Stats_218_HW2

Zihan Lin

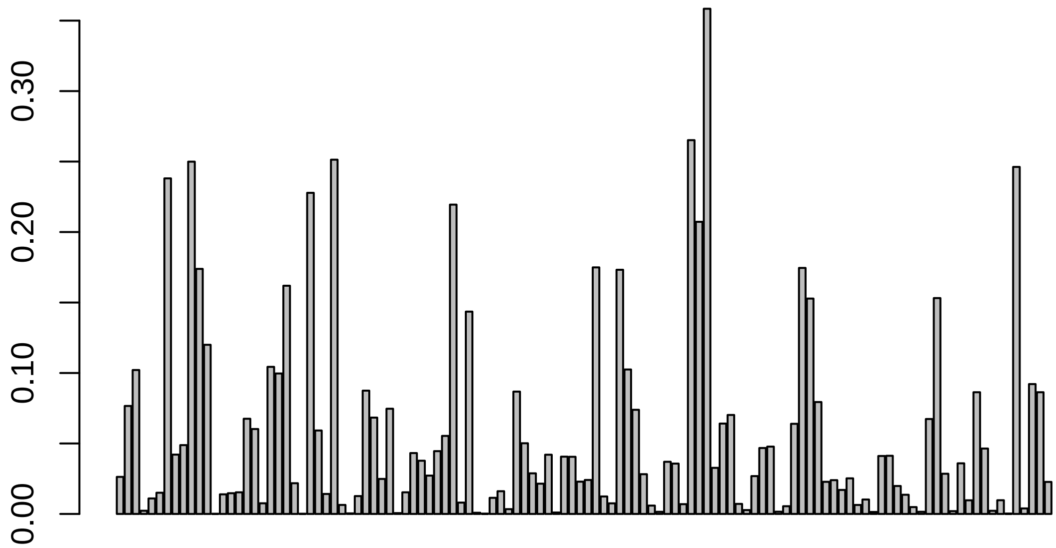
2024-10-15

```r
library(networkdata)
library(sna)
library(network)
library(igraph)
library(degreenet)
library(tidyverse)
data(florentine)
data(butland_ppi)
data(addhealth9)
data(tribes)
```

Question 1

a.

```
only_boy <- which(addhealth9$X[,"female"] == 0)
health_ntwk <- network(addhealth9$E[addhealth9$E[,1] %in% only_boy
                                    & addhealth9$E[,2] %in% only_boy,],
                       directed = FALSE)
largest_health <- component.dist(health_ntwk)$membership == which.max(component.dist(health_ntwk)$csize)
larg_health_ntwk <- network(health_ntwk[largest_health, largest_health], directed = FALSE)

evcent_health <- sna::evcent(larg_health_ntwk, gmode = "graph")
barplot(evcent_health)
```



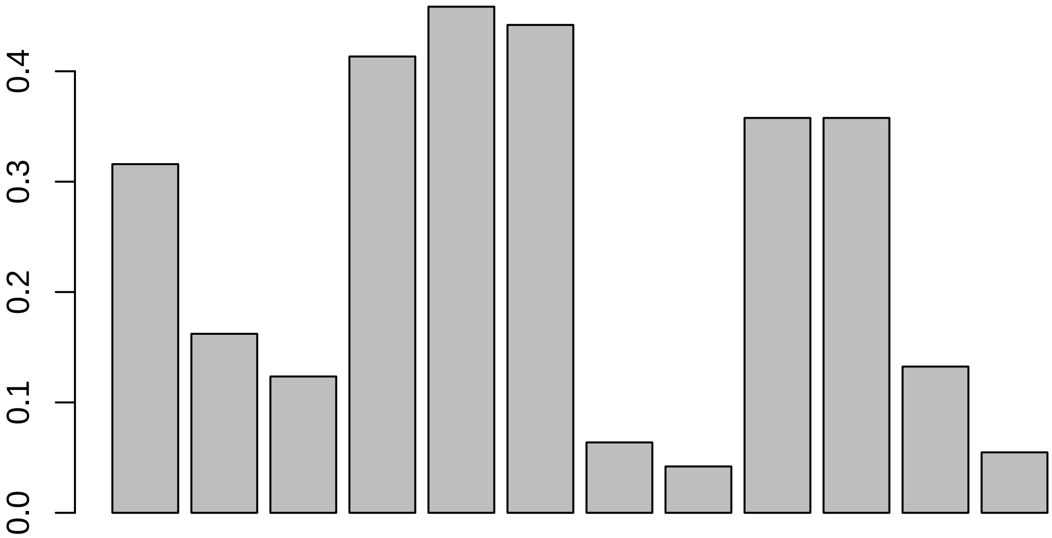

```
summary(evcent_health)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000078 0.009859 0.028670 0.058260 0.074446 0.358421
```

```
centralization(dat = larg_health_ntwk, FUN = sna::evcent, mode = "graph")
```

```
## [1] 0.4318099
```

```r
tribes_pos_ntwk <- network(tribes[,,"pos"], directed = FALSE)
largest_tribes <- component.dist(tribes_pos_ntwk)$membership == which.max(component.dist(tribes_pos_ntwk

larg_tribes_pos_ntwk <- network(tribes_pos_ntwk[largest_tribes, largest_tribes], directed = FALSE)
evcent_tribes_large <- sna::evcent(larg_tribes_pos_ntwk, gmode = "graph")
barplot(evcent_tribes_large)
```



