIN B") # 131

degree(ux.300.proj.char)[order(degree(ux.300.proj.char), decreasing=TRUE)[1:5]]
##CYCLOPS/SCOTT SUMMER STORM/ORORO MUNROE S     WOLVERINE/LOGAN  COLOSSUS II/PETER RA PROFESSOR X/CHARLES
##       361                  358                     355                352                  330


degree(ux.1.proj.char,"CYCLOPS/SCOTT SUMMER") # 6
degree(ux.3.proj.char,"CYCLOPS/SCOTT SUMMER") # 9
```

```
degree(ux.10.proj.char,"CYCLOPS/SCOTT SUMMER") # 28
degree(ux.30.proj.char,"CYCLOPS/SCOTT SUMMER") # 66
degree(ux.100.proj.char,"CYCLOPS/SCOTT SUMMER") # 146
degree(ux.300.proj.char,"CYCLOPS/SCOTT SUMMER") # 361

degree(ux.1.proj.char,"STORM/ORORO MUNROE S") # is not in the first issues
degree(ux.3.proj.char,"STORM/ORORO MUNROE S") # is not in these issues
degree(ux.10.proj.char,"STORM/ORORO MUNROE S") # is not in these issues
degree(ux.30.proj.char,"STORM/ORORO MUNROE S") # is not in these issues
degree(ux.100.proj.char,"STORM/ORORO MUNROE S") # 51
degree(ux.300.proj.char,"STORM/ORORO MUNROE S") # 358

degree(ux.1.proj.char,"WOLVERINE/LOGAN ") # is not in the first issues
degree(ux.3.proj.char,"WOLVERINE/LOGAN ") # is not in these issues
degree(ux.10.proj.char,"WOLVERINE/LOGAN ") # is not in these issues
degree(ux.30.proj.char,"WOLVERINE/LOGAN ") # is not in these issues
degree(ux.100.proj.char,"WOLVERINE/LOGAN ") # 51
degree(ux.300.proj.char,"WOLVERINE/LOGAN ") # 355

degree(ux.1.proj.char,"COLOSSUS II/PETER RA") # is not in the first issues
degree(ux.3.proj.char,"COLOSSUS II/PETER RA") # is not in these issues
degree(ux.10.proj.char,"COLOSSUS II/PETER RA") # is not in these issues
degree(ux.30.proj.char,"COLOSSUS II/PETER RA") # is not in these issues
degree(ux.100.proj.char,"COLOSSUS II/PETER RA") # 51
degree(ux.300.proj.char,"COLOSSUS II/PETER RA") # 352

degree(ux.1.proj.char,"PROFESSOR X/CHARLES ") # 6
degree(ux.3.proj.char,"PROFESSOR X/CHARLES ") # 9
degree(ux.10.proj.char,"PROFESSOR X/CHARLES ") # 28
degree(ux.30.proj.char,"PROFESSOR X/CHARLES ") # 66
degree(ux.100.proj.char,"PROFESSOR X/CHARLES ") # 121
degree(ux.300.proj.char,"PROFESSOR X/CHARLES ") # 330

degree(asm.300.proj.char)[order(degree(asm.300.proj.char), decreasing=TRUE)[1:5]]
##SPIDER-MAN/PETER PAR    JAMESON, J. JONAH          PARKER, MAY       ROBERTSON, JOE   LEEDS, BETTY BRANT
##        265                 197                 167                  167                  163


degree(asm.1.proj.char,"SPIDER-MAN/PETER PAR") # 9
degree(asm.3.proj.char,"SPIDER-MAN/PETER PAR") # 15
degree(asm.10.proj.char,"SPIDER-MAN/PETER PAR") # 30
degree(asm.30.proj.char,"SPIDER-MAN/PETER PAR") # 57
degree(asm.100.proj.char,"SPIDER-MAN/PETER PAR") # 92
degree(asm.300.proj.char,"SPIDER-MAN/PETER PAR") # 265

degree(asm.1.proj.char,"JAMESON, J. JONAH") # 5
degree(asm.3.proj.char,"JAMESON, J. JONAH") # 12
degree(asm.10.proj.char,"JAMESON, J. JONAH") # 28
degree(asm.30.proj.char,"JAMESON, J. JONAH") # 56
degree(asm.100.proj.char,"JAMESON, J. JONAH") # 90
degree(asm.300.proj.char,"JAMESON, J. JONAH") # 197

degree(asm.1.proj.char,"PARKER, MAY") # 5
degree(asm.3.proj.char,"PARKER, MAY") # 8
degree(asm.10.proj.char,"PARKER, MAY") # 25
degree(asm.30.proj.char,"PARKER, MAY") # 55
degree(asm.100.proj.char,"PARKER, MAY") # 75
degree(asm.300.proj.char,"PARKER, MAY") # 167

