Social Network Analysis Demonstration

The LASER Team

August 11, 2021

Welcome to the *social network analysis* demo! To complete this, click the green arrows to the right of each code chunk.

1. Loading, Setting Up

In this section, we load packages with the `library()` function and read data using the `read_csv()` function.

- `d` refers to the data we loaded on teachers’ relations
- `u` refers to “user”-level data (e.g., on teachers’ years of experience)

```
set.seed(0811)

library(tidyverse)
library(tidygraph)
```

```
##
## Attaching package: 'tidygraph'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
library(ggraph)
library(here)

d <- read_csv(here('data', 'teacher-network-data-relations.csv'))
```

```
##
## ── Column specification ────────────────────────────────────────────────
────────────────────────────────────────────────────────────────
## cols(
##   name = col_character(),
##   most_helpful_1 = col_character(),
##   most_helpful_2 = col_character(),
##   most_helpful_3 = col_character()
## )
```

```
## Warning: 1 parsing failure.
## row col  expected    actual
file
##   4  -- 4 columns 3 columns '/Users/joshuarosenberg/aera-workshop/data/teacher
-network-data-relations.csv'
```

```
u <- read_csv(here('data', 'teacher-network-data-users.csv'))
```

```
##
## ── Column specification ────────────────────────────────────────────────
────────────────────────────────────────────────────────────────
## cols(
##   name = col_character(),
##   years_of_experience = col_double(),
##   subject = col_character()
## )
```

Your Turn ⤵

Run the following two code chunks to take a *glimpse* at your data. Below, add a few notes on what you *notice* and *wonder*

```
glimpse(d)
```

```
## Rows: 12
## Columns: 4
## $ name           <chr> "Mikayla", "Dylan", "Ilyaas", "Wayne", "Kaitlyn", "Camr
on", "Melody", "Lakota", "Stephanie", "Jessica", "Faseeha", "Pa…
## $ most_helpful_1 <chr> "Dylan", "Faseeha", "Mikayla", "Melody", "Melody", "Mel
ody", "Wayne", "Camron", "Jessica", "Faseeha", "Kaitlyn", "Step…
## $ most_helpful_2 <chr> NA, NA, "Melody", "Faseeha", NA, "Lakota", "Faseeha", N
A, "Patrick", "Melody", "Mikayla", "Jessica"
## $ most_helpful_3 <chr> NA, NA, NA, NA, NA, NA, NA, NA, "Melody", NA, "Melody",
NA
```

```
glimpse(u)
```

```
## Rows: 12
## Columns: 3
## $ name                <chr> "Mikayla", "Dylan", "Ilyaas", "Wayne", "Kaitlyn",
"Camron", "Melody", "Lakota", "Stephanie", "Jessica", "Faseeha"…
## $ years_of_experience <dbl> 1, 27, 4, 7, 5, 0, 10, 2, 3, 2, 18, 1
## $ subject             <chr> "Chemistry", "Biology", "Biology", "Biology", "Bio
logy", "Chemistry", "Chemistry", "Chemistry", "Biology", "Biolo…
```

What do you *notice* and/or *wonder* about this data? Add a note or two below!

-
-

2. Preparing Data

In this section, we prepare our data to be in edgelist format.

Your Turn ⤵

```
d_long <- d %>%
  pivot_longer(most_helpful_1:most_helpful_3, names_to = "nominee") %>%
  mutate(nominee_rank = str_sub(nominee, start = -1),
         nominee_rank = as.integer(nominee_rank)) %>%
  select(-nominee) %>%
  filter(!is.na(value))

d_long
```

```
## # A tibble: 22 x 3
##    name    value   nominee_rank
##    <chr>   <chr>          <int>
##  1 Mikayla Dylan              1
##  2 Dylan   Faseeha            1
##  3 Ilyaas  Mikayla            1
##  4 Ilyaas  Melody             2
##  5 Wayne   Melody             1
##  6 Wayne   Faseeha            2
##  7 Kaitlyn Melody             1
##  8 Camron  Melody             1
##  9 Camron  Lakota             2
## 10 Melody  Wayne               1
## # … with 12 more rows
```

What is different about the edgelist data now? Add one or more observations (you can add additional observations by adding more dashes):

-
-

3. Creating a Graph ‘Object’ and Preparing to Visualize the Network

This next step is key in that we use the `tbl_graph()` function to create a network "object"; if that sounds a bit vague, it should! An "object" refers to a type of data in R. Here, it's one that is specific to the packages we are using for social network analysis.

