INFX 576: Problem Set 2 - Graph Level Indices and CUG Tests*

Avanti Chande

Due: Thursday, January 26, 2017

Collaborators: Gosuddin Siddiqi, Jay Chauhan.

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 three different 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.

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

```
## Warning: package 'statnet' was built under R version 3.2.5

## Warning: package 'tergm' was built under R version 3.2.5

## Warning: package 'statnet.common' was built under R version 3.2.5

## Warning: package 'ergm' was built under R version 3.2.5

## Warning: package 'network' was built under R version 3.2.5

## Warning: package 'networkDynamic' was built under R version 3.2.5

## Warning: package 'ergm.count' was built under R version 3.2.5

## Warning: package 'sna' was built under R version 3.2.5
```

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

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

```
## [1] "kaptail.ins" "mids_1993"   "sampson"
```

Problem 1: Graph-Level Indices

Consider the Sampson monk data[1]. 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.

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

```r
#Calculating Reciprocity
reciprocity = grecip(sampson, measure = "edgewise.lrr")
reciprocity
```

```
##       Esteem    Disesteem    Influence NegInfluence       LikeT1
##    0.6359888    0.4930708    0.9610551    0.5385964    1.0411232
##       LikeT2       LikeT3      Dislike       Praise        Blame
##    1.0386799    1.0740791    0.9136617    1.0355191    0.7813476
```

```r
#Permutation Vector Test Function
perm.cor.test<-function(x, niter=5000){  #Define a simple test function

  #Defining the relations vector: positive - p; negative - n
  relations = c("p", "n", "p", "n", "p", "p", "p", "n", "p", "n")
  observed_diff = sum(x[relations == "p"]) - sum(x[relations == "n"])
  observed_diff

  c.rep = vector()
  for(i in 1:niter)
    {
    relations_sample = sample(relations)
    c.rep[i]<- sum(x[relations_sample == "p"])-sum(x[relations_sample == "n"])
    }
  pvalue = mean(abs(c.rep)>=observed_diff)  #One-tailed pvalue
  return(pvalue)
}
```

```r
#Calling the Permutation Function
p1 = perm.cor.test(reciprocity)
p1
```

```
## [1] 0.0292
```

Ans: Considering 95% confidence(pvalue 0.05), the calculated pvalue is 0.034,. Hence, we can reject the null hypothesis that positive relations are not more reciprocal than negative relations. We therefore conclude that yes, positive relations are more reciprocal than negative relations in the sampson monastery dataset.

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

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

```r
#Calculating Transitivity
transitivity = log(gtrans(sampson)/gden(sampson))
transitivity
```

```
##       Esteem    Disesteem    Influence NegInfluence       LikeT1
##   0.65660805   0.28254575   0.83323597  -0.31666961  -0.06853888
##       LikeT2       LikeT3      Dislike       Praise        Blame
##   0.45675840   0.64129382  -0.23879386   1.04529265   0.41607931
```

```r
#Calling the Permutation Vector Test Function
p2 = perm.cor.test(transitivity)
p2
```

```
## [1] 0.0242
```

Ans. Considering 95% confidence (pvalue 0.05), the calculated pvalue is 0.0312, Hence, we can reject the null hypothesis that positive relations are not more transitive than negative ones and thus conclude that positive relations are more transitive than negative ones in the sampson monastery dataset.

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

The findings suggest that positive relations tend to be more reciporocal than negative ones. This means that positive networks like Esteem, Influence, Like and Praise, tend to be reciprocal among the sampson monastery monks. Negative ones like Disesteem, Blame and Negative Influence, are not very reciprocal.

In addition to that, evidence also suggests that positive relations tend to be more transitive. This means that if two monks have a positive relation, a third monk also might have a positive relation with the first one. On the other hand, if two monks have a negative relation, there are less chances that a third monk will also have a negative relation with the first one.

Thus, positive relations are more transitive and more reciprocal in the sampson monastery dataset.