degree(asm.1.proj.char,"ROBERTSON, JOE") # is not in the first issues
degree(asm.3.proj.char,"ROBERTSON, JOE") # is not in these issues
degree(ams.10.proj.char,"ROBERTSON, JOE") # is not in these issues
degree(asm.30.proj.char,"ROBERTSON, JOE") # is not in these issues
degree(asm.100.proj.char,"ROBERTSON, JOE") # 47
degree(asm.300.proj.char,"ROBERTSON, JOE") # 167

degree(asm.1.proj.char,"LEEDS, BETTY BRANT") # is not in the first issues
degree(asm.3.proj.char,"LEEDS, BETTY BRANT") # is not in these issues
degree(asm.10.proj.char,"LEEDS, BETTY BRANT") # 21
degree(asm.30.proj.char,"LEEDS, BETTY BRANT") # 52
degree(asm.100.proj.char,"LEEDS, BETTY BRANT") # 72
degree(asm.300.proj.char,"LEEDS, BETTY BRANT") # 163

#### 5 ####

diameter(ff.1.proj.char) # 1
diameter(ff.3.proj.char) # 2
diameter(ff.10.proj.char) # 3
diameter(ff.30.proj.char) # 5
diameter(ff.100.proj.char) # 5
diameter(ff.300.proj.char) # 6

diameter(ux.1.proj.char) # 1
diameter(ux.3.proj.char) # 2
diameter(ux.10.proj.char) # 3
diameter(ux.30.proj.char) # 5
diameter(ux.100.proj.char) # 5
diameter(ux.300.proj.char) # 5

diameter(asm.1.proj.char) # 2
diameter(asm.3.proj.char) # 3
diameter(asm.10.proj.char) # 3
diameter(asm.30.proj.char) # 3
diameter(asm.100.proj.char) # 4
diameter(asm.300.proj.char) # 4

#### 6 ####

### k-core maximal FF

core.ff.1 <- coreness(ff.1.proj.char)
core.ff.3 <- coreness(ff.3.proj.char)
core.ff.10 <- coreness(ff.10.proj.char)
core.ff.30 <- coreness(ff.30.proj.char)
core.ff.100 <- coreness(ff.100.proj.char)
core.ff.300 <- coreness(ff.300.proj.char)
```

```
max.core.ff.1 <- max(coreness(ff.1.proj.char)) # 4 this is k in k-core
max.core.ff.3 <- max(coreness(ff.3.proj.char)) # 4
max.core.ff.10 <- max(coreness(ff.10.proj.char)) # 7
max.core.ff.30 <- max(coreness(ff.30.proj.char)) # 12
max.core.ff.100 <- max(coreness(ff.100.proj.char)) # 21
max.core.ff.300 <- max(coreness(ff.300.proj.char)) # 23

vertices.max.core.ff.1 <- which(core.ff.1 == max.core.ff.1)
kcore.ff.1 <- induced.subgraph(graph=ff.1.proj.char, vids=vertices.max.core.ff.1)
vertices.max.core.ff.3 <- which(core.ff.3 == max.core.ff.3)
kcore.ff.3 <- induced.subgraph(graph=ff.3.proj.char, vids=vertices.max.core.ff.3)
vertices.max.core.ff.10 <- which(core.ff.10 == max.core.ff.10)
kcore.ff.10 <- induced.subgraph(graph=ff.10.proj.char, vids=vertices.max.core.ff.10)
vertices.max.core.ff.30 <- which(core.ff.30 == max.core.ff.30)
kcore.ff.30 <- induced.subgraph(graph=ff.30.proj.char, vids=vertices.max.core.ff.30)
vertices.max.core.ff.100 <- which(core.ff.100 == max.core.ff.100)
kcore.ff.100 <- induced.subgraph(graph=ff.100.proj.char, vids=vertices.max.core.ff.100)
vertices.max.core.ff.300 <- which(core.ff.300 == max.core.ff.300)
kcore.ff.300 <- induced.subgraph(graph=ff.300.proj.char, vids=vertices.max.core.ff.300)

vcount(kcore.ff.1) # 5
vcount(kcore.ff.3) # 6
vcount(kcore.ff.10) # 8
vcount(kcore.ff.30) # 13
vcount(kcore.ff.100) # 22
vcount(kcore.ff.300) # 35

v.ff.1 <- vertex_attr(kcore.ff.1)$name # composition of the k-core
v.ff.3 <- vertex_attr(kcore.ff.3)$name
v.ff.10 <- vertex_attr(kcore.ff.10)$name
v.ff.30 <- vertex_attr(kcore.ff.30)$name
v.ff.100 <- vertex_attr(kcore.ff.100)$name
v.ff.300 <- vertex_attr(kcore.ff.300)$name

