Stratiegic Alliance

Zander Bonnet

March 26, 2025

Given an unweighted adjacency matrix, we must convert the data into a graph. We can use the igrpah package in R to accomplish this.

```r
library(readxl)
library(igraph)
```

```
##
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

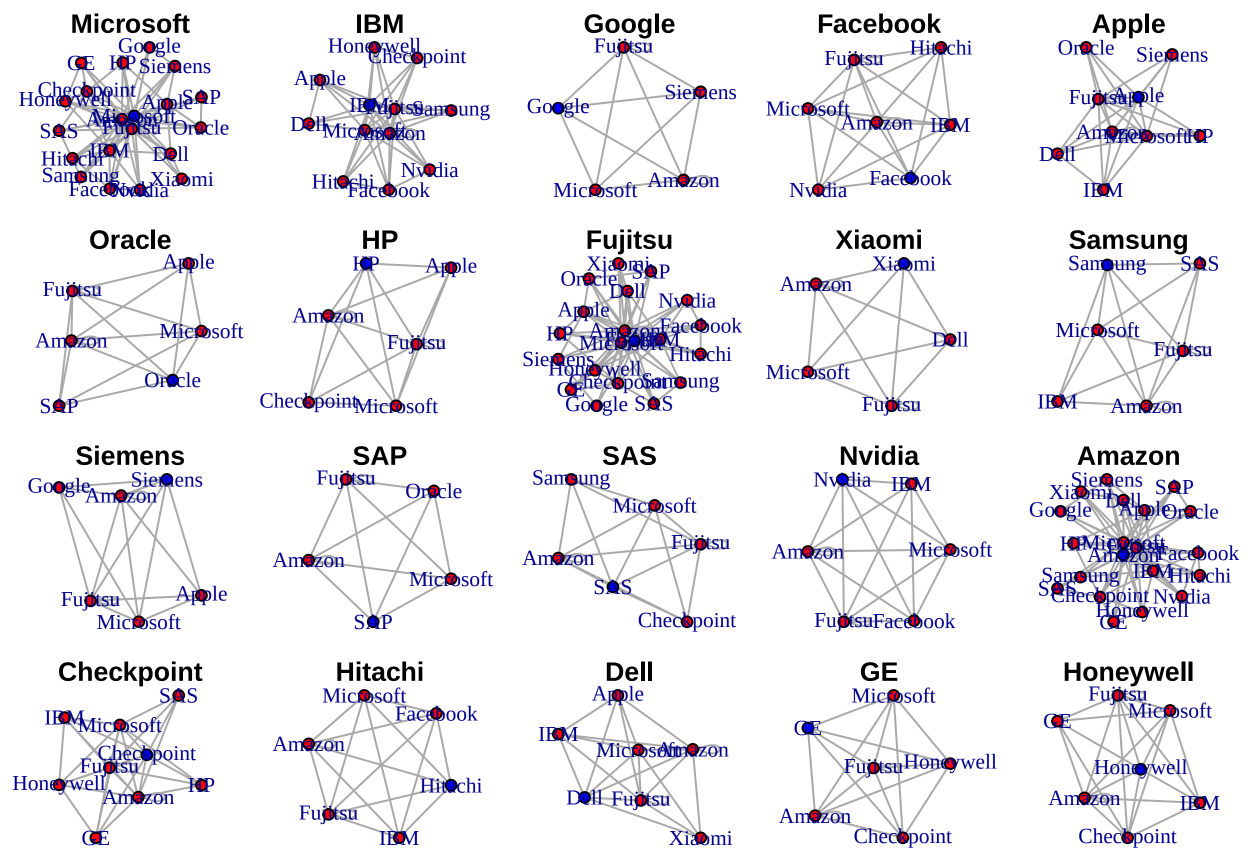
```r
library(ggplot2)
data <-read_xlsx('/Users/zanderbonnet/Desktop/GCU/DSC-570/Topic 4/FinancialTies.xlsx')

mat <- data.matrix(data[,-1])
rownames(mat) <- data[,1]$`0`

(g <- graph_from_adjacency_matrix(mat, mode = 'undirected', weighted = FALSE))
```