Problem 2: Random Graphs

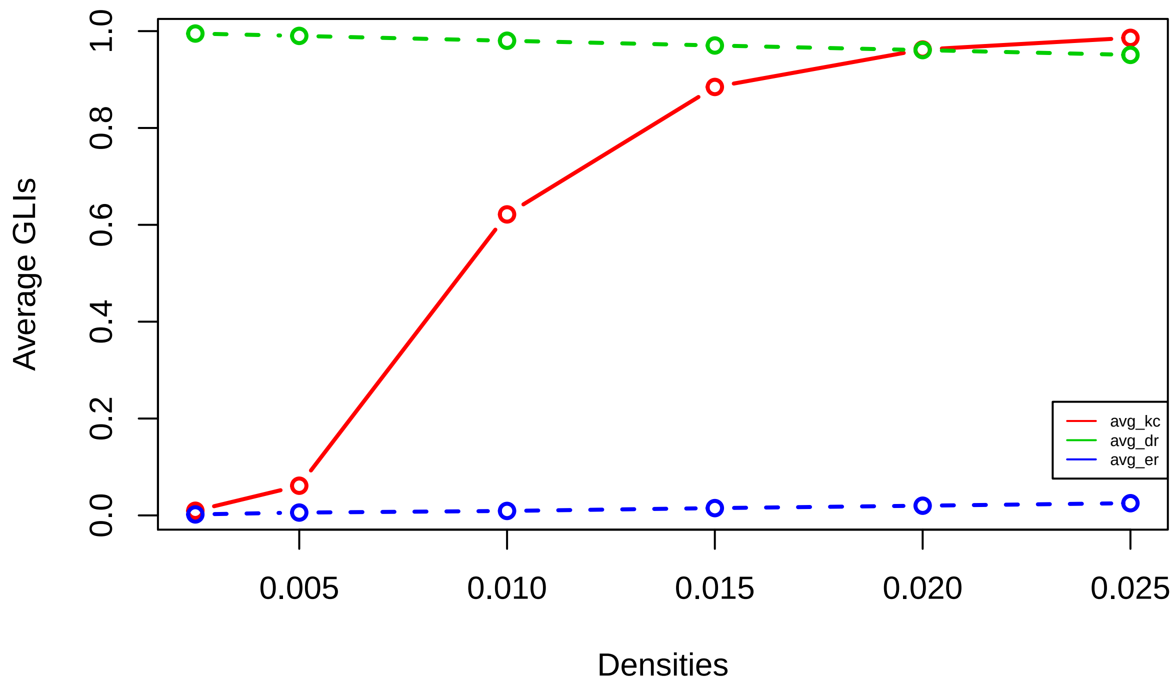
(a) Generating Random Graphs

Generate 100-node random directed graphs with expected densities of 0.0025, 0.005, 0.01, 0.015, 0.02, and 0.025, with at least 500 graphs per sample. Remember the `rgraph` function can draw more than one graph at a time. Plot the average Krackhardt connectedness, dyadic reciprocity, and edgewise reciprocity as a function of expected density. Use these to describe the baseline effect of increasing density on network structure.

```r
densities = c(0.0025, 0.005, 0.01, 0.015, 0.02, 0.025)

#Generating Random Graphs
g1 = rgraph(100, m=500, mode = "digraph", tprob = densities[1])
g2 = rgraph(100, m=500, mode = "digraph", tprob = densities[2])
g3 = rgraph(100, m=500, mode = "digraph", tprob = densities[3])
g4 = rgraph(100, m=500, mode = "digraph", tprob = densities[4])
g5 = rgraph(100, m=500, mode = "digraph", tprob = densities[5])
g6 = rgraph(100, m=500, mode = "digraph", tprob = densities[6])

#KrackHardts' Connectedness
kc1 = mean(connectedness(g1))
kc2 = mean((connectedness(g2)))
kc3 = mean((connectedness(g3)))
```

```r
kc4 = mean((connectedness(g4)))
kc5 = mean((connectedness(g5)))
kc6 = mean(connectedness(g6))

avg_kc = c(kc1,kc2,kc3,kc4,kc5,kc6)

#Dyadic Reciprocity

dr1= mean((grecip(g1)))
dr2= mean((grecip(g2)))
dr3= mean((grecip(g3)))
dr4= mean((grecip(g4)))
dr5= mean((grecip(g5)))
dr6= mean((grecip(g6)))
avg_dr= c(dr1, dr2,dr3,dr4,dr5,dr6)

#Edgewise Reciprocity

er1= mean((grecip(g1, measure = "edgewise")))
er2= mean((grecip(g2, measure = "edgewise")))
er3= mean((grecip(g3, measure = "edgewise")))
er4 = mean((grecip(g4, measure = "edgewise")))
er5= mean((grecip(g5, measure = "edgewise")))
er6= mean((grecip(g6, measure = "edgewise")))

avg_er = c(er1,er2,er3,er4,er5,er6)
```

```r
plot(densities, avg_kc, type="b", col=2, lwd=2, xlab = "Densities", ylab = "Average GLIs")
points(densities, avg_dr, type = "b", col=3, lwd=2, lty=2)
points(densities, avg_er, type = "b", col=4, lwd=2, lty=2)
legend(x="bottomright", inset = c(0,0.1), legend=c("avg_kc","avg_dr", "avg_er"), col=c(2,3,4), lty=1:1,
```



Ans. Average KrackHardt connectedness and density have a positive relationship.As the density of a network increases, its connectedness also increases. This means that with increased probability of possible edges, the fraction of weakly connected dyads also increase.

As the density of a network increases, dyadic reciprocity decreases slightly. This can be explained, since dyadic reciprocity includes mutual as well as null dyads. Therefore, with increase in density, null dyads decrease, with a possibility of assymetric dyads increasing. Therefore, the average dyadic reciprocity decreases as density of a network increases.

As the density of a network increases, edgewise reciprocity increases slightly. This can be explained, since edgewise reciprocity doesnot include the probability of null edges. However, as density increases, the possibility of an edge being assymentric or mutual increases. However, it highly depends on whether the assymmetric dyads increase or the number of mutual dyads increase.I want to say that the relationship cannot be really defined.

