Analyzing centralitty measures for a sexual network of gonorrhea transmission

J Steven Raquel
Department of Statistics, University of California, Irvine

November 12, 2021

Abstract

Keywords: social network analysis, sexually transmitted diseases, epidemiology, centrality, clustering

1 Introduction

Contact tracing for sexually transmitted diseases (STDs) such as gonorrhea, chlamydia, syphilis etc. is a persistent epidemiological problem, as it depends on individuals getting routinely tested as well as informing their sexual partners of their positive diagnosis should they receive one. Compounding this with the fact that many of these positive cases can be symptomless but still contagious creates a serious issue. Gonorrhea in particular is a disease that can be asymptomatic in both men and women who have it, that can go so far as causing infertility or lead to a life-threatening condition.

This dataset, constructed in the form of an adjacency matrix, contains 89 nodes, one of which is the "event" of attending a bar (i.e. when a node has a tie with this bar node, it means they attend the bar). Two of these nodes (denoted by `x` and `x2`) are missing information about their gender which is otherwise indicated by an `m` or `f` in the label of the respective node, followed by a number with which to differentiate them.

The network is directed, and the criteria for some node i to have a tie with some other node j must be that that ith node named node j as a prior sexual partner.

The data was collected from a series of adjacent aboriginal communities located in the province of Alberta, Canada, where public health officials took note of a local gonorrhea outbreak and confirmed that attendance at a local bar in one such community was associated with infection (De 2003).

2 Methods

```
par(mfrow = c(1,1))
set.seed(10)
org_coord <-
  gplot(gonnet, vertex.col = gonnet_df$col,
      vertex.sides = gonnet_df$gender_lty,
      displaylabels = T, label.cex = 0.6,
      boxed.labels = F, pad = 2, usearrows = T,
      vertex.cex = 1.25)
# legend for gender
legend("topleft",
       legend = c("yes", "no"),
       col = c("tomato1", "grey"),
       fill = F, border = "white", pch = 19,
       title = "Bar Attendance", bty = "n")
legend("topright",
       legend = c("m", "f"),
       col = c("black"),
       fill = F, border = "white", pch = c(0, 2),
       title = "Gender", bty = "n")
```

```
# legend for bar attendance
```

Looking at this initial sociogram, where the bar node is firmly in the center, we see that it has outgoing connections to 17 nodes, which themselves have connections to at least one other node in the rest of the network.

The majority of ties are between individuals of the opposite sex, e.g. male-to-female or female-to-male, but there are a minority of instances where individuals have a tie to individuals of the same sex, e.g. m112 has ties with both m106 and m107, who both in turn have ties to at least one female node.

The node m010 is also unique in that it happens to have outgoing ties to one female node (f024) and one male node (m018), the latter of which in turn has a tie with a female node (f023). The idea of men who have sex with men (MSM) or women who have sex with women (WSW) acting as bridging nodes between otherwise disparate sexual networks was something considered in the exploratory analysis but there just wasn't enough data to delve deeper into this subject.

The first approach in the analysis was to look at the *centrality measures* of each of the nodes.

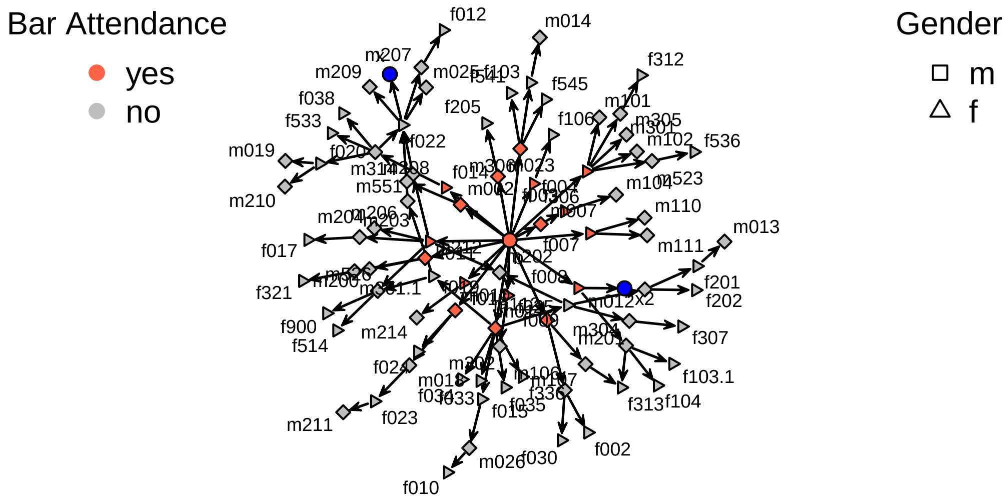
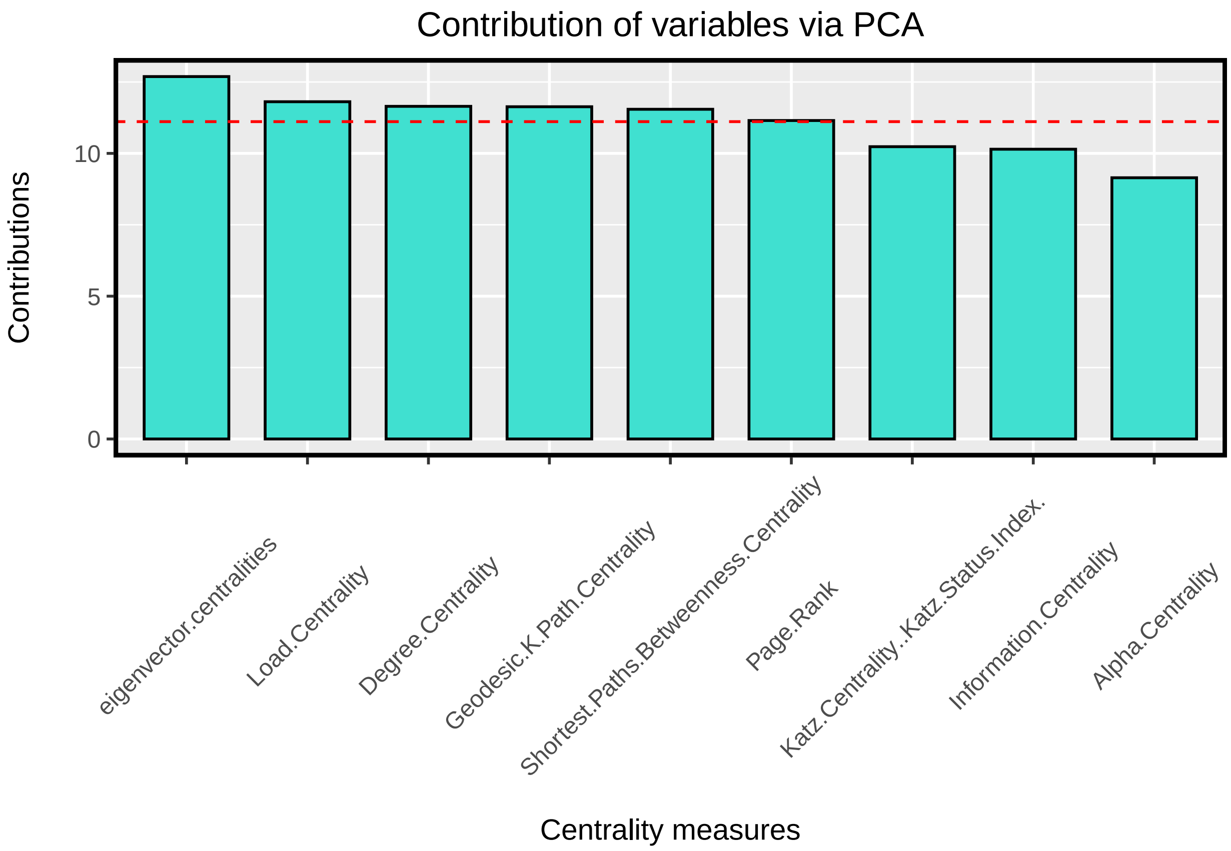


Figure 1: Sociogram of the gonorrhea network.

