Homework 1

Edgar Chicurel

2023-04-17

Intro and Network

The U.S. government agency websites (2018) contains data for website traffic in over 50 U.S. states. The data captures the relationships between different government websites, as well as the frequency with which they are accessed from other government sites. I will focus on the state of **Iowa** for this exercise. This information will be visualized using network analysis techniques to gain insights into patterns of website traffic and connectivity between different government sites.

Website Link

Load Libraries

```
library(igraph)
library(tidyverse)
library(ggraph)

set.seed(6437)
```

Read the data

```
nodes <- read_delim("C:/Users/usuario/Desktop/Master/Social_Network_Analysis/Homeworks/iowa/nodes.txt",
  mutate(
    NodeId = as.character(NodeId)
  )
edges <- read_delim("C:/Users/usuario/Desktop/Master/Social_Network_Analysis/Homeworks/iowa/edges.txt",
   mutate(
    Src = as.character(Src),
    Trg = as.character(Trg)
  )
```

Generate the graph

In this first part the network is generated by using the 2 data frames provided with links and nodes. The network is directed because it shows how on link is clicked from a page to other.

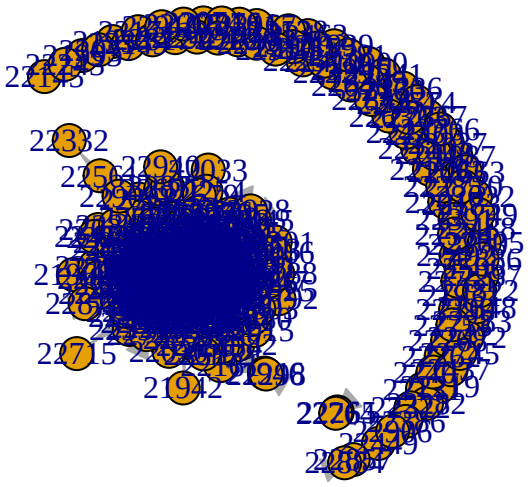
```
g <- graph.data.frame(edges, vertices = nodes, directed = TRUE)
E(g)$weight <- edges$LinkCount

g
```

```
## IGRAPH 8441de7 DNW- 541 3129 --
## + attr: name (v/c), NodeURL (v/c), NodeName (v/c), Year (v/n), pages
## | (v/n), LinkCount (e/n), RelLinkCount (e/n), weight (e/n)
## + edges from 8441de7 (vertex names):
##  [1] 22528->21986 22528->22505 22528->22412 22528->22125 22528->22350
##  [6] 22528->22545 22528->22868 22528->22805 22528->22519 22528->22042
## [11] 22528->22670 22530->22381 22533->22664 22533->22505 22533->22667
## [16] 22533->22125 22533->22868 22533->22714 22535->22042 22540->22560
## [21] 22540->21986 22540->22470 22540->22440 22540->21961 22540->22519
## [26] 22540->22125 22540->22350 22540->21935 22540->22441 22540->22505
## [31] 22540->22042 22540->22941 22540->22206 22540->22143 22541->22877
## + ... omitted several edges
```

It can be seen that the network consists of 541 **nodes** and 3129 **edges.**

A first visualization of the network. It can be seen that there is a giant component and many not connected nodes in the corners.

```
plot(g)
```



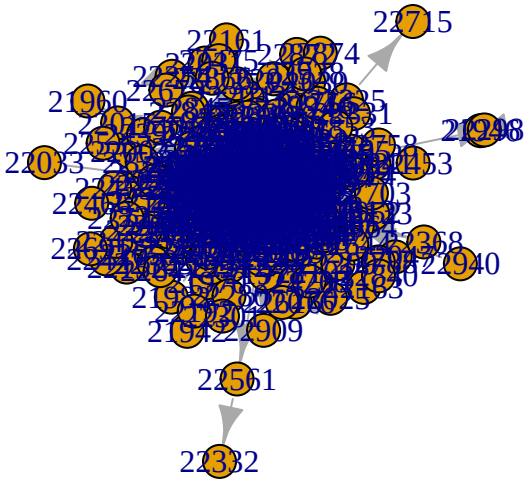
Giant component

Going to work on this for the analysis because there are a lot of disconnected nodes and the relevant part worth analyzing is the giant component in the middle.

```
cc <- components(g)
head(cc$csize,10)
```

```
##  [1]   1 449   1   1   1   1   1   1   1   1
```

```
GC <- igraph::decompose(g)[[2]]
plot(GC)
```