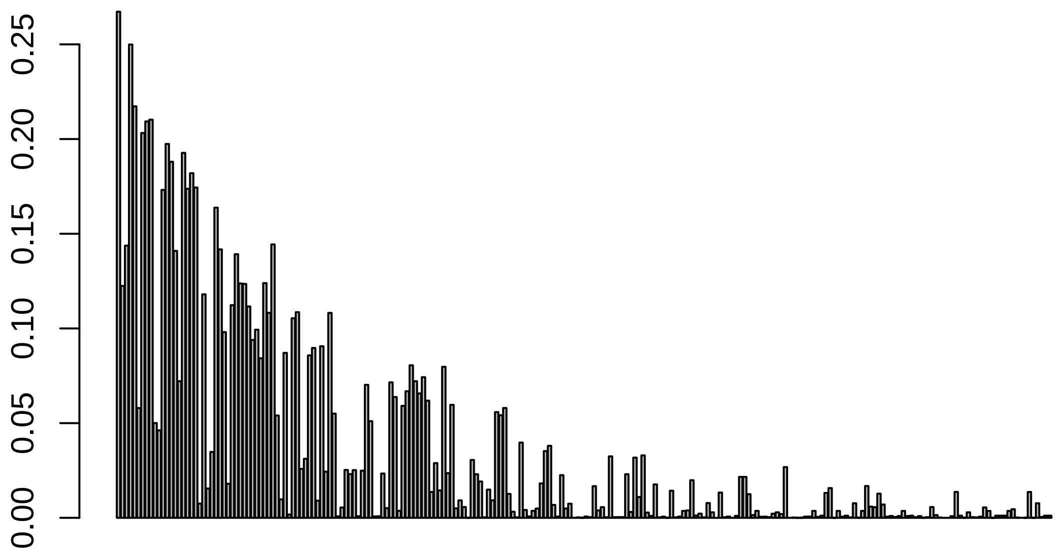

```r
summary(evcent_tribes_large)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.04199 0.10855 0.23901 0.24364 0.37162 0.45849
```

```r
centralization(dat = larg_tribes_pos_ntwk, FUN = sna::evcent, mode = "graph")
```

```
## [1] 0.3646095
```

```r
but_ntwk <- network(butland_ppi, directed = FALSE)
larg_but <- component.dist(but_ntwk)$membership == which.max(component.dist(but_ntwk)$csize)

larg_but_ntwk <- network(but_ntwk[larg_but, larg_but], directed = FALSE)
evcent_but_larg <- sna::evcent(larg_but_ntwk, gmode = "graph")
barplot(evcent_but_larg)
```




```r
summary(evcent_but_larg)
```

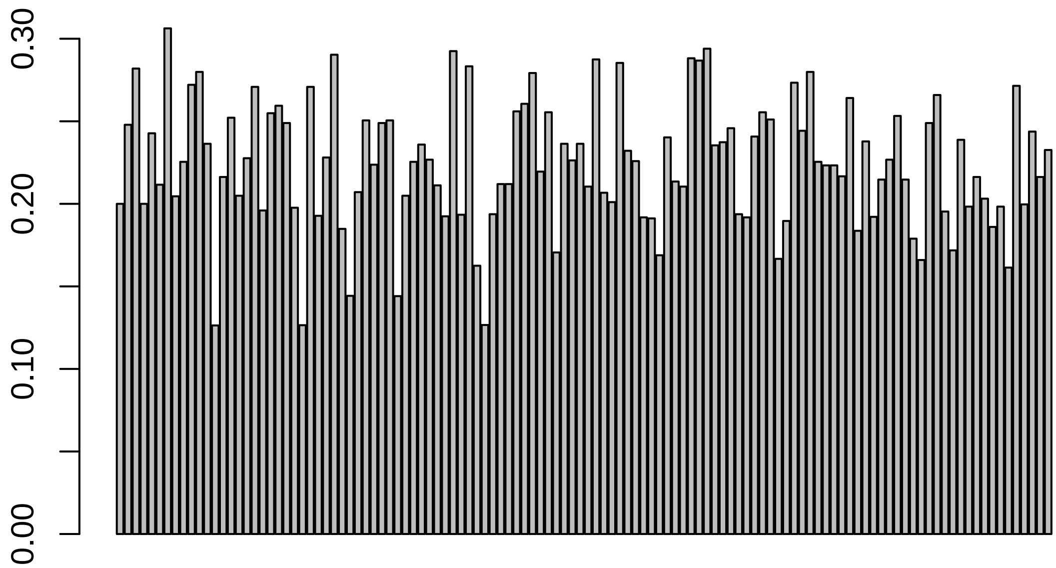
```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 6.000e-08 7.818e-04 6.905e-03 3.539e-02 5.079e-02 2.672e-01
```

```r
centralization(dat = larg_but_ntwk, FUN = sna::evcent, mode = "graph")
```

