

NSUR

Sophia Toubian

03.19.2024

Contents

1	Make a Graph From Data	1
2	Graph Analytics	4
3	Visualization	5

1 Make a Graph From Data

```
library(ggplot2)
library(igraph)
```

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

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

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

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.4
## v forcats    1.0.0      v stringr    1.5.0
## v lubridate  1.9.3      v tibble     3.2.1
## v purrr      1.0.2      v tidyr      1.3.0

## -- Conflicts ----- tidyverse_conflicts() --
## x lubridate::%--%()      masks igraph::%--%()
## x dplyr::as_data_frame() masks tibble::as_data_frame(), igraph::as_data_frame()
## x purrr::compose()      masks igraph::compose()
```

```
## x tidyr::crossing()      masks igraph::crossing()
## x dplyr::filter()       masks stats::filter()
## x dplyr::lag()           masks stats::lag()
## x purrr::simplify()      masks igraph::simplify()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
## 1
df <- read.csv(url("https://raw.githubusercontent.com/ZacharyST/Data/master/PaulRevereAppD"))
```

```
## 2
dim(df) #254 rows x 8 columns
```

```
## [1] 254 8
```

```
## 3
df[194:215, ]
```

```
##           X StAndrewsLodge LoyalNine NorthCaucus LongRoomClub TeaParty
## 194   Proctor.Edward         0         0         1         0         1
## 195   Pulling.John          0         0         1         0         0
## 196   Pulling.Richard        1         0         0         0         0
## 197   Purkitt.Henry          0         0         0         0         1
## 198   Quincy.Josiah          0         0         0         1         0
## 199   Randall.John           0         0         0         0         1
## 200   Revere.Paul            1         0         1         1         1
## 201   Roby.Joseph            0         0         0         0         1
## 202   Roylson.Thomas         0         0         0         0         0
## 203   Ruddock.Abiel          0         0         1         0         0
## 204   Russell.John           0         0         0         0         1
## 205   Russell.William        0         0         0         0         1
## 206   Sessions.Robert        0         0         0         0         1
## 207   Seward.James           1         0         0         0         0
## 208   Sharp.Gibbens          0         0         1         0         0
## 209   Shed.Joseph            0         0         0         0         1
## 210   Sigourney.John         0         0         1         0         0
## 211   Simpson.Benjamin        0         0         0         0         1
## 212   Slater.Peter           0         0         0         0         1
## 213   Sloper.Ambrose         1         0         0         0         0
## 214   Smith.John             0         1         0         0         0
## 215   Spear.Thomas           0         0         0         0         1
##           BostonCommittee LondonEnemies
## 194           0           1
## 195           0           1
## 196           0           0
## 197           0           0
## 198           1           0
## 199           0           0
## 200           0           1
## 201           0           0
## 202           0           1
## 203           0           1
## 204           0           0
## 205           0           0
```

```
## 206      0      0
## 207      0      0
## 208      0      0
## 209      0      0
## 210      0      0
## 211      0      0
## 212      0      0
## 213      0      0
## 214      0      0
## 215      0      0
```

rows are people and columns are places

```
## 4
names(df)[1] <- "Individual"
colnames(df)
```

```
## [1] "Individual"      "StAndrewsLodge"  "LoyalNine"       "NorthCaucus"
## [5] "LongRoomClub"    "TeaParty"        "BostonCommittee" "LondonEnemies"
```

```
## 5
length(which(df$TeaParty == 1)) #97
```

```
## [1] 97
```

```
## 6
rownames(df) <- df$Individual # The first column is the individuals' names
df <- df[, -1] # Drop the names column so that the next line and igraph do not treat it as a node.
temp <- data.matrix(df)
rownames(temp) <- rownames(df)
colnames(temp) <- colnames(df)
class(temp)
```

```
## [1] "matrix" "array"
```

```
boston <- graph_from_biadjacency_matrix(temp, )
```

```
# 7
summary(vertex_attr(boston)) #Length Class Mode type 261 -none- logical name 261 -none- character
```

```
##      Length Class Mode
## type 261    -none- logical
## name 261    -none- character
```

```
## 8
people <- bipartite_projection(boston)$proj1
orgs <- bipartite_projection(boston)$proj2
```

