HW1

HW 1: Exploring network data

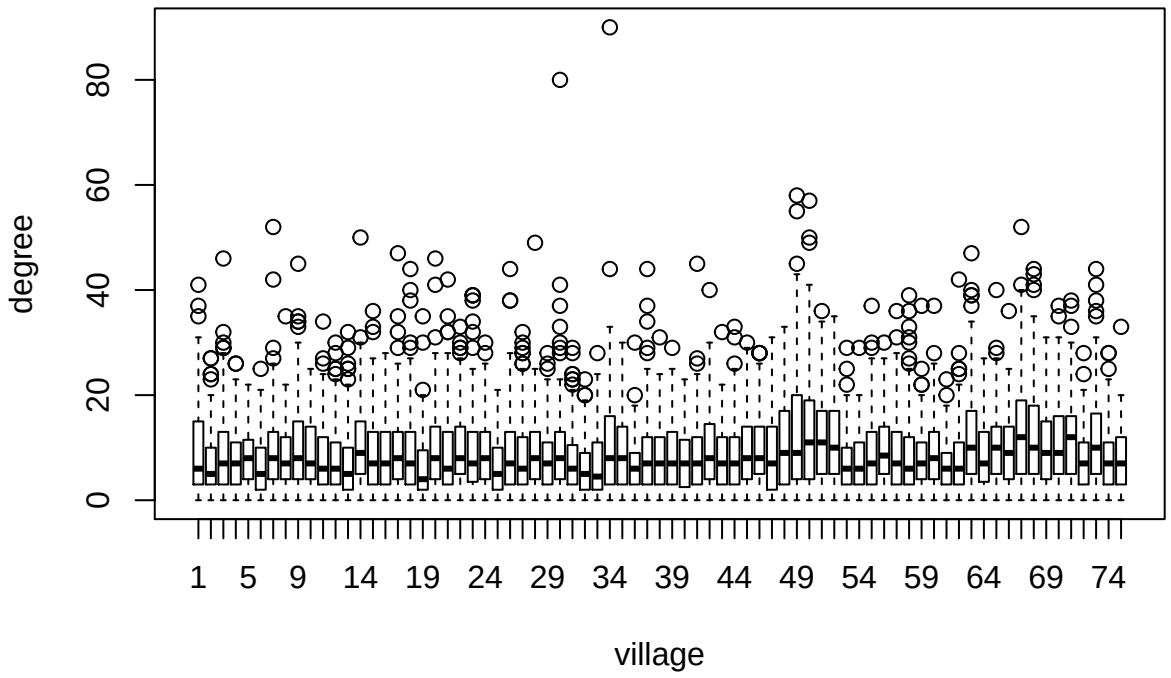
These questions come from Chapter 2-4 of the SANDr book.

1. First, download the Banerjee et al 2013 data we've discussed several times in class and load it into R. Use this data for the questions below (except Q6, where you can either simulate or use these data).