Graph analysis

Average degree and standard deviation

In this part the average degree and standard deviation of the giant component are calculated and interpreted.

```
degrees <- degree(GC)

# Calculate the average degree
avg_degree <- mean(degrees)
```

```r
# Calculate the standard deviation of the degree
degree_sd <- sd(degrees)

cat("Average Degree:", avg_degree, "\n")
```

```
## Average Degree: 13.92873
```

```r
cat("Degree Standard Deviation:", degree_sd, "\n")
```

```
## Degree Standard Deviation: 24.74891
```

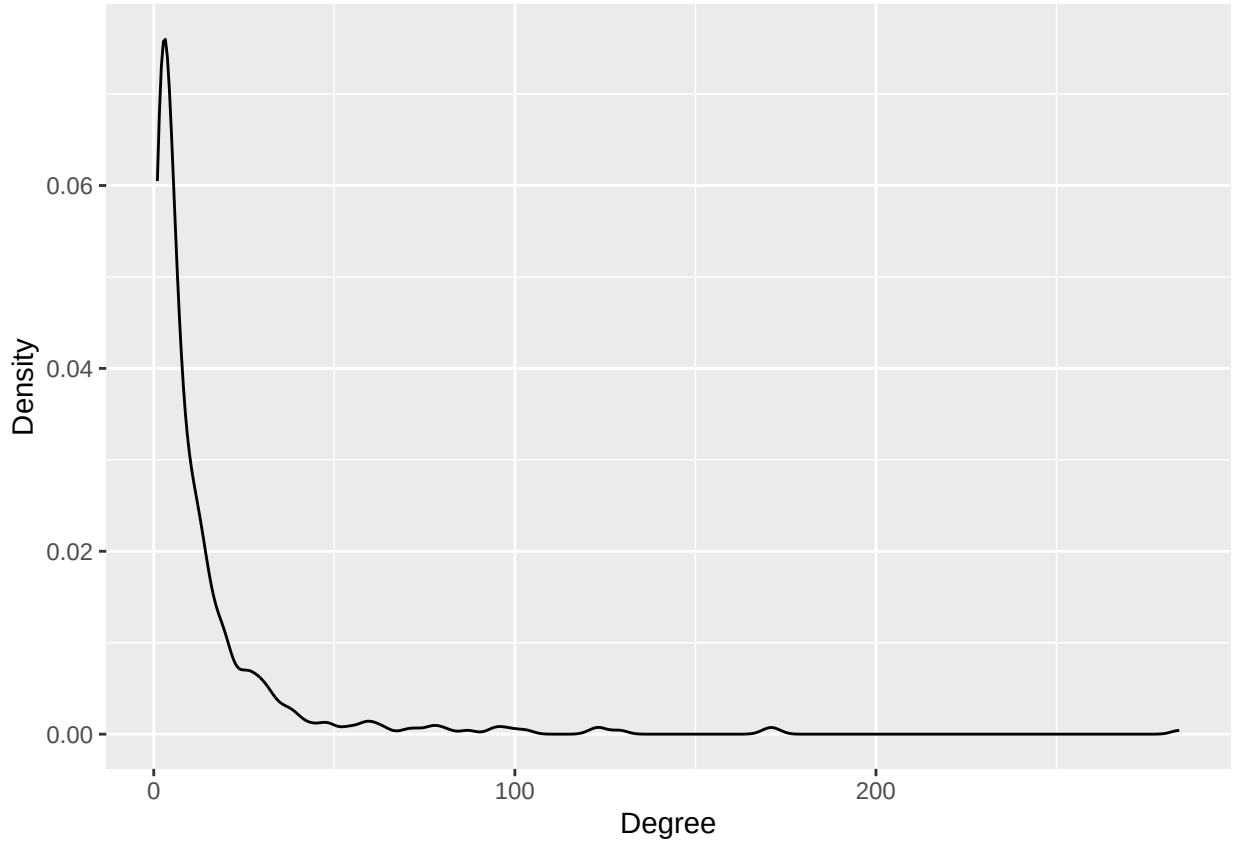
This means that on average, each node in the network is connected to about **14** other nodes.

A higher standard deviation indicates that the degrees of nodes are more spread out, while a lower standard deviation indicates that the degrees of nodes are more clustered around the average degree. In this case, the degree standard deviation is around **24**, which is relatively high. This suggests that there is a lot of variation in the number of connections that nodes have in the network, with some nodes having many connections and others having very few.