```
## 9
vcount(people) #254
```

```
## [1] 254
```

```
vcount(orgs) #7
```

```
## [1] 7
```

2 Graph Analytics

```
## 10a
avg_degree_people <- mean(degree(people))
avg_degree_orgs <- mean(degree(orgs))
difference_avg_degree <- abs(avg_degree_people - avg_degree_orgs)
difference_avg_degree #70.99663
```

```
## [1] 70.99663
```

```
## 10b. The graph with the higher average degree centrality The graph with the
## higher average degree centrality is indicated by the higher value obtained
## from the previous calculations. (the people graph has the higher degree
## centrality)
```

```
## 11
transitivity_people <- transitivity(people)
transitivity_orgs <- transitivity(orgs)
print(transitivity_people)
```

```
## [1] 0.8177028
```

```
print(transitivity_orgs)
```

```
## [1] 0.9069767
```

```
# orgs has the higher transitivity
```

```
## 12
fewest_shared_org <- V(orgs)$name[which.min(degree(orgs))]
print(fewest_shared_org) #LoyalNine
```

```
## [1] "LoyalNine"
```

```
## 13
most_connected_individual <- V(people)$name[which.max(degree(people))]
print(most_connected_individual) #Paul Revere is the most connected individual
```

```
## [1] "Revere.Paul"
```

```
## 14
bc_norm <- betweenness(people, normalized = TRUE)
V(people)$bc_norm <- bc_norm
summary(vertex_attr(people))
```

```
##           Length Class  Mode
## name      254    -none- character
## bc_norm    254    -none- numeric
```

```
## 15
eigenc <- eigen centrality(people)$vector
V(people)$eigenc <- eigenc
summary(vertex_attr(people))
```

```
##           Length Class  Mode
## name      254    -none- character
## bc_norm    254    -none- numeric
## eigenc     254    -none- numeric
```

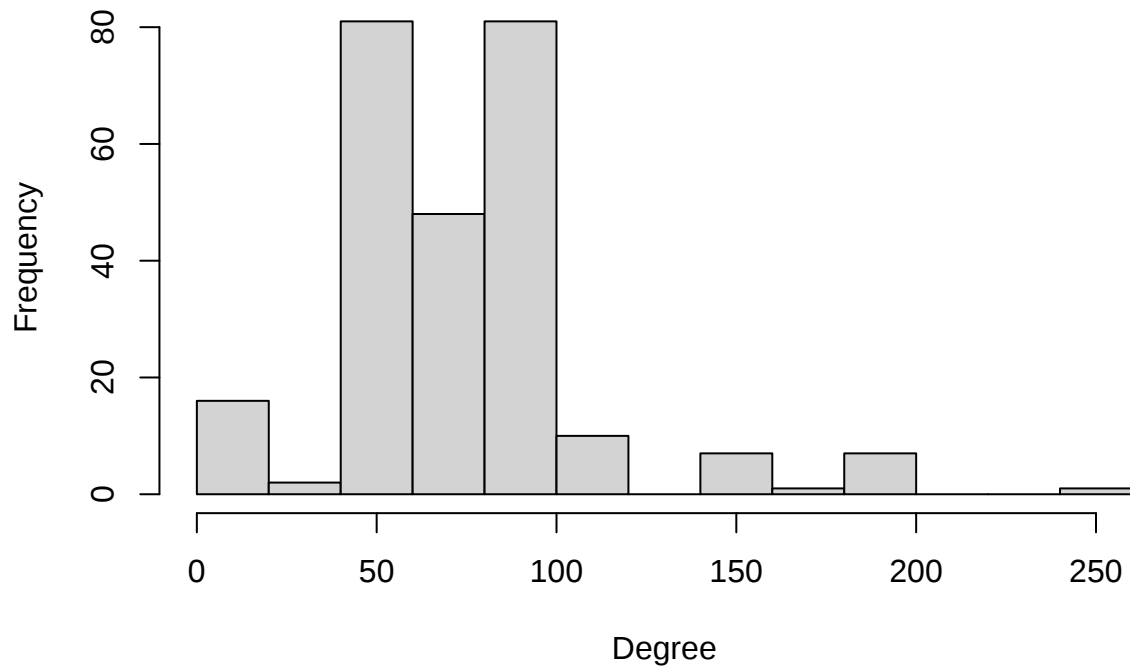
```
## 16
set.seed(277)
community <- cluster_louvain(people)$membership
V(people)$community <- community
summary(vertex_attr(people))
```