```
## IGRAPH 11718e9 UN-- 20 79 --
## + attr: name (v/c)
## + edges from 11718e9 (vertex names):
##  [1] Microsoft--IBM        Microsoft--Google     Microsoft--Facebook
##  [4] Microsoft--Apple      Microsoft--Oracle     Microsoft--HP
##  [7] Microsoft--Fujitsu    Microsoft--Xiaomi     Microsoft--Samsung
## [10] Microsoft--Siemens    Microsoft--SAP        Microsoft--SAS
## [13] Microsoft--Nvidia     Microsoft--Amazon     Microsoft--Checkpoint
## [16] Microsoft--Hitachi    Microsoft--Dell       Microsoft--GE
## [19] Microsoft--Honeywell  IBM      --Facebook   IBM      --Apple
## [22] IBM      --Fujitsu    IBM      --Samsung    IBM      --Nvidia
## + ... omitted several edges
```

Now that we have the data in the proper format I will first visualize each of the nodes as their own egocentric graphs. This will allow us to see what immediate connections each company has, and potentially identify early indicators of strong relationships between companies.

```
egos <- make_ego_graph(g)
ego_dens <- c()
ego_cen <- c()
```

```
par(mfrow = c(4,5),mar=c(1,1,1,1))
for(i in 1:20){
  V(egos[[i]])$color <- NA
  V(egos[[i]])$color<- 'red'
  temp <- which(V(egos[[i]])$name == V(g)$name[i])
  V(egos[[i]])$color[temp]<- 'blue'
  plot(egos[[i]], main = V(g)$name[i], layout = layout_with_fr)
  ego_dens[i] <- edge_density(egos[[i]], loops = TRUE)
  ego_cen[i] <- centr_degree(egos[[i]])$centralization
}
```



Looking at these plots, we can see that there are groups that appear to be very strongly connected, as well as some that have lots of connections. Google and Siemens appear to be two strongly connected companies as they share a connection and share many of the same connections with other companies.

```
d <- data.frame(ego_density = ego_dens, ego_centrality = ego_cen, degree = degree(g), betweenness = bet
     closeness = closeness(g), prestige = eigen_centrality(g)$vector,row.names = V(g)$name)
d[c('ego_density', 'degree')]
```

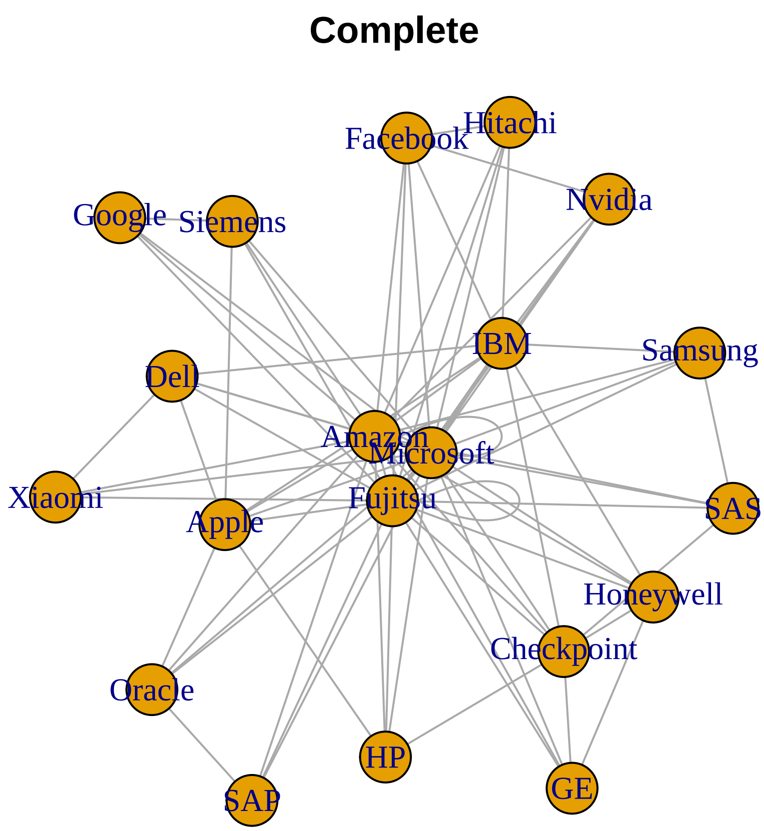
```
##           ego_density degree
## Microsoft   0.3761905     19
```

