

# Project 1 - IntraOrganizational Network (Consultants)

Group 7

```
library(igraph)
library(tnet)
library(intergraph)
library(ergm)
library(ggplot2)
library(scales)
set.seed(1122)
Request <- read.table("http://opsahl.co.uk/tnet/datasets/Cross_Parker-Consulting_info.txt",colClasses=c
Request <- Request[Request$V1!=Request$V2 , ] # delete self loop node
Receive <- read.table("http://opsahl.co.uk/tnet/datasets/Cross_Parker-Consulting_value.txt",colClasses=c
Receive <- Receive[Receive$V1!=Receive$V2 , ]

Level <- read.table("http://opsahl.co.uk/tnet/datasets/Cross_Parker-Consulting-orglevel.txt")
Gender <- read.table("http://opsahl.co.uk/tnet/datasets/Cross_Parker-Consulting-gender.txt")
Region <- read.table("http://opsahl.co.uk/tnet/datasets/Cross_Parker-Consulting-region.txt")
Location <- read.table("http://opsahl.co.uk/tnet/datasets/Cross_Parker-Consulting-location.txt")

Attr <- data.frame(cbind(t(Level),t(Gender),t(Region),t(Location)))
colnames(Attr) <- c("level","gender","region","location")

colnames(Request) <- c("ego","alter","request_tie")
Request$request_tie[Request$request_tie<=3] <- 0
Request$request_tie[Request$request_tie>3] <- 1

colnames(Receive) <- c("ego","alter","receive_tie")
Receive$receive_tie[Receive$receive_tie<=3] <- 0
Receive$receive_tie[Receive$receive_tie>3] <- 1

rb <- c("lightskyblue2","lightpink","orange","palegreen","yellow")
for (i in 1:5){
  Attr$level_color[Attr$level==i] <- rb[i]
}

Attr$gender_color[Attr$gender==1] <- "lightskyblue2"
Attr$gender_color[Attr$gender==2] <- "lightpink"
Attr$gender[Attr$gender==1] <- "M"
Attr$gender[Attr$gender==2] <- "F"

Attr$region_color[Attr$region==1] <- "lightskyblue2"
Attr$region_color[Attr$region==2] <- "darkseagreen3"
Attr$region[Attr$region==1] <- "E"
Attr$region[Attr$region==2] <- "A"

rb <- c("lightskyblue2","lightpink","orange","darkseagreen3","seagreen","yellow","palegreen")
for (i in 1:7){
  Attr$location_color[Attr$location==i] <- rb[i]
}
```

```
RequestDF <- graph.data.frame(Request)
ReceiveDF <- graph.data.frame(Receive)
```

## Introduction to the Network

### Introduction

Social network analytics finds wide range of applications including detecting existing relationships between various components of network thereby leading to clearer understanding of network patterns. Social network analytics help understand,

1. Patterns on individuals interaction and communities formation
2. Common frameworks that are applicable to many systems
3. Connectedness

It is also necessary to keep in mind, the shortcomings of using network analysis and the reliance on data retrieved and analysed using computational models. A good proportion of network analysis studies, especially the concept of connectedness, is an emerging field with lack of standard and agreed upon methodology (lack of a Standard Operating Procedure).

In this project we have successfully attempted to apply network analytics in the field of organizational analysis. The following two sections will describe details of the project, background information and how network helps in deducing crucial relationships to ultimately form global network patterns.

### Project background, scope and question asked:

As mentioned earlier, we used intra-organizational data to identify and study patterns. Network analysis on intra-organizational data helps solve/understand key problems/characteristics pertaining,

1. Organizational structure and hierarchy
2. Training and knowledge transfer during on-boarding and employee exit stages respectively
3. Identification of key value proposition/organization strength in terms of domain expertise
4. Identification of areas of improvement in terms of service quality and domain expertise

For the purpose of this project, we have used data from 46 consultants for both network one and network two analysis. The two network data we used both represent data sets with two mode ties (i.e. two users, two consultants).

**Network one (Request Network)** data comprises information pertaining the frequency of information passage from one user to another. Hence the question asked here is, “How often has one user turned to this one other user person for information or advice on work-related topics in the past three months?”.