```
##           Length Class  Mode
## name      254    -none- character
## bc_norm    254    -none- numeric
## eigenc     254    -none- numeric
## community  254    -none- numeric
```

3 Visualization

```
## 17
hist(degree(people), main = "Degree Distribution of People Graph", xlab = "Degree",
     ylab = "Frequency")
```

Degree Distribution of People Graph

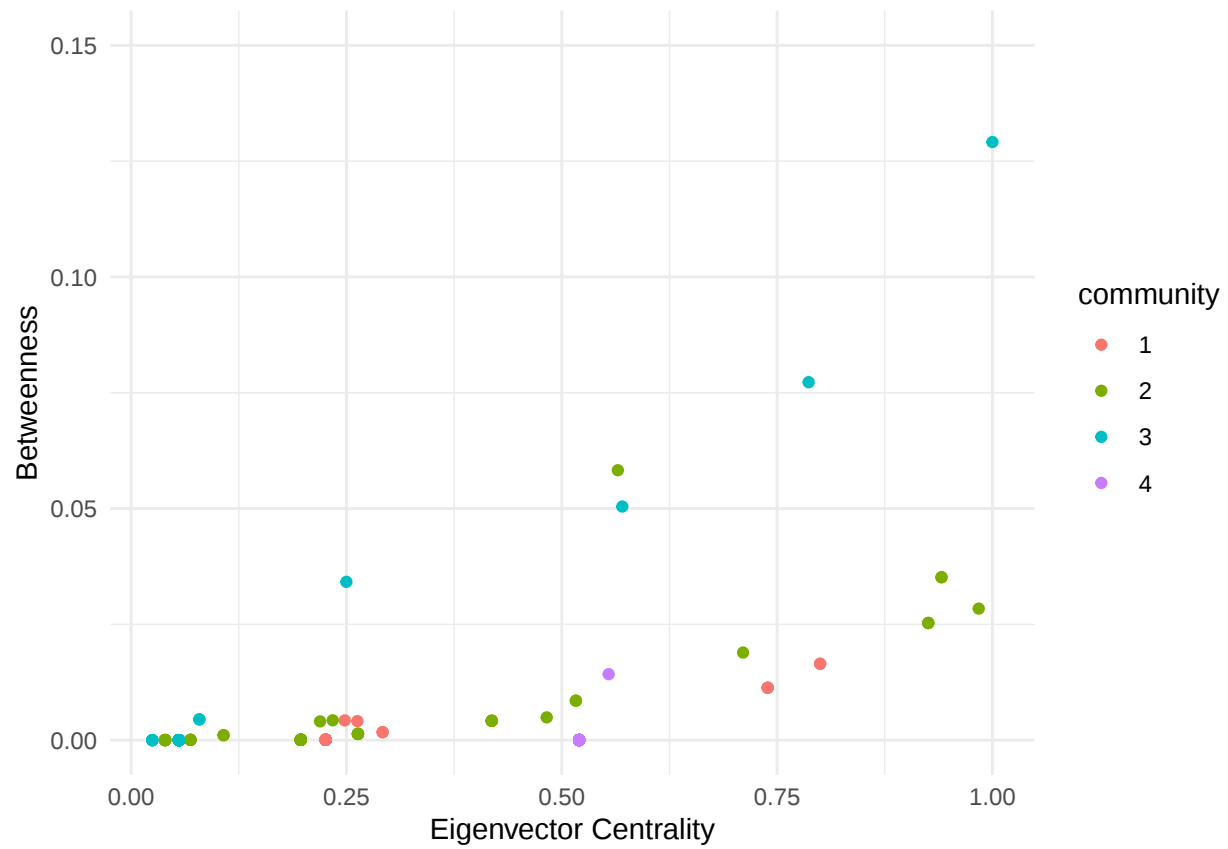


```
ggsave("final_histogram.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
## 18
```

```
plot_data <- data.frame(eigenc = V(people)$eigenc, bc_norm = V(people)$bc_norm, community = factor(V(people)$community))
ggplot(plot_data, aes(x = eigenc, y = bc_norm, color = community)) + geom_point() +
  labs(x = "Eigenvector Centrality", y = "Betweenness") + scale_y_continuous(limits = c(0, 0.15)) + theme_minimal()
```