Linear distributions

In this next section the linear distribution of the degree is plotted.

```r
ggplot() + geom_density(aes(x=degree(GC,mode="all"))) + labs(x="Degree",y="Density")
```

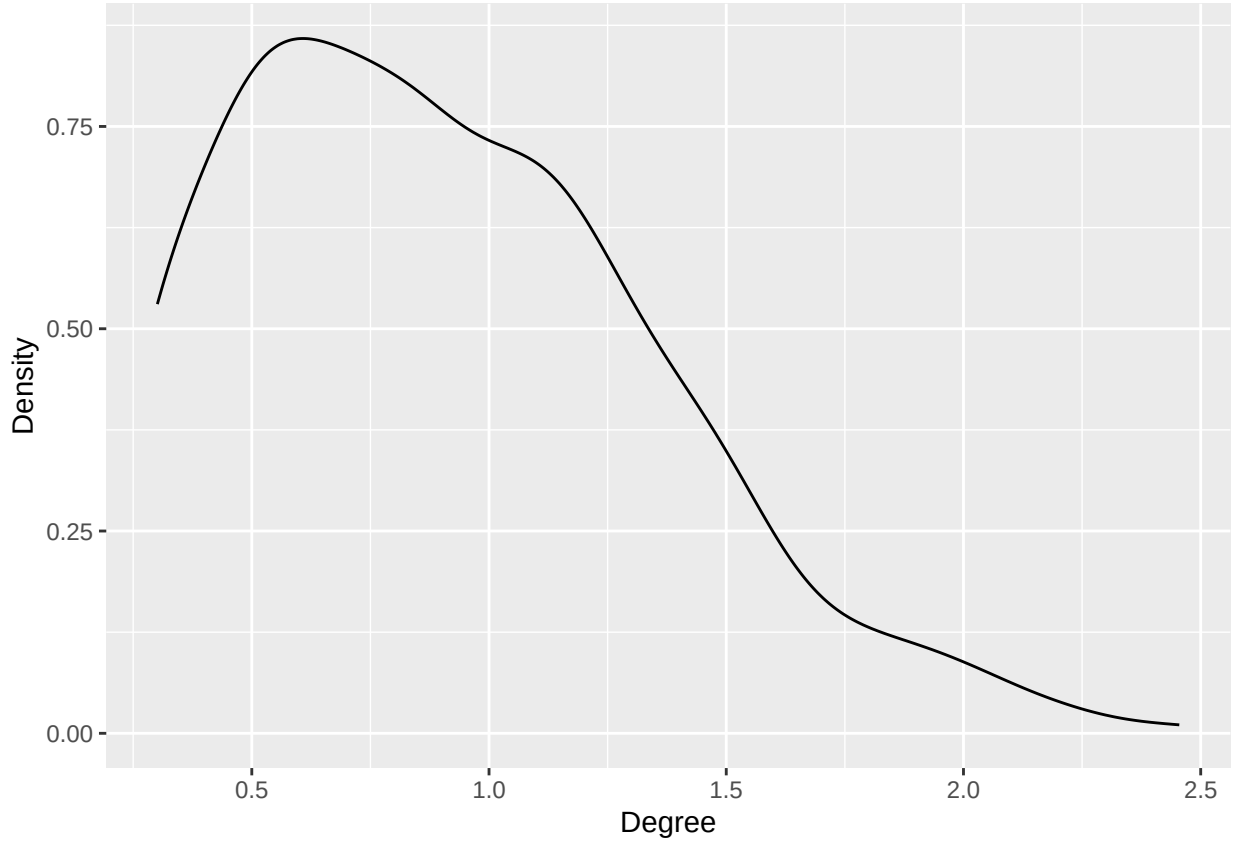


This density draph is very common showing that the majority of the nodes have few links and there are some outliers with more than 200 connections.

Log distribution

Same graph but in log scale.

```
log_degrees <- log10(degrees + 1)
df_log_degrees <- data.frame(Degree = log_degrees, Scale = "Log-Log")

ggplot(df_log_degrees, aes(x = Degree)) + geom_density() + labs(x="Degree",y="Density")
```



More linear but it shows the same thing as the first.

Most Connected Nodes

In this next part, the most connected nodes are extracted from the linear and log distribution.

```
maxlog <- max(log_degrees)
maxlin <- max(degrees)

cat("Degree of the Most Connected Node:", maxlin, "\n")
```

```
## Degree of the Most Connected Node: 284
```

```r
cat("Degree in logs of the Most Connected Node:", maxlog, "\n")
```

```
## Degree in logs of the Most Connected Node: 2.454845
```

It can be seen that there is a node with a lot of conenctions **284** which is the governemnt website from where more links are clicked to access other government websites.