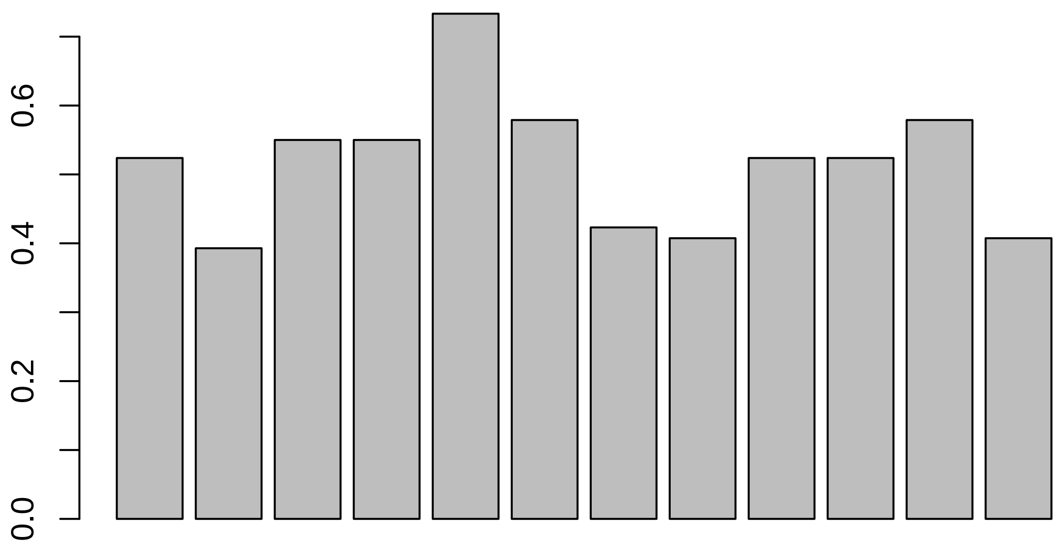
```
## [1] 0.3307625
```

b.

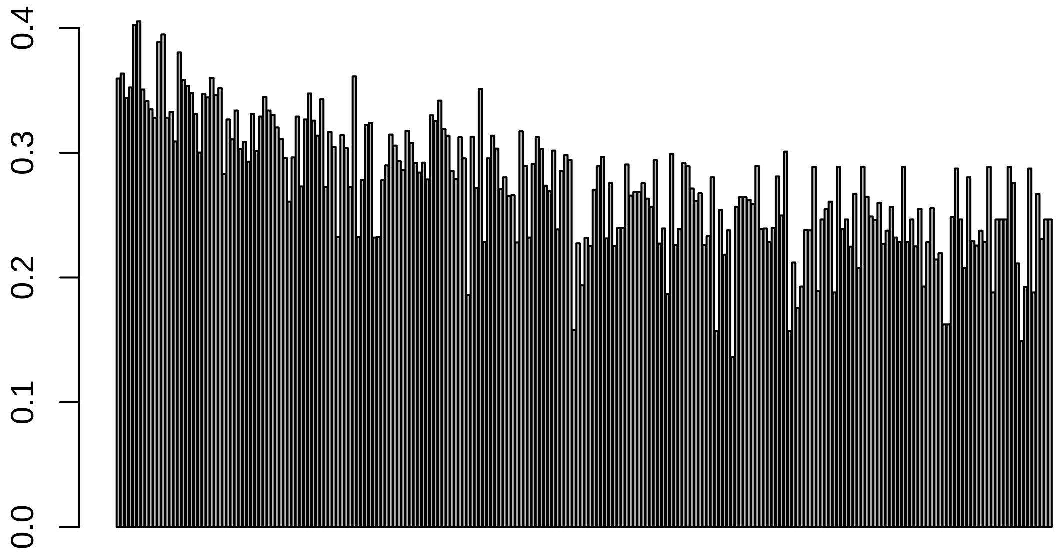
```r
close_health <- sna::closeness(larg_health_ntwk, gmode = "graph")
close_tribes <- sna::closeness(larg_tribes_pos_ntwk, gmode = "graph")
close_but <- sna::closeness(larg_but_ntwk, gmode = "graph")

barplot(close_health)
```




```r
barplot(close_tribes)
```




```r
barplot(close_but)
```




```r
summary(close_health)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1263  0.1983  0.2254  0.2241  0.2505  0.3063
```

```r
summary(close_tribes)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3929  0.4192  0.5238  0.5161  0.5572  0.7333
```

```r
summary(close_but)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1363  0.2382  0.2756  0.2744  0.3103  0.4053
```

```r
centralization(dat = larg_health_ntwk, FUN = sna::closeness, mode = "graph")
```

```
## [1] 0.166436
```

```r
centralization(dat = larg_tribes_pos_ntwk, FUN = sna::closeness, mode = "graph")
```

