Practice Igraph

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Creating the environment

Set working directroy

```
setwd("C:/Users/Sebastian/Google Drive/DOCTORADO/INTERNSHIP/Kentucky University/Activities Internship/F
```

Upload packages

```
library(igraph)
```

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

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

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

Create functions

Getting Data

```
karate.vertices <- read.csv("karate_vertices.csv", stringsAsFactors = FALSE)
karate.edges <- read.csv("karate_edges.csv", stringsAsFactors = FALSE)

head(karate.vertices)
```

```
##   Faction    name label color
## 1       1   Mr Hi     H     1
## 2       1 Actor 2     2     1
## 3       1 Actor 3     3     1
## 4       1 Actor 4     4     1
## 5       1 Actor 5     5     1
## 6       1 Actor 6     6     1
```

```
head(karate.edges)
```

```
##    from      to weight
## 1 Mr Hi Actor 2      4
## 2 Mr Hi Actor 3      5
## 3 Mr Hi Actor 4      3
## 4 Mr Hi Actor 5      3
## 5 Mr Hi Actor 6      3
## 6 Mr Hi Actor 7      3
```

Cleaning Data

```
net.karate <- graph.data.frame(karate.edges, directed = FALSE)

summary(net.karate)
```

```
## IGRAPH UNW- 34 78 --
## + attr: name (v/c), weight (e/n)
```

Tidying Data

```
net.karate.1 <- set.vertex.attribute(net.karate, name = "Faction",
                                     value = karate.vertices$Faction)

net.karate.2 <- set.vertex.attribute(net.karate, name = "label",
                                       value = karate.vertices$label)

net.karate.3 <- set.vertex.attribute(net.karate, name = "color",
                                       value = karate.vertices$color)

net.karate.tidied <- net.karate.3

summary(net.karate.tidied)
```

```
## IGRAPH UNW- 34 78 --
## + attr: name (v/c), color (v/n), weight (e/n)
```

Exploratory Analysis

Global properties

Density

```
edge_density(net.karate.tidied, loops = FALSE)
```

```
## [1] 0.1390374
```

Transitivity

```
transitivity(net.karate.tidied, type = "global")
```

```
## [1] 0.2556818
```

Diameter

```
diameter(net.karate.tidied, directed = TRUE, weights = NA)
```

```
## [1] 5
```

Centralization

```
centr_degree(net.karate.tidied, mode = "all")$centralization
```

```
## [1] 0.3761141
```

Local properties

Degree

```
V(net.karate.tidied)$degree <- degree(net.karate.tidied, mode = "all")
```

Betweenness

```
V(net.karate.tidied)$bet <- betweenness(net.karate.tidied)
```

Bonacich

```
V(net.karate.tidied)$bonacich <- power_centrality(net.karate.tidied)
```

Transitivity

```
V(net.karate.tidied)$transitivity <- transitivity(net.karate.tidied, type = "local")
head(as_data_frame(net.karate.tidied, what = "vertices"))
```

```
##            name color degree        bet   bonacich transitivity
## Mr Hi     Mr Hi     1     16 250.150000 -1.5664590    0.1500000
## Actor 2 Actor 2     1      9  33.800000 -0.8916767    0.3333333
## Actor 3 Actor 3     1     10  36.650000 -1.1085710    0.2444444
## Actor 4 Actor 4     1      6   1.333333 -0.1686956    0.6666667
## Actor 5 Actor 5     1      3   0.500000 -1.5664590    0.6666667
## Actor 6 Actor 6     1      4  15.500000 -0.8916767    0.5000000
```

Subgroups and communities

Based on greedy optimization of modularity

```
community <- cluster_fast_greedy(as.undirected(net.karate.tidied))
V(net.karate.tidied)$community <- community$membership
```

```
table(V(net.karate.tidied)$community)
```

```
##
##  1  2  3
## 18 11  5
```