(b) Comparing GLIs

In this problem we will use the well-known social network dataset, collected by Bruce Kapferer in Zambia from June 1965 to August 1965, involves interactions among workers in a tailor shop as observed by Kapferer himself.[2] Here, an interaction is defined by Kapferer as "continuous uninterrupted social activity involving the participation of at least two persons"; only transactions that were relatively frequent are recorded.

Generate 500 random directed graphs whose dyad census is the same as that of `kaptail.ins`. Plot histograms for total degree centralization, betweenness centralization, transitivity, and Krackhardt connectedness from this random sample. On your plot mark the observed values of these statistics (from the `kaptail.ins` data)

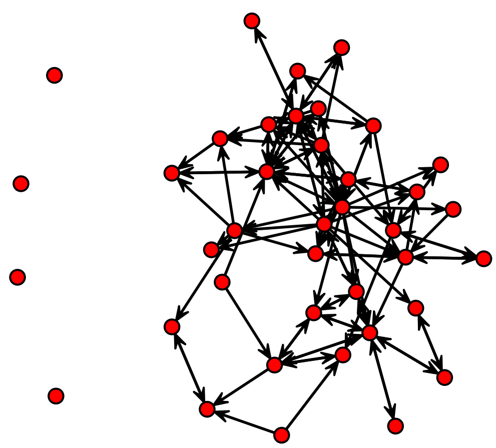
[2]Kapferer B. (1972). Strategy and transaction in an African factory. Manchester: Manchester University Press.

using a vertial line You mgith find the abline function helpful here. Try modifying the lwd argument to the plot function to make the vertical line stand out. How do the replicated graphs compare to the observed data.

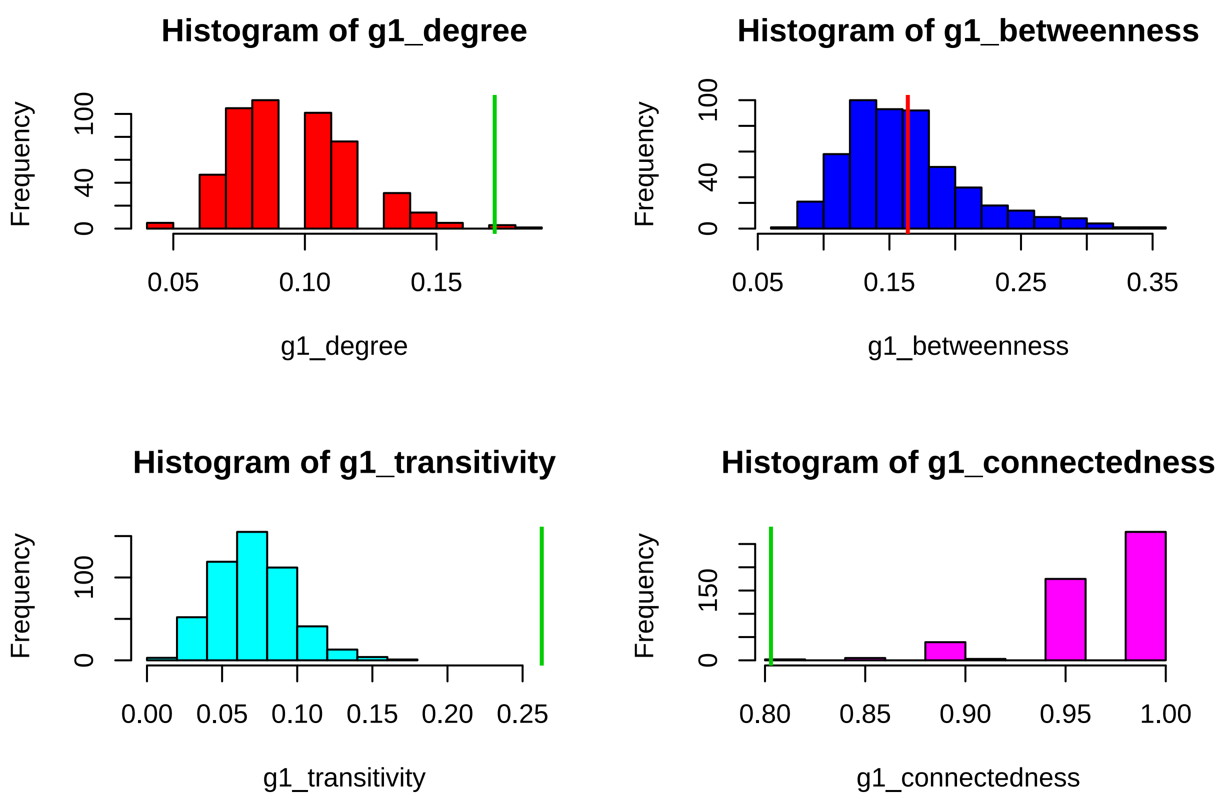
```
#Kaptail Dyad Census and Size Measures
dyad_census_kapt = dyad.census(kaptail.ins)
size_kapt = network.size(kaptail.ins)
size_kapt
```

```
## [1] 39
```

```
g1 = rguman(500,size_kapt,mut=dyad_census_kapt[1], asym = dyad_census_kapt[2], null = dyad_census_kapt[3

#Random Graph Centralization Measures
g1_degree = centralization(g1, degree)
g1_betweenness = centralization(g1, betweenness)
g1_transitivity = gtrans(g1)
g1_connectedness = connectedness(g1)


#Kaptail Centralization Measures
kapt_degree = centralization(kaptail.ins, degree)
kapt_betweenness = centralization(kaptail.ins, betweenness)
kapt_transitivity = gtrans(kaptail.ins)
kapt_connectedness = connectedness(kaptail.ins)


gplot(kaptail.ins)
```



```r
#Plots
par(mfrow = c( 2, 2 ))
hist(g1_degree, col=2)
abline(v=kapt_degree, col=3, lwd=2)

hist(g1_betweenness, col=4)
abline(v=kapt_betweenness, col=2, lwd=2)

hist(g1_transitivity, col=5,xlim = c(0, kapt_transitivity))
abline(v=kapt_transitivity, col=3, lwd=2)

hist(g1_connectedness, col=6, xlim = c(min(g1_connectedness, kapt_connectedness),
                                        max(g1_connectedness, kapt_connectedness)))
abline(v=kapt_connectedness,col=3, lwd=2)
```