2.1 Centrality Measures

```
##               outdegree   indegree betweenness eigenvector
## outdegree    1.0000000 -0.1970015  0.40612528  0.79067530
## indegree    -0.1970015  1.0000000  0.52061632 -0.37578288
## betweenness  0.4061253  0.5206163  1.00000000 -0.05333083
## eigenvector  0.7906753 -0.3757829 -0.05333083  1.00000000
```

```
##               outdegree   indegree betweenness eigenvector
## outdegree    1.0000000  1.0000000   0.9543577  -0.8172166
## indegree     1.0000000  1.0000000   0.9543577  -0.8172166
## betweenness  0.9543577  0.9543577   1.0000000  -0.8603699
## eigenvector -0.8172166 -0.8172166  -0.8603699   1.0000000
```



According to Borgatti and Everett (2006), *centrality* is a summary index of a node's position in a graph, based on sums or averages of one of several things: 1) the number of edges the node has, 2) the length of the paths that end up at the node, or 3) the proportion of paths that contain the node inside of it (not as an endpoint).

Different measures of centrality depend on functions of one of these aspects and communicate different things about a node, depending on the algorithm for the centrality measure. Among the centrality measures used in this analysis were that of *degree centrality*, *eigenvector centrality*, *load centrality*, and *information centrality*.

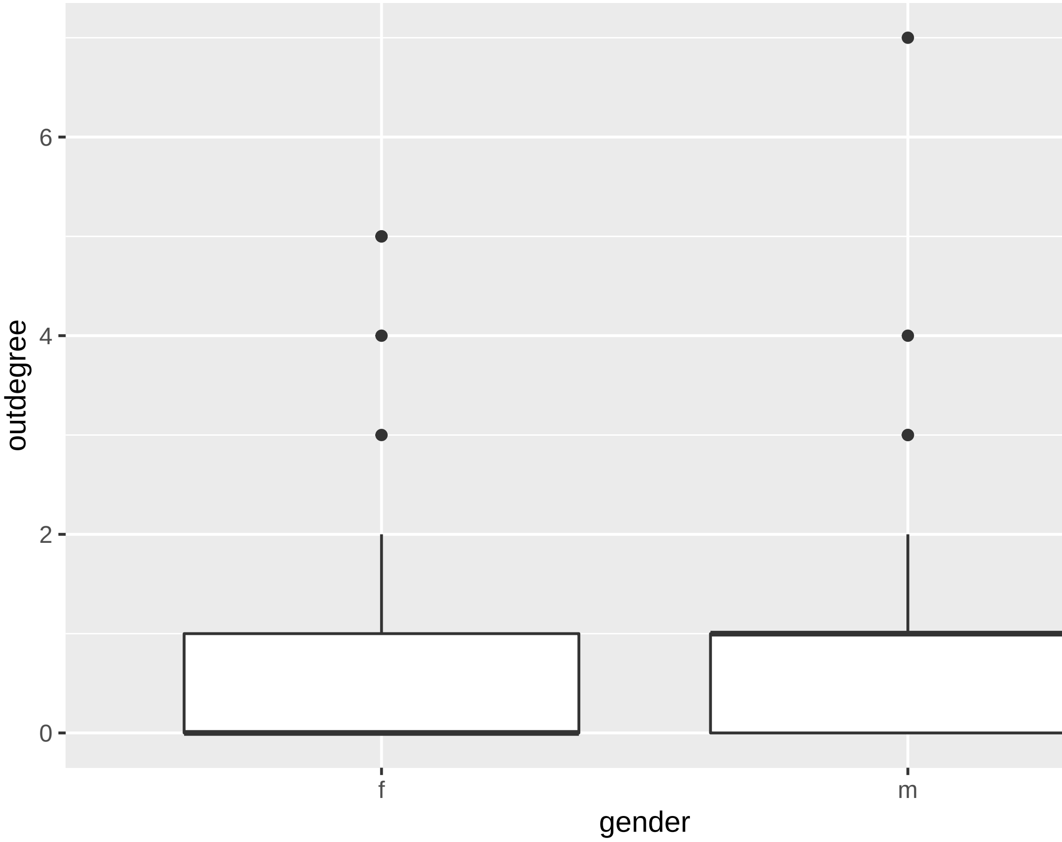
2.1.1 Degree Centrality

Degree centrality can be measured in multiple ways; the first is *indegree* which is a count of the number of incoming ties that a node has. The second is *outdegree* which conversely is the count of the number of outgoing ties that a node has.

The vast majority of nodes in the directed network have an indegree of 1, i.e. only one individual nominated that person as a sexual partner when interviewed. The outdegree was *at least* 1 for most nodes in the network, but could be zero for nodes on the very outside of the network who did not name any sexual partners.

The overall degree of a node can be taken as the total amount of ties a node has, either incoming or outgoing, but in the case of an undirected graph, these are one and the same.

```r
gonnet_df2 <- cbind(gonnet_df, cen) %>% select(-id)

# boxplot comparing outdegree
outdegree_gender <- gonnet_df2 %>%
  select(nodes, gender, outdegree) %>%
  drop_na()

# boxplot of outdegree by gender
ggplot(outdegree_gender, aes(x=gender, y=outdegree)) +
  geom_boxplot()
```



Outdegree Centrality

```
# average outdegree overall
gonnet_df2$outdegree %>% mean()
```

```
## [1] 1.044944
```

```
# average outdegree among males in network, overall
outdegree_m <- gonnet_df2 %>%
  select(nodes, gender, outdegree) %>%
  filter(gender == "m") %>%
  select(outdegree) %>%
  unlist()
outdegree_m %>% mean()
```

```
## [1] 0.9767442
```

```r
outdegree_m %>% median()
```

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

```r
# male outdegree that excludes nodes with zero outdegree
gonnet_df2 %>%
  select(nodes, gender, outdegree) %>%
  filter(gender == "m" & outdegree != 0) %>%
  select(outdegree) %>%
  unlist() %>% mean()
```

```
## [1] 1.826087
```

```r
# average outdegree among females
outdegree_f <- gonnet_df2 %>%
  select(nodes, gender, outdegree) %>%
  filter(gender == "f") %>%
  select(outdegree) %>%
  unlist()
outdegree_f %>% mean()
```

```
## [1] 0.7906977
```

```r
outdegree_f %>% median()
```

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

```r
# female outdegree that excludes nodes with zero outdegree
gonnet_df2 %>%
  select(nodes, gender, outdegree) %>%
  filter(gender == "f" & outdegree != 0) %>%
  select(outdegree) %>%
  unlist() %>% mean()
```

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

The analysis gave that the overall average outdegree for the directed graph is approximately 1, implying that on average individuals named one sexual partner. It's important to note that many outer edges have an outdegree of 0, which skews down the mean calculation somewhat.

