Assignment 1

Smitesh Patil

2023-02-01

1. Basic measures

library(igraph)

## Warning: package 'igraph' was built under R version 4.1.3

library(kableExtra)

## Warning: package 'kableExtra' was built under R version 4.1.3

dib\_graph<-read.graph("dib2.graphml",format="graphml")

1.1 Give the number of nodes and edges

cat("num vertices:", vcount(dib\_graph), "\n")

## num vertices: 8969

cat("num edges :", ecount(dib\_graph), "\n")

## num edges : 46750

1.2 Is the network strongly or weakly connected. If neither, what is the distribution of component sizes.

1. Strongly connected components

strong\_component = as.data.frame(table(factor(components(dib\_graph, mode="strong")$csize)))  
names(strong\_component)[1] = "Component Size"  
kbl(strong\_component)

Component Size

Freq

1

3024

2

180

3

25

4

5

5

1

6

1

5479

1

Answer: The table above shows strongly connected components in the directed network. There is one component with 5479, 6 and 5 nodes and a distribution of component with sizes varying from 1 to 4.

1. Weakly connected components

weak\_component = as.data.frame(table(factor(components(dib\_graph, mode="weak")$csize)))  
names(weak\_component)[1] = "Component Size"  
kable(weak\_component)

Component Size

Freq

2

30

3

11

4

1

8872

1

Answer: The table above shows weakly connected components in the directed network. There is one component with 8872 and 4 nodes with 30 weakly connected components of size 2 and 11 components with size 3.

1.3 What is the diameter of the network ?

cat("The diameter of the network is : ", diameter(dib\_graph, directed = T, unconnected = TRUE, weights=NA), "\n")

## The diameter of the network is : 18

1.4 What is the average path length of the network ?

cat("The average path length of the network :", mean\_distance(dib\_graph, directed = T), "\n")

## The average path length of the network : 6.017593

1.5 What is the clustering coefficient of the network ?

cat("The clustering coeff of the graph is :", transitivity(dib\_graph, type="localaverage"), "\n")

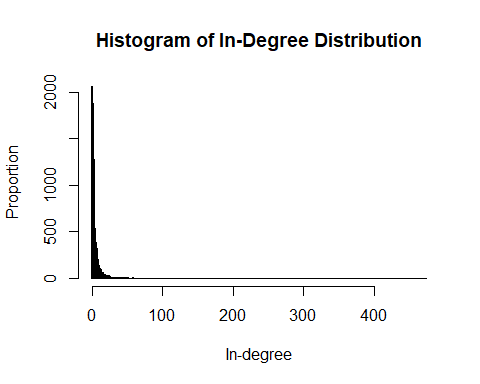
## The clustering coeff of the graph is : 0.2300017

1.6 What is the in- and out-degree distribution ?

deg<-degree(dib\_graph, mode = "in")  
cat("The in-degree distribution of the graph varies from ", min(deg), "to ", max(deg))

## The in-degree distribution of the graph varies from 0 to 473

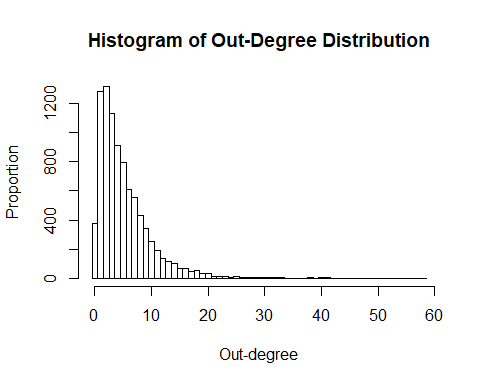
hist(deg,   
 breaks=(min(deg)-1):(max(deg))+0.5,  
 xlab = "In-degree",   
 ylab = "Proportion",   
 main = "Histogram of In-Degree Distribution",  
 border="black",   
 col="white",   
 )



deg<-degree(dib\_graph, mode = "out")  
cat("The in-degree distribution of the graph varies from ", min(deg), "to ", max(deg))

## The in-degree distribution of the graph varies from 0 to 58

hist(deg,   
 breaks=(min(deg)-1):(max(deg))+0.5,  
 xlab = "Out-degree",   
 ylab = "Proportion",   
 main = "Histogram of Out-Degree Distribution",  
 border="black",   
 col="white")

 Section 2 - Bowtie Analysis

2.1 What percentage of the network comprises the strongly connected component, the incomponent, the out-component, and tube.