```
ggsave("final_3d.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
## 19  
library(ggraph)  
plot.new()
```

```

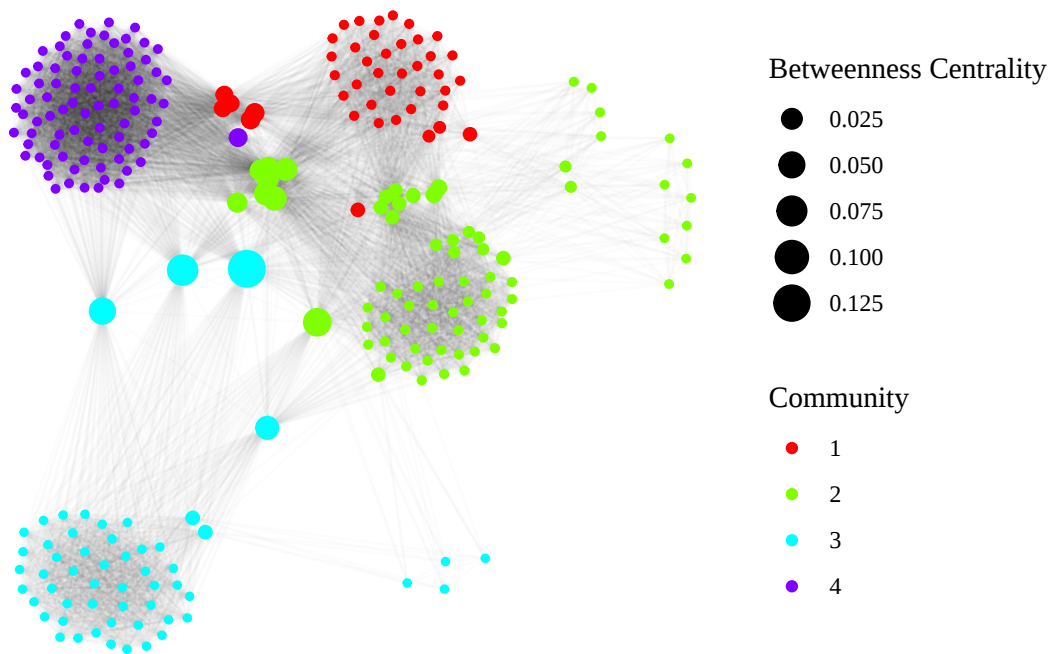
num_communities <- max(V(people)$community)
community_palette <- rainbow(num_communities)
ggraph(people, layout = "fr") + geom_edge_link(alpha = 0.01) + geom_node_point(aes(size = bc_norm,
  color = as.character(community))) + scale_color_manual(values = community_palette) +
  theme_graph(base_family = "Times") + labs(size = "Betweenness Centrality", color = "Community")

```

```

## Warning: Using the 'size' aesthetic in this geom was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' in the 'default_aes' field and elsewhere instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

```

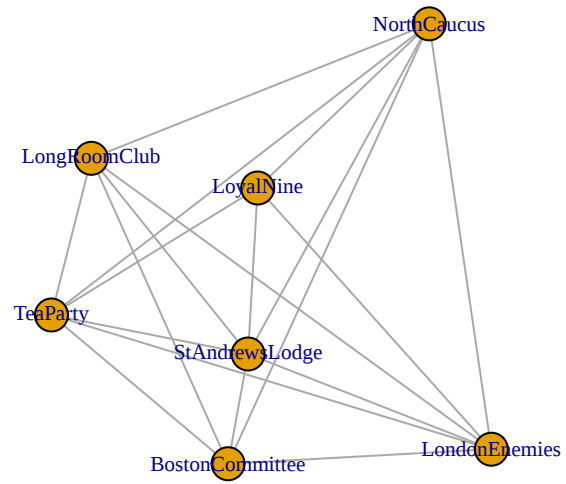


```
ggsave("final_network_people.png", "~/Desktop/PP C277/Assignments/Final")
```

```
## Saving 6.5 x 4.5 in image
```

```
## Error in x$theme: $ operator is invalid for atomic vectors
```

```
## 20
layout <- layout_with_kk(orgs)
plot(orgs, layout = layout, vertex.label = V(orgs)$name, vertex.label.cex = 0.7)
```



```
jpeg("final_network_orgs.jpeg", width = 800, height = 600)
```