SNA-R

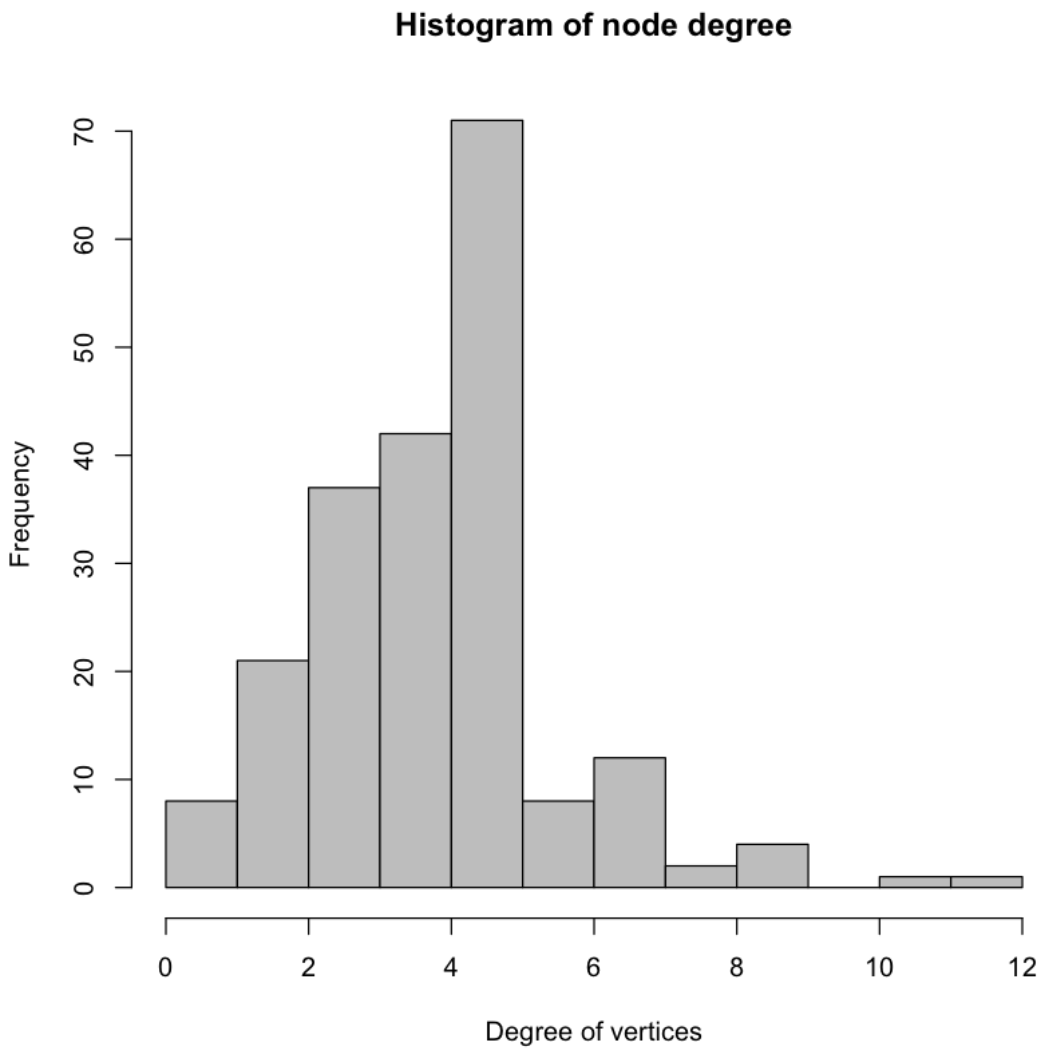
December 3, 2021

[ ]:
```
library(tidyverse)
library(igraph)
```

0.1 Internal - 2020

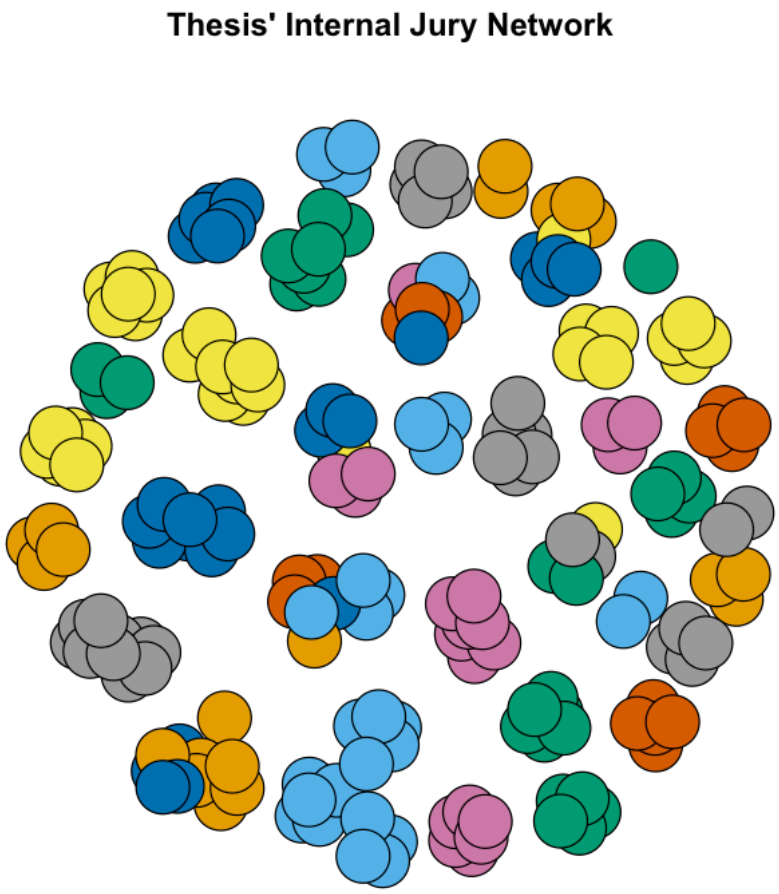
[2]:
```
# Read graph from edges list
edges_int = read_csv("datasets/graph_int.csv", col_names = c("from", "to",␣
↪"weight"), show_col_types = FALSE)
nodes_int = read_csv("datasets/nodes_int.csv", show_col_types = FALSE)
g <- graph_from_data_frame(edges_int, directed = FALSE, vertices = nodes_int)
```

[3]:
```
#Histogram of node degree
V(g)$degree <- degree(g)
hist(V(g)$degree,
     col="grey",
     main='Histogram of node degree',
     ylab='Frequency',
     xlab='Degree of vertices')
```



[4]:
```
encode_ordinal <- function(x, order = unique(x)) {