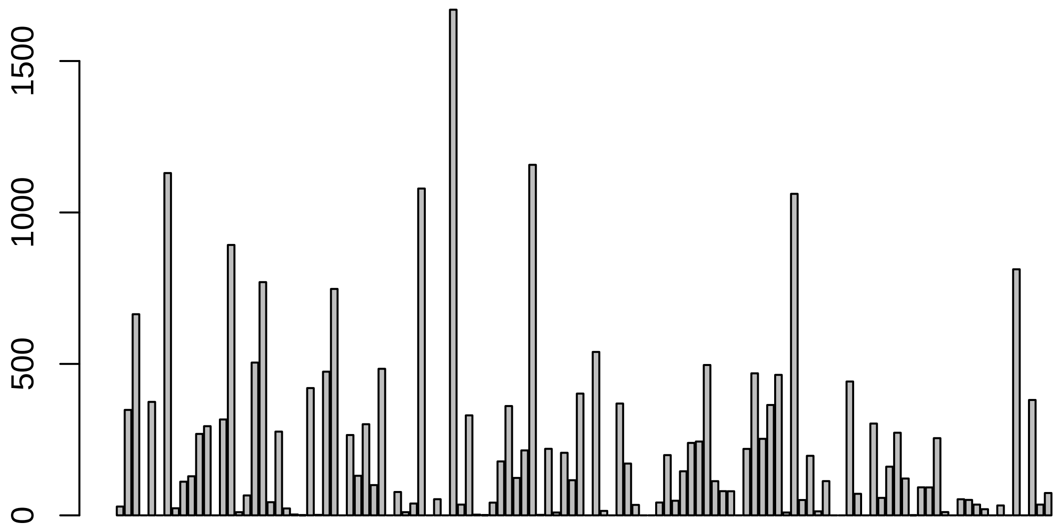
```
## [1] 0.4976226
```

```r
centralization(dat = larg_but_ntwk, FUN = sna::closeness, mode = "graph")
```

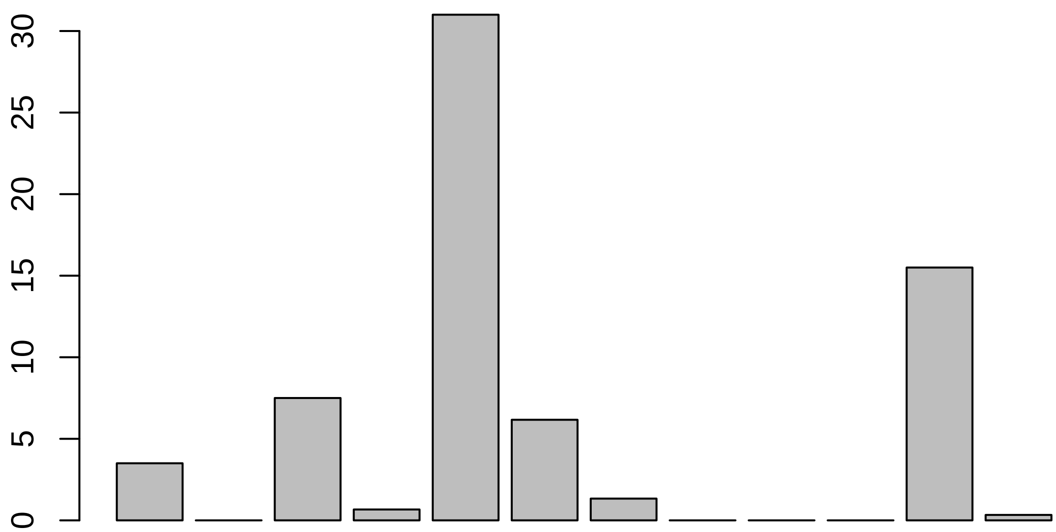
```
## [1] 0.2634439
```

c.

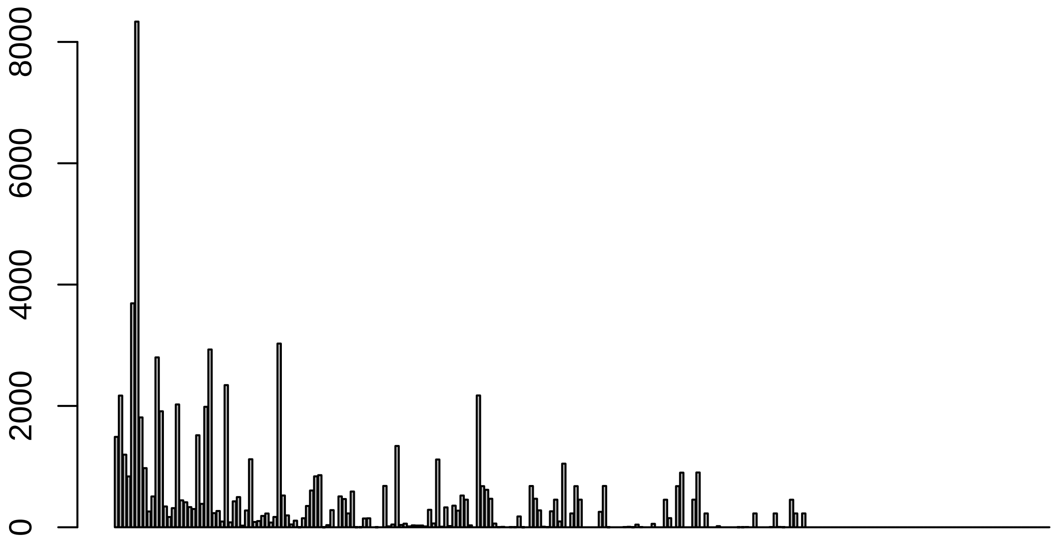
```
between_health <- sna::betweenness(larg_health_ntwk, gmode = "graph")
between_tribes <- sna::betweenness(larg_tribes_pos_ntwk, gmode = "graph")
between_but <- sna::betweenness(larg_but_ntwk, gmode = "graph")

barplot(between_health)
```




```
barplot(between_tribes)
```




```
barplot(between_but)
```




```r
summary(between_health)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   10.63   92.37  211.32  299.28 1669.50
```

```r
summary(between_tribes)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0     1.0     5.5     6.5    31.0
```

```r
summary(between_but)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    0.000    0.000    7.093  318.809  311.891 8334.841
```

```r
centralization(dat = larg_health_ntwk, FUN = sna::betweenness, mode = "graph")
```

```
## [1] 0.2167172
```

```r
centralization(dat = larg_tribes_pos_ntwk, FUN = sna::betweenness, mode = "graph")
```