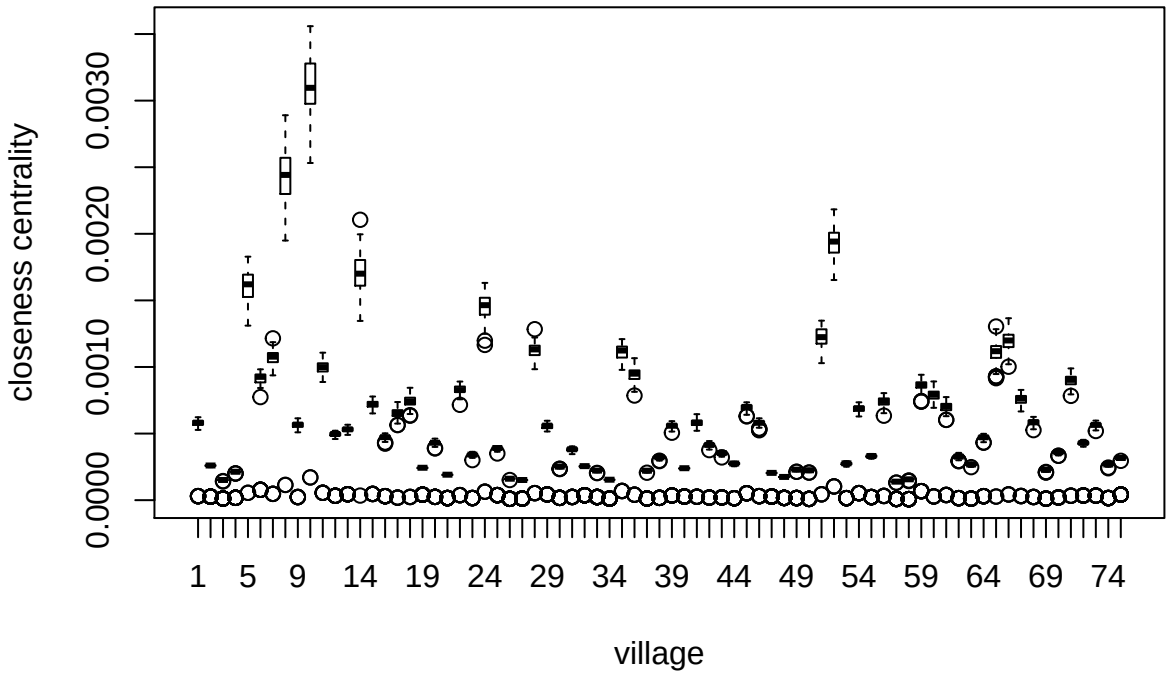
```
library(igraph)
library(readstata13)
data=list()
directory="/Users/Mengjie/Downloads/datav4.0/Data/"
for (i in c(1:12,14:21,23:77)){
  data_vlg=read.csv(paste0(directory,"1. Network Data/Adjacency Matrices/adj_allVillageRelationships_HH
  data=c(data,list(data_vlg))
}
```

2. For this exercise, let's compare graph structure across the 75 villages. Within each village, compute some (2-3) individual level graph statistics (e.g. degree) and plot the distribution for each village. To do this, I recommend a boxplot where every box represents the village and in each box are individual level statistics individuals in each graph.

```
degrees=list()
centralities=list()
for (i in 1:75){
  graph=graph.adjacency(as.matrix(data[[i]]),mode="undirected")
  degrees=c(degrees,list(degree(graph)))
  centralities=c(centralities,list(closeness(graph)))
}
boxplot(degrees,xlab='village',ylab='degree')
```



```
boxplot(centralities,xlab='village',ylab='closeness centrality')
```



3. For the individual level graph statistics above, identify individuals who are extreme (either on the high or low end). Use covariates to provide aggregate summaries of the characteristics of these individuals.

Getting outliers for degrees

```
outlier_degree=list()
for (i in 1:75){
  outlier_degree=c(outlier_degree,list(which(degrees[[i]]> quantile(degrees[[i]],0.75)+1.5*IQR(degrees[
}
```

Below is the code to get the caste distribution for households with extreme degrees. You could do the same thing with other covariates.

```
covariates=read.dta13(paste0(directory,"2. Demographics and Outcomes/household_characteristics.dta"))
```