Transivity

In this next part the transitivity of the Giant component is calculated, or as we saw in class the probability of a node forming a traingle.

```r
tran <- transitivity(GC)

cat("The transivity in the network is:", tran, "\n")
```

```
## The transivity in the network is: 0.1775517
```

In this case, the transitivity of the network is a relatively small proportion of possible triangles actually exist in the network, indicating that nodes in the network are not strongly clustered or connected in groups.

Assortatvity

In this section the assortativity is calculated. As we saw in class it is the a measure of the tendency for nodes in a network to be connected to other nodes with similar or dissimilar characteristics.

```r
ass <- assortativity_degree(GC)

cat("The assortativity degree in the network is:", ass, "\n")
```

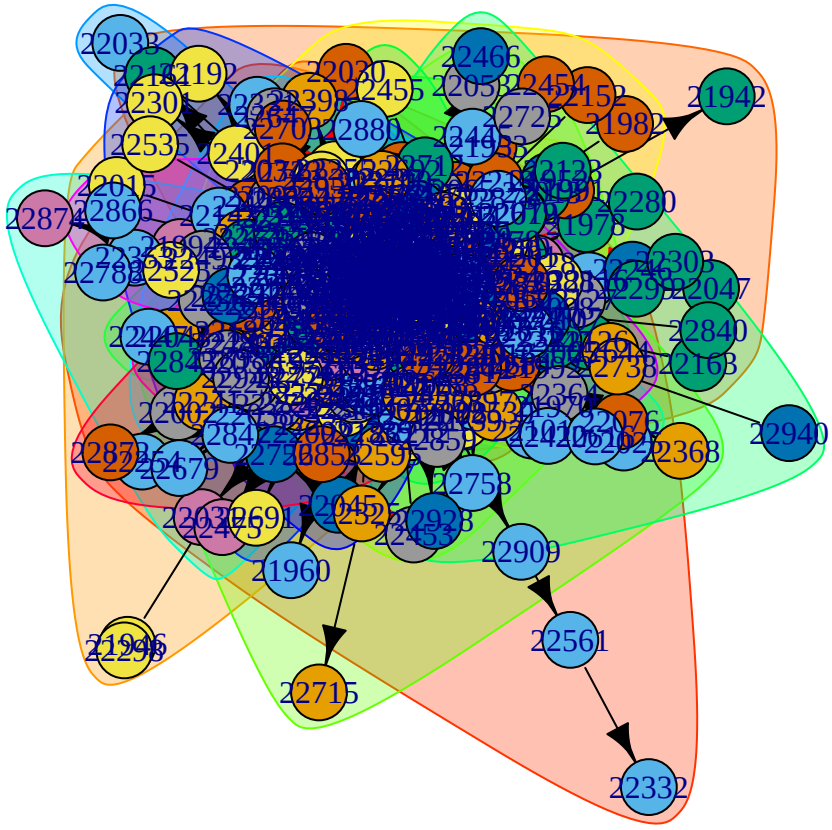
```
## The assortativity degree in the network is: -0.2628465
```

This negative degree assortativity coefficient indicates that nodes with high degree tend to be connected to nodes with low degree, and vice vers not showing really social networks characteristics.

Louvian Method

In this section **The Louvain Method** is calculated for detecting communities or clusters in this network. In order to apply this the network should be read as not directed. For the sake of the analysis going to proceed this way but I could have also used the **Walk Trap Method** and kept the directions.

```r
par(mar=c(0, 0, 0, 0))
louvain <- cluster_louvain(as.undirected(GC), weights = E(g)$weight)
plot(louvain,GC)
```



The network is really dense so it is visually not really appreciate. If filters where used then probably it could be seen better but still going to analyze the clusters.

Communities sizes

The communities of the louvian method are analyzed in this next section.

```
sizes(louvain)
```

```
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
## 24 65 47 27 17 44  7 22 32 20  3 12 13  2 13 30  5  2  2 21  4  3  5  3  6  8
## 27 28 29 30
##  3  2  2  5
```

It can be seen that there are 23 communities 10 of them with a high number of nodes and the rest with low number of nodes. This fact that there are 10 communities with a high number of nodes suggests that there may be some relatively large and relevant group of webpages within the network.