```
## IBM          0.5641026     11
## Google       0.8000000      4
## Facebook     0.7857143      6
## Apple        0.6444444      8
## Oracle       0.7619048      5
## HP           0.7619048      5
## Fujitsu      0.3761905     21
## Xiaomi       0.8000000      4
## Samsung      0.7619048      5
## Siemens      0.7619048      5
## SAP          0.8000000      4
## SAS          0.7619048      5
## Nvidia       0.8095238      5
## Amazon       0.3761905     21
## Checkpoint   0.6666667      8
## Hitachi      0.8095238      5
## Dell         0.7500000      6
## GE           0.8095238      5
## Honeywell    0.7857143      6
```

From these graphs, we aim to identify a couple of things. We want to find the density of the egocentric graph so that we can see if the local network has a strong connection, and we also want the degree of the company. The density measures the proportion of edges in the graph compared to the maximum number of edges that could be in the graph. So, a larger number would mean that a company is strongly connected within its network. The degree, on the other hand the degree is the number of edges leaving the company. From this we can see if a company is popular.

Looking deeper into the data, we can look at the density of the egocentric graph and the centrality of the companies. From this, we can see that Google has a very high density (.8) and a low degree(4). This means that Google does not have a lot of connections, but its network is very connected. This is a similar story for Siemens as they have a high density (.76) and a low degree (5). Looking at these two, we can see that they are well connected within their small networks but do not have nearly as many connections as other companies do. To explore this, we will look at the complete graph and other metrics that will help explain the companies more effectively.

```r
par(mar = c(0,0,2,0))
plot(g, layout = layout_with_fr, main = 'Complete')
```



From the initial plot, we can start to see that Amazon, Microsoft, and Fujitsu are very connected within the network. This is reinforced by the fact that they each have a degree of 19 or higher. There is still a lot more information that can be extracted from the companies, though. The graph can also be improved so we can get a clearer picture of what is going on.