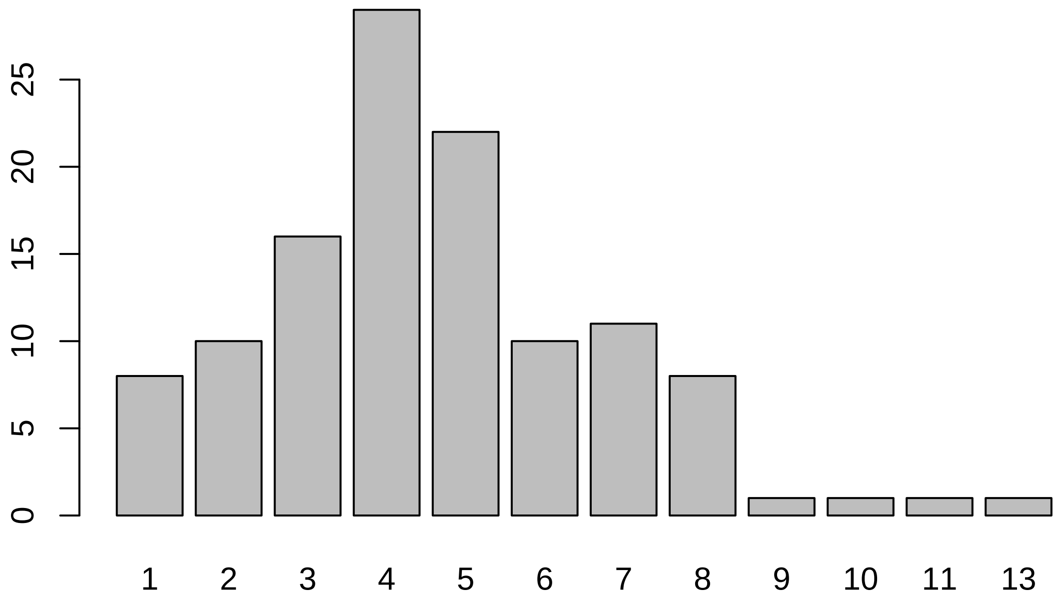
```
## [1] 0.5057851
```

```r
centralization(dat = larg_but_ntwk, FUN = sna::betweenness, mode = "graph")
```

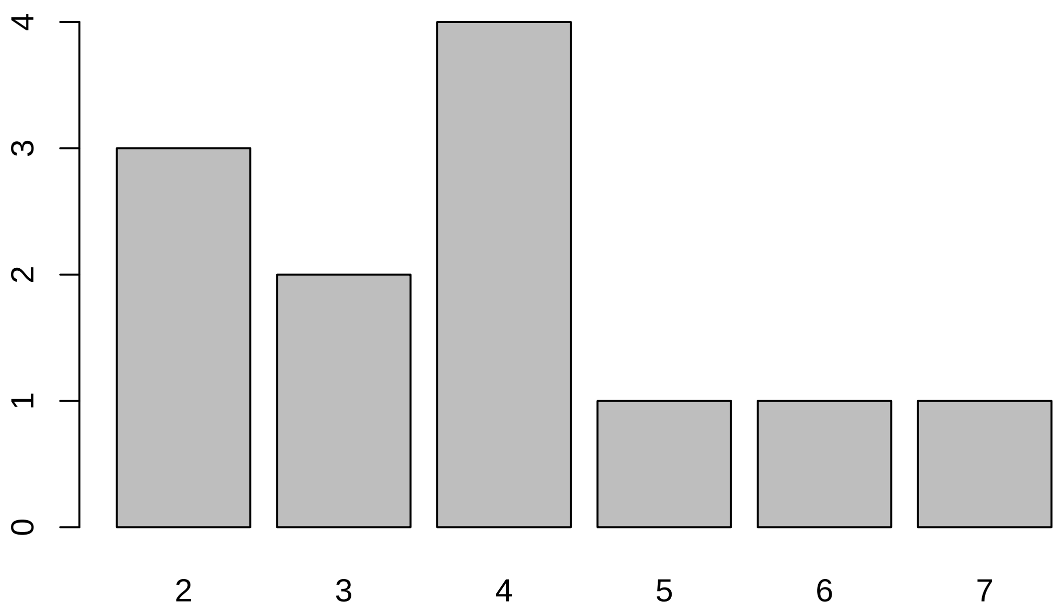
```
## [1] 0.3083979
```

d.