The implication of this observation is two-fold. For one, rather than most individuals in the network having many sexual partners, the implication of how outdegree is distributed in this network is that most individuals have as few as one, but there is a minority of individuals (of both genders) who named more.

Behavioral Considerations There are some behavioral considerations taken into account when considering this data. The information regarding past sexual partners is entirely self-nominated on the part of the individual who represents the node in the graph, and as such, it's subject to an individual possibly withholding information, for a number of reasons.

There is a social stigma attached to promiscuity (having a high number of sexual partners), as well as living with an STD, so it's important to note that the data may be skewed by dishonesty on the part of the individuals comprising the dataset. For example, a person who receives a positive test result for a sexually transmitted disease may refrain from naming *all* of their recent sexual partners, either to avoid having to communicate the uncomfortable truth of either having contracted or transmitted a disease, or to avoid judgment for divulging what may be perceived as a high number of sexual partners. Societal attitudes towards sex and sexual health both in a Western context and also in an indigenous/Aboriginal context can and should be kept in mind when drawing conclusions from this data.

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.2092 0.6486
##       84
```

```r
# Student's t-test comparing the mean outdegree of men and women
# H0: mu_x - mu_y = 0
t.test(x = outdegree_m,
       y = outdegree_f,
       var.equal = T, alternative = "two.sided")
```

```
##
##  Two Sample t-test
##
## data:  outdegree_m and outdegree_f
## t = 0.6411, df = 84, p-value = 0.5232
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.3910458  0.7631389
## sample estimates:
## mean of x mean of y
## 0.9767442 0.7906977
```

```r
# p-value = 0.5232 > 0.05, we fail to reject H0
```

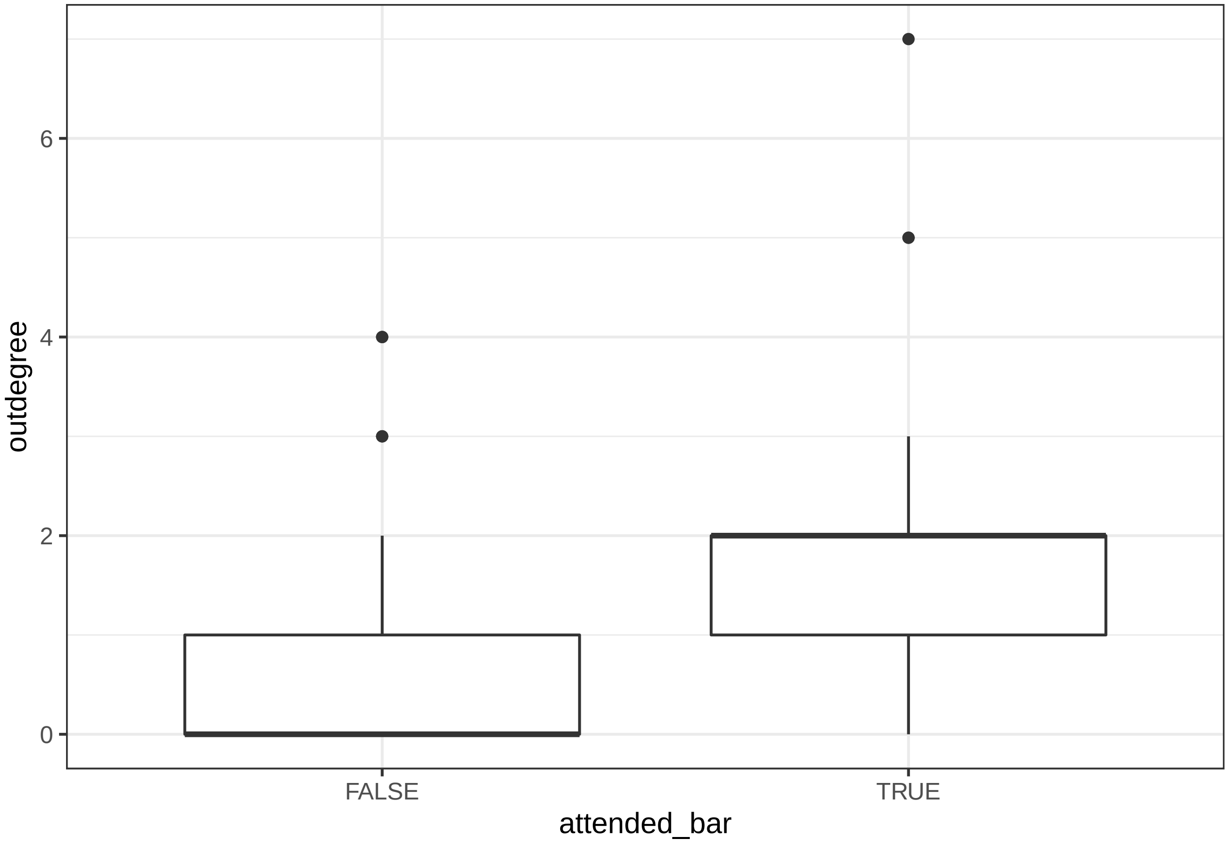
2.1.2 Eigenvector Centrality

Eigenvector centrality is calculated both as a function of a node's degree but also as a function of the degree of the nodes it is connected to. In other words, a node with a high eigenvector centrality is well-connected to nodes that are themselves well-connected.

2.1.3 Load Centrality

2.1.4 Information Centrality