**Network two (Receive Network)** data comprises information pertaining the value of advice/information received by a user from another user. The question asked here is, “How strongly does a user agree or disagree that the other user has expertise in areas that are important in the kind of work former user does?”.

### Network description:

This section of the paper gives a relatively detailed description of data, its characteristics and key annotations.

As mentioned in the earlier section, network one (Request network) describes how often does one user turn to another for advice/information on work-related topics in the last three months. In the first network, the ties are differentiated on a scale from 0 to 5 in terms of frequency of information or advice requests.

\*0= Not Know, 1=Never, 2=Seldom, 3=Sometime, 4=Often, 5=Very Often\*

The raw dataset, which was represented in a three number format such as X1 X2 X3 depicts that X1 turned to X2 for advice at a frequency defined by X3. For eg., data represented as 2 38 4 depicts that user 2 turned user 38 for advice/information often (defined as per table above as 4=Often).

Network two (Receive network) describes how much value does one user's advice/information add to the other user's question. In the second network, the ties are differentiated on a scale from 0 to 5 in terms of value of information or advice received.

\*0= Strongly Disagree, 1=Disagree, 2=Neutral, 3=Sometimes, 4=Agree, 5=Strongly Agree\*

The raw dataset, which was represented in a three number format such as X1 X2 X3 depicts that X1 received advice from X2 that had quality defined by X3. For eg., data represented as 2 38 4 depicts that user 2 received data from user 38 that user 2 agrees with (defined as per table above as 4=Agree).

Network analysis primarily comprised of four major attributes that helped us understand key relationships and patterns.

**Level** (1=Research Assistant, 2=Junior Consultant, 3=Senior Consultant, 4=Managing Consultant and 5=Partner)

**\*\*Gender\*\*** (1=Male and 2=Female)

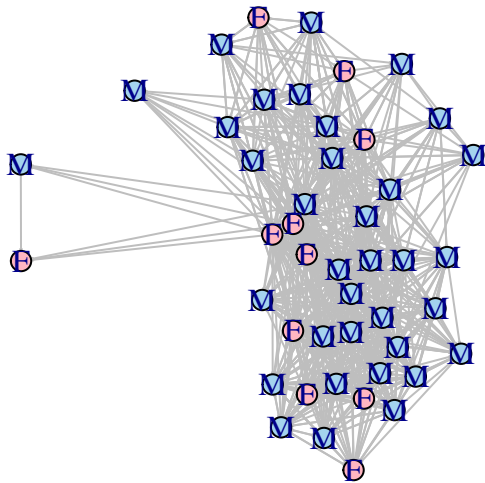
**\*\*Region\*\*** (1=Europe and 2=USA) and Location (1=Boston, 2=London, 3=Paris, 4=Rome, 5=Madrid, 6=Oslo and 7=Copenhagen)

**Dataset: Network 8-11: Intra-organisational networks : [https://toreopsahl.com/datasets/#Cross\\_Parker](https://toreopsahl.com/datasets/#Cross_Parker)**

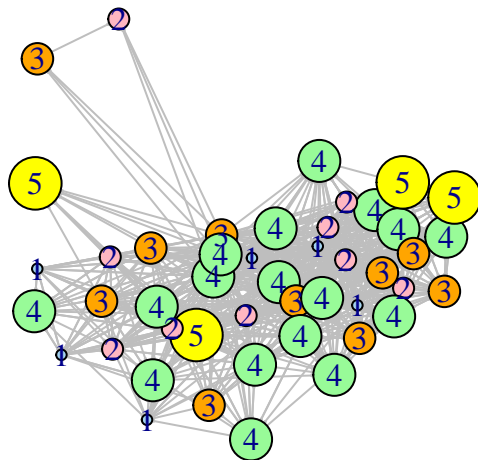
## Analysis

The network analysis and the identified patterns and relationships have been detailed in this section. The results of analysis has been performed in terms of plot distribution, nodes, Exponential Random Graph Model (ERGM). It is important to note that for the analysis, we have considered key Centrality metrics i.e Degree Centrality (in-degree and out-degree), Betweenness Centrality and Eigenvector Centrality. Degree Centrality particularly in important if the node has many neighbours. Depending on the direction, the network can be in-degree or out-degree. Betweenness centrality is primarily observed when nodes play an intermediary role or as a gatekeeper. Eigenvector centrality is a measure of the influence of a node in a network. It assigns relative scores to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes

