

lab_3

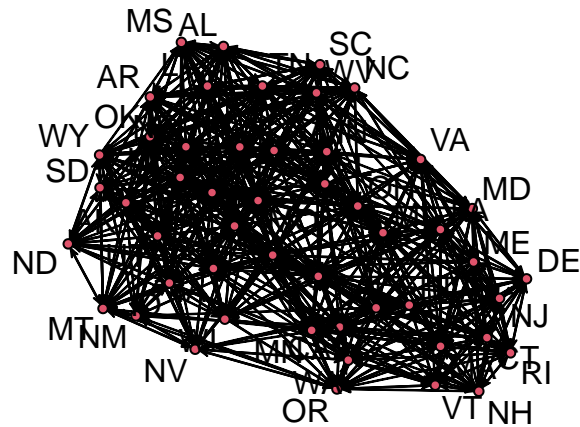
Naomi Liftman

2023-02-28

Question 1

Load `similarity_scores.dta` into R, convert into network object. Plot the network (note- start with `statnet`, then when switching to community detection detach `sna` and `statnet` in a chunk).

```
#load packages
library(rio)
library(statnet)
#load data as a df
sim_as_df <- import("similarity_scores.dta")
#convert to network
sim <- as.network(sim_as_df, mode=c("directed"))
#plot the network
plot.network(sim, label = network.vertex.names(sim))
```



Question 2

Tell me the density of the network, number of nodes, level of reciprocity, whether there are any isolates, and whether it is directed.

The density is 0.0221308, there are 50 nodes, the level of reciprocity is 0.8710204, which tells us it is a directed network and there are no isolates.

```
#summary gives us density and nodes
summary(sim, print.adj=FALSE)
```

```
## Network attributes:
##   vertices = 50
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges = 994
##   missing edges = 0
##   non-missing edges = 994
##   density = 0.4057143
##
## Vertex attributes:
```

```

## vertex.names:
##   character valued attribute
##   50 valid vertex names
##
## Edge attributes:
##
## dif_inc:
##   numeric valued attribute
##   attribute summary:
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      12    2674    5791    6789    9504    25728
##
## dif_log_pop:
##   numeric valued attribute
##   attribute summary:
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.001486 0.401950 0.800069 0.990721 1.430133 4.112955
##
## dif_pop:
##   numeric valued attribute
##   attribute summary:
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1976 1599068 3553909 6148154 7123640 38297952
##
## dif_prof:
##   numeric valued attribute
##   attribute summary:
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.0000 0.0000 1.0000 0.8883 1.0000 3.0000
##
## dif_urb:
##   numeric valued attribute
##   attribute summary:
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.06   4.65   10.32   13.55   18.31   56.29
##
## dif_white:
##   numeric valued attribute
##   attribute summary:
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   5.125  11.950  14.505  21.000  71.600
##
## log_dif_pop:
##   numeric valued attribute
##   attribute summary:
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   7.589  14.285  15.084  14.946  15.779  17.461
##
## same_censusreg:
##   numeric valued attribute
##   attribute summary:
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.0000 0.0000 0.0000 0.4779 1.0000 1.0000
##

```

```
## same_elazar3:
##   numeric valued attribute
##   attribute summary:
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.0000 0.0000 0.0000 0.4628 1.0000 1.0000
##
## same_elazar8:
##   numeric valued attribute
##   attribute summary:
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.0000 0.0000 0.0000 0.1932 0.0000 1.0000
##
## same_leg:
##   numeric valued attribute
##   attribute summary:
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.0000 0.0000 1.0000 0.5513 1.0000 1.0000
##
## sim_score:
##   numeric valued attribute
##   attribute summary:
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.04717 0.06509 0.09629 0.12236 0.16039 0.41667
```

```
#isolates gives us integer for isolates
isolates(sim, diag=FALSE)
```

```
## integer(0)
```

```
#grecip gives us reciprocity
grecip(sim)
```

```
##      Mut
## 0.8710204
```