```
# boxplot comparing distribution of outdegree between bar (not)-attended
ggplot(gonnet_df2 %>% filter(!(nodes %in% c("b", "x", "x2"))),
       aes(x = attended_bar, outdegree)) +
  geom_boxplot() + theme_bw()
```

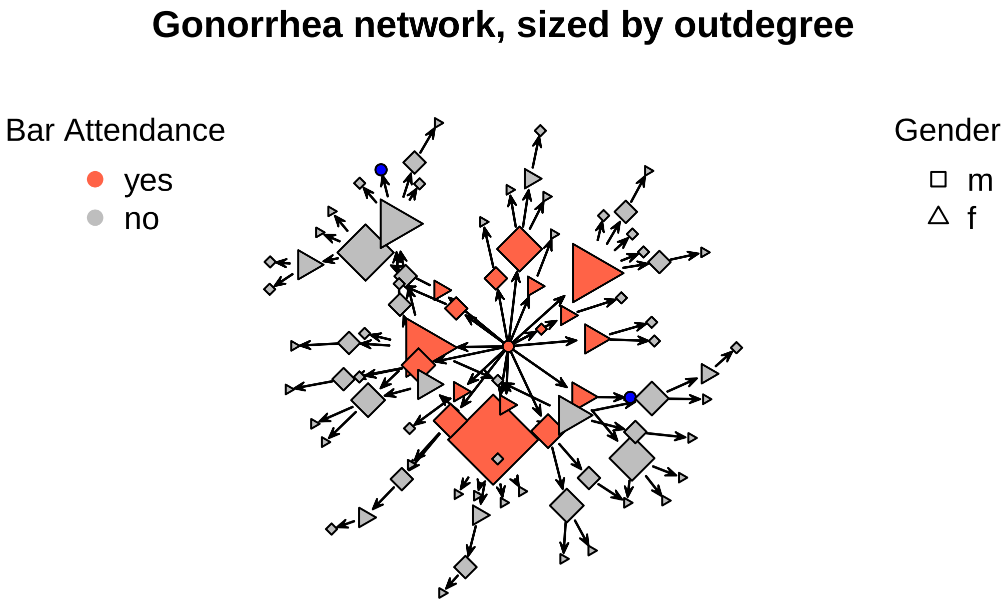


```
# average outdegree among bar attendees
outdegree_bar <- gonnet_df2 %>%
  select(nodes, attended_bar, outdegree) %>%
  filter(!(nodes %in% c("b", "x2", "x"))) %>%
  filter(attended_bar == TRUE)


# average outdegree among non-bar attendees
outdegree_nobar <- gonnet_df2 %>%
  select(nodes, attended_bar, outdegree) %>%
  filter(!(nodes %in% c("b", "x2", "x"))) %>%
```

```r
  filter(attended_bar == FALSE)

# sample sizes are unequal, so we cannot assume equal variance
# try two-sided Welch's t.test
# H0: mu_x - mu_y = 0
t.test(x = outdegree_bar$outdegree,
       y = outdegree_nobar$outdegree,
       var.equal = FALSE, alternative = "two.sided")
```

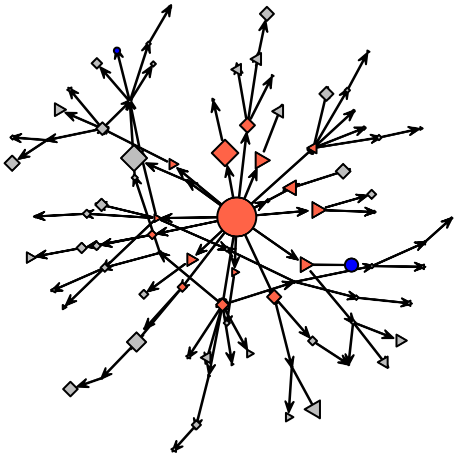
```
##
##  Welch Two Sample t-test
##
## data:  outdegree_bar$outdegree and outdegree_nobar$outdegree
## t = 3.4857, df = 18.197, p-value = 0.002605
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.6408486 2.5816578
## sample estimates:
## mean of x mean of y
## 2.1764706 0.5652174
```

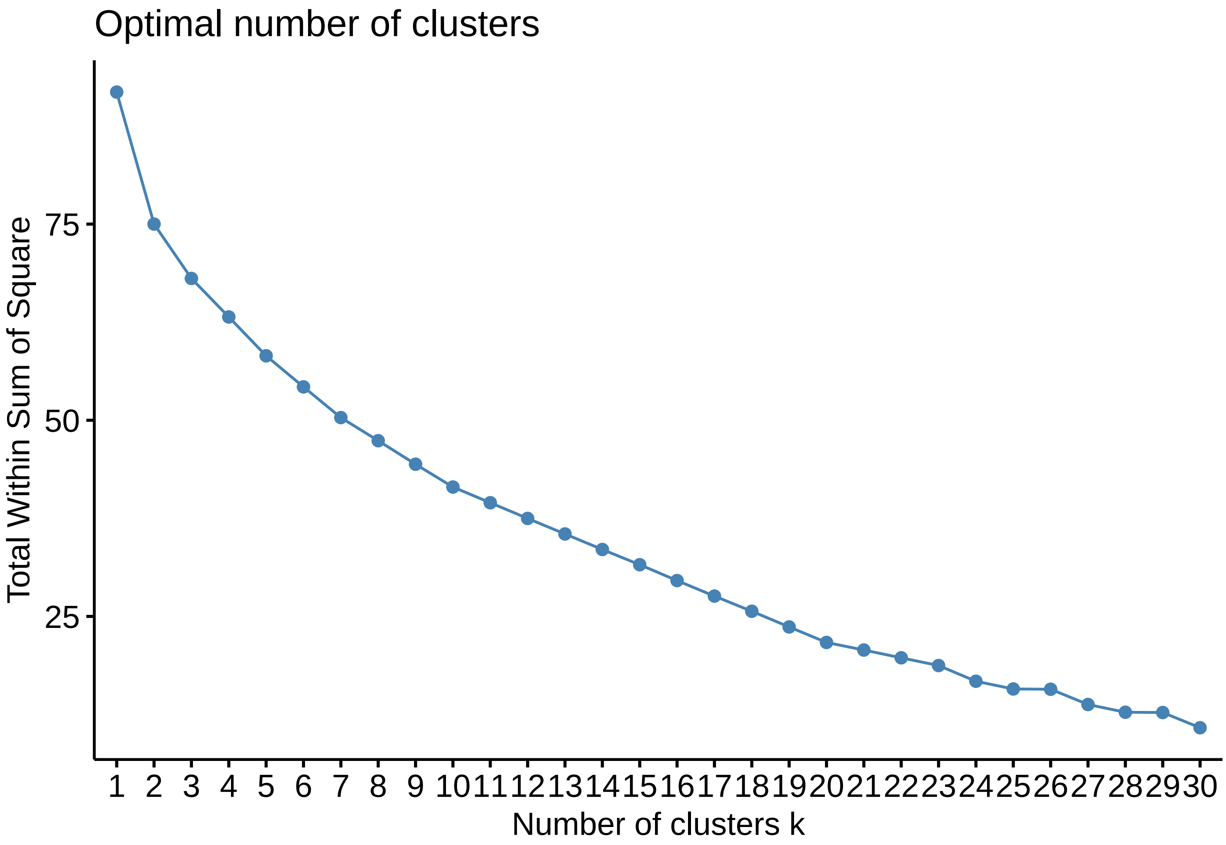
```r
# p-value < 0.05, reject H0
# conclude the difference in mean outdegree is not equal to zero
```

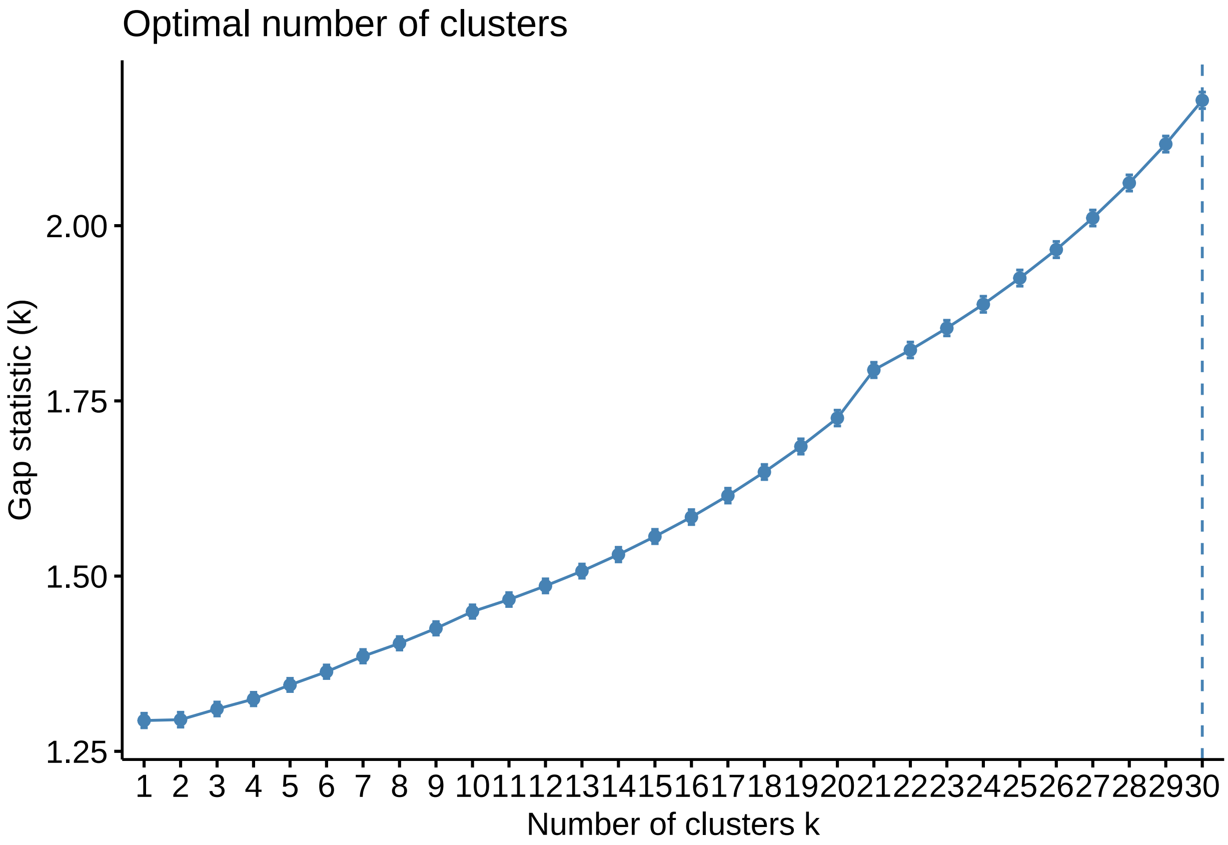


2.2 Principal Component Analysis