### k-core maximal UX

core.ux.1 <- coreness(ux.1.proj.char)
core.ux.3 <- coreness(ux.3.proj.char)
core.ux.10 <- coreness(ux.10.proj.char)
core.ux.30 <- coreness(ux.30.proj.char)
core.ux.100 <- coreness(ux.100.proj.char)
core.ux.300 <- coreness(ux.300.proj.char)

max.core.ux.1 <- max(coreness(ux.1.proj.char)) # 6 this is k in k-core
max.core.ux.3 <- max(coreness(ux.3.proj.char)) # 7
max.core.ux.10 <- max(coreness(ux.10.proj.char)) # 13
max.core.ux.30 <- max(coreness(ux.30.proj.char)) # 14
max.core.ux.100 <- max(coreness(ux.100.proj.char)) # 24
max.core.ux.300 <- max(coreness(ux.300.proj.char)) # 40

vertices.max.core.ux.1 <- which(core.ux.1 == max.core.ux.1)
kcore.ux.1 <- induced.subgraph(graph=ux.1.proj.char, vids=vertices.max.core.ux.1)
vertices.max.core.ux.3 <- which(core.ux.3 == max.core.ux.3)
kcore.ux.3 <- induced.subgraph(graph=ux.3.proj.char, vids=vertices.max.core.ux.3)
vertices.max.core.ux.10 <- which(core.ux.10 == max.core.ux.10)
kcore.ux.10 <- induced.subgraph(graph=ux.10.proj.char, vids=vertices.max.core.ux.10)
vertices.max.core.ux.30 <- which(core.ux.30 == max.core.ux.30)
kcore.ux.30 <- induced.subgraph(graph=ux.30.proj.char, vids=vertices.max.core.ux.30)
vertices.max.core.ux.100 <- which(core.ux.100 == max.core.ux.100)
kcore.ux.100 <- induced.subgraph(graph=ux.100.proj.char, vids=vertices.max.core.ux.100)
vertices.max.core.ux.300 <- which(core.ux.300 == max.core.ux.300)
kcore.ux.300 <- induced.subgraph(graph=ux.300.proj.char, vids=vertices.max.core.ux.300)

vcount(kcore.ux.1) # 7
vcount(kcore.ux.3) # 8
vcount(kcore.ux.10) # 14
vcount(kcore.ux.30) # 15
vcount(kcore.ux.100) # 42
vcount(kcore.ux.300) # 118

v.ux.1 <- vertex_attr(kcore.ux.1)$name # composition of the k-core
v.ix.3 <- vertex_attr(kcore.ux.3)$name
v.ux.10 <- vertex_attr(kcore.ux.10)$name
v.ux.30 <- vertex_attr(kcore.ux.30)$name
v.ux.100 <- vertex_attr(kcore.ux.100)$name
v.ux.300 <- vertex_attr(kcore.ux.300)$name

### k-core maximal ASM

core.asm.1 <- coreness(asm.1.proj.char)
core.asm.3 <- coreness(asm.3.proj.char)
core.asm.10 <- coreness(asm.10.proj.char)
core.asm.30 <- coreness(asm.30.proj.char)
core.asm.100 <- coreness(asm.100.proj.char)
core.asm.300 <- coreness(asm.300.proj.char)

max.core.asm.1 <- max(coreness(asm.1.proj.char)) # 5 this is k in k-core
max.core.asm.3 <- max(coreness(asm.3.proj.char)) # 6
max.core.asm.10 <- max(coreness(asm.10.proj.char)) # 9
max.core.asm.30 <- max(coreness(asm.30.proj.char)) # 23
max.core.asm.100 <- max(coreness(asm.100.proj.char)) # 23
max.core.asm.300 <- max(coreness(asm.300.proj.char)) # 24