From the histograms, degree centralization of the Kaptail network is concentrated on a single vertex to an extent more than one would expect from random graphs with the same dyad census.

Betweenness centralization of Kaptail network is concentrated on a single vertex to the same extent as one would expect from random graphs with the same dyad census.

Transitivity of Kaptail network is much more than one would expect from random graphs with the same dyad census.This means that interactions in this network are highly transitive.

Krachardt Connectedness of this network much lesser than one would expect from random graphs with the same dyad census. This can be explained, since the network has a high degree centralization. Thus, there are less weakly connected dyads, hence a lower connectedness than that of random graphs.

Problem 3: Testing Structural Hypotheses

Consider the following set of propositions, which may or may not be true of given dataset. For each, do the following:

1. Identify a statistic (e.g. GLI) whose value should deviate from a random baseline if the proposition is true.
2. Identify the approporate baseline distribution to which the statistic should be compared.
3. Determine whether the proposition implies that the statistic should be greater or lower than its baseline distribution would indicate.
4. Conduct a conditional uniform graph test based on your conclusions in 1-3. In reporting your results, include appropriate summary output from the `cug.test` function as well as the resulting distributional plots. Based on the results, indicate whether the data appears to support or undermine the proposition in question. Be sure to justify your conclusion.

(a) In militarized interstate disputes, hostile acts are disproportionately likely to be responded to in kind.

1. Statistic - Edgewise Reciprocity
2. Baseline Distribution - Size and Edges
3. Direction of Deviation - Higher

```r
# Edit me.

dyad.census(mids_1993)
```

```
##      Mut Asym  Null
## [1,]   3   64 17138
```