```
print(d)
```

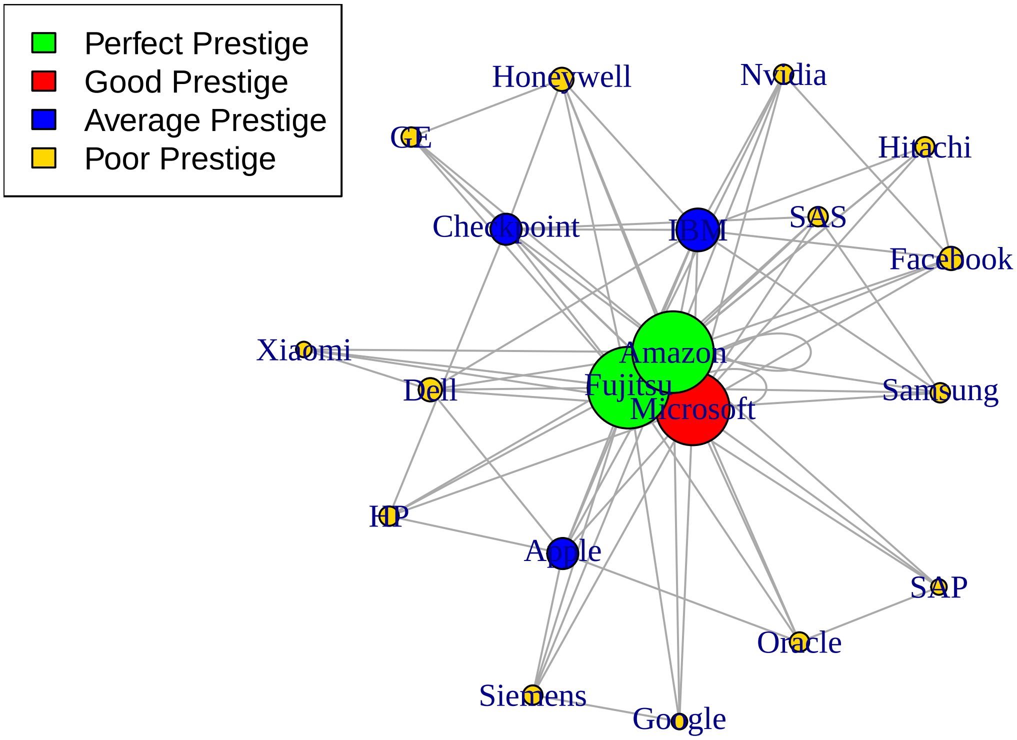
```
##            ego_density ego_centrality degree betweenness  closeness  prestige
## Microsoft    0.3761905      0.6894737     19       33.70 0.05263158 0.8239499
## IBM          0.5641026      0.5151515     11        5.85 0.03703704 0.5868615
## Google       0.8000000      0.3000000      4        0.00 0.02941176 0.3061152
## Facebook     0.7857143      0.2857143      6        0.20 0.03125000 0.4002260
## Apple        0.6444444      0.4444444      8        2.20 0.03333333 0.4704596
## Oracle       0.7619048      0.3333333      5        0.25 0.03030303 0.3475274
## HP           0.7619048      0.3333333      5        0.20 0.03030303 0.3635811
## Fujitsu      0.3761905      0.6894737     21       33.70 0.05263158 1.0000000
## Xiaomi       0.8000000      0.3000000      4        0.00 0.02941176 0.3116340
## Samsung      0.7619048      0.3333333      5        0.20 0.03030303 0.3633111
## Siemens      0.7619048      0.3333333      5        0.25 0.03030303 0.3475274
## SAP          0.8000000      0.3000000      4        0.00 0.02941176 0.3061152
## SAS          0.7619048      0.3333333      5        0.20 0.03030303 0.3532390
## Nvidia       0.8095238      0.2666667      5        0.00 0.03030303 0.3678464
## Amazon       0.3761905      0.6894737     21       33.70 0.05263158 1.0000000
## Checkpoint   0.6666667      0.4166667      8        1.85 0.03333333 0.4724384
## Hitachi      0.8095238      0.2666667      5        0.00 0.03030303 0.3678464
## Dell         0.7500000      0.3333333      6        0.50 0.03125000 0.4047047
```

