# Part1_Q3

April 15, 2020

[1]:
```
# # Install Necessary Packages
install.packages("igraph")
install.packages("Matrix")
install.packages("pracma")

# Load Packages
library('igraph')
library('Matrix')
library('pracma')
```

```
Updating HTML index of packages in '.Library'
Making 'packages.html' … done
Updating HTML index of packages in '.Library'
Making 'packages.html' … done
Updating HTML index of packages in '.Library'
Making 'packages.html' … done

Attaching package: ‘igraph’

The following objects are masked from ‘package:stats’:

    decompose, spectrum

The following object is masked from ‘package:base’:

    union


Attaching package: ‘pracma’

The following objects are masked from ‘package:Matrix’:

    expm, lu, tril, triu
```

[2]:
```
# 3a) Probability of linking new vertices

n = 1000
```

```
m = 1
alpha = 1
beta = -1
a = c = d = 1
b = 0

# Create graph with given parameters
g = sample_pa_age(n=n, pa.exp=alpha, ,m=m, aging.exp=beta, zero.deg.appeal=a,
                  zero.age.appeal=b, deg.coef=c, age.coef=d, aging.bin=1000,␣
 ↪directed=FALSE)

deg_distribution = degree.distribution(g)
# take log
idx = which(deg_distribution != 0, arr.ind=TRUE) #remove 0s
x = log(seq(1:length(deg_distribution)))[idx]
y = log(deg_distribution)[idx]

cat(paste("For n = ", n))

# Solve linear Equation:
relation = lm(y ~ x)
print(relation)

png(sprintf("plots/part1/question3/q3a_degree_dist.png"))
plot(degree.distribution(g),main="Degree distribution of the␣
 ↪network",xlab="Degree",ylab="Probability")
dev.off()

png(sprintf("plots/part1/question3/q3a_degree_dist_logplot.png"))
plot(x,y,abline(relation),main=sprintf("Degree distribution of the network for␣
 ↪n=%d (log-log plot)",n),xlab="log(Degree)",ylab="log(Probability)")
dev.off()

png(sprintf("plots/part1/question3/q3a_degree_dist_hist.png"))
hist(degree(g),col=rgb(0.1,0.5,1,.6),main=sprintf("Degree distribution of the␣
 ↪network for n=%d (Histogram)",n),xlab="Degree",ylab="Frequency" )
dev.off()
```

```
For n =  1000
Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
      2.406       -3.555
```

**png:** 2

**png:** 2

**png:** 2

[4]:
```
# 3b) Find community structure and modularity

print(sprintf("Is the graph always connected : %s",is.connected(g)))

png(sprintf("plots/part1/question3/q3b_noetwork.png"))
plot(g, vertex.label="", vertex.size=2, main="Preferential attachment Network")
dev.off()

g_comm = cluster_fast_greedy(g)
community_size = sizes(g_comm)
g_modularity = modularity(g_comm)
cat(paste("\nModularity is ", g_modularity))

png(sprintf("plots/part1/question3/q3b_commstruct.png"))
plot(g_comm, g, main="Community Structure",vertex.size=2, vertex.label=NA)
dev.off()
```

```
[1] "Is the graph always connected : TRUE"
```

**png:** 2

```
Modularity is  0.935354273192115
```

**png:** 2