Modularity

```
modul <- modularity(louvain)

cat("The modularity in the network is:", modul, "\n")
```

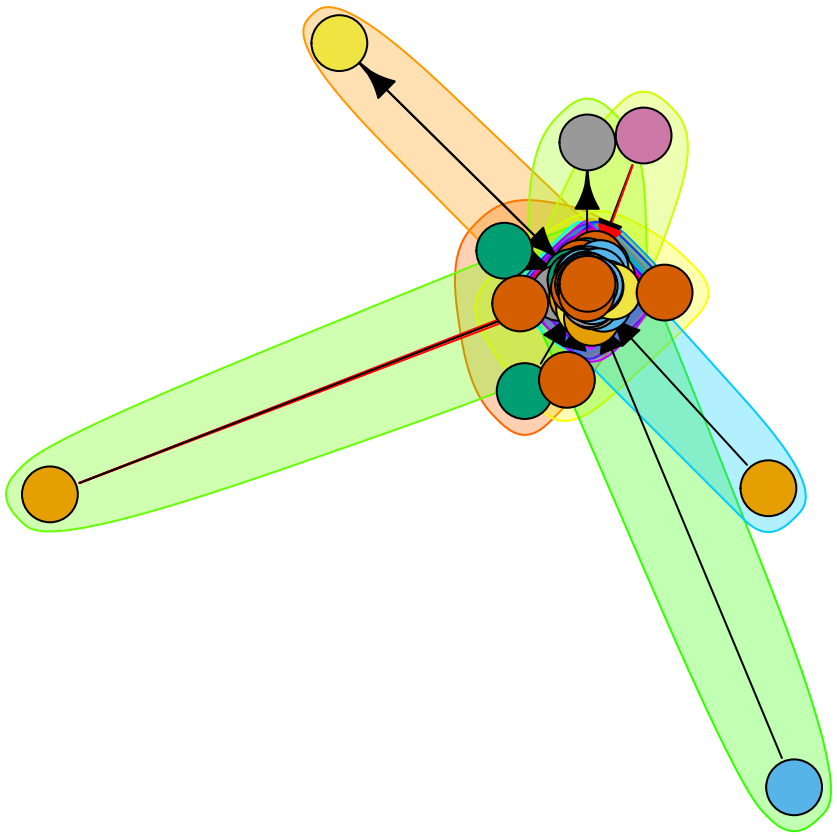
```
## The modularity in the network is: 0.8591668
```

This value is relatively high and suggests that the Louvain method has successfully identified a structure in the network where there is a higher density of edges within communities than between them.

Plot communities

In this next part I tried to plot the communities in a different layout so it can be seen more clearly. There are some communities in the corners and many in the center confirming what I said before.

```
par(mar=c(0, 0, 0, 0))
ll <- layout.kamada.kawai(GC)
plot(louvain,GC,layout=ll,vertex.label="")
```



Test Clustering Coefficient

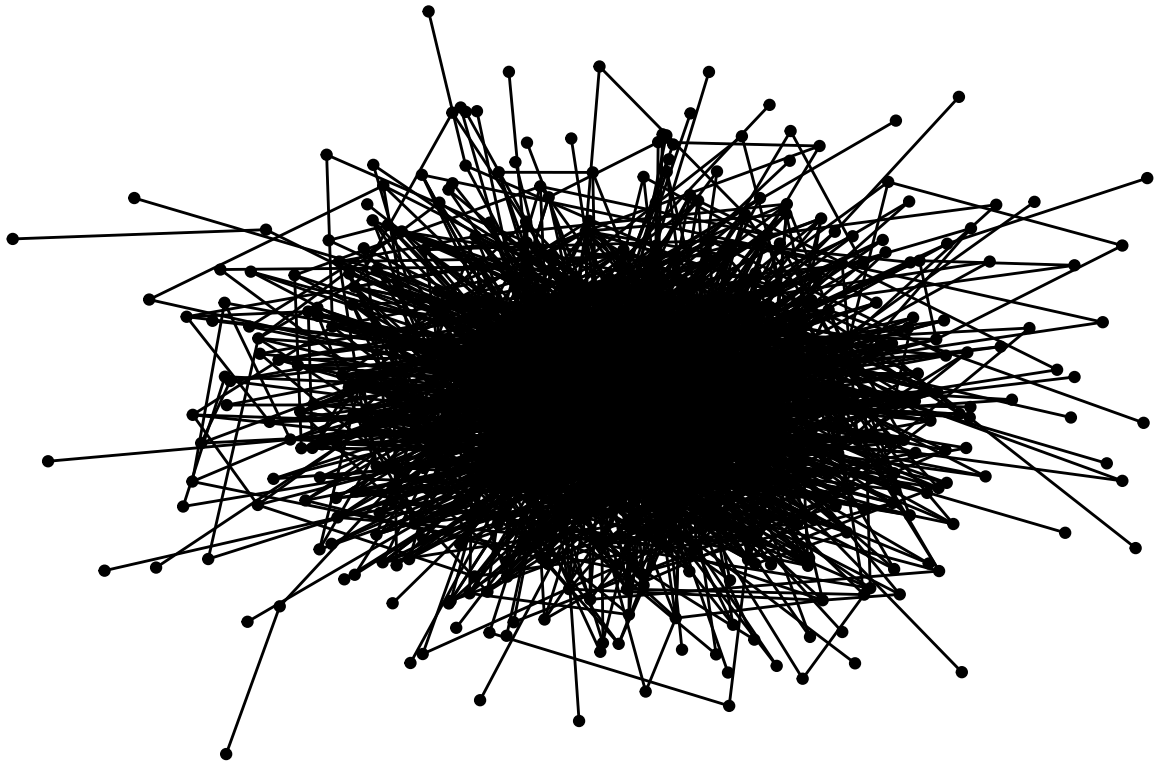
In this next section I tested that the clustering coefficient in the network cannot be statistically explain by a configuration model in which the nodes have the same degree as the original