Question 3

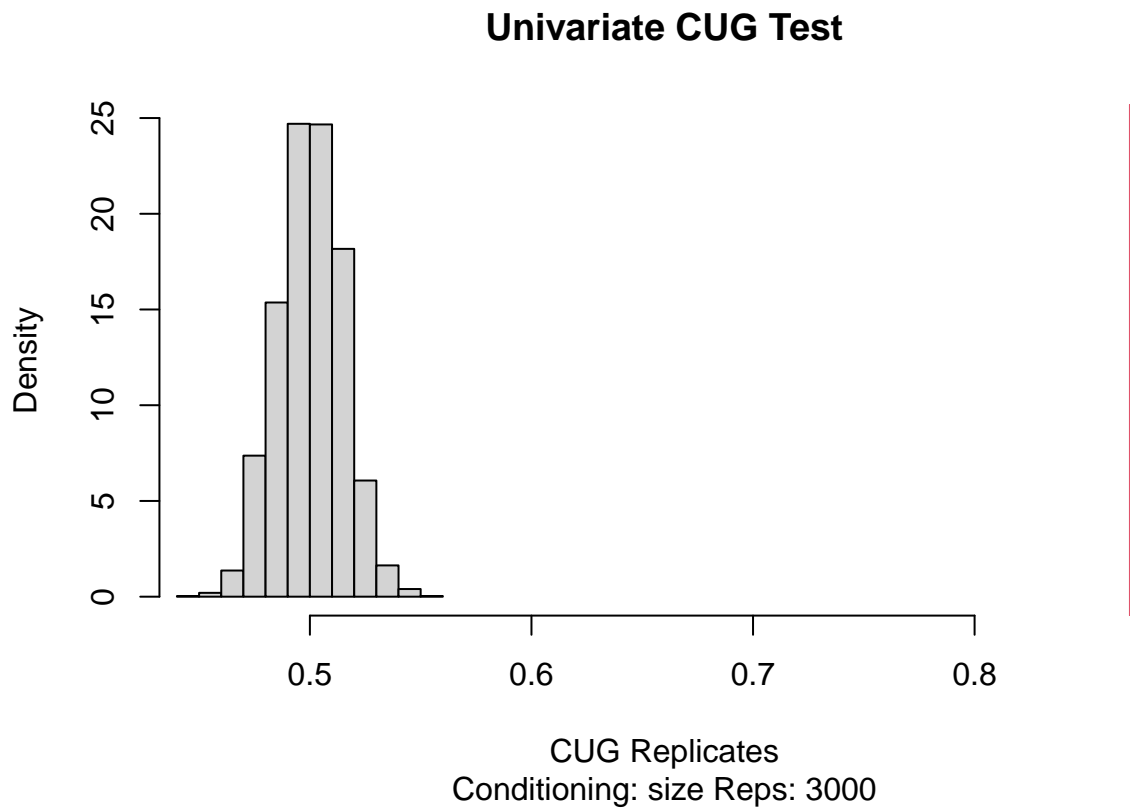
Conduct a cug-test for reciprocity. Make sure you include a plot and let us know what we can conclude about the level of reciprocity observed in the network. Set the number of simulated networks to 3000.

```
#conduct the test
cug_sim_recip <- cug.test(sim, grecip, reps = 3000)
#print out the test
cug_sim_recip
```

```
##
## Univariate Conditional Uniform Graph Test
##
## Conditioning Method: size
```

```
## Graph Type: digraph
## Diagonal Used: FALSE
## Replications: 3000
##
## Observed Value: 0.8710204
## Pr(X>Obs): 0
## Pr(X<=Obs): 1
```

```
#give a pretty graph!
plot(cug_sim_recp)
```



We can conclude that the level of reciprocity is much higher than what we would expect for a random network with the same number of nodes. The probability of this level of reciprocity occurring in a random network of the same size is nearly zero.