### Strongly connected components  
clu<-components(dib\_graph,mode = "strong")  
  
scc\_index <- which.max(clu$csize)  
scc<- V(dib\_graph)[which(clu$membership==scc\_index)]$name  
  
scc\_career<- V(dib\_graph)[which(clu$membership==scc\_index)]$career  
scc\_century<- V(dib\_graph)[which(clu$membership==scc\_index)]$century  
  
  
### IN components  
IN\_component=c()  
IN\_component\_career = c()  
IN\_component\_century = c()  
vertices<-V(dib\_graph)  
  
non\_SCC <-vertices[!(vertices$name %in% scc)]  
  
for (v in non\_SCC){  
   
 dist<- bfs(dib\_graph, root=v, mode="out",unreachable =F, dist=T)$dist  
   
 connected\_to\_SCC<-!is.nan(dist[scc])  
   
 if(TRUE %in% connected\_to\_SCC){  
 IN\_component<-c(IN\_component,V(dib\_graph)[v]$name)  
 IN\_component\_career <- c(IN\_component\_career, V(dib\_graph)[v]$career)  
 IN\_component\_century <- c(IN\_component\_century, V(dib\_graph)[v]$century)  
 }  
}  
  
### OUT components  
OUT\_component = c()  
OUT\_component\_career = c()  
OUT\_component\_century = c()  
  
non\_SCC2 <- vertices[!(vertices$name %in% c(scc, IN\_component))]  
  
for (v in non\_SCC2){  
 dist<- bfs(dib\_graph, root=v, mode="in",unreachable =F, dist=T)$dist  
 connected\_to\_non\_SCC<-!is.nan(dist[scc])  
 if(TRUE %in% connected\_to\_non\_SCC){  
 OUT\_component<-c(OUT\_component,V(dib\_graph)[v]$name)  
 OUT\_component\_career <- c(OUT\_component\_career, V(dib\_graph)[v]$career)  
 OUT\_component\_century <- c(OUT\_component\_century, V(dib\_graph)[v]$century)  
 }  
}  
  
### Tube components  
tube = c()  
tube\_career = c()  
tube\_century = c()  
nodes\_minus\_SCC <- vertices[!(vertices$name %in% scc)]  
g\_minus\_SCC<-induced\_subgraph(dib\_graph ,nodes\_minus\_SCC)  
out = c()  
  
for (v in V(g\_minus\_SCC)[IN\_component]){  
 paths<-all\_simple\_paths(g\_minus\_SCC, from=v,to = V(g\_minus\_SCC)[OUT\_component], mode = "out", cutoff = -1)  
 tube<-c(tube, names(unlist(paths)))  
}  
  
tube<-unique(tube)  
  
tube<-tube[!(tube %in% c(IN\_component,OUT\_component))]  
  
  
tube\_data = c()  
tube\_data <- V(dib\_graph)[V(dib\_graph)$name %in% tube]  
tube\_data\_career <- tube\_data$career  
tube\_data\_century <- tube\_data$century  
  
  
### tendril  
  
in\_tendril = c()  
Tendril = vertices[!(vertices$name %in% c(scc, IN\_component, OUT\_component, tube))]$name  
  
  
nodes\_IN\_component\_Tendrils <- vertices[vertices$name %in% c(Tendril, IN\_component)]  
g\_IN\_component\_Tendrils <- induced.subgraph(dib\_graph, nodes\_IN\_component\_Tendrils)  
  
  
for (v in Tendril){  
 paths <- all\_simple\_paths(g\_IN\_component\_Tendrils, from=v,to = V(g\_IN\_component\_Tendrils)[IN\_component], mode = "in", cutoff = -1)  
   
 if(length(paths) > 0){  
 in\_tendril <- c(in\_tendril, v)  
 }  
}  
  
out\_tendril <- c()  
nodes\_OUT\_component\_Tendrils <- vertices[vertices$name %in% c(Tendril, OUT\_component)]  
  
g\_OUT\_component\_Tendrils <- induced.subgraph(dib\_graph, nodes\_OUT\_component\_Tendrils)  
  
for (v in Tendril){  
 paths <- all\_simple\_paths(g\_OUT\_component\_Tendrils, from=v,to = V(g\_OUT\_component\_Tendrils)[OUT\_component], mode = "out", cutoff = -1)  
   
 if (length(paths) > 0){  
 out\_tendril <- c(out\_tendril, v)  
 }  
}

library(dplyr)  
  
components\_prop <- c(length(scc), length(IN\_component), length(OUT\_component), length(tube))  
  
prop\_table <- prop.table(components\_prop) \*100  
  
names(prop\_table) <- c("SCC", "IN Component", "OUT component", "Tube")  
  
kable(prop\_table, col.names = "Percentage")

