INFX 576: Problem Set 2 - Structural Properties*

Harkar Talwar

Due: Friday, April 13, 2018

Collaborators: N/A

Instructions:

Before beginning this assignment, please ensure you have access to R and RStudio.

1. Download the `problemset2.Rmd` file from Canvas. You will also need the `problemset2_data.Rdata` file which contains the network datasets needed for this assignment.
2. Replace the "Insert Your Name Here" text in the `author:` field with your own full name. Any collaborators must be listed on the top of your assignment.
3. Be sure to include well-documented (e.g. commented) code chucks, figures and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text.
4. Collaboration on problem sets is acceptable, and even encouraged, but each student must turn in an individual write-up in his or her own words and his or her own work. The names of all collaborators must be listed on each assignment. Do not copy-and-paste from other students' responses or code.
5. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click `Knit PDF`, rename the R Markdown file to `YourLastName_YourFirstName_ps2.Rmd`, knit a PDF and submit the PDF file on Canvas.

Setup:

In this problem set you will need, at minimum, the following R packages.

```r
# Load standard libraries
library(statnet)
library(xtable)

# Load data
load("problemset2_data.Rdata")
ls() # Print objects in workspace to see what is available
```

```
## [1] "sampson"        "silsys.ad.ilas" "silsys.fr.ilas"
```

Problem 1: Node-Level Indices and Hypothesis Tests

In the data for this assignment, you will find the following network objects: `silsys.ad.ilas` and `silsys.fr.ilas`. These are network objects containing data from David Krackhardt's famous Silican Valley Systems study.[1] The two networks consist of advice-seeking ties and friendship ties (respectively). In addition each network contains several other attributes.

*Problems originally written by C.T. Butts (2009)

[1] Krackhardt, David. (1990) "Assessing the Political Landscape: Structure, Cognition, and Power in Organizations." *ASQ*, 35(2): 342-369.

(a) Computing Node-Level Indices

Compute indegree, outdegree, betweenness and eigenvector centrality scores for all individuals in each of the two networks. A useful trick to combine vectors or matrices `a`, `b`, and `c` into a single matrix using the `cbind` command as follows: `cbind(a,b,c)`. Print the centrality scores.

Centrality scores for Advice Network

```r
# Computations for advice seeking network
indeg <- degree(silsys.ad.ilas, cmode = "indegree") # compute indegree
outdeg <- degree(silsys.ad.ilas, cmode = "outdegree") # compute outdegree
bet <- betweenness(silsys.ad.ilas) # compute betweenness
evcent <- evcent(silsys.ad.ilas) # compute Eigenvector Centrality
# Concatenate the values along columns
centrality.ad <- cbind(Indegree = indeg, Outdegree = outdeg,
                       Betweenness = round(bet, 3),
                       EigenVectorCentrality = round(evcent, 3))
# Extract the names of actors in the network and concatenate with the measures
centrality.ad.names <- cbind(Name = silsys.ad.ilas %v% "vertex.names", centrality.ad)
# Print the results
print(centrality.ad.names)
```

```
##       Name     Indegree Outdegree Betweenness EigenVectorCentrality
##  [1,] "Abe"    "0"      "4"       "0"         "0.085"
##  [2,] "Bob"    "1"      "2"       "10"        "0.046"
##  [3,] "Carl"   "1"      "1"       "0"         "0.011"
##  [4,] "Dale"   "0"      "0"       "0"         "0"
##  [5,] "Ev"     "19"     "1"       "49.128"    "0.035"
##  [6,] "Fred"   "9"      "3"       "33"        "0.053"
##  [7,] "Gary"   "0"      "2"       "0"         "0.017"
##  [8,] "Hal"    "2"      "3"       "0.583"     "0.029"
##  [9,] "Ivo"    "0"      "1"       "0"         "0.01"
## [10,] "Jack"   "5"      "2"       "2.267"     "0.013"
## [11,] "Ken"    "0"      "0"       "0"         "0"
## [12,] "Len"    "3"      "7"       "80.283"    "0.296"
## [13,] "Mel"    "5"      "15"      "124.628"   "0.464"
## [14,] "Nan"    "0"      "1"       "0"         "0.007"
## [15,] "Ovid"   "2"      "4"       "6.461"     "0.092"
## [16,] "Pat"    "7"      "9"       "143.494"   "0.337"
## [17,] "Quincy" "10"     "3"       "48.167"    "0.046"
## [18,] "Robin"  "2"      "3"       "25.5"      "0.05"
## [19,] "Steve"  "19"     "7"       "370.928"   "0.173"
## [20,] "Tom"    "4"      "3"       "5.211"     "0.081"
## [21,] "Upton"  "7"      "3"       "16.583"    "0.053"
## [22,] "Vic"    "1"      "6"       "8.333"     "0.075"
## [23,] "Walt"   "0"      "4"       "0"         "0.116"
## [24,] "Rick"   "6"      "11"      "190.711"   "0.174"
## [25,] "York"   "5"      "15"      "57.328"    "0.473"
## [26,] "Zoe"    "1"      "2"       "0.167"     "0.046"
## [27,] "Alex"   "11"     "2"       "36.611"    "0.02"
## [28,] "Ben"    "2"      "4"       "0"         "0.293"
## [29,] "Chris"  "7"      "6"       "73.667"    "0.192"
## [30,] "Dan"    "7"      "3"       "38.5"      "0.049"
## [31,] "Earl"   "0"      "2"       "0"         "0.103"
## [32,] "Fran"   "0"      "2"       "0"         "0.046"
## [33,] "Gerry"  "0"      "5"       "0"         "0.13"
```

```
## [34,] "Hugh"   "0"      "2"       "0"         "0.012"
## [35,] "Irv"    "6"      "5"       "19.567"    "0.243"
## [36,] "Jim"    "5"      "4"       "8.883"     "0.162"
```