```
# extracting the centralities that were important based on the PCA
centrality_eigen <- centralities$`eigenvector centralities`
centrality_load <- centralities$`Load Centrality`
centrality_degree <- centralities$`Degree Centrality`
centrality_geodesic <- centralities$`Geodesic K-Path Centrality`
centrality_shortest <- centralities$`Shortest-Paths Betweenness Centrality`
centrality_info <- centralities$`Information Centrality`
```

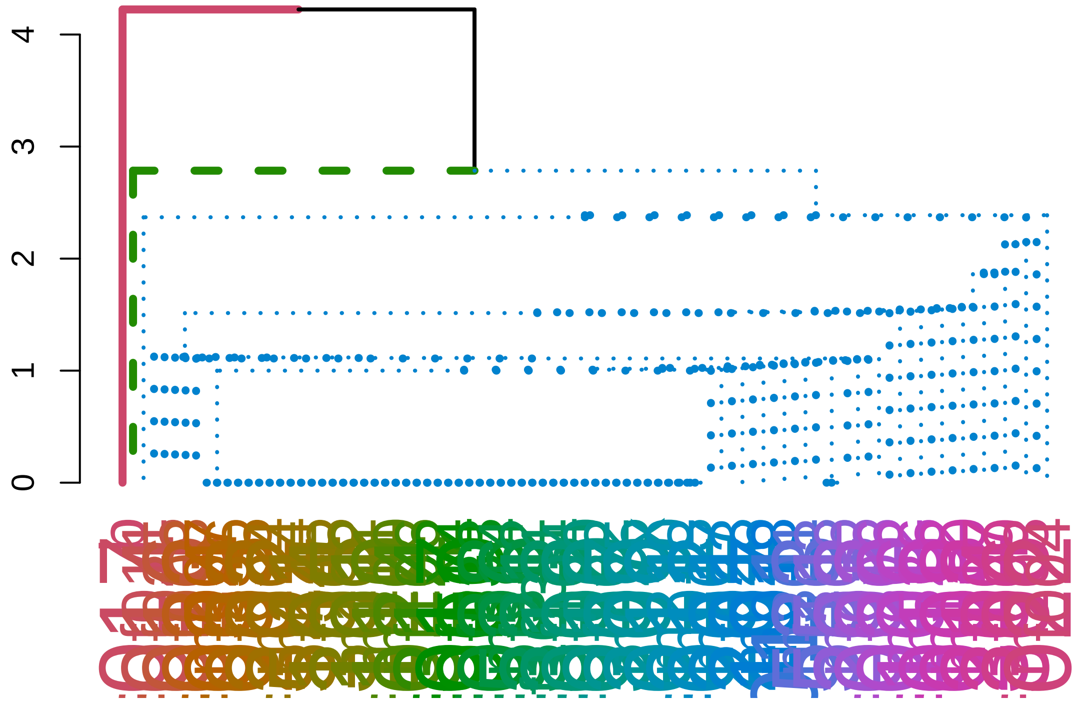



Optimal number of clusters
Total Within Sum of Square
75
50
25
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Number of clusters k



```
km <- kmeans(gonnet, centers = 21, nstart = 25)

# fviz_cluster(km, data = gonnet_nob)

# library(netdiffuseR) is loaded
gonnet_edgelist <- adjmat_to_edgelist(gonnet, undirected = F)

# cluster_membership
km_cluster_mem <- km$cluster %>% as.data.frame() %>%
  tibble::rownames_to_column() %>%
  rename(node = 'rowname', cluster = '.')
```