  x <- as.numeric(factor(x, levels = order, exclude = NULL))
  x
}
V(g)$color = encode_ordinal(V(g)$institutions)
```

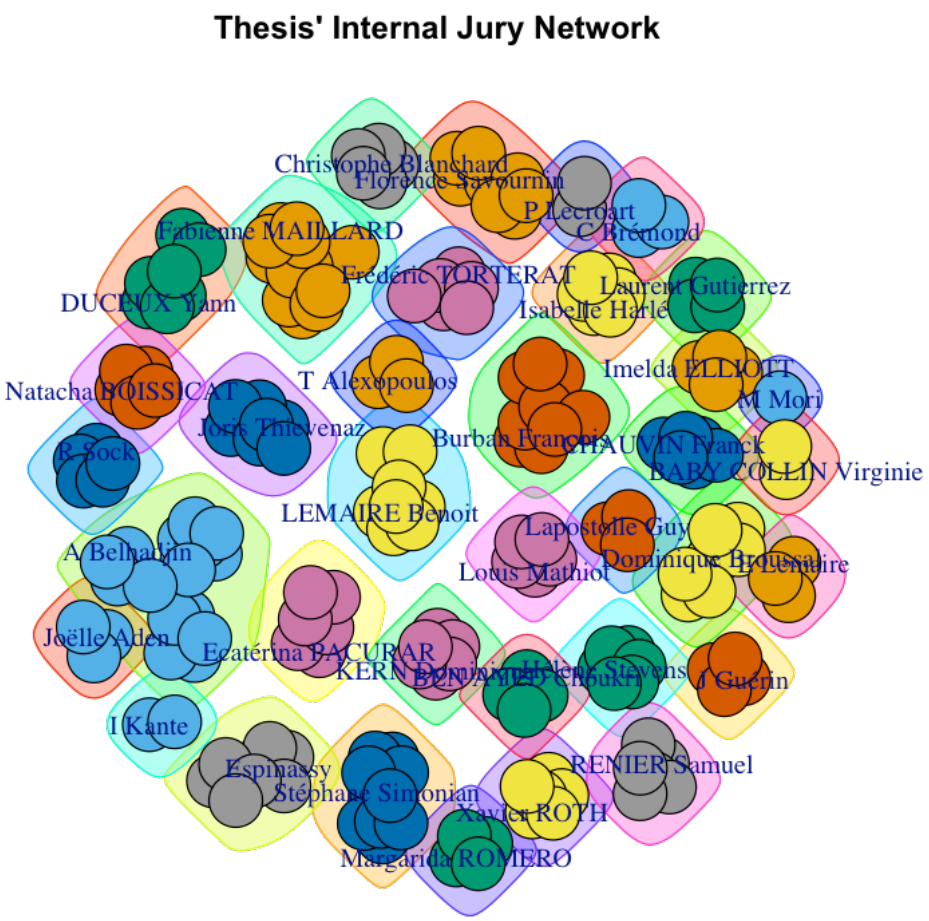
[5]:
```
# Display graph
l <- layout_with_fr(g)
plot(g, layout=l,
     vertex.size = 15,
     vertex.label = NA,
     main = "Thesis' Internal Jury Network")
```



[6]:
```
set.seed(2680)
# Community Detection - Louvain
lc <- cluster_louvain(g)
communities = as.data.frame(communities(lc))[,1]
vector <- character(length(communities))
for (i in 1:length(communities)) {
    vector[i] = unlist(communities[i])[1]
}

plot(lc, g,
     vertex.label = ifelse(V(g)$name %in% vector, V(g)$Name, NA),
     vertex.size = 15,
     vertex.label.cex = 1,
```