Coreness

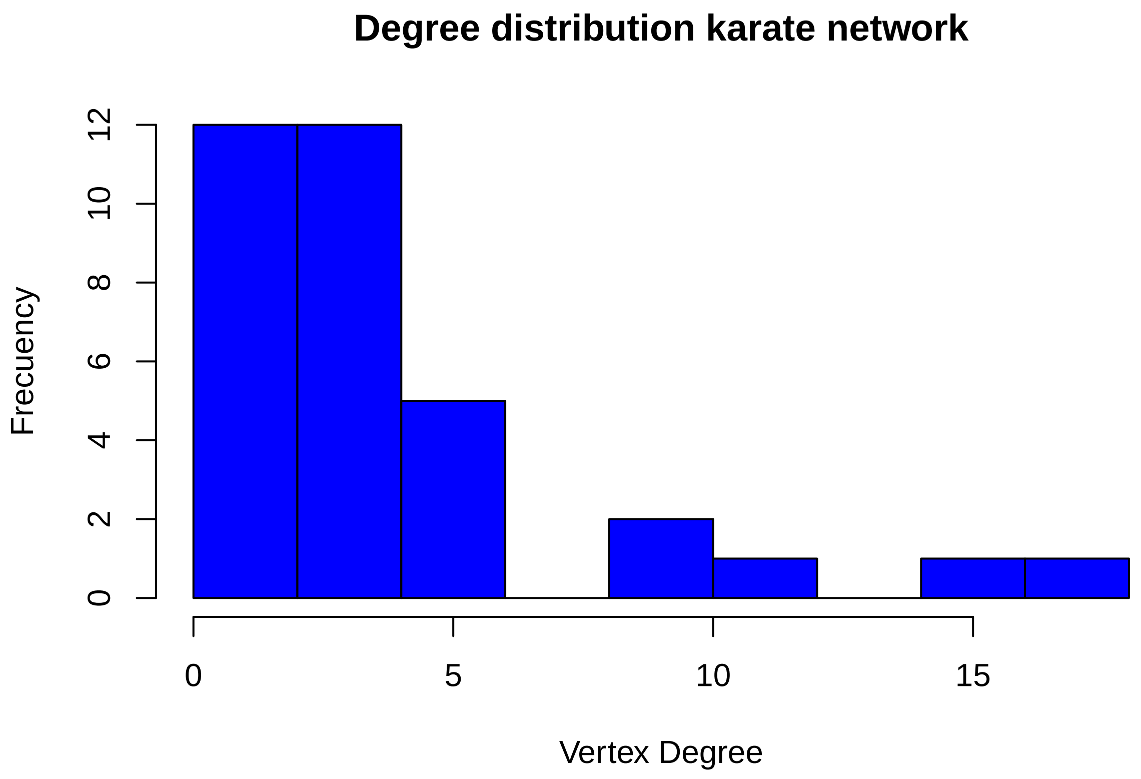
```
V(net.karate.tidied)$coreness <- coreness(net.karate.tidied)

table(V(net.karate.tidied)$coreness)
```

```
##
##  1  2  3  4
##  1 11 12 10
```

Topological properties

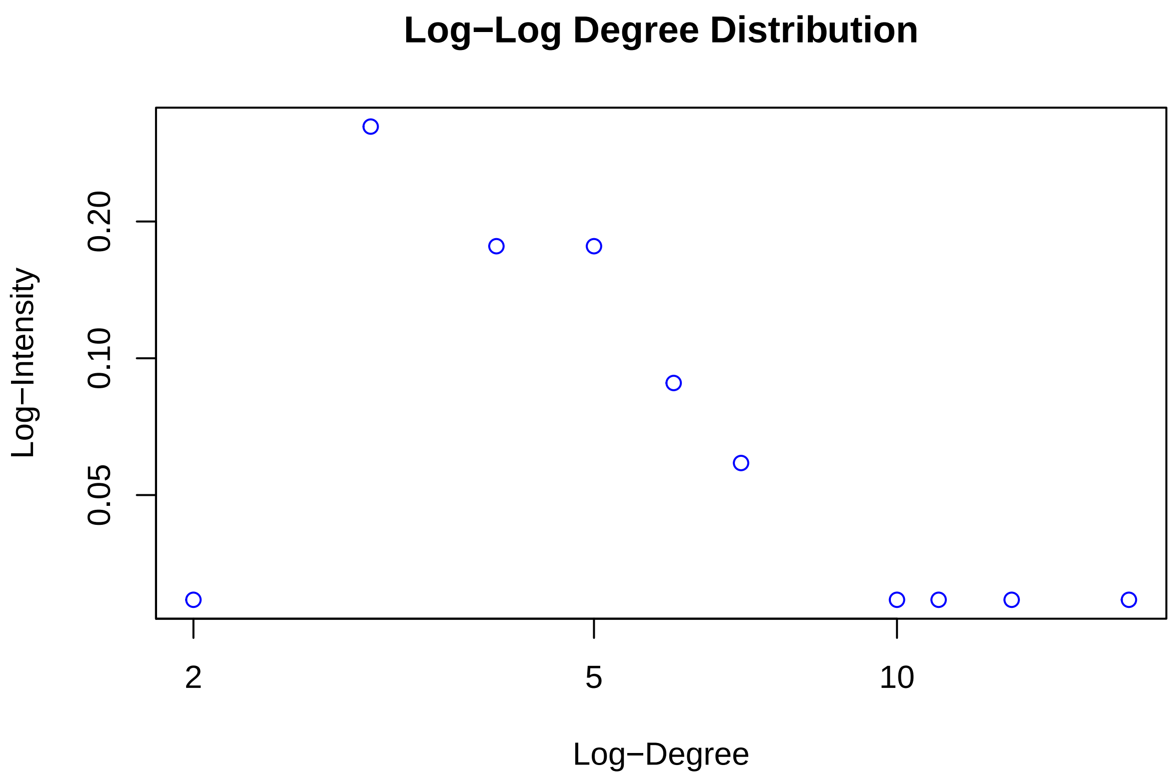
Degree distribution

```
hist(degree(net.karate.tidied, mode = "all"), col="blue",
     main = "Degree distribution karate network",
     xlab = "Vertex Degree", ylab = "Frecuency")
```