Here, we create a network object with only the edgelist.

```
g <- tbl_graph(edges = d_long)

g
```

```
## # A tbl_graph: 12 nodes and 22 edges
## #
## # A directed simple graph with 1 component
## #
## # Node Data: 12 x 1 (active)
##   name
##   <chr>
## 1 Mikayla
## 2 Dylan
## 3 Faseeha
## 4 Ilyaas
## 5 Melody
## 6 Wayne
## # … with 6 more rows
## #
## # Edge Data: 22 x 3
##    from    to nominee_rank
##   <int> <int>        <int>
## 1     1     2            1
## 2     2     3            1
## 3     4     1            1
## # … with 19 more rows
```

Your Turn ⤵

Let's create this object again, but also adding *nodes* information, or the user-level information we also loaded earlier on.

```
g <- tbl_graph(edges = d_long, nodes = u)

g
```

```
## # A tbl_graph: 12 nodes and 22 edges
## #
## # A directed simple graph with 1 component
## #
## # Node Data: 12 x 3 (active)
##   name    years_of_experience subject
##   <chr>                 <dbl> <chr>
## 1 Mikayla                   1 Chemistry
## 2 Dylan                    27 Biology
## 3 Ilyaas                    4 Biology
## 4 Wayne                     7 Biology
## 5 Kaitlyn                   5 Biology
## 6 Camron                    0 Chemistry
## # … with 6 more rows
## #
## # Edge Data: 22 x 3
##    from    to nominee_rank
##   <int> <int>        <int>
## 1     1     2            1
## 2     2    11            1
## 3     3     1            1
## # … with 19 more rows
```

```
g <- g %>%
  mutate(popularity = centrality_degree(mode = 'in')) %>%
  activate("edges") %>%
  mutate(nominee_rank = as.factor(nominee_rank))

g
```

```
## # A tbl_graph: 12 nodes and 22 edges
## #
## # A directed simple graph with 1 component
## #
## # Edge Data: 22 x 3 (active)
##    from    to nominee_rank
##   <int> <int> <fct>
## 1     1     2 1
## 2     2    11 1
## 3     3     1 1
## 4     3     7 2
## 5     4     7 1
## 6     4    11 2
## # … with 16 more rows
## #
## # Node Data: 12 x 4
##   name    years_of_experience subject   popularity
##   <chr>                 <dbl> <chr>          <dbl>
## 1 Mikayla                   1 Chemistry          2
## 2 Dylan                    27 Biology            1
## 3 Ilyaas                    4 Biology            0
## # … with 9 more rows
```

What do you notice about the `g` network object? How does it appear different from either the edgelist or the user-level data we loaded?