## Receive Network - Gender



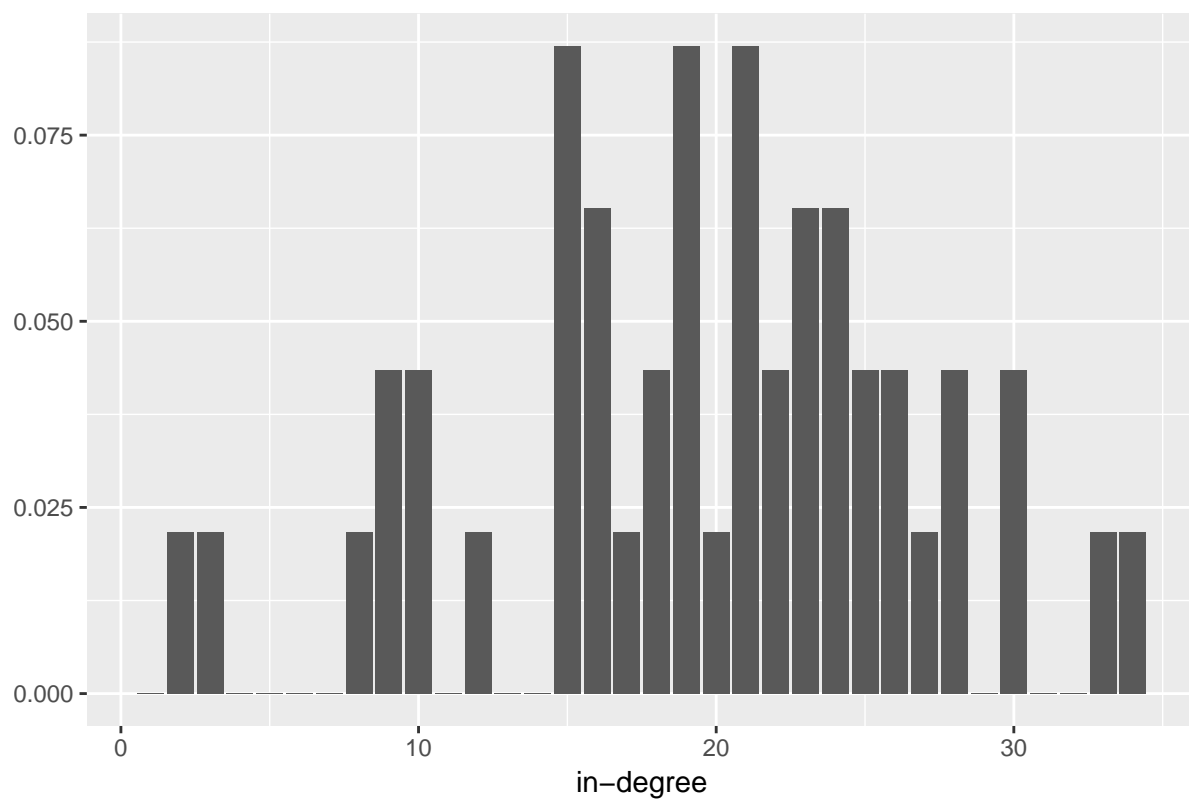
## Receive Network - Level



## Degree Distributions

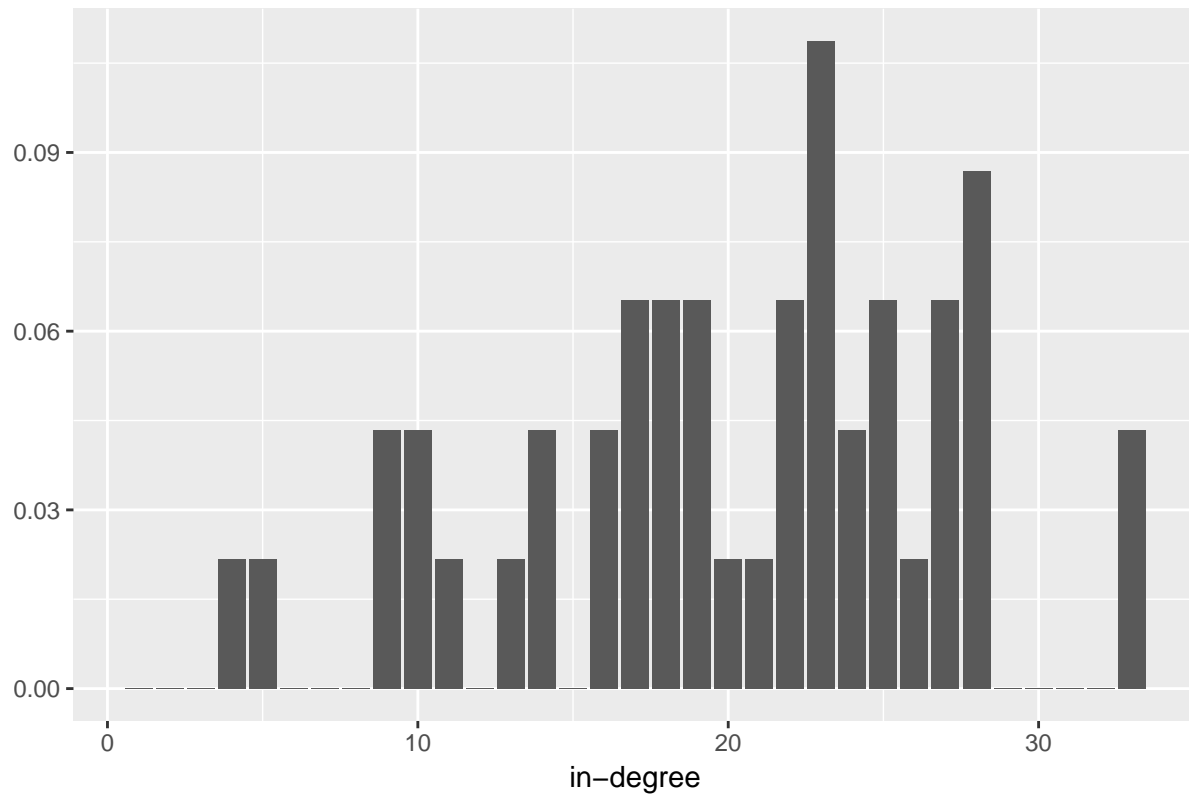
```
degDist <- as.data.frame(degree_distribution(ReceiveDF,mode="in"))
len <- length(degree_distribution(ReceiveDF,mode="in"))-1
degDist$degree <- seq(0:len)
colnames(degDist) <- c("dist","degree")
ggplot(degDist,aes(degree,dist))+geom_bar(stat="identity")+xlab("in-degree") + ylab("") + ggtitle("Receive Network - Degree Distribution")
```

Receive Network – Degree Distribution



```
degDist <- as.data.frame(degree_distribution(RequestDF,mode="in"))
len <- length(degree_distribution(RequestDF,mode="in"))-1
degDist$degree <- seq(0:len)
colnames(degDist) <- c("dist","degree")
ggplot(degDist,aes(degree,dist))+geom_bar(stat="identity")+xlab("in-degree") + ylab("") + ggtitle("Requ
```

## Request Network – Degree Distribution



## Analyzing the Network - Nodes & ERGMS

### Nodes

#### Request network

(How often you have turned to this person for information or advice on work-related topics in the past three months)

Metric Used : **In Degree**

A lot of people have often turned to him for advice in the past three-months

```
Attr[which.max(degree(RequestDF, mode='in')),][1:4]
```

```
##      level gender region location
## V2      4      M      A      1
```

Metric Used : **Out Degree**

The person has often asked for advice in the past three months

```
Attr[which.max(degree(RequestDF, mode='out')),][1:4]
```

```
##      level gender region location
## V12     4      M      E      3
```

Metric Used : **Betweenness Centrality**

People from different knowledge domains turn to this person for information,  
NodesKnowledge1 <-> This User <-> NodesKnowledge2

```
Attr[which.max(betweenness(RequestDF,directed = TRUE)),][1:4]
```

```
##      level gender region location
## V12      4      M      E      3
```