Percentage

SCC

64.6718602

IN Component

32.5542965

OUT component

2.6322002

Tube

0.1416431

2.2 What are the top five careers for people in the in-component, out-component, scc and tube components?

library(dplyr)  
  
scc\_career <- trimws(unlist(strsplit(scc\_career, split = ",")))  
scc\_career <- as\_tibble(table(scc\_career)) %>% arrange(desc(n)) %>% head()  
  
  
IN\_component\_career <- trimws(unlist(strsplit(IN\_component\_career, split = ",")))  
IN\_component\_career <- as\_tibble(table(IN\_component\_career)) %>% arrange(desc(n)) %>% head()  
  
OUT\_component\_career <- trimws(unlist(strsplit(OUT\_component\_career, split = ",")))  
OUT\_component\_career <- as\_tibble(table(OUT\_component\_career)) %>% arrange(desc(n)) %>% head()  
  
  
tube\_component\_career <- trimws(unlist(strsplit(tube\_data\_career, split = ",")))  
tube\_component\_career <- as\_tibble(table(tube\_component\_career)) %>% arrange(desc(n)) %>% head()

kable(list(scc\_career, IN\_component\_career, OUT\_component\_career,tube\_component\_career), )

scc\_career

n

Politics

1909

Religion

1004

Literature

587

Military

502

Gentry and Aristocracy

486

Administration and Diplomacy

443

IN\_component\_career

n

Politics

591

Religion

487

Literature

312

Business and Finance

240

Journalism and Broadcasting

224

Administration and Diplomacy

209

OUT\_component\_career

n

Religion

37

Science and Technology

26

Politics

25

Sport

20

Administration and Diplomacy

18

Business and Finance

18

tube\_component\_career

n

Military

3

Science and Technology

2

The Sea

2

Travel and Exploration

2

Administration and Diplomacy

1

Archaeology and Antiquarianism

1

2.3 What are the top five centuries represented in the in-component, out-component, scc and tube components?

library(dplyr)  
  
scc\_century <- as\_tibble(table(scc\_century)) %>% arrange(desc(n)) %>% head()  
  
IN\_component\_century <- as\_tibble(table(IN\_component\_century)) %>% arrange(desc(n)) %>% head()  
  
OUT\_component\_century <- as\_tibble(table(OUT\_component\_century)) %>% arrange(desc(n)) %>% head()  
  
tube\_component\_century <- as\_tibble(table(tube\_data\_century)) %>% arrange(desc(n)) %>% head()

kable(list(scc\_century, IN\_component\_century, OUT\_component\_century,tube\_component\_century)) %>%  
 kable\_styling(latex\_options = "hold\_position")

scc\_century

n

19

1871

18

1133

20

597

17

595

16

474

15

118

IN\_component\_century

n

19

1090

20

789

18

419

17

170

16

120

13

36

OUT\_component\_century

n

19

112

20

52

18

39

17

10

16

4

13

3

tube\_data\_century

n

20

8

19

3

18

1

Section 3 - Centrality /Authority

3.1 Produce a table that shows the most influential people in each century using 3 different measures of centrality/authority

library(sjmisc)

## Warning: package 'sjmisc' was built under R version 4.1.3

century\_data <- list(induced.subgraph(dib\_graph, V(dib\_graph)[V(dib\_graph)$century == 9]),  
 induced.subgraph(dib\_graph, V(dib\_graph)[V(dib\_graph)$century == 10]),  
 induced.subgraph(dib\_graph, V(dib\_graph)[V(dib\_graph)$century == 11]),  
 induced.subgraph(dib\_graph, V(dib\_graph)[V(dib\_graph)$century == 12]),  
 induced.subgraph(dib\_graph, V(dib\_graph)[V(dib\_graph)$century == 13]),  
 induced.subgraph(dib\_graph, V(dib\_graph)[V(dib\_graph)$century == 14]),  
 induced.subgraph(dib\_graph, V(dib\_graph)[V(dib\_graph)$century == 15]),  
 induced.subgraph(dib\_graph, V(dib\_graph)[V(dib\_graph)$century == 16]),  
 induced.subgraph(dib\_graph, V(dib\_graph)[V(dib\_graph)$century == 17]),  
 induced.subgraph(dib\_graph, V(dib\_graph)[V(dib\_graph)$century == 18]),  
 induced.subgraph(dib\_graph, V(dib\_graph)[V(dib\_graph)$century == 19]),  
 induced.subgraph(dib\_graph, V(dib\_graph)[V(dib\_graph)$century == 20]))  
  