```
## GE            0.8095238      0.2666667      5         0.00 0.03030303 0.3576818
## Honeywell     0.7857143      0.2857143      6         0.20 0.03125000 0.4093403
```

Some key metrics that we can look at to further explain the relationships between the companies are betweenness, closeness, and prestige (also referred to as eigenvector centrality). These centrality measures will give us information like what nodes act as a "bridge' to other clusters of companies, as well as companies that are close to all other companies in the network. Betweenness centrality is a measure of how often a node is on the shortest path to another group of nodes. A high value in this would mean that that specific node is a"bridge" and connects node groups. This can be used to bring nodes together that wouldn't usually interact. Closeness centrality measures the average distance from one node to all other nodes in the network. A large value in this would show that this node is very well connected and would be useful if we needed to get information out as fast as possible. Prestige centrality measures the importance of a node based on its connections and the connections of its neighbors. So if a node is connected to a highly influential node, the prestige value would rise, as well as if it was a highly influential node itself. We can use this to find the nodes that best connect to the rest of the network. This would allow us to find a node that is not only popular, so other nodes want to connect to it, but also can bridge information to other popular nodes, so they can continue to spread the information. An example of this would be if you had two unique social media influences that followed each other but did not necessarily have large overlapping followings. One of them could post something, and then the other could re-post it, exposing it to two unique networks of people. In this example, both of the influencers would have high prestige centrality.

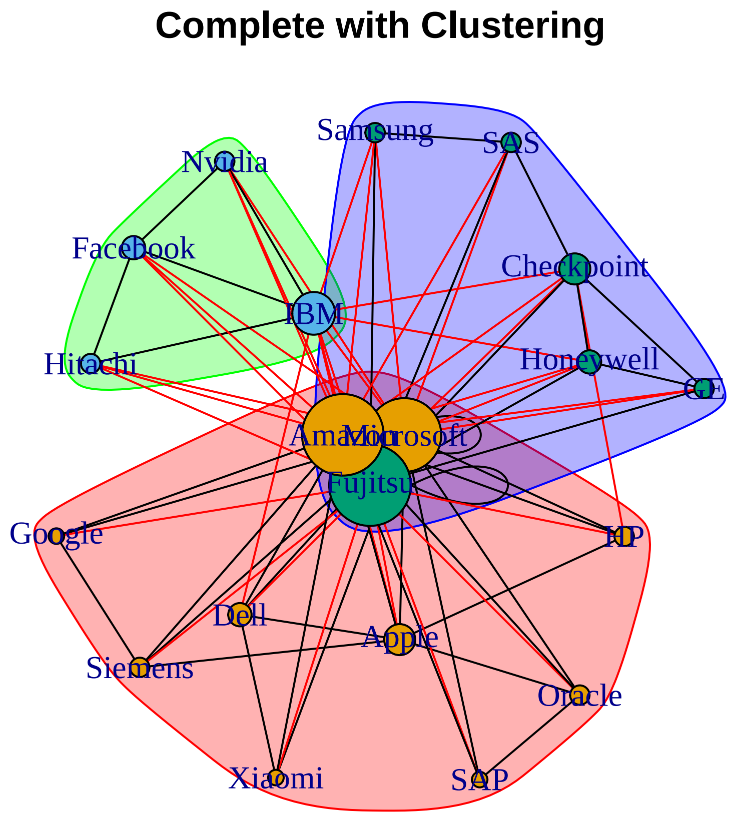
```r
V(g)$color <- NA
V(g)$color[d$prestige > .9] <- 'green'
V(g)$color[d$prestige  <= .9] <- 'red'
V(g)$color[d$prestige < .6] <- 'blue'
V(g)$color[d$prestige < .45] <- 'gold'

par(mar = c(0,0,2,0))
V(g)$size <- d$degree *1.2
plot(g, layout = layout_with_fr, main = 'Complete With Prestige and Scaled by Degree')
legend(legend = c('Perfect Prestige','Good Prestige', 'Average Prestige', 'Poor Prestige'), x= 'topleft
```

Complete With Prestige and Scaled by Degree



Looking at this graph, we can see that the big three are Fujitsu, Microsoft, and Amazon. This was to be expected as they all have a connection to every company in the network. The only thing holding down Microsoft is the fact that it doesn't have a self-connection. Otherwise, it is also perfect. We do gather some useful information, though, as Apple, Checkpoint, and IBM do have a higher level of prestige. All three of these companies have above average betweenness, closeness and have a degree of 8 or higher. This shows that even though they are not the major players in the network, they are acting as the second level of importance. To take a closer look, we can use a clustering algorithm to find clusters within the graph. The fact that Apple, Checkpoint, and IBM stand out may indicate that they are acting as a bridge to smaller clusters of companies, so this should give us more information on that.

```
par(mar = c(0,0,2,0))
clust <- cluster_optimal(g)
plot(clust,g, layout = layout_with_fr, main = 'Complete with Clustering')
```



As we saw and hypothesized based on the previous graph, we can see that IBM, Apple, and Checkpoint are acting as bridges for their respective clusters. Because of the very big three companies connecting to everyone, there are a lot of companies that link to companies out of their cluster, but there is fairly minimal crossover outside of that.