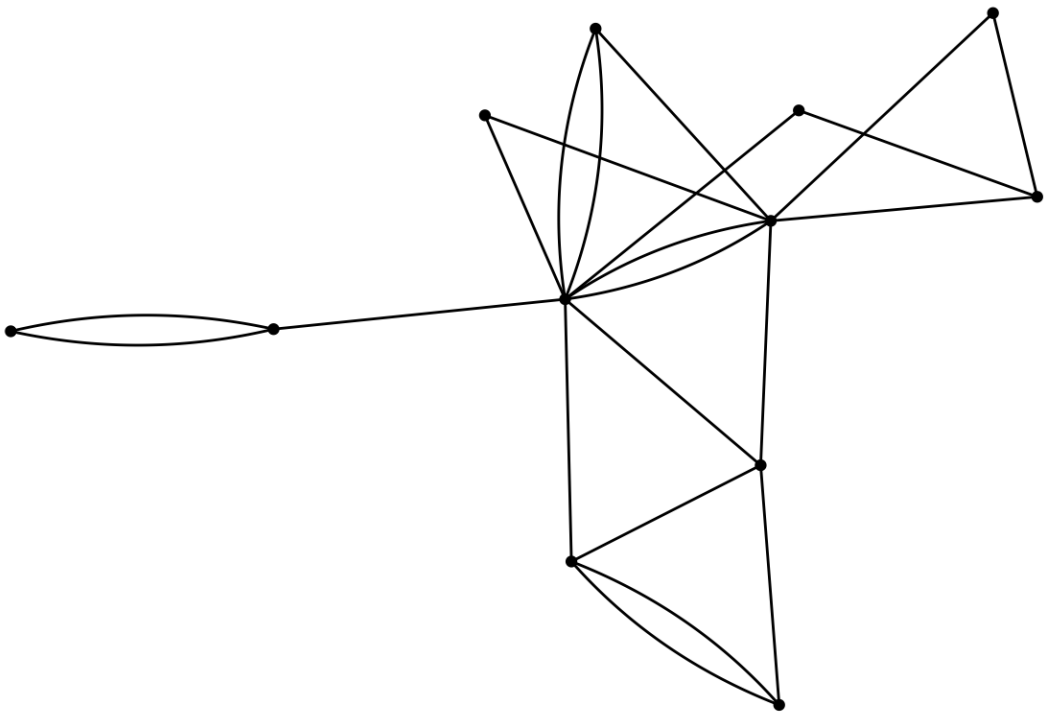
-
-

4. Creating and Refining a Network Visualization

Perhaps our question has to do with who is most central (and, possibly, the most influential) within our network.

Let's start with a simple visualization of our network using `geom_edge_fan()` and `geom_node_point()`.

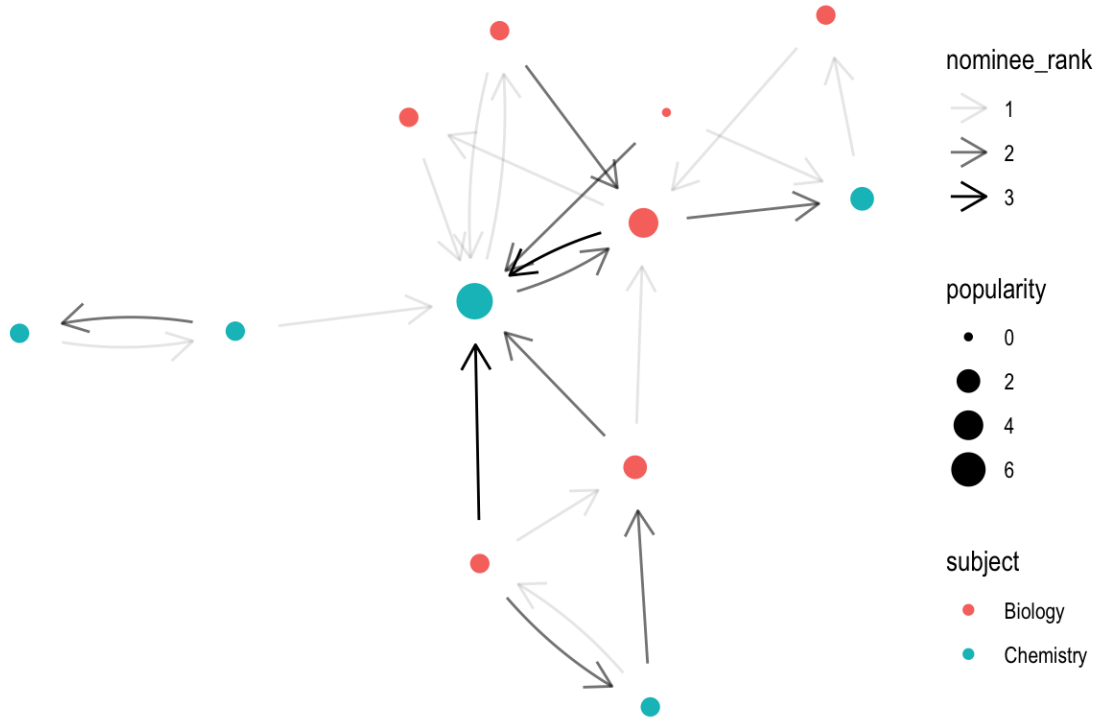
```
ggraph(g, layout = 'kk') +
  geom_edge_fan() +
  geom_node_point() +
  theme_graph()
```



We can enhance this visualization in numerous ways, such as by:

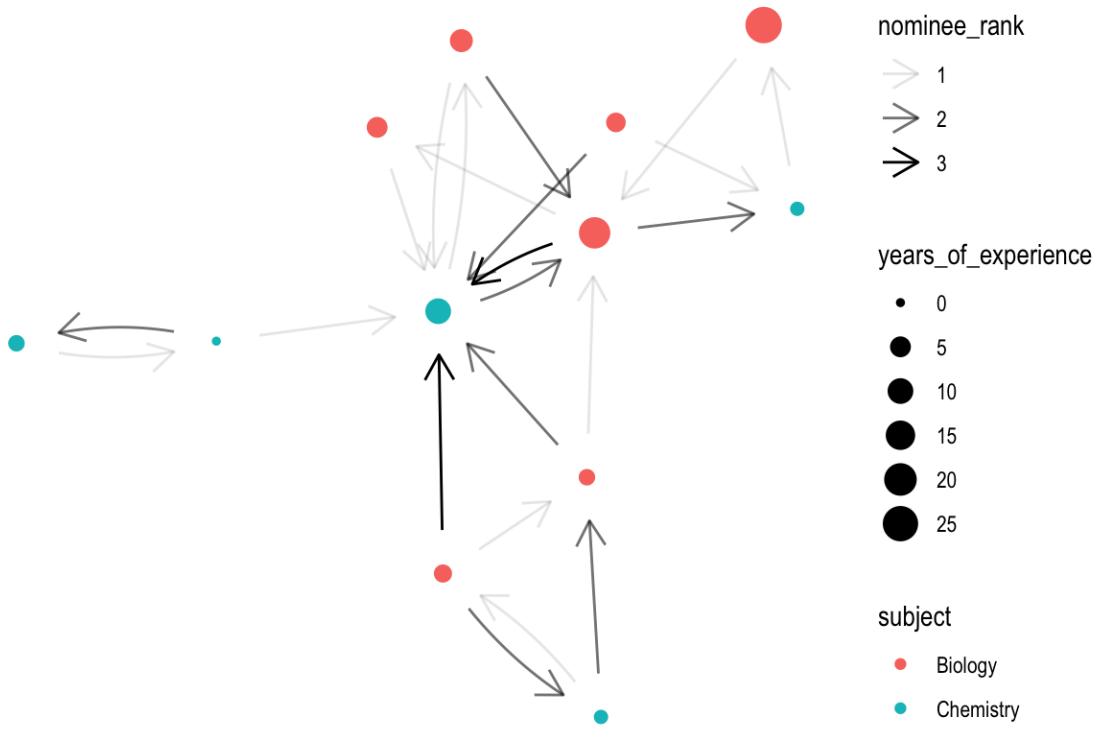
- Sizing the points based on popularity, or in-degree centrality (how many times a person was a nominee)
- Coloring the points by subject
- Changing the hue of the edges based upon the order in which one was nominated

```
ggraph(g, layout = 'kk') +
  geom_edge_fan(aes(alpha = nominee_rank),
                arrow = arrow(length = unit(4, 'mm')),
                start_cap = circle(6, 'mm'),
                end_cap = circle(6, 'mm')) +
  geom_node_point(aes(size = popularity, color = subject)) +
  theme_graph()
```