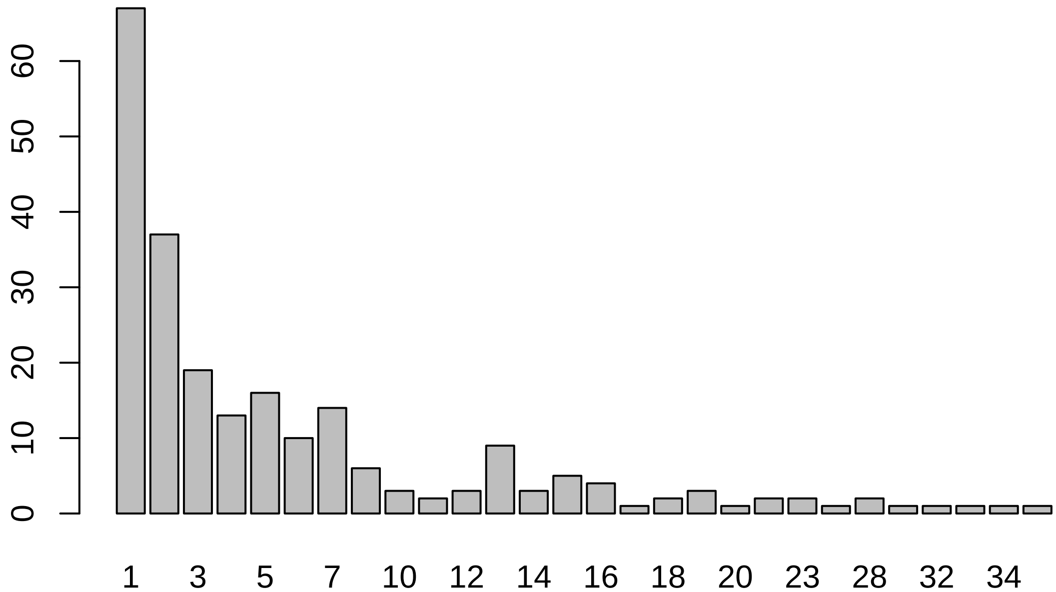
```
degree_health <- sna::degree(larg_health_ntwk, gmode = "graph") %>% table()
degree_tribes <- sna::degree(larg_tribes_pos_ntwk, gmode = "graph") %>% table()
degree_but <- sna::degree(larg_but_ntwk, gmode = "graph") %>% table()

barplot(degree_health)
```




```
barplot(degree_tribes)
```




```
barplot(degree_but)
```




```r
summary(degree_health)
```

```
## Number of cases in table: 118
## Number of factors: 1
```

```r
summary(degree_tribes)
```

```
## Number of cases in table: 12
## Number of factors: 1
```

```r
summary(degree_but)
```

```
## Number of cases in table: 230
## Number of factors: 1
```

```r
health_summary <- data.frame(
  Centrality = c("Degree", "Eigenvector", "Closeness", "Betweenness"),
  Min = c(min(degree_health), min(evcent_health), min(close_health), min(between_health)),
  Max = c(max(degree_health), max(evcent_health), max(close_health), max(between_health)),
  Mean = c(mean(degree_health), mean(evcent_health), mean(close_health), mean(between_health)),
  SD = c(sd(degree_health), sd(evcent_health), sd(close_health), sd(between_health))
)

tribes_summary <- data.frame(
  Centrality = c("Degree", "Eigenvector", "Closeness", "Betweenness"),
  Min = c(min(degree_tribes), min(evcent_tribes_large), min(close_tribes), min(between_tribes)),
  Max = c(max(degree_tribes), max(evcent_tribes_large), max(close_tribes), max(between_tribes)),
  Mean = c(mean(degree_tribes), mean(evcent_tribes_large), mean(close_tribes), mean(between_tribes)),
  SD = c(sd(degree_tribes), sd(evcent_tribes_large), sd(close_tribes), sd(between_tribes))
)

but_summary <- data.frame(
  Centrality = c("Degree", "Eigenvector", "Closeness", "Betweenness"),
  Min = c(min(degree_but), min(evcent_but_larg), min(close_but), min(between_but)),
  Max = c(max(degree_but), max(evcent_but_larg), max(close_but), max(between_but)),
  Mean = c(mean(degree_but), mean(evcent_but_larg), mean(close_but), mean(between_but)),
  SD = c(sd(degree_but), sd(evcent_but_larg), sd(close_but), sd(between_but))
)

health_summary
```

```
##    Centrality          Min          Max         Mean           SD
## 1      Degree 1.000000e+00   29.0000000   9.83333333   8.91202796
## 2 Eigenvector 7.798632e-05    0.3584209   0.05826042   0.07158017
## 3   Closeness 1.263499e-01    0.3062827   0.22412407   0.03862193
## 4 Betweenness 0.000000e+00 1669.5021025 211.32203390 293.77610372
```

```r
tribes_summary
```

```
##    Centrality        Min        Max      Mean        SD
## 1      Degree 1.00000000  4.0000000 2.0000000 1.2649111
## 2 Eigenvector 0.04199026  0.4584902 0.2436420 0.1617133
## 3   Closeness 0.39285714  0.7333333 0.5161171 0.0977180
## 4 Betweenness 0.00000000 31.0000000 5.5000000 9.2790630
```

```r
but_summary
```

```
##    Centrality          Min          Max         Mean           SD
## 1      Degree 1.000000e+00   67.0000000   8.21428571  13.97446651
## 2 Eigenvector 6.203945e-08    0.2672418   0.03539118   0.05575668
## 3   Closeness 1.363095e-01    0.4053097   0.27444746   0.05069430
## 4 Betweenness 0.000000e+00 8334.8406842 318.80869565 785.54773091
```