Metric Used : **Eigenvector Centrality**

Influential people have turned to this person for advice.

```
Attr[which.max(evcent(RequestDF)$vector),][1:4]
```

```
##      level gender region location
## V19      3      M      A      1
```

```
g <- RequestDF
V(g)$level <- Attr$level
V(g)$gender <- Attr$gender
V(g)$region <- Attr$region
V(g)$location <- Attr$location
V(g)$level_color <- Attr$level_color
V(g)$gender_color <- Attr$gender_color
V(g)$region_color <- Attr$region_color
V(g)$location_color <- Attr$location_color

count = 0
for (i in which(V(g)$level == 5)){
  if (RequestDF[i,which.max(evcent(RequestDF)$vector)]){
    count = count +1;
  }
}
paste(count,paste("out of",paste(length(which(V(g)$level == 5)),"'level-5' connections.")))

## [1] "2 out of 4 'level-5' connections."
```

## Receive network

(How strongly you agree or disagree that this person has expertise in areas that are important in the kind of work I do.)

Metric Used : **In Degree**

A lot of people think this person is really experienced in their field.

```
Attr[which.max(degree(ReceiveDF, mode='in'))][1:4]
```

```
##      level gender region location
## V20      5      M      A      1
```

Metric Used : **Out Degree**

This person thinks a lot of people in the organization are experienced in this field.

```
Attr[which.max(degree(ReceiveDF, mode='out')),][1:4]
```

```
##      level gender region location
## V22      2      M      E      6
```

Metric Used : **Eigenvector Centrality**

A lot of influential (high-scoring) people in the network think the person has expertise their field.

```
Attr[which.max(evcent(ReceiveDF)$vector),][1:4]
```

```
##      level gender region location
## V20      5      M      A      1
```

This makes sense as the people tend to work in an organization that aligns and matches with their own interests.

## ERGM

### Request network

When asking for advice, do nodes prefer the same Gender or the opposite Gender?

```
g <- RequestDF
V(g)$level <- Attr$level
V(g)$gender <- Attr$gender
V(g)$region <- Attr$region
V(g)$location <- Attr$location
V(g)$level_color <- Attr$level_color
V(g)$gender_color <- Attr$gender_color
V(g)$region_color <- Attr$region_color
V(g)$location_color <- Attr$location_color
```

```
ergmModel <- ergm(asNetwork(g)~edges+nodematch("gender",diff=TRUE,keep=c(1)))
```

```
## Evaluating log-likelihood at the estimate.
```

```
val_fm <- ergmModel$coef["edges"]
val_ff <- ergmModel$coef["nodematch.gender.F"] + ergmModel$coef["edges"]
prob_fm <- exp(val_fm)/(1+exp(val_fm))
prob_ff <- exp(val_ff)/(1+exp(val_ff))
```

```
ergmModel <- ergm(asNetwork(g)~edges+nodematch("gender",diff=TRUE,keep=c(2)))
```

```
## Evaluating log-likelihood at the estimate.
```

```
val_mf <- ergmModel$coef["edges"]
val_mm <- ergmModel$coef["nodematch.gender.M"] + ergmModel$coef["edges"]
prob_mf <- exp(val_mf)/(1+exp(val_mf))
prob_mm <- exp(val_mm)/(1+exp(val_mm))
```

```
## [1] "F-M -> 0.421"
```

```
## [1] "F-F -> 0.473"
```

```
## [1] "M-F -> 0.426"
```

```
## [1] "M-M -> 0.422"
```



Females tend to ask Females for advice as compared to asking Males. Whereas Males are equilikely to ask Males, Females.

Let us now take a look as to how well the Females appreciate or value the advice of Females. We will do this by Modelling the modelling on the Receive Netowrk.

