BUS 41201 Homework 4 Assignment

Group 24: Shihan Ban, Yi Cao, Shri Lekkala, Ningxin Zhang

16 April 2024

Setup

```
## microfinance network
## data from BANERJEE, CHANDRASEKHAR, DUFLO, JACKSON 2012

## data on 8622 households
hh <- read.csv("microfi_households.csv", row.names="hh")
hh$village <- factor(hh$village)

## We'll kick off with a bunch of network stuff.
## This will be covered in more detail in lecture 6.
## get igraph off of CRAN if you don't have it
## install.packages("igraph")
## this is a tool for network analysis
## (see http://igraph.sourceforge.net/)
library(igraph)
```

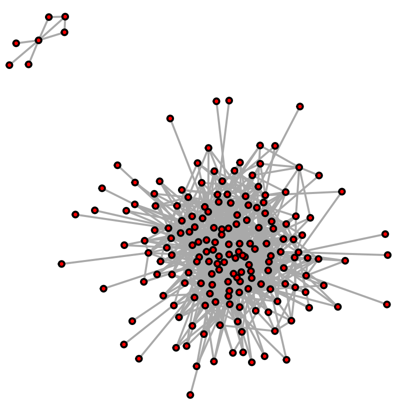
```
##
## Attaching package: 'igraph'

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

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

```
edges <- read.table("microfi_edges.txt", colClasses="character")
## edges holds connections between the household ids
hhnet <- graph.edgelist(as.matrix(edges))
hhnet <- as.undirected(hhnet) # two-way connections.

## igraph is all about plotting.
V(hhnet) ## our 8000+ household vertices
```

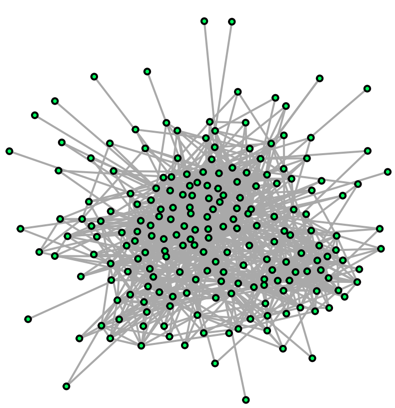
```
## + 8182/8182 vertices, named, from 128e388:
##    [1] 1002  1001  1020  1042  1053  1163  1003  1004  1026  1029  1076  1159
##   [13] 1106  1031  1048  1081  1006  1005  1008  1016  1021  1024  1089  1103
##   [25] 1007  1019  1155  1015  1040  1044  1045  1078  1088  1110  1115  1140
##   [37] 1145  1009  1018  1060  1064  1073  1153  1067  1099  1010  1162  1012
##   [49] 1143  1013  1023  1028  1034  1065  1117  1139  1154  1157  1173  1014
##   [61] 1068  1071  1148  1017  1036  1062  1112  1118  1120  1129  1134  1165
##   [73] 1183  1126  1122  1049  1058  1093  1108  1114  1119  1022  1043  1079
```

```
##   [85] 1033  1102  1104  1105  1152  1169  1171  1025  1027  1147  1032  1035
##   [97] 1037  1039  1041  1113  1174  1069  1116  1132  1178  1146  1080  1086
##  [109] 1101  1172  1059  1141  1142  1038  1094  1052  1092  1082  1095  1158
## + ... omitted several vertices
```

```r
## Each vertex (node) has some attributes, and we can add more.
V(hhnet)$village <- as.character(hh[V(hhnet),'village'])
## we'll color them by village membership
vilcol <- rainbow(nlevels(hh$village))
names(vilcol) <- levels(hh$village)
V(hhnet)$color = vilcol[V(hhnet)$village]
## drop HH labels from plot
V(hhnet)$label=NA

# graph plots try to force distances proportional to connectivity
# imagine nodes connected by elastic bands that you are pulling apart
# The graphs can take a very long time, but I've found
# edge.curved=FALSE speeds things up a lot.  Not sure why.

## we'll use induced.subgraph and plot a couple villages
village1 <- induced.subgraph(hhnet, v=which(V(hhnet)$village=="1"))
village33 <- induced.subgraph(hhnet, v=which(V(hhnet)$village=="33"))