```
dismat <- distances(g,1)
for(i in 2:length(V(g)$name)){
  dismat <- rbind(dismat, distances(g,i))
}
(dismat <- data.frame(dismat))
```

```
##             Microsoft IBM Google Facebook Apple Oracle HP Fujitsu Xiaomi Samsung
## Microsoft           0   1      1        1     1      1  1       1      1       1
## IBM                 1   0      2        1     1      2  2       1      2       1
## Google              1   2      0        2     2      2  2       1      2       2
## Facebook            1   1      2        0     2      2  2       1      2       2
## Apple               1   1      2        2     0      1  1       1      2       2
## Oracle              1   2      2        2     1      0  2       1      2       2
## HP                  1   2      2        2     1      2  0       1      2       2
## Fujitsu             1   1      1        1     1      1  1       0      1       1
## Xiaomi              1   2      2        2     2      2  2       1      0       2
## Samsung             1   1      2        2     2      2  2       1      2       0
## Siemens             1   2      1        2     1      2  2       1      2       2
## SAP                 1   2      2        2     2      1  2       1      2       2
## SAS                 1   2      2        2     2      2  2       1      2       1
## Nvidia              1   1      2        1     2      2  2       1      2       2
```

```
## Amazon              1   1      1        1     1      1  1       1      1       1
## Checkpoint          1   1      2        2     2      2  1       1      2       2
## Hitachi             1   1      2        1     2      2  2       1      2       2
## Dell                1   1      2        2     1      2  2       1      1       2
## GE                  1   2      2        2     2      2  2       1      2       2
## Honeywell           1   1      2        2     2      2  2       1      2       2
##            Siemens SAP SAS Nvidia Amazon Checkpoint Hitachi Dell GE Honeywell
## Microsoft        1   1   1      1      1          1       1    1  1         1
## IBM              2   2   2      1      1          1       1    1  2         1
## Google           1   2   2      2      1          2       2    2  2         2
## Facebook         2   2   2      1      1          2       1    2  2         2
## Apple            1   2   2      2      1          2       2    1  2         2
## Oracle           2   1   2      2      1          2       2    2  2         2
## HP               2   2   2      2      1          1       2    2  2         2
## Fujitsu          1   1   1      1      1          1       1    1  1         1
## Xiaomi           2   2   2      2      1          2       2    1  2         2
## Samsung          2   2   1      2      1          2       2    2  2         2
## Siemens          0   2   2      2      1          2       2    2  2         2
## SAP              2   0   2      2      1          2       2    2  2         2
## SAS              2   2   0      2      1          1       2    2  2         2
## Nvidia           2   2   2      0      1          2       2    2  2         2
## Amazon           1   1   1      1      0          1       1    1  1         1
## Checkpoint       2   2   1      2      1          0       2    2  1         1
## Hitachi          2   2   2      2      1          2       0    2  2         2
## Dell             2   2   2      2      1          2       2    0  2         2
## GE               2   2   2      2      1          1       2    2  0         1
## Honeywell        2   2   2      2      1          1       2    2  1         0
```

```r
sum(dismat==0)-length(g); max(dismat)
```

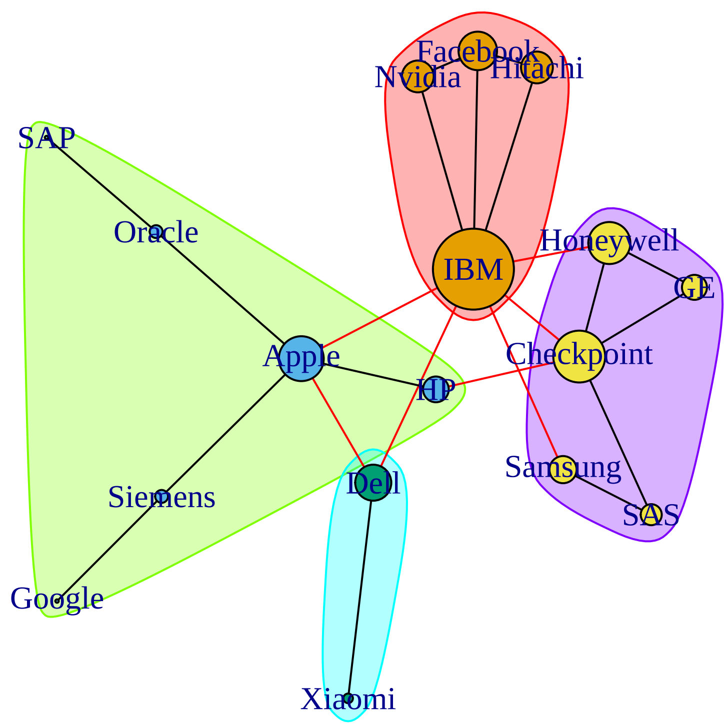
```
## [1] 0
```