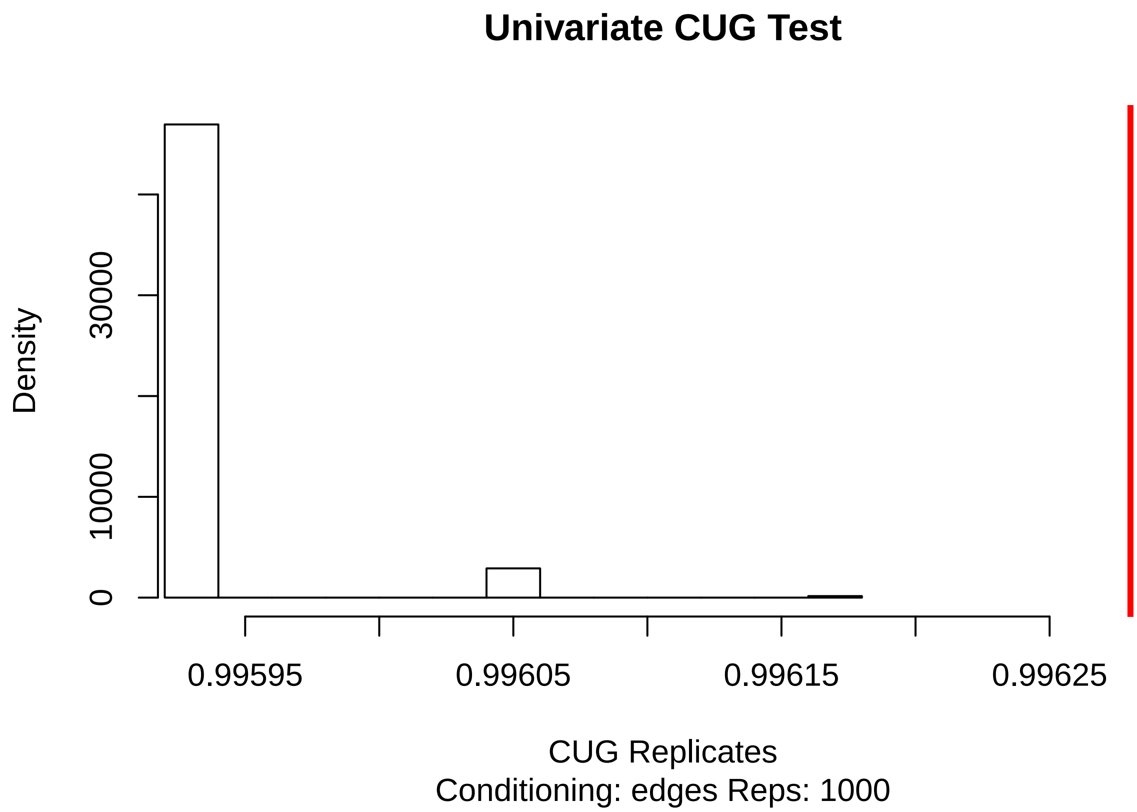
```r
grecip(mids_1993,measure = "edgewise")
```

```
##        Mut
## 0.08571429
```

```r
#CUG Test with Edges as baseline model
cug1_edges <- cug.test(mids_1993, grecip, FUN.args = (measure = "edgewise"), cmode = "edges")
names(cug1_edges)
```

```
## [1] "obs.stat" "rep.stat" "mode"     "diag"     "cmode"    "plteobs"
## [7] "pgteobs"  "reps"
```

```r
plot(cug1_edges)
```




```r
cug1_edges
```

```
##
## Univariate Conditional Uniform Graph Test
##
## Conditioning Method: edges
## Graph Type: digraph
## Diagonal Used: FALSE
## Replications: 1000
##
## Observed Value: 0.9962802
## Pr(X>=Obs): 0
## Pr(X<=Obs): 1
```

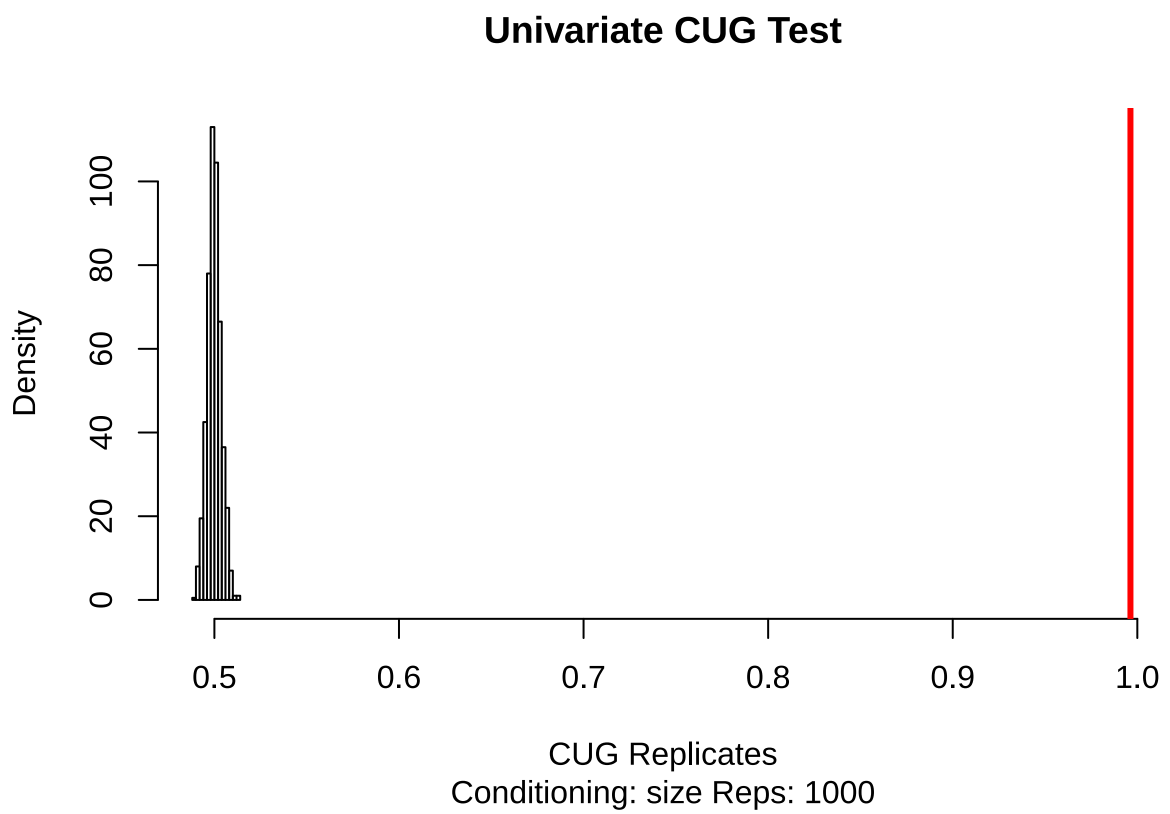
```r
summary(cug1_edges)
```

```
##          Length Class  Mode
## obs.stat    1   -none- numeric
## rep.stat 1000   -none- numeric
## mode        1   -none- character
## diag        1   -none- logical
## cmode       1   -none- character
## plteobs     1   -none- numeric
## pgteobs     1   -none- numeric
## reps        1   -none- numeric
```

```r
#CUG Test with Edges as baseline model
cug1_size <- cug.test(mids_1993, grecip, FUN.args = (measure = "edgewise"), cmode = "size")
names(cug1_size)
```

```
## [1] "obs.stat" "rep.stat" "mode"     "diag"     "cmode"    "plteobs"
## [7] "pgteobs"  "reps"
```

```r
plot(cug1_size)
```




```r
cug1_size
```

```
##
## Univariate Conditional Uniform Graph Test
##
## Conditioning Method: size
## Graph Type: digraph
## Diagonal Used: FALSE
## Replications: 1000
##
## Observed Value: 0.9962802
## Pr(X>=Obs): 0
## Pr(X<=Obs): 1
```

```r
summary(cug1_size)
```

```
##          Length Class  Mode
## obs.stat    1   -none- numeric
## rep.stat 1000   -none- numeric
```

```
## mode        1   -none- character
## diag        1   -none- logical
## cmode       1   -none- character
## plteobs     1   -none- numeric
## pgteobs     1   -none- numeric
## reps        1   -none- numeric
```