Centrality scores for Frienship Network

```r
# Computations for friendship network
indeg <- degree(silsys.fr.ilas, cmode = "indegree") # compute indegree
outdeg <- degree(silsys.fr.ilas, cmode = "outdegree") # compute outdegree
bet <- betweenness(silsys.fr.ilas) # compute betweenness
evcent <- evcent(silsys.fr.ilas) # compute Eigenvector Centrality
# Concatenate the values along columns
centrality.fr <- cbind(Indegree = indeg, Outdegree = outdeg,
                       Betweenness = round(bet, 3),
                       EigenVectorCentrality = round(evcent, 3))
# Extract the names of actors in the network and concatenate with the measures
centrality.fr.names <- cbind(Name = silsys.fr.ilas %v% "vertex.names", centrality.fr)
# Print the results
print(centrality.fr.names)
```

```
##       Name     Indegree Outdegree Betweenness EigenVectorCentrality
##  [1,] "Abe"    "0"      "1"       "0"         "0.005"
##  [2,] "Bob"    "6"      "4"       "90.8"      "0.128"
##  [3,] "Carl"   "2"      "4"       "46.029"    "0.041"
##  [4,] "Dale"   "8"      "7"       "101.62"    "0.247"
##  [5,] "Ev"     "2"      "1"       "0"         "0.021"
##  [6,] "Fred"   "4"      "2"       "46.629"    "0.035"
##  [7,] "Gary"   "3"      "2"       "0.976"     "0.076"
##  [8,] "Hal"    "1"      "2"       "0"         "0.011"
##  [9,] "Ivo"    "3"      "2"       "6.849"     "0.118"
## [10,] "Jack"   "1"      "1"       "0"         "0.019"
## [11,] "Ken"    "9"      "7"       "30.202"    "0.254"
## [12,] "Len"    "2"      "4"       "67.555"    "0.078"
## [13,] "Mel"    "6"      "7"       "75.163"    "0.231"
## [14,] "Nan"    "5"      "7"       "19.664"    "0.217"
## [15,] "Ovid"   "4"      "3"       "3.25"      "0.142"
## [16,] "Pat"    "5"      "4"       "75.932"    "0.08"
## [17,] "Quincy" "0"      "0"       "0"         "0"
## [18,] "Robin"  "4"      "4"       "7.493"     "0.154"
## [19,] "Steve"  "8"      "6"       "124.702"   "0.143"
## [20,] "Tom"    "10"     "10"      "104.753"   "0.34"
## [21,] "Upton"  "5"      "4"       "61.725"    "0.094"
## [22,] "Vic"    "1"      "3"       "0"         "0.131"
## [23,] "Walt"   "1"      "2"       "0"         "0.106"
## [24,] "Rick"   "11"     "9"       "108.648"   "0.264"
## [25,] "York"   "0"      "0"       "0"         "0"
## [26,] "Zoe"    "3"      "2"       "6.783"     "0.037"
## [27,] "Alex"   "3"      "3"       "2.532"     "0.099"
## [28,] "Ben"    "0"      "1"       "0"         "0.019"
## [29,] "Chris"  "12"     "16"      "203.9"     "0.453"
## [30,] "Dan"    "5"      "5"       "21.357"    "0.155"
## [31,] "Earl"   "0"      "2"       "0"         "0.033"
## [32,] "Fran"   "0"      "0"       "0"         "0"
## [33,] "Gerry"  "8"      "7"       "89.149"    "0.292"
## [34,] "Hugh"   "6"      "7"       "16.1"      "0.291"
```

```
## [35,] "Irv"    "8"      "7"       "169.187"   "0.153"
## [36,] "Jim"    "1"      "1"       "0"         "0.012"
```