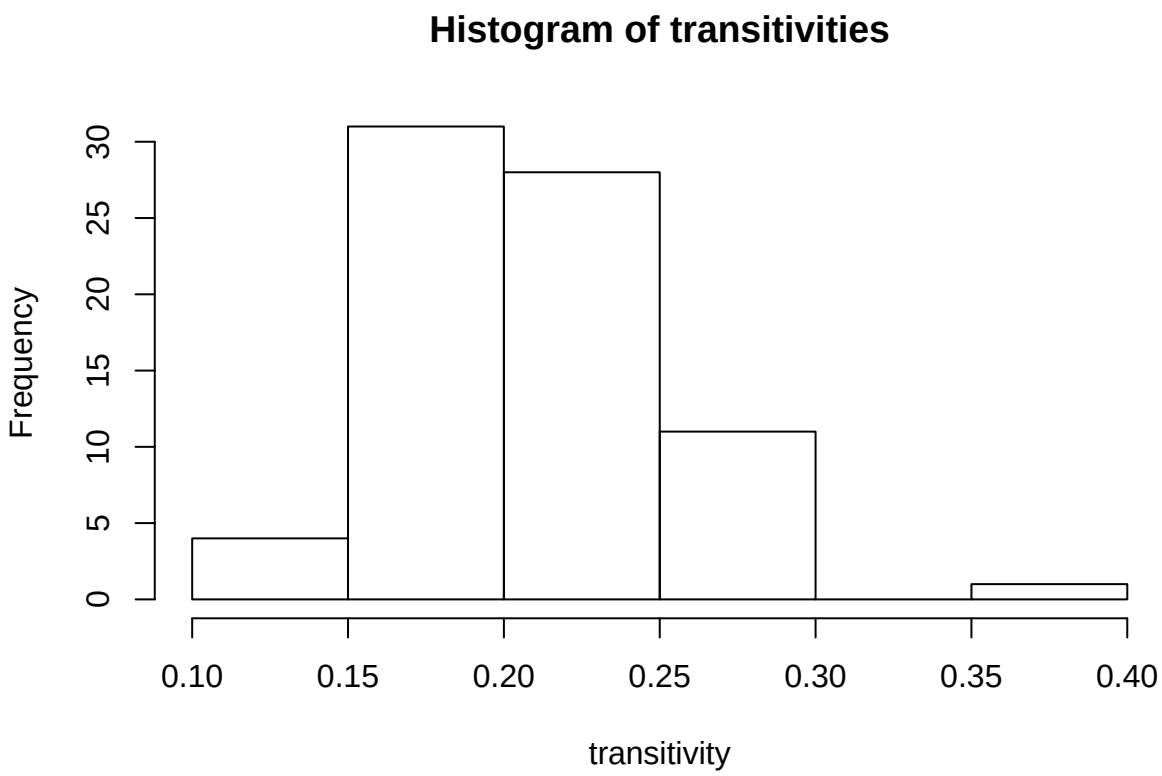
```
## Warning in read.dta13(paste0(directory, "2. Demographics and Outcomes/household_characteristics.dta"
##   ownrent:
##   Missing factor labels - no labels assigned.
##   Set option generate.factors=T to generate labels.
```

```
caste_outlier=NULL
for (i in 1:75){
  caste_outlier=c(caste_outlier, covariates$castesubcaste[covariates$village==c(1:12,14:21,23:77)[i] & 
}
table(caste_outlier)
```

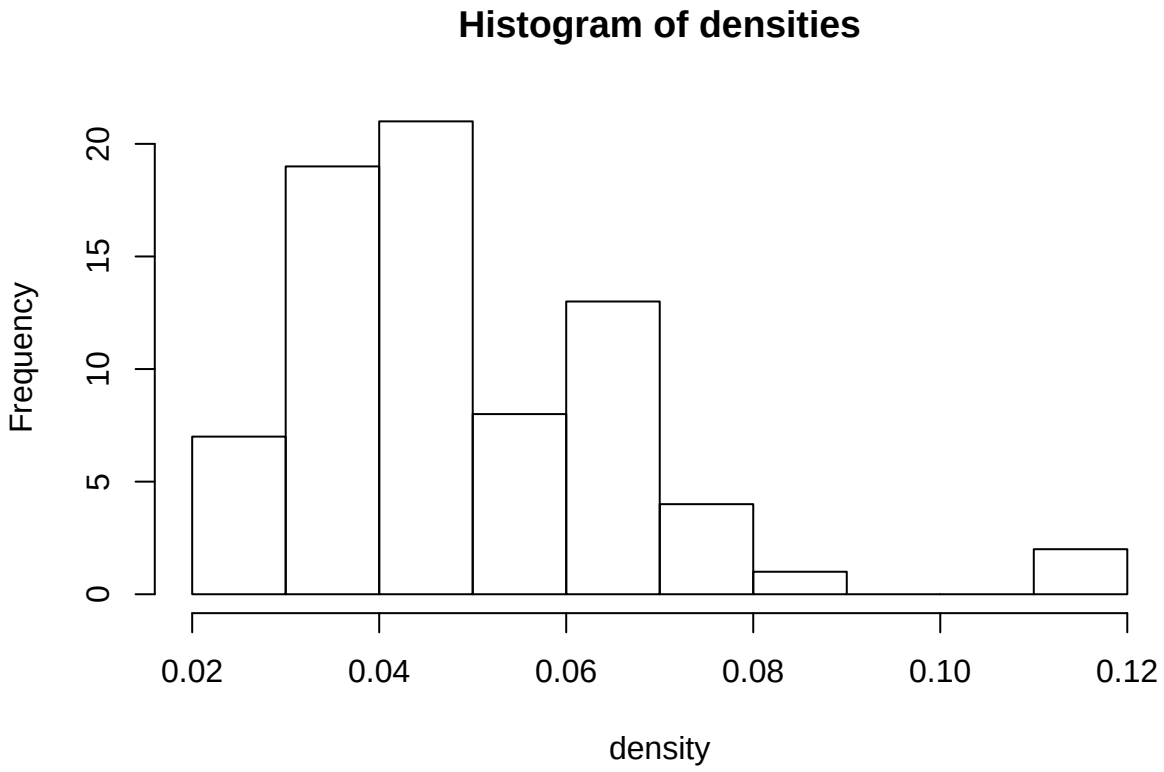
```
## caste_outlier
##                       GENERAL       MINORITY            OBC SCHEDULE CASTE
##             83             15              3             79             23
## SCHEDULE TRIBE
##              5
```