The plots show us that there is a noteworthy departure from the baseline model. The pvalue calculated is less than the critical value(0.05), Therefore, we reject the null hypothesis and conclude that hostile acts are disproportionately likely to be responded to in kind in military disputes.

(b) When engaging in disputes, nations behave in accordance with the notion that "the enemy of my enemy is not my enemy".

1.Statistic - Transitivity 2. Baseline Distribution - Edges 3. Direction of Deviation - Lower

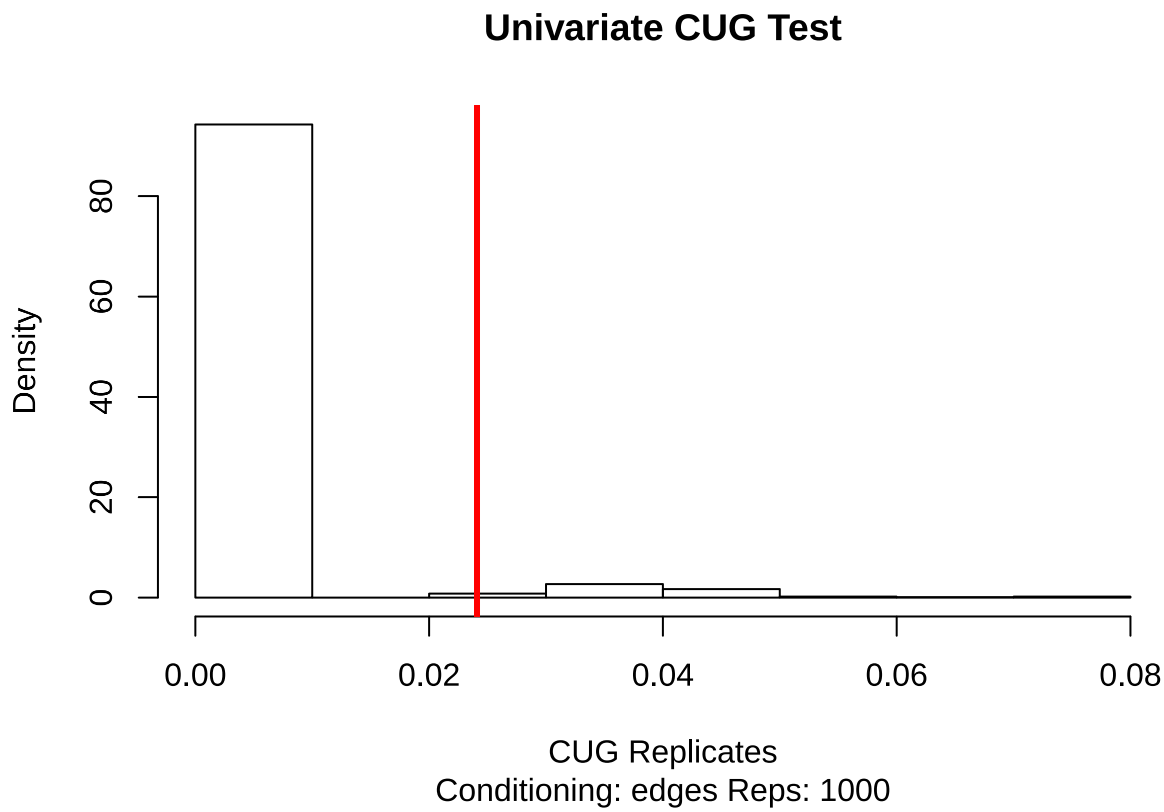
```r
cug2 <- cug.test(mids_1993, gtrans, cmode = "edges")
names(cug2)
```

```
## [1] "obs.stat" "rep.stat" "mode"     "diag"     "cmode"    "plteobs"
## [7] "pgteobs"  "reps"
```

```r
cug2
```

```
##
## Univariate Conditional Uniform Graph Test
##
## Conditioning Method: edges
## Graph Type: digraph
## Diagonal Used: FALSE
## Replications: 1000
##
## Observed Value: 0.02409639
## Pr(X>=Obs): 0.057
## Pr(X<=Obs): 0.943
```

```r
plot(cug2)
```




```r
summary(cug2)
```

```
##          Length Class  Mode
## obs.stat    1   -none- numeric
## rep.stat 1000   -none- numeric
## mode        1   -none- character
## diag        1   -none- logical
## cmode       1   -none- character
## plteobs     1   -none- numeric
## pgteobs     1   -none- numeric
## reps        1   -none- numeric
```