Question 4

Conduct a cug-test for density, and condition it by the size of the network. Set the number of simulated networks to 1500.

```
cug.test(sim, gden, cmode="size", reps = 1500)
```

```
##
## Univariate Conditional Uniform Graph Test
```

```
##
## Conditioning Method: size
## Graph Type: digraph
## Diagonal Used: FALSE
## Replications: 1500
##
## Observed Value: 0.4057143
## Pr(X>=Obs): 1
## Pr(X<=Obs): 0
```

Question 5

Detach the packages `statnet` and `sna`. Convert the original dataframe into a network object that `igraph` can use.

```
#detach statnet and sna
detach(package:statnet)
detach(package:sna)
#load igraph
library(igraph)
```

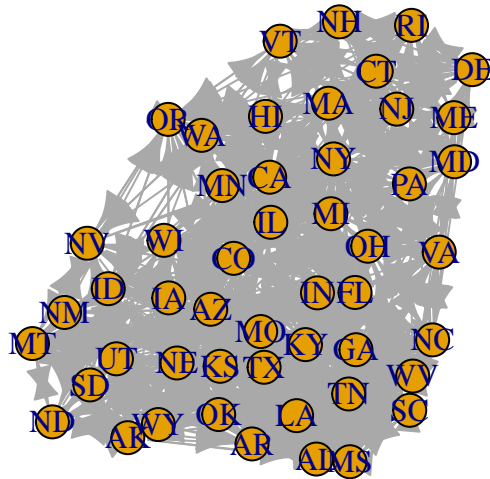
```
##
## Attaching package: 'igraph'

## The following objects are masked from 'package:network':
##
##      %c%, %s%, add.edges, add.vertices, delete.edges, delete.vertices,
##      get.edge.attribute, get.edges, get.vertex.attribute, is.bipartite,
##      is.directed, list.edge.attributes, list.vertex.attributes,
##      set.edge.attribute, set.vertex.attribute

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

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

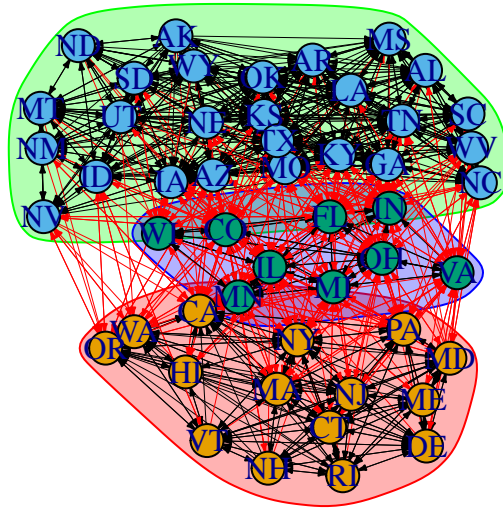
```
#use data frame to turn into an igraph network object
sim_graph <- graph_from_data_frame(sim_as_df, directed = TRUE)
#silly little plot to make sure it worked
plot(sim_graph)
```



Question 6

Estimate the community detection algorithm for walktrap algorithm. Plot the network with community membership included in the graph. How many communities are there?

```
#create the walktrap
walktrap_sim <- cluster_walktrap(sim_graph)
#make a pretty plot
plot(walktrap_sim, sim_graph, edge.width = .5, edge.arrow.size = .2)
```



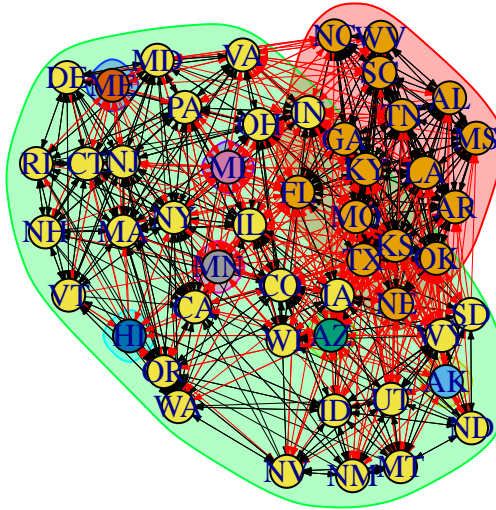
#had to make the edges and arrow size smaller or you couldnt see anything

There are three clear and distinct communities.

Question 7

Estimate the community detection algorithm for the betweenness community algorithm. Plot the network with community membership included in the graph. How many communities are there?