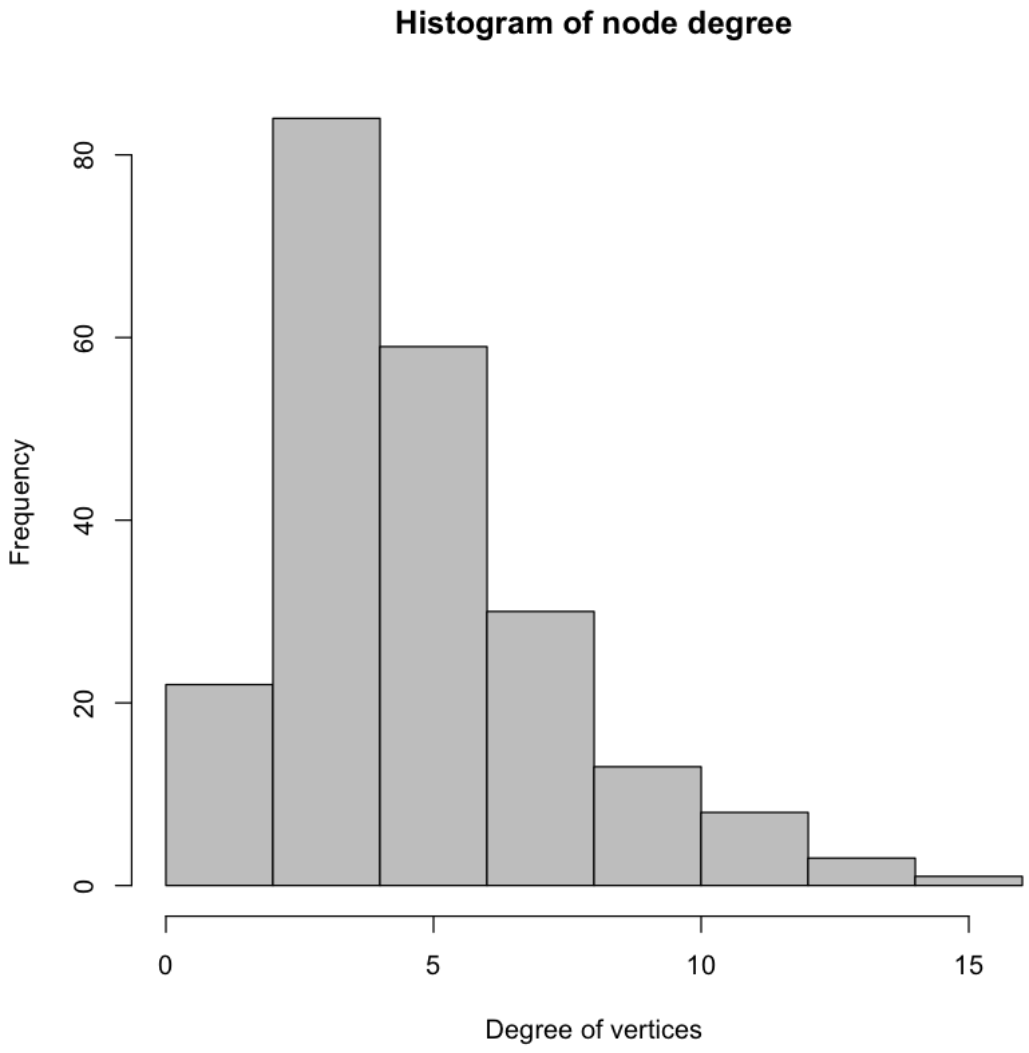
```
    vertex.label = NA,
    main = "Thesis' Internal Jury Network",
   )
```



0.2 External - 2020

[7]:
```
# Read graph from edges list
edges_ext = read_csv("datasets/graph_ext.csv", col_names = c("from", "to",␣
 ↪"weight"), show_col_types = FALSE)
nodes_ext = read_csv("datasets/nodes_ext.csv", show_col_types = FALSE)
g <- graph_from_data_frame(edges_ext, directed = FALSE, vertices = nodes_ext)
```

[8]:
```
#Histogram of node degree
V(g)$degree <- degree(g)
hist(V(g)$degree,
     col="grey",
     main='Histogram of node degree',
     ylab='Frequency',
     xlab='Degree of vertices')
```



[9]:
```
set.seed(2680)
# Community Detection - Louvain
lc <- cluster_louvain(g)
communities = as.data.frame(communities(lc))[,1]
vector <- character(length(communities))
```

```r
for (i in 1:length(communities)) {
    vector[i] = unlist(communities[i])[1]
}

plot(lc, g,
     vertex.label = ifelse(V(g)$name %in% vector, V(g)$Name, NA),
     vertex.size = 15,
     vertex.label.cex = 1,
     vertex.label = NA,
     vertex.label.dist = 0,
     main = "Thesis' External Jury Network",
    )
```