- Who are some of the most central individuals in the advice-seeking network? In the friendship network?

1. Advice Seeking Network

- On the basis of 'Indegree', Ev (19) and Steve (19) seem to be the most central, i.e., a large number of people seek advice from them.
- In terms of 'Outdegree', Mel (15) and York (15) are the two most central people, i.e., they seek advice from the highest number of people.
- From the perspective of 'Betweenness', Steve (371) has the highest score, followed by Rick (190). This implies that they act as key conduits of information and can influence the group by withholding or distorting information.
- Lastly, Eigenvector centrality is high for Mel (0.46) and York (0.47), implying that both are influential in the network by virtue of being linked to other influential individuals.

2. Friendship Network

- On the basis of 'Indegree', Chris (12), followed by Rick (11), seem to be the most central persons, with a lot of friendship ties coming in.
- In terms of 'Outdegree', Chris (16), followed by Tom (10) are the two most central people, i.e., they have a high number of friendship ties going outward.
- From the perspective of 'Betweenness', Chris (204), followed by Irv (169), have the highest scores. This implies that they act as key conduits of information in the friendship network and can influence the group by withholding or distorting information.
- Lastly, Eigenvector centrality is high for Chris (0.45) followed by Tom (0.34), implying that both are influential in the network by virtue of being friends with other influential individuals.

(b) Comparing Node-Level Indices

The `cor` command calculates correlations. You can apply this function to a matrix to compute the correlation matrix - correlations for all pairs of columns. Compute the within and between network correlation matrices for the centrality scores you computed in part (a). Print this table and answer the following:

Correlation within advice network

```r
round(cor(centrality.ad), 3)
```

```
##                        Indegree Outdegree Betweenness EigenVectorCentrality
## Indegree                  1.000     0.205       0.653                 0.161
## Outdegree                 0.205     1.000       0.555                 0.893
## Betweenness               0.653     0.555       1.000                 0.428
## EigenVectorCentrality     0.161     0.893       0.428                 1.000
```

Correlation within friendship network

```r
round(cor(centrality.fr), 3)
```

```
##                        Indegree Outdegree Betweenness EigenVectorCentrality
## Indegree                  1.000     0.911       0.823                 0.883
## Outdegree                 0.911     1.000       0.815                 0.941
## Betweenness               0.823     0.815       1.000                 0.663
## EigenVectorCentrality     0.883     0.941       0.663                 1.000
```

Correlation between the two networks

```r
round(cor(centrality.ad, centrality.fr), 3)
```

```
##                        Indegree Outdegree Betweenness EigenVectorCentrality
```

```
## Indegree                   0.133     0.022       0.233                      -0.099
## Outdegree                  0.118     0.148       0.277                       0.095
## Betweenness                0.355     0.262       0.445                       0.143
## EigenVectorCentrality      0.029     0.090       0.291                       0.007
```

- Does centrality in the advice-seeking network correspond (or not) to centrality in the friendship network?
  The correlation matrix between the centrality measures in the advice-seeking and friendship networks proves that centrality in one of the networks does not imply the same in the other. To illustrate, the correlation between indegree in one network and all other centrality measures in the other remains below 0.25. This means that a vertex having say a high indegree (and thus centrality) in one network is unlikely to have a high value for any of the centrality measures in the other network, including indegree. In other words, a person giving advice to a lot of people at Silicon Valley Systems may not have a lot of friends and vice versa.