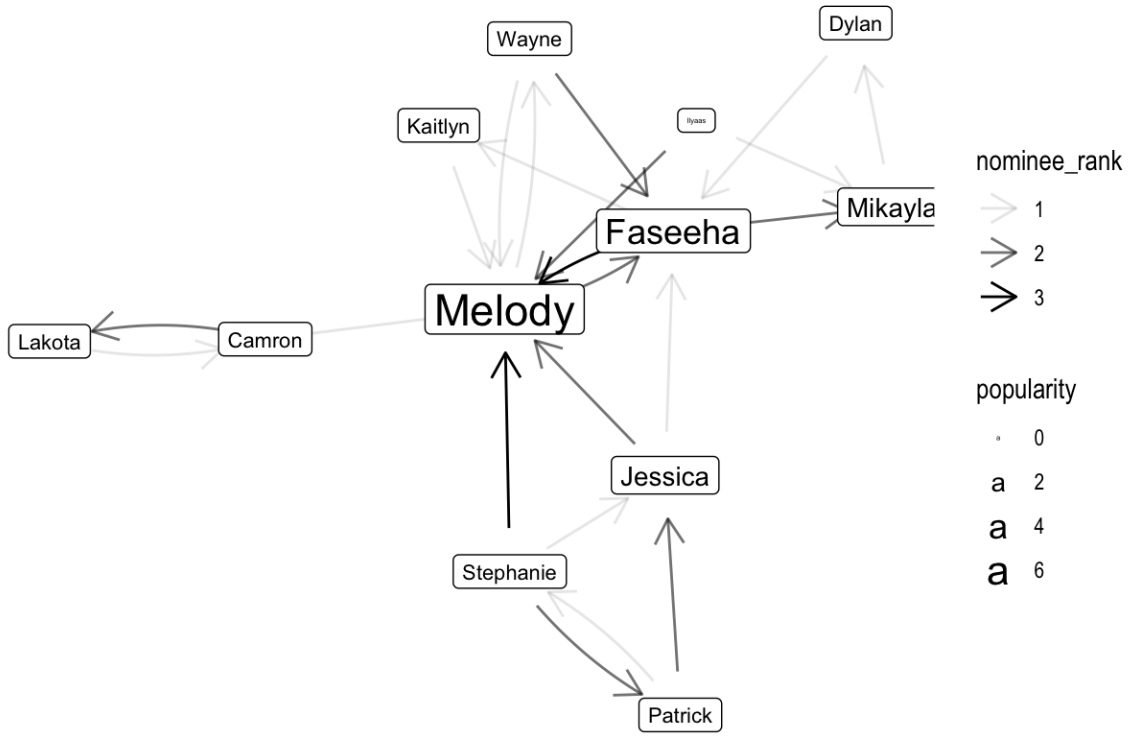
We might wish instead to size the points by years of experience, as we do below.

```
ggraph(g, layout = 'kk') +
  geom_edge_fan(aes(alpha = nominee_rank),
                arrow = arrow(length = unit(4, 'mm')),
                start_cap = circle(6, 'mm'),
                end_cap = circle(6, 'mm')) +
  geom_node_point(aes(size = years_of_experience, color = subject)) +
  theme_graph()
```



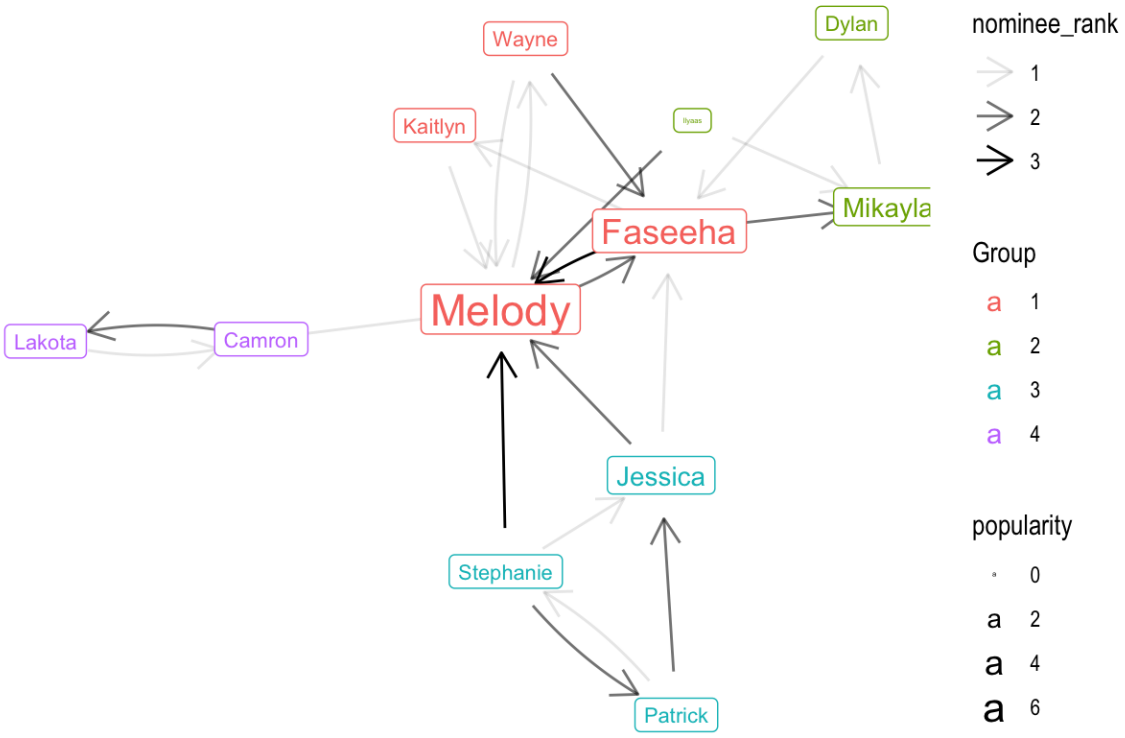
We can use names (if anonymized or otherwise ethically appropriate for our analysis), as below.

```
ggraph(g, layout = 'kk') +
  geom_edge_fan(aes(alpha = nominee_rank),
                arrow = arrow(length = unit(4, 'mm')),
                start_cap = circle(6, 'mm'),
                end_cap = circle(6, 'mm')) +
  geom_node_label(aes(label = name, size = popularity)) +
  theme_graph()
```