4. Now, let's look at some graph level statistics. Make a histogram of 2-3 village level statistics. Use covariates to describe villages that are high or low on either end.

```
transitivities=NULL
densities=NULL
for (i in 1:75){
  graph=graph.adjacency(as.matrix(data[[i]]),mode="undirected")
  transitivities=c(transitivities,transitivity(graph))
  densities=c(densities,edge_density(graph))
}
hist(transitivities,xlab="transitivity")
```




```
hist(densities,xlab="density")
```



To find index of extreme values of each measure:

```
outlier_trans=which(transitivities>0.3)
outlier_den=which(densities>0.1)
c(1:12,14:21,23:77)[outlier_trans]
```

```
## [1] 54
```

```
c(1:12,14:21,23:77)[outlier_den]
```

```
## [1] 10 54
```

Therefore, village 54 has extreme transitivity value, and villages 10 and 54 have extreme density value.

To get a summary of caste (categorical) and number of rooms (continuous):

```
table(covariates$castesubcaste[covariates$village %in% c(10,54)])
```

```
##
##                         GENERAL             OBC SCHEDULE CASTE SCHEDULE TRIBE
##              77              41               1              56               1
```

```
mean(covariates$room_no[covariates$village %in% c(10,54)])
```

```
## [1] 2.664773
```

5. Use regression models to explore the association between village-level covariates (you can aggregate individuals ones if you'd like) and graph statistics.

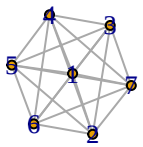
Below is the code to regress degree density on average number of rooms:

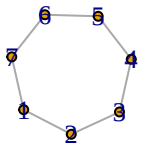
```
room_no_df=aggregate(room_no ~ village, covariates, mean)
model=lm(densities~room_no_df$room_no)
summary(model)
```

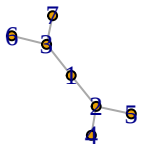
```
##
## Call:
## lm(formula = densities ~ room_no_df$room_no)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.027192 -0.011711 -0.000605  0.008419  0.060197
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.011525   0.016385   0.703   0.4841
## room_no_df$room_no 0.015936   0.006889   2.313   0.0235 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01737 on 73 degrees of freedom
## Multiple R-squared:  0.06831,    Adjusted R-squared:  0.05554
## F-statistic: 5.352 on 1 and 73 DF,  p-value: 0.02352
```

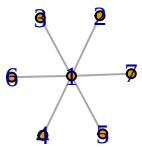
6. Replicate Figure 2.2. Describe a data setting where you would expect to see each of the four types of graphs.

```
g.full <- graph.full(7)
g.ring <- graph.ring(7)
g.tree <- graph.tree(7, children=2, mode="undirected")
g.star <- graph.star(7, mode="undirected")
par(mfrow=c(2, 2))
plot(g.full)
plot(g.ring)
plot(g.tree)
plot(g.star)
```