The plot shows us that there is a not a noteworthy departure from the baseline model. The pvalue calculated is greater than the critical value(0.05), Therefore, we fail to reject the null hypothesis and conclude that nations do not behave in accordance with the notion that "the enemy of my enemy is not my enemy"

(c) Given the number of disputes at any given time, as small number of nations will receive a disproportionate share of aggressive acts.

1.Statistic - Indegree 2.Baseline Distribution - Size and edges 3. Direction of Deviation - Higher

```r
#CUG Test with Edges as condition
cug3_edges <- cug.test(mids_1993, centralization, FUN.arg = list(FUN = degree,cmode="indegree"), cmode=
cug3_edges
```

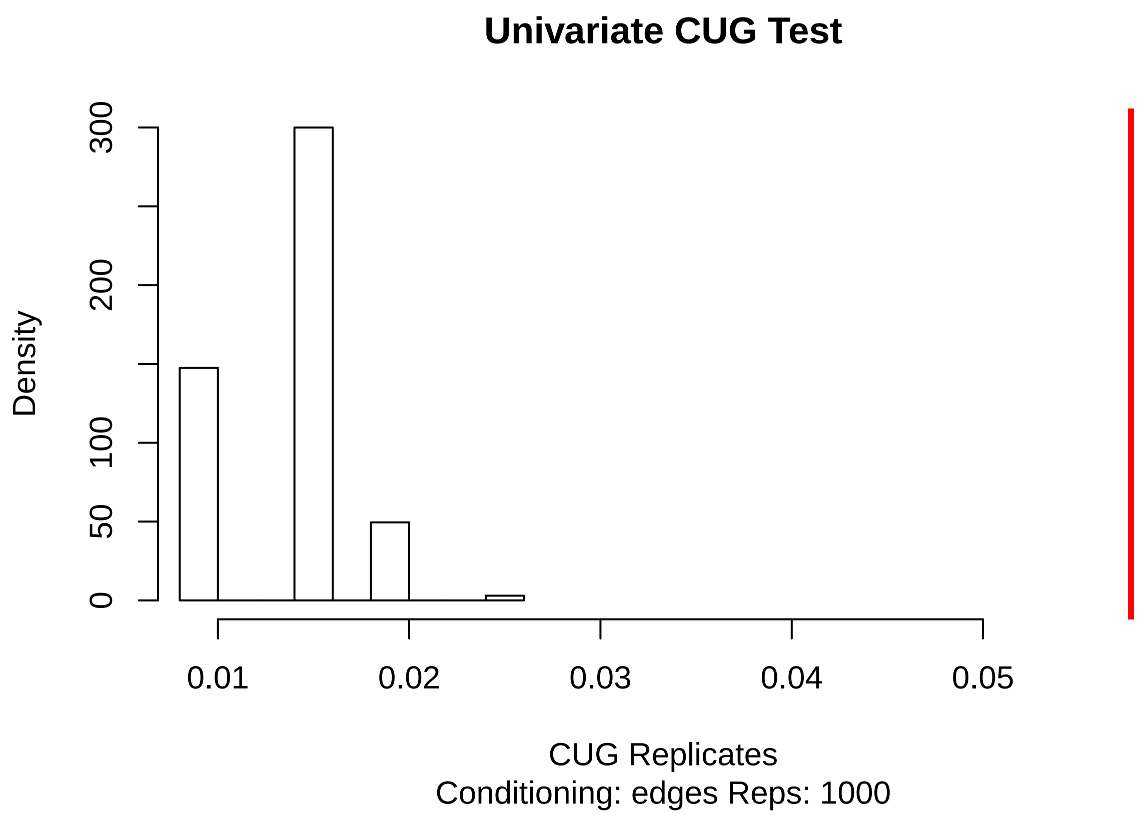
```
##
## Univariate Conditional Uniform Graph Test
##
```

```
## Conditioning Method: edges
## Graph Type: digraph
## Diagonal Used: FALSE
## Replications: 1000
##
## Observed Value: 0.05773557
## Pr(X>=Obs): 0
## Pr(X<=Obs): 1
```