Lastly, we can identify sub-groups within our network and use color to indicate which individuals are a part of which sub-groups, as below.

```
g %>%
  activate(nodes) %>%
  mutate(group = group_spinglass()) %>%
  ggraph(layout = 'kk') +
  geom_edge_fan(aes(alpha = nominee_rank),
                arrow = arrow(length = unit(4, 'mm')),
                start_cap = circle(6, 'mm'),
                end_cap = circle(6, 'mm')) +
  geom_node_label(aes(label = name, size = popularity, color = as.factor(group)))
+
  theme_graph() +
  scale_color_discrete("Group", type = "qual")
```



Your Turn ⤵

Which visualization is most helpful for understanding who may be influential in the network? Why?

-
-

5. Calculating Descriptive Statistics

Finally, we can calculate a range of network statistics, as below.

```
g <- g %>%
  activate("nodes") %>%
  rename(in_degree_centrality = popularity) %>%
  mutate(out_degree = centrality_degree(mode = 'out')) %>%
  mutate(betweenness_centrality = centrality_betweenness()) %>%
  mutate(centrality_eigen = centrality_eigen())

g %>%
  as_tibble() %>%
  skimr::skim()
```

Data summary

Name	Piped data
Number of rows	12

Number of columns	7
_____	
Column type frequency:	
character	2
numeric	5
_____	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
name	0	1	5	9	0	12	0
subject	0	1	7	9	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
years_of_experience	0	1	6.67	8.14	0.00	1.75	3.50	7.75	27	
in_degree_centrality	0	1	1.83	1.90	0.00	1.00	1.00	2.00	7	
out_degree	0	1	1.83	0.72	1.00	1.00	2.00	2.00	3	
betweenness_centrality	0	1	7.00	10.44	0.00	0.00	2.50	7.75	31	
centrality_eigen	0	1	0.44	0.27	0.11	0.26	0.34	0.56	1	

We could also group our data by either years of experience or subject to begin to understand differences in centrality (and, potentially, influence), as below.

```
g %>%
  as_tibble() %>%
  group_by(subject) %>%
  select(-name) %>%
  summarize(mean_in_degree_centrality = mean(in_degree_centrality),
            sd_in_degree_centrality = sd(in_degree_centrality))
```

```
## # A tibble: 2 x 3
##   subject   mean_in_degree_centrality sd_in_degree_centrality
##   <chr>                        <dbl>                   <dbl>
## 1 Biology                       1.43                    1.27
## 2 Chemistry                     2.4                     2.61
```

```
g %>%
  as_tibble() %>%
  mutate(high_experience = if_else(years_of_experience > 5, 1, 0)) %>%
  group_by(high_experience) %>%
  summarize(mean_in_degree_centrality = mean(in_degree_centrality),
            sd_in_degree_centrality = sd(in_degree_centrality))
```

```
## # A tibble: 2 x 3
##   high_experience mean_in_degree_centrality sd_in_degree_centrality
##             <dbl>                     <dbl>                   <dbl>
## 1               0                      1.12                   0.641
## 2               1                      3.25                   2.87
```

Your Turn ⤵

Based on the descriptive statistics, what can we say is associated with an individual being more (or less) central in the network?

-
-