```
## [1] 2
```

Because of the main three companies there companies we can see that there is very little separation between any one node. With the max being two steps to any company. There are also no companies that have no way of reaching each other. What if the main three companies didn't exist, though, or even prevent other companies from using them as a middleman? How would the network react?

```r
par(mar = c(0,0,2,0))
g2 <- delete_vertices(g, c('Microsoft', 'Fujitsu', 'Amazon'))
V(g2)$size <- eigen_centrality(g2)$vector*25
clust2 <- cluster_optimal(g2)
plot(clust2,g2, layout = layout_with_fr, main = 'Complete with Clustering Sized by Prestige')
```

Complete with Clustering Sized by Prestige



If we remove the big three companies altogether, it is easier to see that Apple, IBM, and Checkpoint really are the bridge points to their clusters. We also discover that Xiaomi and Dell are potentially off on their own. When sized by prestige, we can see that IBM is the most important company as it has connections to the bridge of all clusters. We can also see that Google, SAP, and Xiaomi are now very insignificant as they are not providing any additional paths to take.

```
dismat2 <- distances(g2,1)
for(i in 2:length(V(g2)$name)){
  dismat2 <- rbind(dismat2, distances(g2,i))
}
(dismat2 <- data.frame(dismat2))
```

```
##            IBM Google Facebook Apple Oracle HP Xiaomi Samsung Siemens SAP SAS
## IBM          0      3        1     1      2  2      2       1       2   3   2
## Google       3      0        4     2      3  3      4       4       1   4   5
## Facebook     1      4        0     2      3  3      3       2       3   4   3
## Apple        1      2        2     0      1  1      2       2       1   2   3
## Oracle       2      3        3     1      0  2      3       3       2   1   4
## HP           2      3        3     1      2  0      3       3       2   3   2
## Xiaomi       2      4        3     2      3  3      0       3       3   4   4
## Samsung      1      4        2     2      3  3      3       0       3   4   1
## Siemens      2      1        3     1      2  2      3       3       0   3   4
## SAP          3      4        4     2      1  3      4       4       3   0   5
## SAS          2      5        3     3      4  2      4       1       4   5   0
## Nvidia       1      4        1     2      3  3      3       2       3   4   3
## Checkpoint   1      4        2     2      3  1      3       2       3   4   1
```

```
## Hitachi      1      4        1      2       3   3      3        2        3   4   3
## Dell         1      3        2      1       2   2      1        2        2   3   3
## GE           2      5        3      3       4   2      4        3        4   5   2
## Honeywell    1      4        2      2       3   2      3        2        3   4   2
##            Nvidia Checkpoint Hitachi Dell GE Honeywell
## IBM             1          1       1    1  2         1
## Google          4          4       4    3  5         4
## Facebook        1          2       1    2  3         2
## Apple           2          2       2    1  3         2
## Oracle          3          3       3    2  4         3
## HP              3          1       3    2  2         2
## Xiaomi          3          3       3    1  4         3
## Samsung         2          2       2    2  3         2
## Siemens         3          3       3    2  4         3
## SAP             4          4       4    3  5         4
## SAS             3          1       3    3  2         2
## Nvidia          0          2       2    2  3         2
## Checkpoint      2          0       2    2  1         1
## Hitachi         2          2       0    2  3         2
## Dell            2          2       2    0  3         2
## GE              3          1       3    3  0         1
## Honeywell       2          1       2    2  1         0
```