First a visualization of a **Configuration model** with the same degrees as in the U.S. website network.

```
g_ds <- sample_degseq(degree(GC))

g4 <- ggraph(g_ds,layout="kk")+geom_edge_link()+geom_node_point()+theme_void()+labs(title="Degree Seq."
```

```
g4
```

Degree Seq.



Transitivty Comparisons

Now, going to calculate the transitivity for this model, compare it with the original network and test if it is significantly different.

```
t1 <- transitivity(GC)
t2 <- transitivity(sample_degseq(degree(GC)))


cat("The transivity in the original network is:", t1, "\n")
```

```
## The transivity in the original network is: 0.1775517
```

```
cat("The transivity in the configuration model network is:", t2, "\n")
```

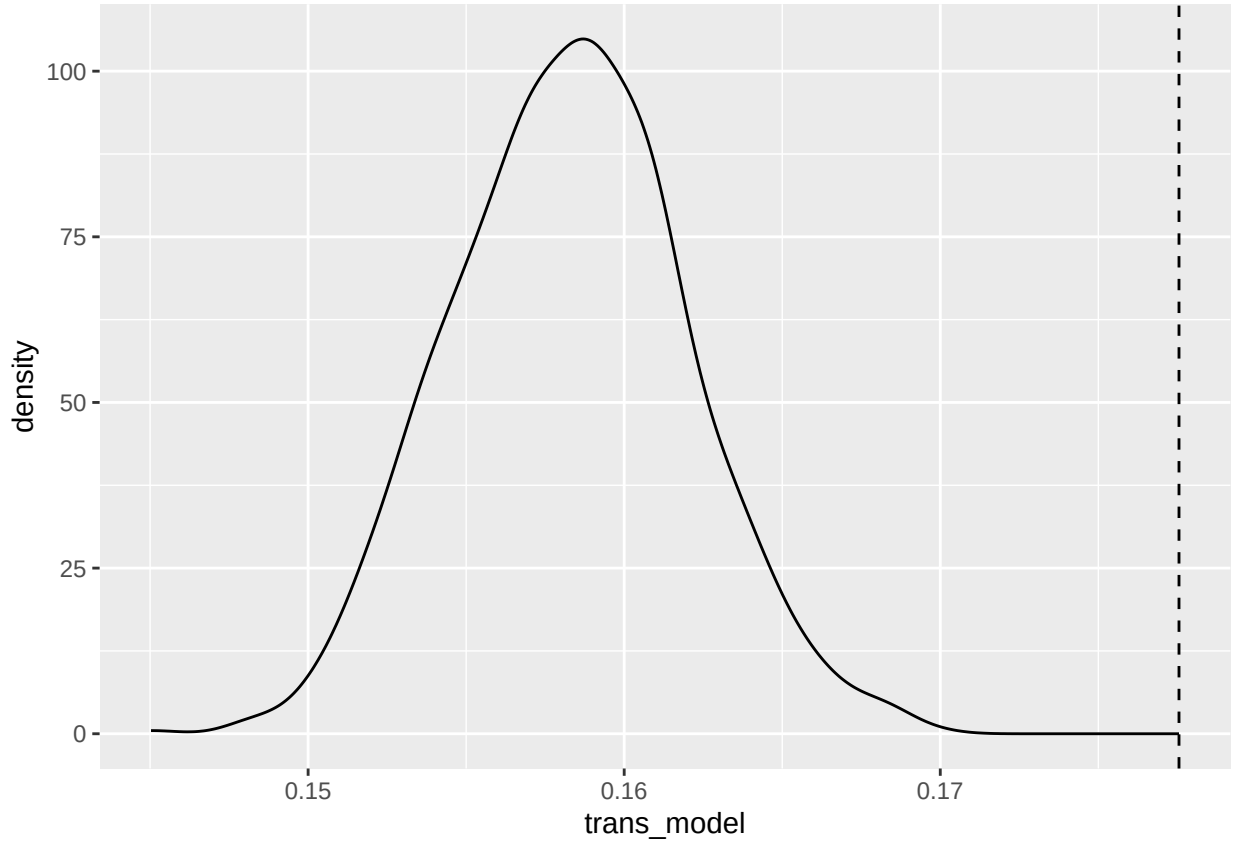
```
## The transivity in the configuration model network is: 0.162622
```