## Receive network

Do Females value the advice they receive from Females?

```
g <- ReceiveDF
V(g)$level <- Attr$level
V(g)$gender <- Attr$gender
V(g)$region <- Attr$region
V(g)$location <- Attr$location
V(g)$level_color <- Attr$level_color
V(g)$gender_color <- Attr$gender_color
V(g)$region_color <- Attr$region_color
V(g)$location_color <- Attr$location_color

ergmModel <- ergm(asNetwork(g)~edges+nodematch("gender",diff=TRUE,keep=c(1)))

## Evaluating log-likelihood at the estimate.
val_fm <- ergmModel$coef["edges"]
val_ff <- ergmModel$coef["nodematch.gender.F"] + ergmModel$coef["edges"]
prob_fm <- exp(val_fm)/(1+exp(val_fm))
prob_ff <- exp(val_ff)/(1+exp(val_ff))

ergmModel <- ergm(asNetwork(g)~edges+nodematch("gender",diff=TRUE,keep=c(2)))

## Evaluating log-likelihood at the estimate.
val_mf <- ergmModel$coef["edges"]
val_mm <- ergmModel$coef["nodematch.gender.M"] + ergmModel$coef["edges"]
prob_mf <- exp(val_mf)/(1+exp(val_mf))
prob_mm <- exp(val_mm)/(1+exp(val_mm))

## [1] "F-M -> 0.42"
## [1] "F-F -> 0.291"
## [1] "M-F -> 0.323"
## [1] "M-M -> 0.48"
```

Females believe Males have higher expertise, as compared to Females in their field. Surprisingly, Males too believe that Males have higher expertise in their field as compared to Females. The number of Females compared to number of Males are lower in the network, hence we could assume that we are getting this result due to a class imbalance in Gender.

## Receive network

How do people in the two different regions interact when asking for advice?

```
g <- ReceiveDF
V(g)$level <- Attr$level
V(g)$gender <- Attr$gender
```

```

V(g)$region <- Attr$region
V(g)$location <- Attr$location
V(g)$level_color <- Attr$level_color
V(g)$gender_color <- Attr$gender_color
V(g)$region_color <- Attr$region_color
V(g)$location_color <- Attr$location_color

ergmModel <- ergm(asNetwork(g)~edges+nodematch("region",diff=TRUE,keep=c(1)))

## Evaluating log-likelihood at the estimate.
val_AE <- ergmModel$coef["edges"]
val_AA <- ergmModel$coef["edges"] + ergmModel$coef["nodematch.region.A"]
prob_AE <- exp(val_AE)/(1+exp(val_AE))
prob_AA <- exp(val_AA)/(1+exp(val_AA))

ergmModel <- ergm(asNetwork(g)~edges+nodematch("region",diff=TRUE,keep=c(2)))

## Evaluating log-likelihood at the estimate.
val_EA <- ergmModel$coef["edges"]
val_EE <- ergmModel$coef["edges"] + ergmModel$coef["nodematch.region.E"]
prob_EA <- exp(val_EA)/(1+exp(val_EA))
prob_EE <- exp(val_EE)/(1+exp(val_EE))

## [1] "A-E -> 0.339"
## [1] "A-A -> 0.575"
## [1] "E-E -> 0.426"
## [1] "E-A -> 0.41"

```

Nodes in Europe are equally likely to ask nodes in America and Europe for advice. Whereas Nodes in America prefer to ask advice from Nodes in America only. The reason this could be happening is that there could be about equal number of nodes in Europe and America working in the same field.