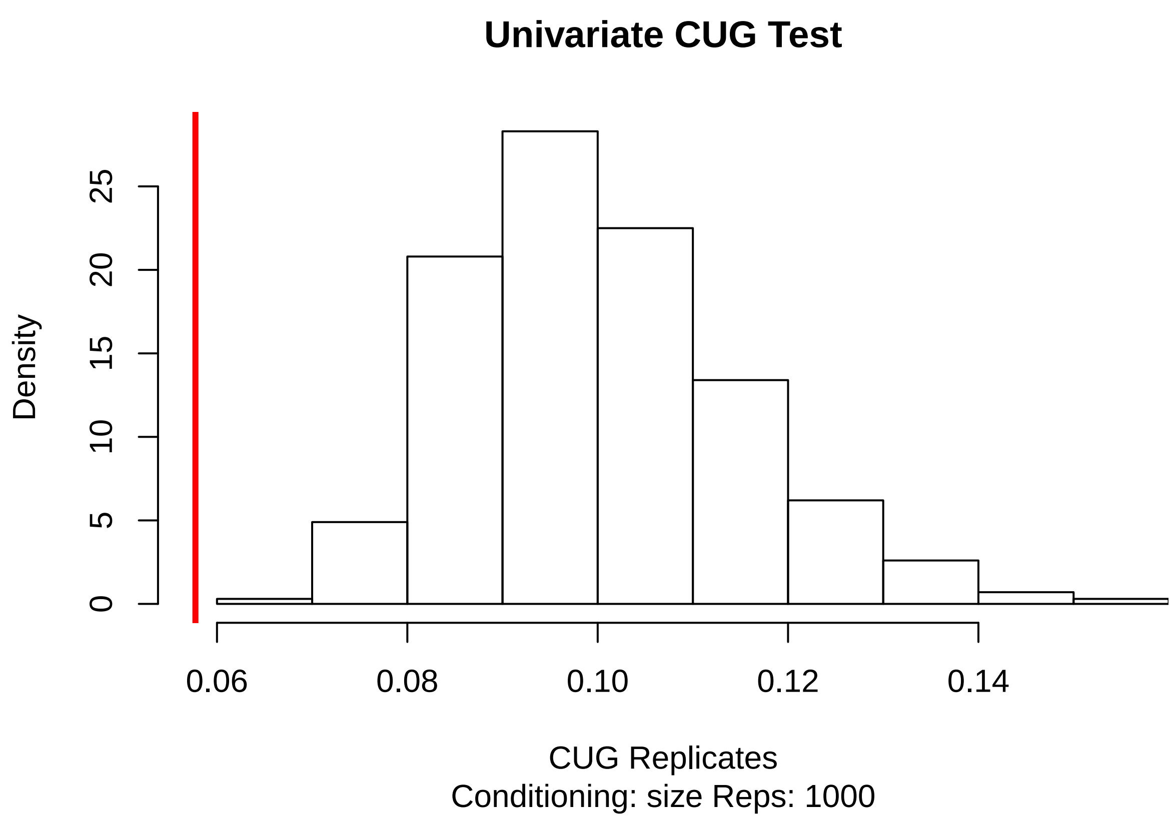
```r
#CUG Test with Edges as size
cug3_size <- cug.test(mids_1993, centralization, FUN.arg = list(FUN = degree,cmode="indegree"), cmode="s
cug3_size
```

```
##
## Univariate Conditional Uniform Graph Test
##
## Conditioning Method: size
## Graph Type: digraph
## Diagonal Used: FALSE
## Replications: 1000
##
## Observed Value: 0.05773557
## Pr(X>=Obs): 1
## Pr(X<=Obs): 0
```

```r
plot(cug3_edges)
```



```r
plot(cug3_size)
```




```r
summary(cug3_edges)
```

```
##          Length Class  Mode
## obs.stat    1   -none- numeric
## rep.stat 1000   -none- numeric
## mode        1   -none- character
## diag        1   -none- logical
## cmode       1   -none- character
## plteobs     1   -none- numeric
## pgteobs     1   -none- numeric
## reps        1   -none- numeric
```

```r
summary(cug3_size)
```

```
##          Length Class  Mode
## obs.stat    1   -none- numeric
## rep.stat 1000   -none- numeric
## mode        1   -none- character
## diag        1   -none- logical
## cmode       1   -none- character
## plteobs     1   -none- numeric
## pgteobs     1   -none- numeric
## reps        1   -none- numeric
```

The plots show us that there is a noteworthy departure from the baseline model. The pvalue calculated is less than the critical value(0.05), Therefore, we reject the null hypothesis and conclude that given the number of disputes at any given time, as small number of nations will receive a disproportionate share of aggressive acts.