- What centrality measures are most strong correlated? Least strongly correlated?
  Following are the strong correlations (>=0.5) in centrality measures:
  - **Within the Advice Network**
    1. Indegree & Betweenness (0.65)
    2. Outdegree & Betweenness (0.55)
    3. Outdegree & Eigenvector Centrality (0.89, strongest)
  - **Within the Friendship Netork**
    All centrality measures are strongly correlated to each other in the friendship network, with a correlation of 0.8 or above. The only exception is the association among betweenness and EigenVectorCentrality (0.66).
  - **Between the two networks**
    As noted in the previous response, there is no strong correlation in the measures between the two networks. The highest value of correlation is for the betweenness measures in the two networks (0.44).

(c) Relating Node-Level Indices to Covariates

In the in-class demo you were given a function for testing the correlation between vectors using a permutation test. Using this function, assess the relationship between the "Charisma" (charisma, as rated by fellow employees) and "Potency" (ability to overcome opposition in order to achieve goals, as rated by fellow employees) vertex attributes and the centrality scores you computed in part (a).

Remember you can extract vertex attributes from network objects with the `%v%` operator or the `get.vertex.attribute` function. Report the results of these tests as a table showing the observed correlation of each attribute with each centrality measure, along with the two-sided p-value for the appropriate test in each case.

```r
# Create vectors for the covariates and centrality measures to be used
covariates <- c("Charisma", "Potency")
cent.scores <- c("Indegree", "Outdegree", "Betweenness", "EigenVectorCentrality")

# Define a simple test function
perm.cor.test<-function(x,y,niter=5000){
  # Observed test statistic
  c.obs<-cor(x,y,use="complete.obs")
  c.rep<-vector()
  for(i in 1:niter)
    # Test statistic for different permuted samples
    c.rep[i]<-cor(x,sample(y),use="complete.obs")
  # Return a list with the observed statistic and the p-value
  list(obs=c.obs, p.val=mean(abs(c.rep)>=abs(c.obs)))
```

```r
}

# Define a runner function that passes apt network parameters to the simple test function
perm.test.network<-function(network.obj, centrality, output.vectors){
  # Iterate for each covariate and centrality score
  for (i in 1:length(covariates)) {
    for (j in 1:length(cent.scores)) {
      # invoke the simple test function with vectors representing the
      # covariate and the centrality measures for all nodes in the network
      test <- perm.cor.test(network.obj %v% covariates[i],
                            centrality[,cent.scores[j]])
      # store the results in an output vector
      output.vectors[i,(j*2)-1] <- round(test$obs, digits = 3)
      output.vectors[i,j*2] <- test$p.val
    }
  }
  return(output.vectors)
}
# Initialize the output vector
output.vectors <- array(dim = c(2,8))
# Invoke the test runner for advice network
output.vectors = perm.test.network(silsys.ad.ilas, centrality.ad, output.vectors)
# Initialize the output vector for the second invocation
output.vectors2 <- array(dim = c(2,8))
# Invoke the test runner for the friendship network
output.vectors2 = perm.test.network(silsys.fr.ilas, centrality.fr, output.vectors2)
# Concatenate the outputs (row-wise)
output.vectors.merged <- rbind(output.vectors, output.vectors2)
# Create vectors for static display data
covariates.names <- rep(covariates, 2)
test.param.names <- c("Indegree", "p-value", "Outdegree", "p-value",
                      "Betweenness", "p-value", "EigenVectorCentrality", "p-value")
network.names <- c(rep("silsys.ad.ilas",2), rep("silsys.fr.ilas",2))

# Create the final merged output
merged.output.final <- cbind(Network = network.names,
                             Covariate = covariates.names,
                             Indegree = output.vectors.merged[,1],
                             pvalue = output.vectors.merged[,2],
                             Outdegree = output.vectors.merged[,3],
                             pvalue = output.vectors.merged[,4],
                             Betweenness = output.vectors.merged[,5],
                             pvalue = output.vectors.merged[,6],
                             EigenVectorCentrality = output.vectors.merged[,7],
                             pvalue = output.vectors.merged[,8])
```

```r
# Render the results as a table
x.rescale <- xtable(merged.output.final,
                    caption='Correlation Between Covariates and Centrality')
print(x.rescale, scalebox = 0.8, caption.placement = "top", type = "latex")
```

% latex table generated in R 3.4.3 by xtable 1.8-2 package % Fri Apr 13 12:20:11 2018

- How do charisma and potency appear to relate to positional structure at Silicon Valley Systems?