vertices.max.core.asm.1 <- which(core.asm.1 == max.core.asm.1)
kcore.asm.1 <- induced.subgraph(graph=asm.1.proj.char, vids=vertices.max.core.asm.1)
vertices.max.core.asm.3 <- which(core.asm.3 == max.core.asm.3)
kcore.asm.3 <- induced.subgraph(graph=asm.3.proj.char, vids=vertices.max.core.asm.3)
vertices.max.core.asm.10 <- which(core.asm.10 == max.core.asm.10)
kcore.asm.10 <- induced.subgraph(graph=asm.10.proj.char, vids=vertices.max.core.asm.10)
vertices.max.core.asm.30 <- which(core.asm.30 == max.core.asm.30)
kcore.asm.30 <- induced.subgraph(graph=asm.30.proj.char, vids=vertices.max.core.asm.30)
vertices.max.core.asm.100 <- which(core.asm.100 == max.core.asm.100)
kcore.asm.100 <- induced.subgraph(graph=asm.100.proj.char, vids=vertices.max.core.asm.100)
vertices.max.core.asm.300 <- which(core.asm.300 == max.core.asm.300)
kcore.asm.300 <- induced.subgraph(graph=asm.300.proj.char, vids=vertices.max.core.asm.300)
```

```
vcount(kcore.asm.1) # 6
vcount(kcore.asm.3) # 7
vcount(kcore.asm.10) # 15
vcount(kcore.asm.30) # 24
vcount(kcore.asm.100) # 24
vcount(kcore.asm.300) # 54

v.asm.1 <- vertex_attr(kcore.asm.1)$name # composition of the k-core
v.asm.3 <- vertex_attr(kcore.asm.3)$name
v.asm.10 <- vertex_attr(kcore.asm.10)$name
v.asm.30 <- vertex_attr(kcore.asm.30)$name
v.asm.100 <- vertex_attr(kcore.asm.100)$name
v.asm.300 <- vertex_attr(kcore.asm.300)$name

### plotting k as a function of number of issues

k.asm <- c(5,6,9,23,23,24)
k.ux <- c(6,7,13,14,24,40)
k.ff <- c(4,4,7,12,21,23)
issues <- c(1,3,10,30,100,300)

plot(issues, k.ff,type = "l", xlab = "issue", ylab = "k", main = "k against issues - FF")
plot(issues, k.ux,type = "l", xlab = "issue", ylab = "k", main = "k against issues - UX")
plot(issues, k.asm,type = "l", xlab = "issue", ylab = "k", main = "k against issues - ASM")

### plotting degrees as a function of the group of issues for wolverine, cyclops, joe robinson,spider-man, franklin b and johnny s

d.wolverine <- c(0,0,0,0,51, 355)
d.cyclops <- c(6,9,28,66,146,361)
d.joerob <- c(0,0,0,0,47,167)
d.spidman <- c(9,15,30,57,92,265)
d.franklin <- c(0,0,0,0,25,131)
d.johnny <- c(4,5,13,49,109,237)

plot(issues, d.wolverine, type = "l", xlab = "issue", ylab = "degree", main = "degree against issues - Wolverine")
plot(issues, d.cyclops, type = "l", xlab = "issue", ylab = "degree", main = "degree against issues - Cyclops")
plot(issues, d.joerob, type = "l", xlab = "issue", ylab = "degree", main = "degree against issues - Robby")
plot(issues, d.spidman, type = "l", xlab = "issue", ylab = "degree", main = "degree against issues - Spider-Man")
plot(issues, d.franklin, type = "l", xlab = "issue", ylab = "degree", main = "degree against issues - Franklin B")
plot(issues, d.johnny, type = "l", xlab = "issue", ylab = "degree", main = "degree against issues - Johnny S")

### plotting dynamics of the average degree

deg.ff <- c(4, 4.67, 8, 12.64, 19.33, 24.28)
deg.ux <- c(6, 8, 14.97, 17.73, 23.97, 47.03)
deg.asm <- c(5, 6.75, 10.84, 19.48, 22.97, 25.51)

plot(issues,deg.ff, type = "l", xlab = "issue", ylab = "av.degree", main = "av.degree against issues - FF")
plot(issues,deg.ux, type = "l", xlab = "issue", ylab = "av.degree", main = "av.degree against issues - UX")
plot(issues,deg.asm, type = "l", xlab = "issue", ylab = "av.degree", main = "av.degree against issues - ASM")

### plotting evolution of the size of maximal k-core over time

size.k.ff <- c(5,6,8,13,22,35)
size.k.ux <- c(7,8,14,15,42,118)
size.k.asm <- c(6,7,15,24,24,54)

plot(issues, size.k.ff,type = "l",xlab = "issue",ylab = "number of vertices in max k-core",main = "vertices in max k-core against issues - FF")
plot(issues, size.k.ux,type = "l",xlab = "issue",ylab = "number of vertices in max k-core",main = "vertices in max k-core against issues - UX")
plot(issues, size.k.asm,type = "l",xlab = "issue",ylab = "number of vertices in max k-core",main = "vertices in max k-core against issues - ASM")
```