```
## [1] 0.8444488
## [1] 0.8528613
## [1] 0.8285892
## [1] 0.8514371
```

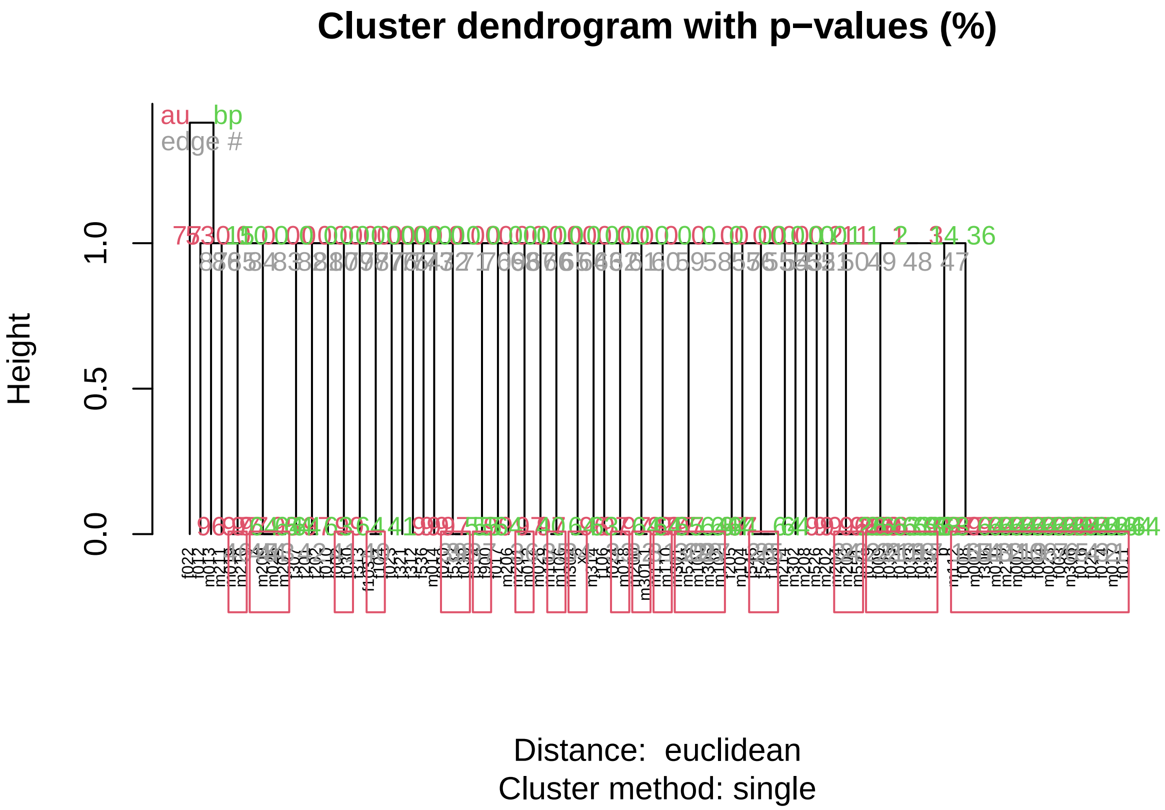


```
fit <-
  pvclust(gonnet,
  method.hclust = "single",
  method.dist = "euclidean",
  iseed = 10, # to get same results
  parallel = T, # to use all but one CPU thread
  nboot = 1000)
```

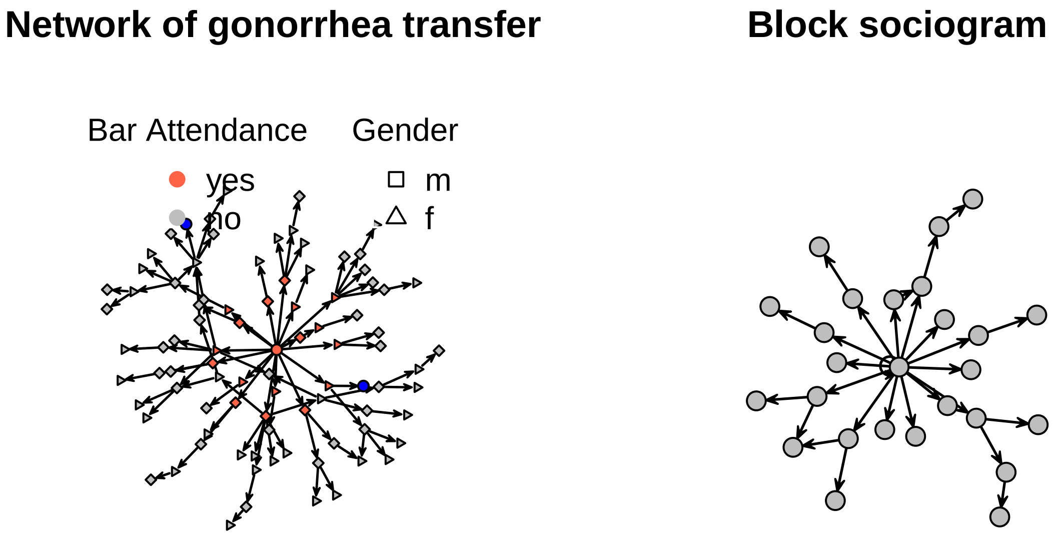
```
## Creating a temporary cluster...done:
## socket cluster with 15 nodes on host 'localhost'
## Multiscale bootstrap... Done.
```

```r
fit_hclust <- fit$hclust

fit_hclust %>% cutreeDynamicTree(deepSplit = F)
```

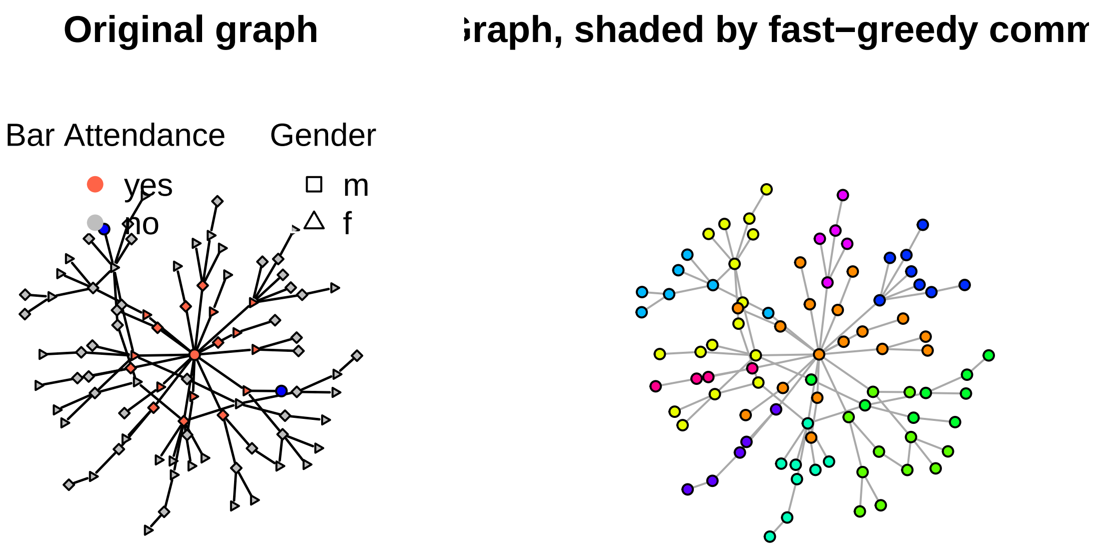
```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [39] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1
## [77] 1 1 1 1 1 1 1 1 1 1 1 1 1
```



```
## [1] "The mean overall network density is 0.013."
```



```
## Note: fast-greedy community algorithm is for undirected graphs.
```



```
par(mfrow = c(1,1))
set.seed(10)
edgelist <- as.edgelist(gonnet, n = dim(gonnet)[1])
plot_kcores(edgelist, sym = F, mode = "digraph",
            coord = org_coord,
            cmode = "outdegree")
```

3 References

4 Appendix