## Receive network

How does the credibility of a person change when people give him/her a vote of confidence?

```

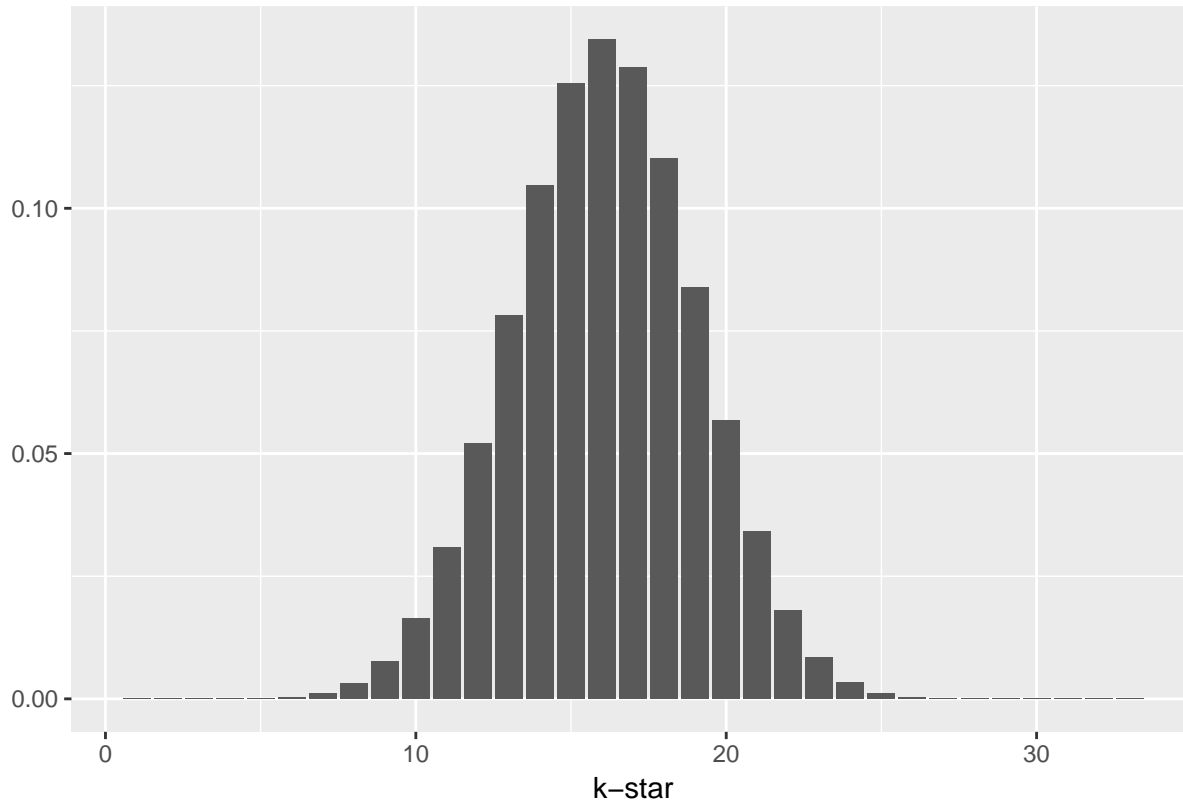
g <- ReceiveDF
V(g)$level <- Attr$level
V(g)$gender <- Attr$gender
V(g)$region <- Attr$region
V(g)$location <- Attr$location
V(g)$level_color <- Attr$level_color
V(g)$gender_color <- Attr$gender_color
V(g)$region_color <- Attr$region_color
V(g)$location_color <- Attr$location_color

num_kStar <- c()
kstar <- c()
for (i in seq(1,33)){
  num_kStar <- append(num_kStar,c(as.numeric(summary(asNetwork(g)~istar(i)))))
  kstar <- append(kstar,c(i))
}

```

```
starDF <- as.data.frame(cbind(kstar,num_kStar))
starDF$num_kStarNo <- num_kStar/sum(num_kStar)
ggplot(starDF,aes(kstar,num_kStarNo))+geom_bar(stat="identity") + scale_y_continuous(labels = comma) +
```

k-star Distribution



```
edgesP <- c()
istarP <- c()
istarNum <- c()
nzKStar <- kstar[num_kStar!=0]
for (i in nzKStar){
  result = tryCatch({
    ergmModel <- ergm(asNetwork(g)~edges + istar(as.numeric(i))) #
    if ( !( ergmModel$coef[paste("edges")] == -Inf | ergmModel$coef[paste("istar",i,sep="")] == -Inf) ) {
      edgesP <- append(edgesP,c(ergmModel$coef["edges"]))
      istarP <- append(istarP,c(ergmModel$coef[paste("istar",i,sep="")])) #
      istarNum <- append(istarNum, c(i))
    }
  }, warning = function(w) {
    print("Warning.....")
  }, error = function(e) {
    print("Error.....")
  })
}
```

## MCMLE estimation did not converge after 20 iterations. The estimated coefficients may not be accurate

```
val_temp <- edgesP+ istarP
prob <- exp(val_temp)/(1+exp(val_temp))
istarNum
```

```
## [1] 2 3 4 32 33
```

```
prob
```