  
page\_rank <- vector(mode = "list", length = 12)  
katz\_centrality <- vector(mode = "list", length = 12)  
eigen\_centrality <- vector(mode = "list", length = 12)  
between <- vector(mode = "list", length = 12)  
close <- vector(mode = "list", length = 12)   
  
century\_list <- c("9th century",  
 "10th century",  
 "11th century",  
 "12th century",  
 "13th century",  
 "14th century",  
 "15th century",   
 "16th century",  
 "17th century",  
 "18th century",  
 "19th century",  
 "20th century")  
  
names(century\_data) <- century\_list  
  
  
for(i in 1:length(century\_data)){  
 page\_rank[[i]] <- page\_rank(century\_data[[i]], directed = TRUE, damping = 0.85)$vector  
 page\_rank[[i]] <- page\_rank[[i]] %>% sort(decreasing = TRUE) %>% head(1)  
 page\_rank[[i]] <- paste(names(page\_rank[[i]]),"\n",round(page\_rank[[i]],2))  
 eigen\_centrality[[i]] <- eigen\_centrality(century\_data[[i]], weights=NA)$vector  
 eigen\_centrality[[i]] <- eigen\_centrality[[i]] %>% sort(decreasing = TRUE) %>% head(1)  
 eigen\_centrality[[i]] <- paste(names(eigen\_centrality[[i]]),"\n",round(eigen\_centrality[[i]],2))  
 between[[i]] <- betweenness(century\_data[[i]], weights=NA)  
 between[[i]] <- between[[i]] %>% sort(decreasing = TRUE) %>% head(1)  
 between[[i]] <- paste(names(between[[i]]),"\n",round(between[[i]],2))  
  
}  
  
out <- tibble("Page Rank" = unlist(page\_rank),   
 "Eigen Centrality" = unlist(eigen\_centrality),   
 "Between" = unlist(between))  
out <- rotate\_df(out)  
  
colnames(out) <- century\_list  
  
out <- as.data.frame(out)  
  
kbl(out[c('9th century', '10th century', '11th century', '12th century')],   
 caption = "1. Centrality for centuries from 9th to 12th", format = "latex") %>%  
 kable\_styling(full\_width = T, latex\_options = "striped", position = "center") %>%  
 kable\_material()

kbl(out[c('10th century', '11th century', '12th century', '13th century')],  
 caption = "2. Centrality for centuries from 10th to 13th", format = "latex") %>%  
 kable\_styling(full\_width = T, latex\_options = "striped", position = "center") %>%  
 kable\_classic\_2()

kbl(out[c('14th century', '15th century', '16th century', '17th century')],  
 caption = "3. Centrality for centuries from 14th to 17th", format = "latex") %>%  
 kable\_styling(full\_width = T, latex\_options = "striped", position = "center") %>%  
 kable\_minimal()

kbl(out[c('18th century', '19th century', '20th century')],  
 caption = "4. Centrality for centuries from 18th to 20th", format = "latex") %>%  
 kable\_styling(full\_width = T, latex\_options = "striped", position = "center") %>%  
 kable\_paper()

dib\_graph<- dib\_graph + vertices("Smitesh Patil")  
dib\_graph<- dib\_graph + edges("Margaret Alice Joyce","Smitesh Patil")  
  
vertices <- V(dib\_graph)  
  
in\_tendril = c()  
Tendril = vertices[!(vertices$name %in% c(scc, IN\_component, OUT\_component, tube))]$name  
  
  
nodes\_IN\_component\_Tendrils <- vertices[vertices$name %in% c(Tendril, IN\_component)]  
g\_IN\_component\_Tendrils <- induced.subgraph(dib\_graph, nodes\_IN\_component\_Tendrils)  
  
  
for (v in Tendril){  
 paths <- all\_simple\_paths(g\_IN\_component\_Tendrils, from=v,to = V(g\_IN\_component\_Tendrils)[IN\_component], mode = "in", cutoff = -1)  
   
 if(length(paths) > 0){  
 in\_tendril <- c(in\_tendril, v)  
 }  
}  
  
  
for (v in in\_tendril){  
 paths <- all\_simple\_paths(g\_IN\_component\_Tendrils, from= v, to = V(g\_IN\_component\_Tendrils)[IN\_component], mode = "out", cutoff = -1)  
   
 if(length(paths) > 0){  
 in\_tendril <- in\_tendril[!v]  
 }  
}  
  
for (v in out\_tendril){  
 paths <- all\_simple\_paths(g\_OUT\_component\_Tendrils, from= v, to = V(g\_OUT\_component\_Tendrils)[OUT\_component], mode = "in", cutoff = -1)  
   
 if(length(paths) > 0){  
 out\_tendril <- out\_tendril[!v]  
 }  
}