The value seems similar so going to test it.

Test

In order to test this,as in class, I need to generate the distribution of clustering values from the degree-sequence models with the same degree sequence as the U.S. original network. Do it 1000 times

```
trans_model <- replicate(1000,transitivity(sample_degseq(degree(GC))))

ggplot() + geom_density(aes(x=trans_model)) +
  geom_vline(xintercept = transitivity(GC),linetype=2)
```



It can be seen that it is probably not the same based on the density of the transitivity but in next part calculating formally the p-value of our hypothesis.

```
dnorm(transitivity(GC),mean=mean(trans_model),sd=sd(trans_model))
```

```
## [1] 0.0001511243
```

Close to 0 so can reject and assume they are different.

Centrality

In this next part, different measures for calculating centrality is done in order to graph the neighborhood of the **most central** node.

```
cent_degree <- degree(GC,mode="all")
cent_streng <- strength(GC,weights=E(GC)$LinkCount)
cent_betw <- betweenness(GC, directed = F, weights=E(GC)$LinkCount)
cent_page <- page_rank(GC, weights=E(GC)$LinkCount)$vector

table_cent <- data.frame(cent_degree,cent_streng,
           cent_betw,cent_page)
```

Strenght

```
table_cent %>% arrange(-cent_streng) %>% head(5)
```

```
##       cent_degree cent_streng cent_betw    cent_page
## 22125         284       36594 18513.608 0.1737830914
## 22602         172       24792 12652.447 0.0299648889
## 22069          77       17531  3556.846 0.0034625575
## 22592          64       17317  2841.984 0.0003810841
## 22441         129       16569  2703.701 0.0340083776
```

Betweness

```
table_cent %>% arrange(-cent_betw) %>% head(5)
```

```
##       cent_degree cent_streng cent_betw   cent_page
## 22125         284       36594 18513.608 0.173783091
## 22602         172       24792 12652.447 0.029964889
## 22505         170        4048 10997.684 0.041924599
## 22659          81         586  6994.604 0.022032292
## 22553          99         808  6931.115 0.008654494
```

PageRank

```
table_cent %>% arrange(-cent_page) %>% head(5)
```

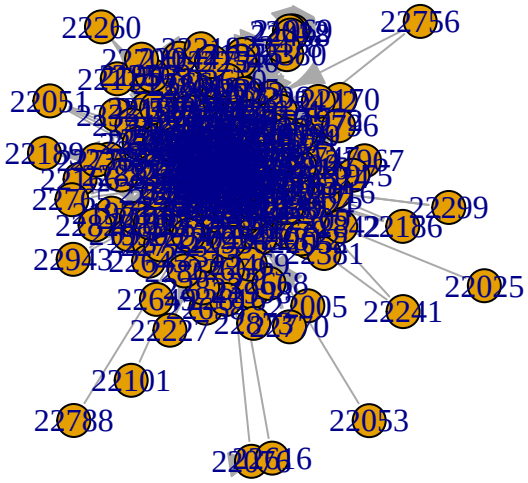
```
##       cent_degree cent_streng cent_betw  cent_page
## 22125         284       36594 18513.608 0.17378309
## 22042         124       11749  4206.743 0.06107679
## 21986         103        3935  4178.508 0.05041249
## 22505         170        4048 10997.684 0.04192460
## 22545          32        1993  1194.054 0.03471218
```

In all measures of centrality, the top or most central node is the website **21125** so going to focus on the visualization for the neighborhood of this specific node in the next section.