# vertex.size=3 is small.  default is 15
plot(village1, vertex.size=3, edge.curved=FALSE)
```



```r
plot(village33, vertex.size=3, edge.curved=FALSE)
```



```r
library(gamlr)
```

```
## Loading required package: Matrix
```

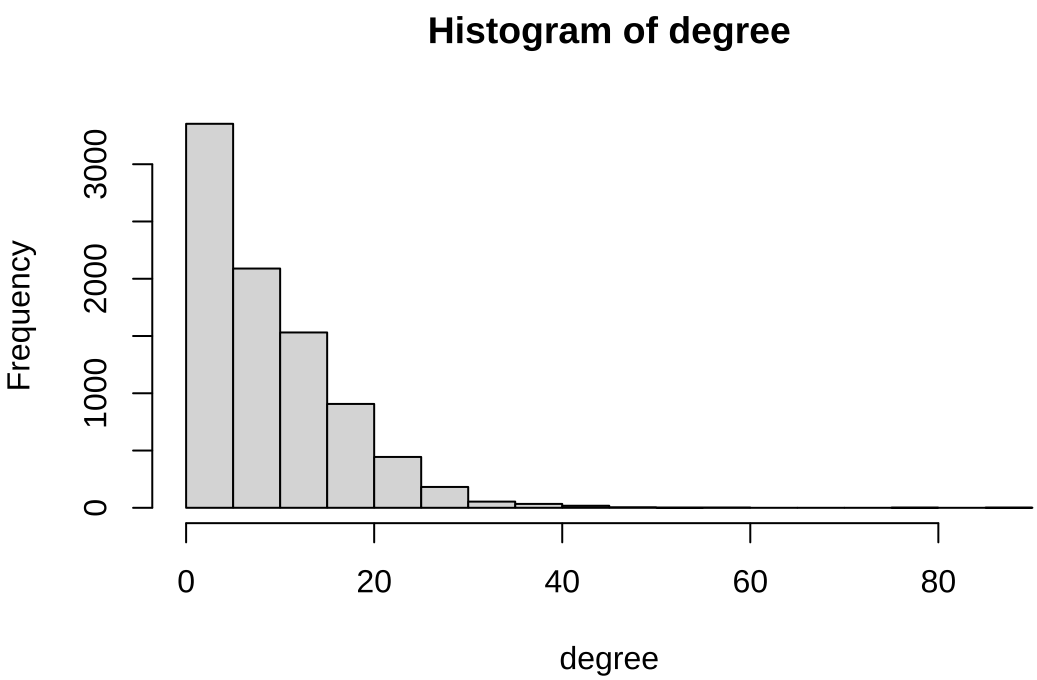
```r
## match id's; I call these 'zebras' because they are like crosswalks
zebra <- match(rownames(hh), V(hhnet)$name)

## calculate the `degree' of each hh:
##  number of commerce/friend/family connections
degree <- degree(hhnet)[zebra]
names(degree) <- rownames(hh)
degree[is.na(degree)] <- 0 # unconnected houses, not in our graph

## if you run a full glm, it takes forever and is an overfit mess
# > summary(full <- glm(loan ~ degree + .^2, data=hh, family="binomial"))
# Warning messages:
# 1: glm.fit: algorithm did not converge
# 2: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Question 1

I'd transform degree to create our treatment variable d. What would you do and why?

We can first plot a histogram of the degree variable to get an idea of it's structure:

```
hist(degree)
```



From the graph, it might be appropriate to perform a logarithmic transformation for the following reasons:

- It appears that the degree frequency is highly skewed to the right as there are many nodes with few connections (degree < 20), but few nodes with many connections (degree > 40). So by taking a log transformation, we can normalize the distribution, making it more symmetric and more suitable to statistical analyses.
- The histogram appears to follow an exponential / multiplicative relationship. So transforming the data logarithmically can make the relationship more linear, which is easier to model and interpret in regression models.
- We can reduce the range of variability in degree values, effectively performing a dimensionality reduction. This is useful to prevent the model being overly effected by outliers, i.e. households with a very high number of connections.

```
# Transform degree and add it to the hh dataset
hh$log_degree = log1p(degree)
head(hh)
```

```
##      loan village religion  roof rooms beds electricity ownership leader
## 1001    0       1    hindu  tile     3    4           0     OWNED      0
## 1002    0       1    hindu  tile     1    1           1     OWNED      1
## 1003    0       1    hindu   rcc     3    4           1     OWNED      1
## 1004    0       1    hindu  tile     2    6           1     OWNED      0
```

```
## 1005    0       1    hindu  tile      3    4            1     OWNED      0
## 1006    0       1    hindu stone      2    1            1     OWNED      0
##      log_degree
## 1001   1.791759
## 1002   2.079442
## 1003   1.098612
## 1004   1.609438
## 1005   2.197225
## 1006   2.302585
```

Question 2

Build a model to predict d from x, our controls.

Comment on how tight the fit is, and what that implies for estimation of a treatment effect.

Question 3

Use predictions from [2] in an estimator for effect of d on loan.

Question 4

Compare the results from [3] to those from a straight (naive) lasso for loan on d and x.

Explain why they are similar or different.

Question 5

Bootstrap your estimator from [3] and describe the uncertainty.

[+]

Can you think of how you'd design an experiment to estimate the treatment effect of network degree?