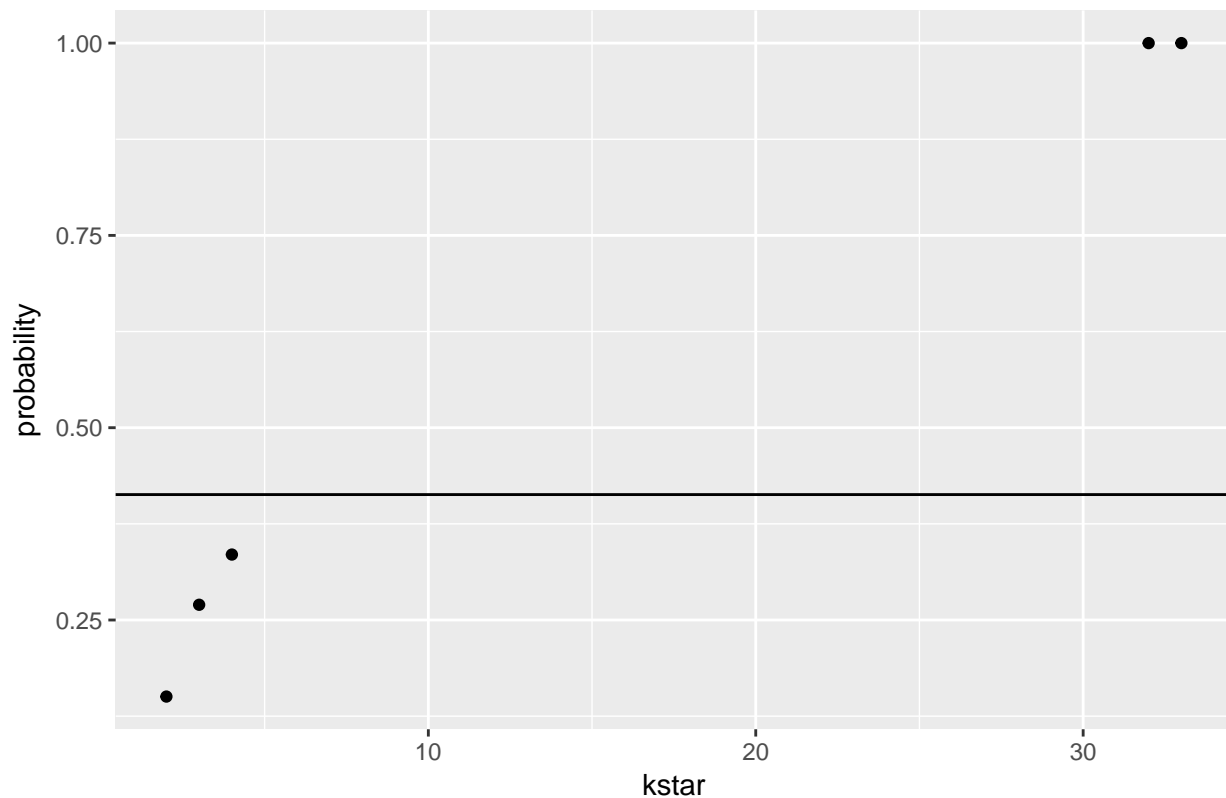
```
## edges edges edges edges edges
```

```
## 0.1504647 0.2698040 0.3350993 1.0000000 1.0000000
```

```
probDegree <- as.data.frame(cbind(istarNum,prob))
```

```
ggplot(probDegree,aes(istarNum,prob))+geom_point()+ geom_hline(aes(yintercept=network.density(asNetworkk
```

## Receive Network – k Star Configuration



The Horizontal line represents the Probability of Link formation using the Network Density only. After running the code multiple times the only k-star configuration that had a high probability were the 32-star and 33-star configuration. While running this code again the model might not converge. But, we can clearly saw that for kstar-32 change in probability of link from 0.0006 (baseline) to 1.00 and for kstar-33 change in probability of link from 0.27(baseline) to 1.00. The Estimate Coefficients for both these k-stars were huge, greater that 800 always.