

Data 620 Assignment 3 Graph Visualization

Benjamin Bravo

2026-02-23

Introduction About Dataset

The dataset represents email communication within a large European research institution. Each node represents a person, and a directed edge means that one person sent an email to another. The data only includes emails exchanged between members of the institution, not messages sent outside the organization. If an email was sent to multiple recipients, a separate edge is created for each person who received it.

There are also four smaller sub-networks that show communication within individual departments. The static version of the network represents the largest connected group of people in the organization.

I chose this dataset because it clearly represents real-world communication patterns and is well-suited for network analysis. Email networks naturally form directed graphs, making it easy to analyze metrics like diameter and centrality while also providing meaningful insights into how information flows within an organization.

Dataset link: <https://snap.stanford.edu/data/email-Eu-core-temporal.html>

Loading Data and Graph

```
install.packages("igraph")

## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.5'
## (as 'lib' is unspecified)
library(igraph)

##
## Attaching package: 'igraph'

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

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

url <- "https://raw.githubusercontent.com/bb2955/Data-620/main/email-Eu-core-temporal-Dept1.txt"

data <- read.table(url, header = FALSE)
```

Code Explanation:

This code installs the igraph package which allows us to create and analyze networks. The library code allows us to load the igraph package so we can use its functions. Then I stored the Github file link in a variable

called url, which tells R where the dataset is located online. I picked only the department 1 dataset. I then stored the data in an object called ‘data’.

```
colnames(data) <- c("from", "to", "time")
```

Code Explanation:

This code renames the columns.

```
data_small <- data[1:3000, ]
```

Code Explanation:

This code selects the first 3,000 rows to keep the graph smaller and easier to visualize and stores it in data_small.

```
g <- graph_from_data_frame(data_small[,1:2], directed = TRUE)
```

```
summary(g)
```

```
## IGRAPH 377d640 DN-- 43 3000 --
## + attr: name (v/c)
```

Code Explanation:

This code creates the network graph by first specifying to only use columns 1 and 2 (which are the sender and receiver columns) and has the arrows show email direction (directed = TRUE) and stores it in g. Then it shows a quick summary telling you the number of nodes (people) and the number of edges (emails).

Graph Analysis

Graph Diameter

```
diameter(g, directed = TRUE, unconnected = TRUE)
```

```
## [1] 4
```

Code Explanation:

This code calculates the diameter, which is the longest shortest path between two nodes. The (unconnected = TRUE) allows calculation even if the graph isn’t fully connected. This tells us the maximum number of steps needed for communication across the network.

Degree Centrality

```
deg <- degree(g)
```

```
summary(deg)
```

```
##      Min. 1st Qu. Median    Mean 3rd Qu.    Max.
##      1.0   17.0   95.0 139.5 174.0 940.0
```

Code Explanation:

Deg calculates how many connections each person has. Then summary(deg) shows the statistics such as the minimum connections, maximum connections, and average connections.

```
which.max(deg)
```

```
## 28
## 4
```

Code Explanation:

This code finds the person with the most connections, which is likely a central communicator.

Number of Nodes and Edges

```
vcount(g)
```

```
## [1] 43
ecount(g)
```

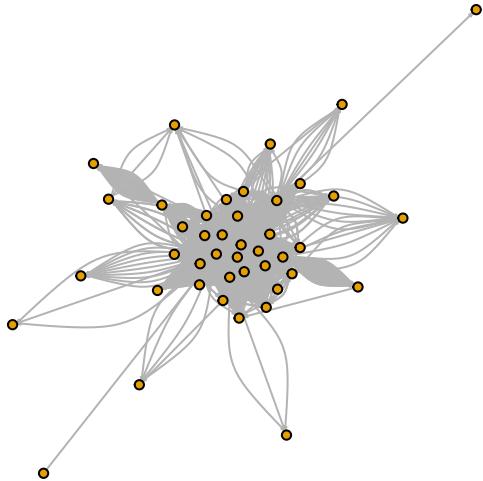
```
## [1] 3000
```

Code Explanation:

This code counts the number of nodes (people), and counts the number of edges (emails).

Visualization

```
plot(g,
      vertex.size = 4,
      vertex.label = NA,
      edge.arrow.size = 0.1,
      edge.color = "gray70",
      layout = layout_with_fr)
```



Code Explanation:

This code draws the graph, specifies the nodes to be small, then removes names to reduce clutter, makes the arrows smaller, and finally uses a force-directed layout that spreads nodes naturally. This creates the visual network.