These measures capture different aspects of node importance and centrality, so while there may be correlations between them, they are not measuring the same thing. Eigenvector centrality might emphasize the influence of connected nodes, while closeness focuses on proximity, and betweenness highlights the node's control over information flow.

e.

```
deg_cent_health <- centralization(larg_health_ntwk, FUN = sna::degree, mode = "graph")
deg_cent_tribes <- centralization(larg_tribes_pos_ntwk, FUN = sna::degree, mode = "graph")
deg_cent_but <- centralization(larg_but_ntwk, FUN = sna::degree, mode = "graph")

close_cent_health <- centralization(larg_health_ntwk, FUN = sna::closeness, mode = "graph")
close_cent_tribes <- centralization(larg_tribes_pos_ntwk, FUN = sna::closeness, mode = "graph")
close_cent_but <- centralization(larg_but_ntwk, FUN = sna::closeness, mode = "graph")

between_cent_health <- centralization(larg_health_ntwk, FUN = sna::betweenness, mode = "graph")
between_cent_tribes <- centralization(larg_tribes_pos_ntwk, FUN = sna::betweenness, mode = "graph")
between_cent_but <- centralization(larg_but_ntwk, FUN = sna::betweenness, mode = "graph")

evcent_cent_health <- centralization(larg_health_ntwk, FUN = sna::evcent, mode = "graph")
evcent_cent_tribes <- centralization(larg_tribes_pos_ntwk, FUN = sna::evcent, mode = "graph")
evcent_cent_but <- centralization(larg_but_ntwk, FUN = sna::evcent, mode = "graph")

centralization_summary <- data.frame(
  Network = c("ppi", "addhealth", "tribes"),
  Degree = round(c(deg_cent_but, deg_cent_health, deg_cent_tribes),2),
  Closeness = round(c(close_cent_but, close_cent_health, close_cent_tribes),2),
  Betweenness = round(c(between_cent_but, between_cent_health, between_cent_tribes),2),
  Eigenvector = round(c(evcent_cent_but, evcent_cent_health, evcent_cent_tribes),2)
)

centralization_summary
```

```
##     Network Degree Closeness Betweenness Eigenvector
## 1       ppi   0.13      0.26        0.31        0.33
## 2 addhealth   0.07      0.17        0.22        0.43
## 3    tribes   0.35      0.50        0.51        0.36
```

Question 2

a.

```
flom_ntwk <- network(flomarriage, directed = FALSE)
larg_flom <- component.dist(flom_ntwk)$membership == which.max(component.dist(flom_ntwk)$csize)

larg_flom_ntwk <- network(flom_ntwk[larg_flom, larg_flom], directed = FALSE)
flom_igraph <- graph_from_adjacency_matrix(as.matrix(as.sociomatrix(larg_flom_ntwk)))
```

```
ver_flom <- vertex_connectivity(as.undirected(flom_igraph))
edge_flom <- edge_connectivity(as.undirected(flom_igraph))
min_ver_flom <- min_separators(as.undirected(flom_igraph))
min_edge_flom <- min_cut(as.undirected(flom_igraph), value.only = FALSE)$cut

ver_flom
```

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

```
edge_flom
```

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

```
min_ver_flom
```

```
## [[1]]
## + 1/15 vertex, named, from 3c740db:
## [1] Albizzi
##
## [[2]]
## + 1/15 vertex, named, from 3c740db:
## [1] Guadagni
##
## [[3]]
## + 1/15 vertex, named, from 3c740db:
## [1] Medici
##
## [[4]]
## + 1/15 vertex, named, from 3c740db:
## [1] Salviati
```

```
min_edge_flom
```

```
## + 1/20 edge from 9089597 (vertex names):
## [1] Pazzi--Salviati
```