Table 1: Correlation Between Covariates and Centrality

	Network	Covariate	Indegree	pvalue	Outdegree	pvalue	Betweenness	pvalue	EigenVectorCentrality	pvalue
1	silsys.ad.ilas	Charisma	0.081	0.641	-0.456	0.0066	-0.134	0.413	-0.388	0.0198
2	silsys.ad.ilas	Potency	0.583	2e-04	-0.249	0.1464	0.241	0.164	-0.212	0.2246
3	silsys.fr.ilas	Charisma	-0.028	0.8718	-0.06	0.7222	-0.114	0.512	0.001	0.9952
4	silsys.fr.ilas	Potency	-0.051	0.7682	-0.11	0.525	-0.062	0.727	-0.136	0.4374

The association between positional structure and non-structural attributes (Charisma, Potency) of the two networks based on Table 1. is described below. The analysis is based on a significance value of $\alpha = 0.05$.

1. Advice Network

- Charisma
  - Charisma seems to be inversely correlated to the Outdegree, with a p-value less than our significance level α. This implies that charismatic individuals are less likely to seek advice than individuals who are not rated highly on the charisma attribute.
  - Moreover, charisma is also slightly inversely correlated with the Eigenvector centrality with a statistically significant p-value, a finding which could indicate that charismatic leaders are less likely to have ties with other influential leaders.
- Potency
  - Potency and Indegree are positively correlated, with a statistically significant p-value. Thus we can infer that at Silicon Valley Systems, individuals with high potency are likely to have more people seeking advice from them (an intuitive outcome). Other correlations in the advice network are not statistically significant.

2. Friendship Network

- In the friendship network, we see that Charisma and Potency do not have a statistically significant association with any of the centrality measures. Therefore, people at Silicon Valley Systems, do not make friends based on an individuals charisma or potency.

Problem 2: Graph-Level Indices

Consider the Sampson monk data[2]. Sampson collected various relationships between several monks at a monastery. Suppose we divide the types of social ties into positive and negative relationship types as follows:

- Positive Relationships: Esteem, Influence, LikeT1, LikeT2, LikeT3, and Praise
- Negative Relationships: Disesteem, NegInfluence, Dislike, and Blame

Using a vector permutation test, evaluate the questions below.

Note: The test statistic chosen for vector permutation test is computed by subtracting the sum of graph level index values for negative relations from the sum of graph level index values for positive relations. The analysis is based on a significance value of α = 0.05.

```r
# Function to perform permutation test for graph level indexes
perm.test.graph <- function(graph.index, network.positive, niter = 5000) {
  # Observed test statistic
  c.obs<-sum(graph.index[network.positive == TRUE]) -
    sum(graph.index[network.positive == FALSE])
  c.rep<-vector()
  for(i in 1:niter) {
    # Test statistic for different permuted samples
    graph.index.perm <- sample(graph.index)
```

[2] F. S. Sampson. A novitiate in a period of change: An experimental and case study of social relationships. PhD thesis, Cornell University. 1968.

```r
    c.rep[i]<- sum(graph.index.perm[network.positive == TRUE]) -
    sum(graph.index.perm[network.positive == FALSE])
  }
  # Return a list with the observed statistic and the p-value
  return (list(obs=c.obs, rep = c.rep, p.val=mean(abs(c.rep)>=abs(c.obs))))
}
```

(a) Are positive relations more reciprocal (relative to density) than negative ones?