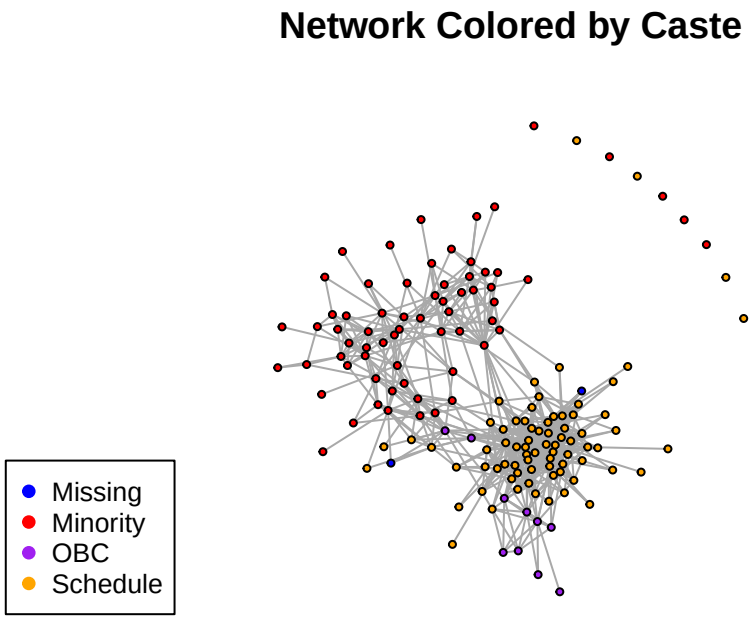


Any reasonable senario would get full credit.

7. For one of the villages, construct a visualization of the graph. Label the nodes based on covariates. Do this for a couple of covariates.

Below is the code for coloring nodes by caste. You may use other covariates for coloring.

```
vlg=29
g <- graph.adjacency(as.matrix(data[[vlg]]), mode="undirected")
covariates_vlg29=covariates[covariates$village==c(1:12,14:21,23:77)[vlg],]
colors <- rep("blue", length(covariates_vlg29$castesubcaste))
colors[covariates_vlg29$castesubcaste=="MINORITY"] <- "red"
colors[covariates_vlg29$castesubcaste=="OBC"] <- "purple"
colors[covariates_vlg29$castesubcaste=="SCHEDULE CASTE"] <- "orange"
plot(g, layout=layout.fruchterman.reingold, vertex.label="", vertex.size=3, vertex.color = colors, main=
legend("bottomleft", col=c("blue", "red", "purple", "orange"), pch=c(19,19), c("Missing", "Minority", "
```



8. Run the samecaste regression we talked about in class. For this, you can assume that $\delta = 0$ (that is, there are no other informative covariates). Describe why standard inferential techniques are or aren't valid here.

```
vlg=52
y_network=as.matrix(data[[vlg]])
x_caste=matrix(NA,nrow=dim(y_network)[1],ncol=dim(y_network)[1])
caste_cov=covariates$castesubcaste[covariates$village==c(1:12,14:21,23:77)[vlg]]
for (i in 1:(dim(y_network)[1]-1)){
  for (j in (i+1):dim(y_network)[1]){
    x_caste[i,j] <-x_caste[j,i] <- as.numeric(caste_cov[i]==caste_cov[j])
  }
}
model=glm(y_network[lower.tri(y_network)]~x_caste[lower.tri(x_caste)],family="binomial")
summary(model)
```

```
##
## Call:
## glm(formula = y_network[lower.tri(y_network)] ~ x_caste[lower.tri(x_caste)],
##     family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6391  -0.6391  -0.3075  -0.3075   2.4801
##
```

```
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -3.02824    0.09547  -31.72   <2e-16 ***
## x_caste[lower.tri(x_caste)]  1.54372    0.10922   14.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3434.0  on 4850  degrees of freedom
## Residual deviance: 3190.6  on 4849  degrees of freedom
## AIC: 3194.6
##
## Number of Fisher Scoring iterations: 5
```

The inference (standard errors and p-value) displayed in summary of model assumes i.i.d. errors, which is violated in this case. Observations involve the same node (for example, an edge between household 1 &2 and an edge between household 2&3) are likely correlated because they involve the same household, and that same household may have individual effects on connectivity, which violates the independent errors assumptions. Refer to dyadic data analysis covered in the lab on a detailed description of how to model the errors.