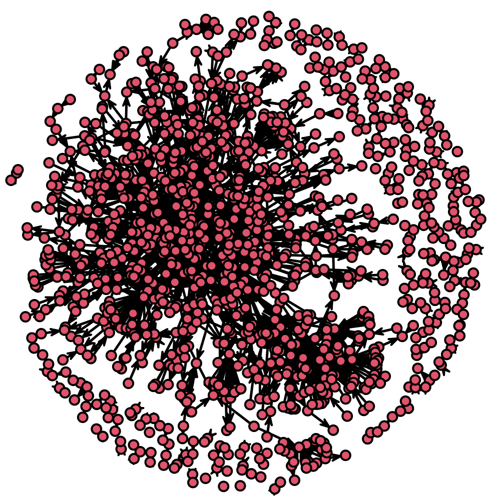
Question 3

a.

```
ccsb <- read.table("CCSB-Y2H.txt")
ccsb <- network(ccsb[, 1:2], directed = TRUE, loops = TRUE)
ccsb[cbind(1:network.size(ccsb), 1:network.size(ccsb))] <- 0
ccsb
```

```
##  Network attributes:
##   vertices = 1278
##   directed = TRUE
##   hyper = FALSE
##   loops = TRUE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges= 1641
##     missing edges= 0
##     non-missing edges= 1641
##
##  Vertex attribute names:
##     vertex.names
##
##  Edge attribute names not shown
```

```
plot(ccsb)
```



b.

```r
out_degrees <- sna::degree(ccsb, cmode = "outdegree")
in_degrees <- sna::degree(ccsb, cmode = "indegree")
total_degrees <- sna::degree(ccsb)
```

```r
zipf <- adpmle(total_degrees)
# summary(zipf)
yule <- ayulemle(total_degrees)
# summary(yule)
waring <- awarmle(total_degrees)
# summary(waring)
possion <- apoimle(total_degrees)
# summary(possion)
cmp <- acmpmle(total_degrees)
# summary(cmp)
```

c.

```r
aic_zipf <- lldpall(zipf$theta, total_degrees)
aic_yule <- llyuleall(yule$theta, total_degrees)
aic_waring <- llwarall(waring$theta, total_degrees)
aic_poisson <- llpoiall(possion$theta, total_degrees)
aic_cmp <- llcmpall(cmp$theta, total_degrees)
```

```r
data.frame(zipf = aic_zipf,
           yule = aic_yule,
           waring = aic_waring,
           possion = aic_poisson,
           cmp = aic_cmp)
```

```
##              zipf      yule    waring   possion       cmp
## np          3.000     3.000     3.000     1.000     3.000
## log-lik -2165.233 -2149.717 -2145.538 -3861.345 -1572.314
## AICC     4336.484  4305.453  4297.095  7724.693  3150.647
## BIC      4351.925  4320.894  4312.535  7729.843  3166.088
```

The CMP model fits the data best according to all the measures we looked at, followed by the Waring, Yule, and Zipf models. The Poisson model does a very poor job and doesn't fit the data well at all. The CMP model works well because it can handle some of the complexity in the data that the other models can't, which is why it comes out on top. The other models still do a decent job, but the CMP model is the best choice based on the results.

Question 4

a.

```
ccsb <- read.table("CCSB-Y2H.txt")
undirected_ccsb_network <- network(ccsb[, 1:2], directed = FALSE, loops = TRUE)
undirected_ccsb_network[cbind(1:network.size(undirected_ccsb_network), 1:network.size(undirected_ccsb_ne
ccsb_comp <- component.dist(undirected_ccsb_network)
comp_size <- ccsb_comp$csize
largest_size <- max(comp_size)

largest_size
```

```
## [1] 964
```

```
length(comp_size)
```

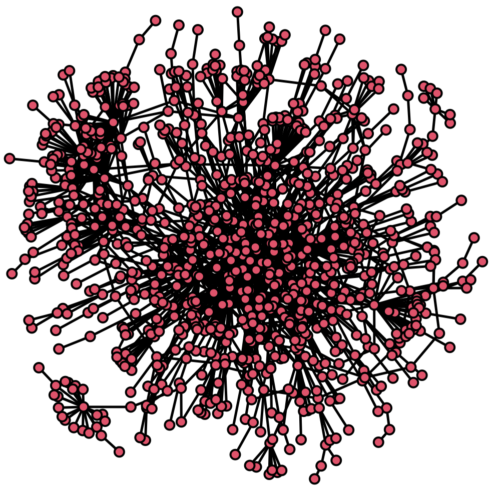
```
## [1] 162
```

The largest component of the network contains 964 nodes.

Yes, the network likely has a giant component, as the largest component includes 964 out of 1278 nodes.

b.

```
larg_ccsb <- component.dist(undirected_ccsb_network)$membership == which.max(component.dist(undirected_c
larg_ccsb_ntwk <- network(undirected_ccsb_network[larg_ccsb, larg_ccsb], directed = FALSE)
plot(larg_ccsb_ntwk)
```



c.

```
geodesic_distances <- geodist(undirected_ccsb_network, inf.replace = NA)$gdist
summary(c(geodesic_distances) %>% na.omit())
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   4.000   5.000   5.357   6.000  14.000
```

```
N = 1278
L = 964

proportion_reachable = ((L * (L - 1)) / 2) / ((N * (N - 1)) / 2)
proportion_reachable
```

```
## [1] 0.5688288
```

```
mean_geodesic_distance <- mean(geodesic_distances, na.rm = TRUE)
mean_geodesic_distance
```

```
## [1] 5.356807
```

```
node_degrees <- sna::degree(undirected_ccsb_network)
num_isolates <- sum(node_degrees == 0)
num_isolates
```

```
## [1] 45
```