Visualization

Visualize the neighborhood of the node with the largest centrality **21125**. First, start with a basic preliminar plot.

```
egoDW <- make_ego_graph(GC,order = 1,"22125")[[1]]
plot(egoDW)
```



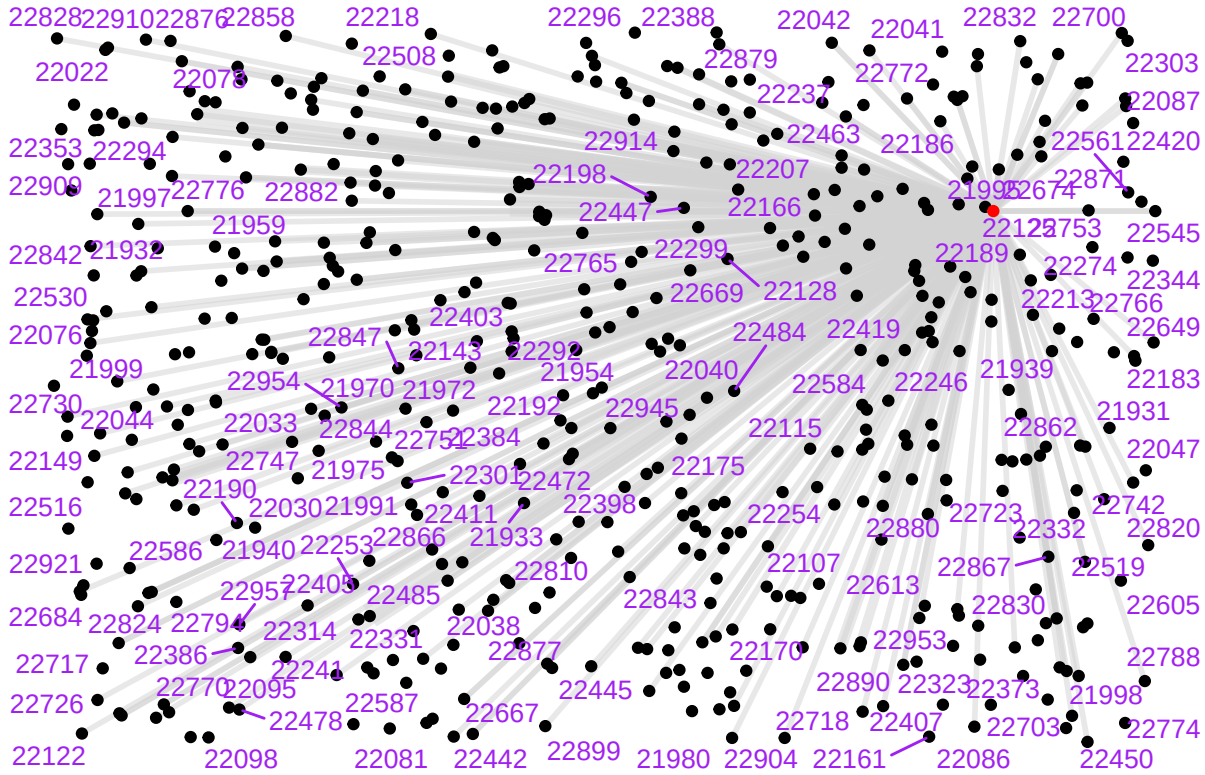
Improved Visualization

Finally, an improved visualization of the neighborhood of the central node was done.

```
edges.filter <- edges %>%
  filter(Src == "22125" | Trg == "22125")

filtergraph <- graph.data.frame(edges.filter, vertices = nodes, directed = F)
E(filtergraph)$weight <- edges.filter$LinkCount


ggraph(filtergraph, layout = "lgl") +
  geom_edge_link(aes(edge_width = LinkCount), color = "lightgrey", alpha = 0.5) +
  geom_node_point(aes(color = ifelse(name == "22125", "red", "black"))) +
  geom_node_text(aes(label = name), size = 3.5, color = "purple", repel = T) +
  scale_color_manual(values = c("black", "red")) +
  labs(title = "22125 neighborhood") +
```

```
theme(legend.position = "none",
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      panel.border = element_blank(),
      panel.background = element_blank())
```

22125 neighborhood



Conclusions

The network analysis has yielded several significant findings. Firstly, the website "22125" stands out as the most influential node in the network based on three measures, indicating its crucial role in the government website network structure. Secondly, the clustering coefficient of the network is not due to random chance but rather influenced by shared characteristics or interests among nodes. Lastly, the network displays high modularity, dividing it into distinct communities or clusters. The Louvain method identified 23 communities, with 10 having a large number of nodes and the remainder with a low number, suggesting the presence of unique subgroups within the network, each with their own distinctive features or connections.