H_o : Networks depicting positive relations are not more reciprocal than those depicting negative relations.

```r
# Compute the graph level index (reciprocity relative to density)
graph.density.vector <- grecip(sampson, measure = "edgewise.lrr")
# Create vector representing positive/negative nature of the sampson networks
sampson.is.positive <- c(TRUE,FALSE,TRUE,FALSE,TRUE,TRUE,TRUE,FALSE,TRUE,FALSE)
# Invoke the vector permutation test function
test_response = perm.test.graph(graph.density.vector, sampson.is.positive)
# Plot the frequency distribution
hist(test_response[['rep']], main = "Frequency Distribution of Repeated Test Statistic",
     xlab = "Value of Test Statistic\n", sub = "\nFigure 1.")
abline(v = test_response[['obs']], col=3)
```

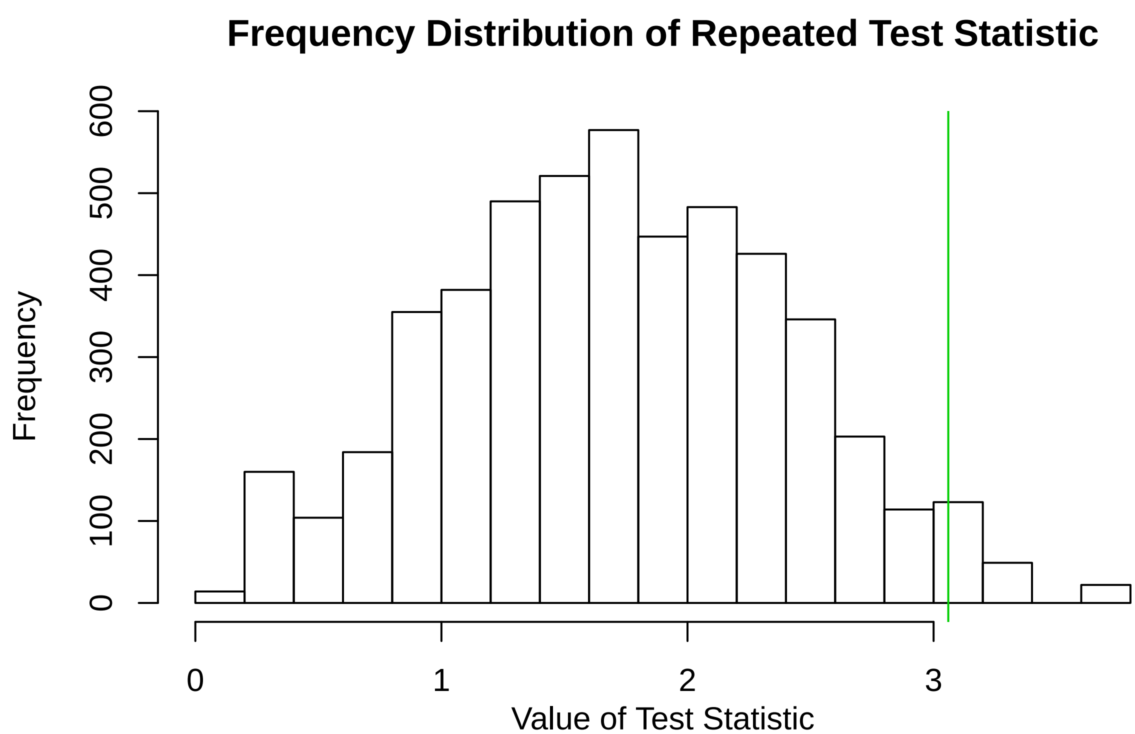

Figure 1.

```r
cat("Observed Test Statistic:", test_response[['obs']], "\n")
```

```
## Observed Test Statistic: 3.059769
```

```r
cat("p-value:", test_response[['p.val']])
```

```
## p-value: 0.0344
```

(b) Are positive relations more transitive (relative to density) than negative ones?

H_o : Networks depicting positive relations are not more transitive than those depicting negative relations.

```r
# Compute the graph level index (transitivity)
graph.trans.vector <- gtrans(sampson)/gden(sampson)
# Invoke the vector permutation test function
test_response = perm.test.graph(graph.trans.vector, sampson.is.positive)
# Plot the frequency distribution
hist(test_response[['rep']], main = "Frequency Distribution of Repeated Test Statistic",
     xlab = "Value of Test Statistic\n", sub = "\nFigure 2.")
abline(v = test_response[['obs']], col=3)
```