```
sort(closeness(g2));sum(dismat2==0)-length(g2); max(dismat2)
```

```
##     Google        SAP     Xiaomi         GE        SAS     Oracle    Siemens
## 0.01754386 0.01754386 0.02083333 0.02083333 0.02127660 0.02380952 0.02380952
##     Nvidia    Hitachi   Facebook    Samsung         HP  Honeywell Checkpoint
## 0.02500000 0.02500000 0.02564103 0.02564103 0.02702703 0.02777778 0.02941176
##       Dell      Apple        IBM
## 0.03030303 0.03448276 0.03846154

## [1] 0

## [1] 5
```

Removing the big three has disconnected a lot of companies from the rest of the network. The most affected are companies like Google and SAP, who saw their closeness centrality significantly drop as they now have multiple companies that they are 4 or 5 steps away from. There are still no companies that are completely separate from each other, though.

#3 - Interlocking Boards

Interlocking boards provide unique challenges as they can cause ethical issues and violate anti-trust laws. Having members on multiple boards can lead to unethical business practices as the members could share confidential information between companies and potentially lead to insider trading. There is also the chance that two competing companies that share board members might elect to collaborate and violate anti-monopoly laws. In this example, if IBM and Apple were to share board member, they would influence a majority of companies in the network. As we have shown, both Apple and IBM act as bridge companies for business ties for two of the largest clusters of companies. Another way that we could potentially see interlocking boards in this data is the companies that have business ties with themselves. In this data, Amazon and Fujitsu have edges that point back to themselves. This could be because a board member serves on both boards, so that member would have business ties in both directions.

#4 - Strategic Alliances

Google and or SAP would very wise to connect to IBM. IBM is a big player in this network, so connecting to it would greatly expand its reach. This is especially true when you look at them outside of the big 3 companies. If those three were to limit collaboration or even sever ties, Google and SAP would be left isolated from the group. A connection to IBM would give them the ability to get around any roadblocks put up by the big three companies. This would also be beneficial for IBM as it gives them easier access to two powerhouses in the tech world. Apple would want to prevent this, though, as it would lower its importance. They would no longer be the bridge to IBM for these companies and would no longer be the big player in their cluster, and most likely be replaced by HP as the most important node in their cluster.

#5 - Strong and Weak Ties

The two strongest ties are Amazon and Fujitsu. They are the strongest because they have many connections within the network and act as the middleman for a lot more connections within the network. The two weakest ties would be SAP and Google, as they only have one connection each outside of the big three. This puts them at risk of being disconnected from the network, as they have so few ties to it. They also are not contributing to the network, so companies may not want to collaborate with them as they are not bringing anything to the table as far as network connections.

#6 - Holes in Network

In the complete network there are no apparent holes because the main three are connected to all companies in the network. This creates redundancy as if one cuts them off there will there is anothe way to reach every node within 2 streps. If we do not account for the big three companies though there there are a couple of structural holes. Apple, Dell, and IBM are holes as if they were to disappear all of their connections would be left isolated.

#7 - Challenges

As it stands companies like SAP, Google and Xiaomi face big challenges in that it would be very expensive for them to netowrk with companies like SAS as they are 4-5 steps away. This could prove to be costly as SAS excels in creating data solutions with AI and in an ever developing world AI is at teh forefront. Yes, these companies are able to create their own AI products but that can be expensive. If they had an easier connection and business relationship they might be able to outsource this project and save money. There also is very string competition between the big three companies as they are all veying to be the most relevant in the network.

#8 - View on Informational Networks

Informational networks are a great way to visualize how comapnienns align themselves with strategic aliances and allows ine to find openings in the network. For example we saw that there is a structural whole between IBM and the rest of the network. We can then develop a company or product that could compete with IBM. This could potentially pull other companies to partener with you instead of IBM. The same is true for apple. These companies that act as bridges to other nodes are great places to exploit to introduce variety. It is also a great tool to look at at risk companies. If we look at cases like Google and SAP if we were able to recreate what they do and create a relationship with another company we could replace them in the network.