```r
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
library(sna)
library(igraph)
library(epimdr) # for gonnet data
library(pvclust) # for pvclust()
library(CINNA) # for pca_centralities()
library(car) # for leveneTest()
library(factoextra) # for nbclust
library(cluster) # more cluster analysis
library(netdiffuseR)
library(dendextend) # for appealing dendrograms
library(dynamicTreeCut)
data(gonnet)
source(here::here("functions.R"))
# data is loaded as 'gonnet'
nodes <- rownames(gonnet)
colnames(gonnet) <- nodes

# symmetrized graph (undirected)
gonnet_sym <- symmetrize(gonnet, rule = "weak")

gonnet_df <- data.frame(nodes) %>%
  mutate(gender = ifelse(grepl("m", nodes), yes = "m", no = NA)) %>%
  mutate(gender = ifelse(grepl("f", nodes), yes = "f", no = gender)) %>%
  mutate(gender_lty = ifelse(gender == "m", yes = 4, no = 3)) %>%
  mutate(gender_lty = ifelse(gender == "f", yes = 3, no = gender_lty)) %>%
  mutate(gender_lty = ifelse(gender %in% c("m", "f"), yes = gender_lty, no = 50)
         ) %>%
  mutate(attended_bar = gonnet["b",] == 1) %>%
  mutate(col = ifelse(attended_bar == T, "tomato1", "grey"))


gonnet_df$col[!(gonnet_df$gender %in% c("m", "f"))] <- "blue"
gonnet_df$col[gonnet_df$nodes == "b"] <- "tomato1"
par(mfrow = c(1,1))
set.seed(10)
org_coord <-
  gplot(gonnet, vertex.col = gonnet_df$col,
      vertex.sides = gonnet_df$gender_lty,
      displaylabels = T, label.cex = 0.6,
      boxed.labels = F, pad = 2, usearrows = T,
      vertex.cex = 1.25)
# legend for gender
legend("topleft",
       legend = c("yes", "no"),
       col = c("tomato1", "grey"),
       fill = F, border = "white", pch = 19,
       title = "Bar Attendance", bty = "n")
legend("topright",
       legend = c("m", "f"),
       col = c("black"),
       fill = F, border = "white", pch = c(0, 2),
       title = "Gender", bty = "n")
# legend for bar attendance
cen <- get_centralities(gonnet, "directed")

# correlations between centrality measures
```

```
centrality_correlations(gonnet, cen)

cen_sym <- get_centralities(gonnet_sym, "undirected")
centrality_correlations(gonnet_sym, cen_sym)
# creating igraph object
gonnet_ig <- graph_from_adjacency_matrix(gonnet)

(pc <- proper_centralities(gonnet_ig)) %>%
  capture.output() %>%
  invisible()

centralities <- calculate_centralities(gonnet_ig,
                       include = pc[c(1, 4, 11, 16, 27,
                                      45, 8, 18, 31, 43, 3, 19)])

# plot of different centralities' contributions via PCA
pca_centralities(centralities)

gonnet_df2 <- cbind(gonnet_df, cen) %>% select(-id)

# boxplot comparing outdegree
outdegree_gender <- gonnet_df2 %>%
  select(nodes, gender, outdegree) %>%
  drop_na()

# boxplot of outdegree by gender
ggplot(outdegree_gender, aes(x=gender, y=outdegree)) +
  geom_boxplot()

# average outdegree overall
```

```
gonnet_df2$outdegree %>% mean()

# average outdegree among males in network, overall
outdegree_m <- gonnet_df2 %>%
  select(nodes, gender, outdegree) %>%
  filter(gender == "m") %>%
  select(outdegree) %>%
  unlist()
outdegree_m %>% mean()
outdegree_m %>% median()
# male outdegree that excludes nodes with zero outdegree
gonnet_df2 %>%
  select(nodes, gender, outdegree) %>%
  filter(gender == "m" & outdegree != 0) %>%
  select(outdegree) %>%
  unlist() %>% mean()

# average outdegree among females
outdegree_f <- gonnet_df2 %>%
  select(nodes, gender, outdegree) %>%
  filter(gender == "f") %>%
  select(outdegree) %>%
  unlist()
outdegree_f %>% mean()
outdegree_f %>% median()

# female outdegree that excludes nodes with zero outdegree
gonnet_df2 %>%
  select(nodes, gender, outdegree) %>%
  filter(gender == "f" & outdegree != 0) %>%
  select(outdegree) %>%
  unlist() %>% mean()
# levene test for equality of variances
# H0: the variance in outdegree between the two genders is equal
outdegree_gender$gender <- as.factor(outdegree_gender$gender)

leveneTest(outdegree ~ gender,
           data = outdegree_gender)
# p-value > 0.05, so we fail to reject H0, conclude the variances are equal
# Student's t-test comparing the mean outdegree of men and women
# H0: mu_x - mu_y = 0
t.test(x = outdegree_m,
       y = outdegree_f,
       var.equal = T, alternative = "two.sided")
# p-value = 0.5232 > 0.05, we fail to reject H0


# boxplot comparing distribution of outdegree between bar (not)-attended
ggplot(gonnet_df2 %>% filter(!(nodes %in% c("b", "x", "x2"))),
       aes(x = attended_bar, outdegree)) +
  geom_boxplot() + theme_bw()

# average outdegree among bar attendees
outdegree_bar <- gonnet_df2 %>%
  select(nodes, attended_bar, outdegree) %>%
  filter(!(nodes %in% c("b", "x2", "x"))) %>%
  filter(attended_bar == TRUE)

# average outdegree among non-bar attendees
outdegree_nobar <- gonnet_df2 %>%
  select(nodes, attended_bar, outdegree) %>%
  filter(!(nodes %in% c("b", "x2", "x"))) %>%
  filter(attended_bar == FALSE)

# sample sizes are unequal, so we cannot assume equal variance
# try two-sided Welch's t.test
# H0: mu_x - mu_y = 0
t.test(x = outdegree_bar$outdegree,
       y = outdegree_nobar$outdegree,
       var.equal = FALSE, alternative = "two.sided")
# p-value < 0.05, reject H0
# conclude the difference in mean outdegree is not equal to zero
# normalizing the outdegree of bar because of course it has a high outdegree
outdegree_m <- cen$outdegree + 1
outdegree_m[1] <- 1

set.seed(10)

# plotting based on outdegree
gplot(gonnet,
      vertex.col = gonnet_df$col,
      vertex.sides = gonnet_df$gender_lty,
      vertex.cex = outdegree_m,
      coord = org_coord)
# legend for gender
legend("topleft",
       legend = c("yes", "no"),
       col = c("tomato1", "grey"),
       fill = F, border = "white", pch = 19,
       title = "Bar Attendance", bty = "n")
```