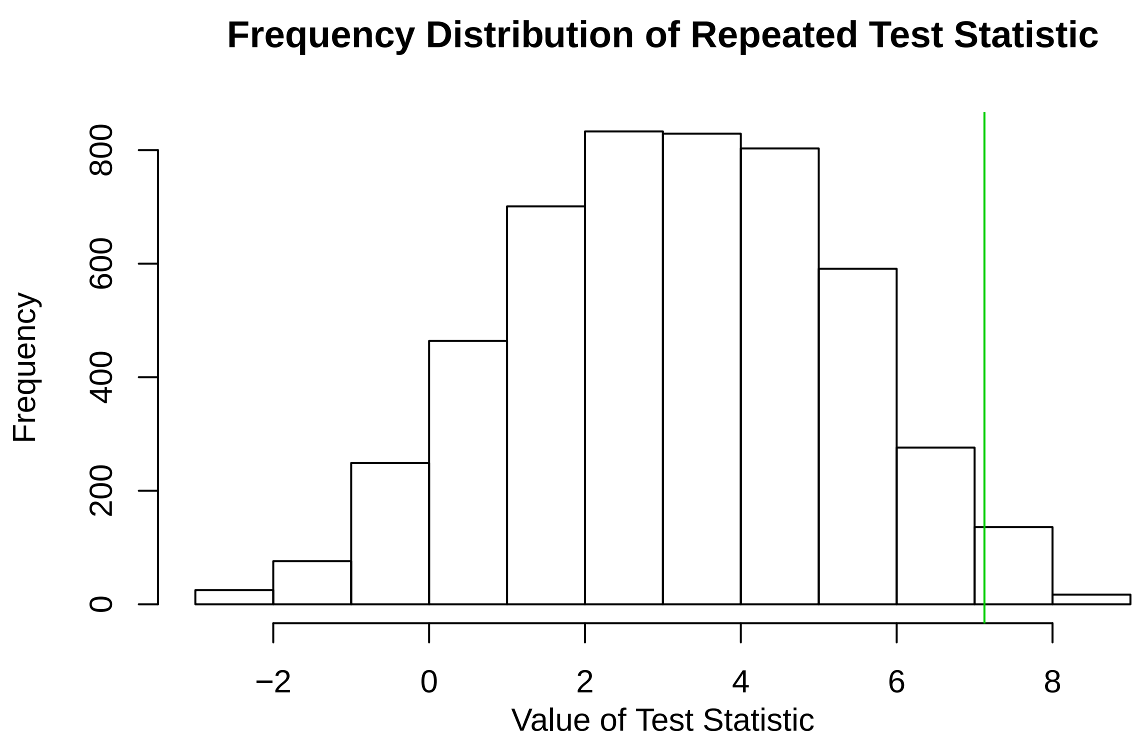

Figure 2.

```r
cat("Observed Test Statistic:", test_response[['obs']], "\n")
```

```
## Observed Test Statistic: 7.126207
```

```r
cat("p-value:", test_response[['p.val']])
```

```
## p-value: 0.0214
```

(c) Discuss the findings from part (a) and part (b).

- Positive Relations and Reciprocity:
  Since we have a positive value (3.06) for our test statistic, and the p-value (0.03) in part (a) is less than the significance level of 0.05, we can reject the null hypothesis that networks depicting positive relations are not more reciprocal than those depicting negative relations.
  This means that positive relations such as Influence, Like and Praise are more likely to be symmetric in nature than negative relations such as Blame and Dislike. So, if Monk A at the monastery praises Monk B, it is likely that Monk B also praises Monk A. It could also imply the presence of null dyads. However, that possibility is ruled out as shown below, since all the symmetric ties for positive relation networks are mutual in nature (no null dyads)

```r
# Check for isolates in each of the positive relations
isolates(sampson[sampson.is.positive])
```

```
## $Esteem
## integer(0)
##
## $Influence
## integer(0)
##
## $LikeT1
## integer(0)
##
## $LikeT2
## integer(0)
##
## $LikeT3
## integer(0)
##
## $Praise
## integer(0)
```

- Positive Relations and Transitivity:
  Here we have a positive value (7.12) shown above for our test statistic and from Figure 2, it can be seen that the probability of observing a more extreme test statistic in the direction of the alternate hypothesis is 0.02. Since the p-value (0.02) in part (b) is less than the significance level of 0.05, we can reject the null hypothesis that networks depicting positive relations are not more transitive than those depicting negative relations. Here the transitive constraint corresponds to a->b->c => a->c. Therefore, if Monk A at the monastery praises Monk B, and Monk B praises Monk C, it is likely that Monk A praises Monk C as well. Further such a transitive relation is more probable in case of positive relations such as Influence and Praise, compared to negative relations like Blame and Dislike.