```
#build betweenness algorithm
between_sim <- cluster_edge_betweenness(sim_graph)
#plot it, but difficult to see!
plot(between_sim, sim_graph, edge.width = .5, edge.arrow.size = .2)
```

```
#make sure we can see the members to count correctly
membership(between_sim)
```

```
## AL AK AZ AR CA CO CT DE FL GA HI ID IL IN IA KS KY LA ME MD MA MI MN MS MO MT
##  1  2  3  1  4  4  4  4  1  1  5  4  4  4  4  1  1  1  6  4  4  7  8  1  1  4
## NE NV NH NJ NM NY NC ND OH OK OR PA RI SC SD TN TX UT VT VA WA WV WI WY
##  1  4  4  4  4  4  1  4  4  1  4  4  4  1  4  1  1  4  4  4  4  1  4  4
```

Based on our graph there appeared to be 8 communities, and we can check this is true using the membership function! So there are 8 communities, but many of them only have themselves.

Question 8

Pick one of the two community detection algorithms, and plot the network as a dendrogram.

```
#upload packages
library(dendextend)
```

```
##
## -----
## Welcome to dendextend version 1.16.0
## Type citation('dendextend') for how to cite the package.
##
```

```
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
## To suppress this message use: suppressPackageStartupMessages(library(dendextend))
## -----
```

```
##
## Attaching package: 'dendextend'
```

```
## The following object is masked from 'package:stats':
##
##   cutree
```

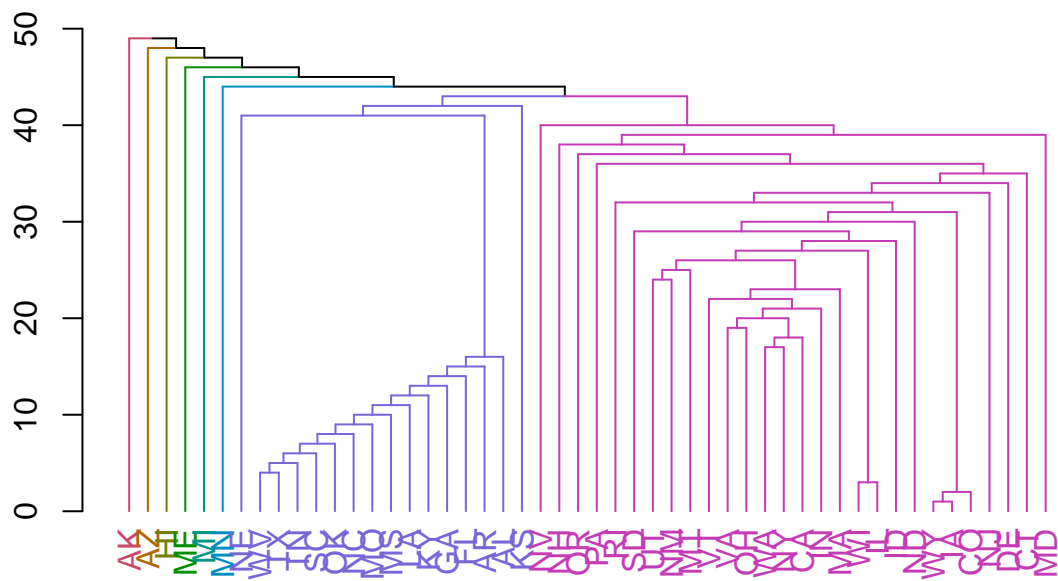
```
library(circlize)
```

```
## =====
## circlize version 0.4.15
## CRAN page: https://cran.r-project.org/package=circlize
## Github page: https://github.com/jokergoo/circlize
## Documentation: https://jokergoo.github.io/circlize\_book/book/
##
## If you use it in published research, please cite:
## Gu, Z. circlize implements and enhances circular visualization
##   in R. Bioinformatics 2014.
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(circlize))
## =====
```

```
##
## Attaching package: 'circlize'
```

```
## The following object is masked from 'package:igraph':
##
##   degree
```

```
#build denogram and add colors!
sim_den <- as.dendrogram(between_sim) %>%
  color_branches(k = 8) %>%
  color_labels(k = 8)
#build a plot of a denogram
plot(sim_den)
```



Question 9

Use the `circlize_dendrogram` to plot the same network as a circle. Don't forget to add colors by community.

```
circlize_dendrogram(sim_den)
```