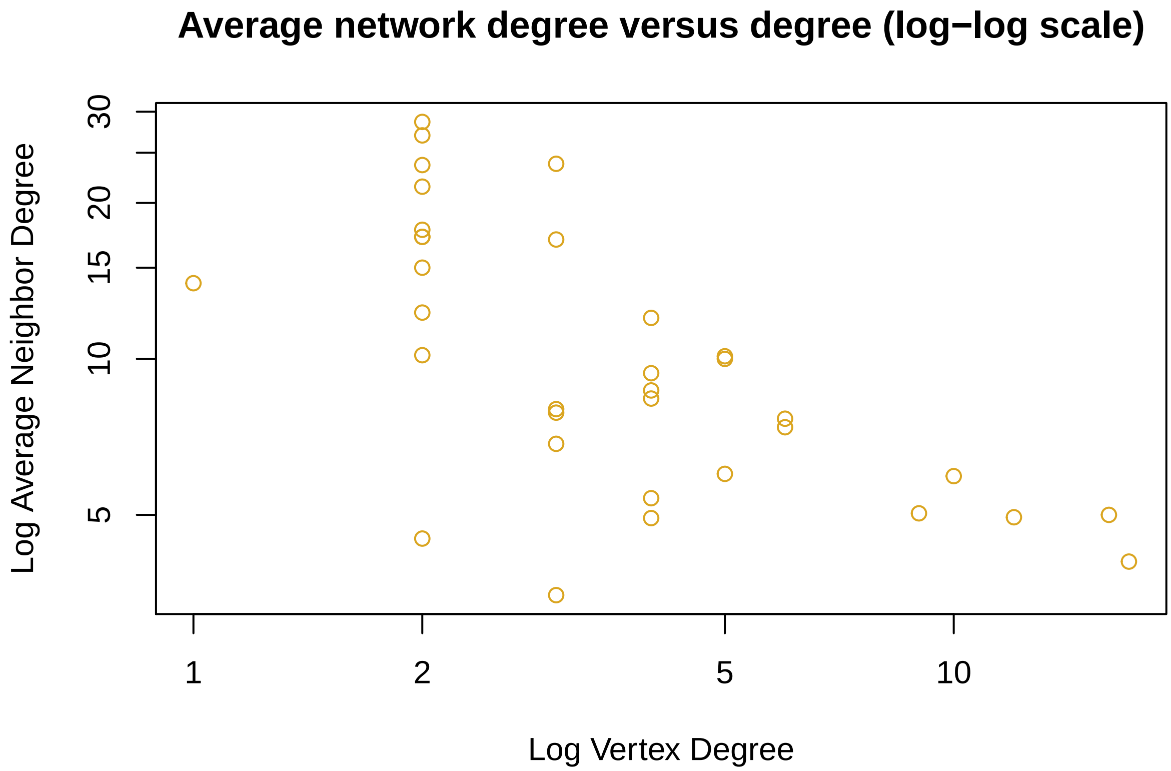
Log-log degree distribution

```
d.net.karate.tidied <- degree(net.karate.tidied, mode = "all")
dd.net.karate.tidied <- degree.distribution(net.karate.tidied)

d <- 1:max(d.net.karate.tidied)
ind <- (dd.net.karate.tidied != 0)
plot(d[ind], dd.net.karate.tidied[ind], log = "xy", col = "blue",
     xlab = c("Log-Degree"), ylab = c("Log-Intensity"),
     main = "Log-Log Degree Distribution")
```



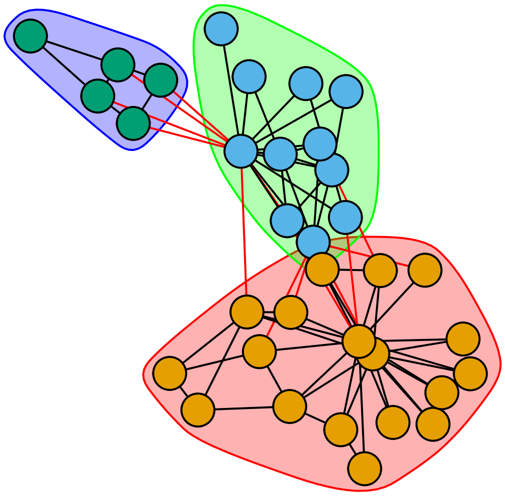
Average network deree versus vertex degree

```
a.nn.deg.net.karate.tidied <- graph.knn(net.karate.tidied,V(net.karate.tidied))$knn
plot(d.net.karate.tidied, a.nn.deg.net.karate.tidied,
     log="xy", col="goldenrod",
     xlab=c("Log Vertex Degree"),
     ylab=c("Log Average Neighbor Degree"),
     main = "Average network degree versus degree (log-log scale)")
```



Visualization

```
kc <- fastgreedy.community(net.karate.tidied)
plot(kc, net.karate.tidied, vertex.label = NA)
```



Export network

```
write.graph(net.karate.tidied, "net_karate_tidied.graphml", "graphml")
```