```r
legend("topright",
       legend = c("m", "f"),
       col = c("black"),
       fill = F, border = "white", pch = c(0, 2),
       title = "Gender", bty = "n")
title("Gonorrhea network, sized by outdegree")
# extracting the centralities that were important based on the PCA
centrality_eigen <- centralities$`eigenvector centralities`
centrality_load <- centralities$`Load Centrality`
centrality_degree <- centralities$`Degree Centrality`
centrality_geodesic <- centralities$`Geodesic K-Path Centrality`
centrality_shortest <- centralities$`Shortest-Paths Betweenness Centrality`
centrality_info <- centralities$`Information Centrality`
set.seed(10)

centrality_degree[1] <- 1
gplot(gonnet, vertex.col = gonnet_df$col,
      vertex.sides = gonnet_df$gender_lty,
      vertex.cex = centrality_info %>% scale(),
      # displaylabels = T, label.cex = 0.6, label.pos = 2,
      boxed.labels = F, pad = 2,
      coord = org_coord)
# library(factoextra) is loaded
fviz_nbclust(gonnet, kmeans, method = "wss", k.max = 30)

# library(cluster) is loaded
gap_stat <- clusGap(gonnet, FUN = kmeans, nstart = 25, K.max = 30, B = 50)
fviz_gap_stat(gap_stat)
km <- kmeans(gonnet, centers = 21, nstart = 25)
```

```r
# fviz_cluster(km, data = gonnet_nob)

# library(netdiffuseR) is loaded
gonnet_edgelist <- adjmat_to_edgelist(gonnet, undirected = F)

# cluster_membership
km_cluster_mem <- km$cluster %>% as.data.frame() %>%
  tibble::rownames_to_column() %>%
  rename(node = 'rowname', cluster = '.')

set.seed(10)
par(mfrow = c(1,1))
clust <- hclust(dist(gonnet), method = "average")
clust_dend <- clust %>% as.dendrogram()
# clust <- equiv.clust(gonnet)

bm_complete <- blockmodel(gonnet,
                          ec = clust, k = 15)

bmR_complete <- reorder_blockmodel(bm_complete)
# overall network density, which we will use for alpha
alpha <- bmR_complete$block.model %>% mean(na.rm = T)
paste0("The mean overall network density is ", round(alpha, 3), ".")

density_matrix <- bmR_complete$block.model
# plotting blockmodel
gplot(density_matrix > alpha,
      diag = T,
      vertex.cex = 1.5,
      label = unique(bmR_complete$block.membership),
      # boxed.labels=F,
      vertex.col="grey",
      pad = 1.25)
title("Block sociogram")
set.seed(10)
par(mfrow = c(1,1))
#define linkage methods
hclust_methods <- c("average", "single", "complete", "ward")
names(hclust_methods) <- hclust_methods
#function to compute agglomerative coefficient
ac <- function(data, method) {
  agnes(data, method)$ac
}

# cluster is loaded

#calculate agglomerative coefficient for each clustering linkage method
ac_list <- list()
for (i in 1:4) {
  ac_list[[i]] <- agnes(gonnet, diss = FALSE, metric = "euclidean",
                        method = hclust_methods[i])

  print(ac_list[[i]]$ac)
}
# maximum agglomerative coefficient is from the single linkage algorithm

# hclust using single and agnes
clust <- agnes(gonnet, "single", diss = F)
clust <- clust %>% as.hclust() %>% as.dendrogram() %>%
  set("labels_color")  %>%
  set("labels_cex", c(2,3)) %>%
  set("branches_lwd", c(2,4)) %>%
  set("branches_k_lty", k=3)  %>%
  set("branches_k_color", k = 3) %>% plot()



#pltree(clust, cex = 0.6, hang = -1)
fit <-
  pvclust(gonnet,
  method.hclust = "single",
  method.dist = "euclidean",
  iseed = 10, # to get same results
  parallel = T, # to use all but one CPU thread
  nboot = 1000)
fit_hclust <- fit$hclust
fit_hclust %>% cutreeDynamicTree(deepSplit = F)
par(mfrow = c(1,1))
plot(fit, hang = -1, cex = 0.5)
pvrect(fit, alpha = 0.95)
set.seed(10)
# if pchisq > 0.05, cluster is significant
pp <- pvpick(fit)
# number of significant clusters
ideal_k <- pp$clusters %>% length()

# note that these are just the "child" clusters and I want to include their parents

# blockmodel using k = 58 and cutting trees at 0.5
bm_complete <- blockmodel(gonnet,
                          ec = fit$hclust,
                           k = ideal_k + 9, mode = "digraph")

bmR_complete <- reorder_blockmodel(bm_complete)
# overall network density, which we will use for alpha
alpha <- bmR_complete$block.model %>% mean(na.rm = T)
paste0("The mean overall network density is ", round(alpha, 3), ".")

density_matrix <- bmR_complete$block.model

par(mfrow = c(1,2))
# original plot, again
gplot(gonnet, vertex.col = gonnet_df$col,
      vertex.sides = gonnet_df$gender_lty,
      # displaylabels = T, label.cex = 0.6, label.pos = 2,
      boxed.labels = F, pad = 2,
      vertex.cex = 1.25)
# legend for gender
legend("topleft",
       legend = c("yes", "no"),
       col = c("tomato1", "grey"),
       fill = F, border = "white", pch = 19,
       title = "Bar Attendance", bty = "n")
legend("topright",
       legend = c("m", "f"),
       col = c("black"),
       fill = F, border = "white", pch = c(0, 2),
       title = "Gender", bty = "n")
# legend for bar attendance
title("Network of gonorrhea transfer")
```

```
# plotting blockmodel
gplot(density_matrix > alpha,
      diag = T,
      vertex.cex = 1.5,
      # label = unique(bmR_complete£block.membership),
      # boxed.labels=F,
      vertex.col="grey",
      pad = 1.25,
      # coord = org_coord
      )
title("Block sociogram")

set.seed(10)
par(mfrow = c(1,2))
# original plot, again
org_coord <- gplot(gonnet, vertex.col = gonnet_df$col,
      vertex.sides = gonnet_df$gender_lty,
      # displaylabels = T, label.cex = 0.6, label.pos = 2,
      boxed.labels = F, pad = 2,
      vertex.cex = 1.25,
      usearrows = F)
# legend for gender
legend("topleft",
       legend = c("yes", "no"),
       col = c("tomato1", "grey"),
       fill = F, border = "white", pch = 19,
       title = "Bar Attendance", bty = "n")
legend("topright",
       legend = c("m", "f"),
       col = c("black"),
       fill = F, border = "white", pch = c(0, 2),
       title = "Gender", bty = "n")
title("Original graph")
# adding in shape arguments for igraph to correspond to original sna gplot


# fast and greedy community detection
plot_fastgreedy_cd(gonnet, layout = org_coord,
                   legend = F, vertex.size = 6)
title("Graph, shaded by fast-greedy community")
# labeling only the original bar patrons
par(mfrow = c(1,1))
set.seed(10)
edgelist <- as.edgelist(gonnet, n = dim(gonnet)[1])
plot_kcores(edgelist, sym = F, mode = "digraph",
            coord = org_coord,
            cmode = "outdegree")
labs <- knitr::all_labels()
```