Project 5

Logan Bolton

2025-03-30

Acknowledgement: This code was created through the repurposing of code found in the lecture notes and through collaboration with ChatGPT-4o and Gemini 2.5 Pro. AI tools were very helpful for me while fixing errors and determining the correct syntax to plot graphs.

```
# knitr::opts_chunk$set(echo = TRUE)
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(poweRlaw)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:igraph':
##
##     as_data_frame, groups, union
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
url1 <- "https://raw.githubusercontent.com/JeffreyAlanSmith/Integrated_Network_Science/master/data/affil
url2 <- "https://raw.githubusercontent.com/JeffreyAlanSmith/Integrated_Network_Science/master/data/affil

affiliations96 <- read.delim(file = url1, check.names = FALSE)
affiliations97 <- read.delim(file = url2, check.names = FALSE)
dim(affiliations96)
```

```
## [1] 1295   91
```

1 - 1996 Dataset

a - Which student clubs serve to integrate the school and which are more peripheral?

The most core clubs are the Spanish Club, the Pep Club and NHS. The Choir barbershop quartet (4 men), the Cross Country girls 8th grade team and the boys Swim & Dive Team are the most peripheral clubs.

```
G96 <- graph_from_incidence_matrix(as.matrix(affiliations96))
```

```
## Warning: `graph_from_incidence_matrix()` was deprecated in igraph 1.6.0.
## i Please use `graph_from_biadjacency_matrix()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
# Split nodes into types by rows versus columns
# V(G96)$type # TRUE = club, FALSE = student,
clubs <- V(G96)[V(G96)$type == TRUE]

club_degrees <- degree(G96, v = clubs)
# Identify top & bottom
cat("top clubs\n")
```

```
## top clubs
```

```
head(sort(club_degrees, decreasing = TRUE), 3)
```

```
## Spanish Club     Pep Club          NHS
##          199          157          124
```

```
cat("Peripheral clubs\n")
```

```
## Peripheral clubs
```

```
head(sort(club_degrees, decreasing = FALSE), 3)
```

```
## Choir, barbershop quartet (4 men)          Cross Country, girls 8th
##                                 3                                 3
##            Swim & Dive Team, boys
##                                 3
```

b - Which student clubs tend to share members at high rates?

The following pairs of clubs share the most amount of members:

- Pep Club & Spanish Club
- Debate club & Forensics (National Forensics League)
- Forensics club & Forensics (National Forensics League)
  - "Forensics club" and "Forensics (National Forensics League)" are listed as separate clubs in the dataset, despite having similar names

```
projections <- bipartite_projection(G96)
club_graph <- projections$proj2  # This is the club-club graph

# Weight = number of shared members
E(club_graph)$weight
```

```
##    [1]  9  2  6  6  4  4  1  2  5  4  5  1  1  5  2  1  2  4  2 10  4  4  2  1
```

```
##   [25]  2  2  1  3  3  1  2  4  4  5  2  3  2  2  2  5  1  2  1  3  1  3  1  8
##   [49]  5  3  2  1  2  7  8  2  1  2  3  1  1  1  1  1  2  1  5  1  1  1  2  1
##   [73]  2  1  1  4  1  2  1  2  1  1  2  1  1  1  1  1  2  1  2  2  1  2  1  1
##   [97]  1  3  2  1  1  1  1  1  1  1  1 11 13  6  1  7  4 10  4  2  1  1  5  4
##  [121]  1  4  2  3  4  3  1  1  2  2  1  1  2  1  1  5  1  1  1  1  1  1  1  2
##  [145]  1  1  1  1  1  1  3  2  1  2  2  3  1  1  1  4  1  1  2  1  1  2  2  1
##  [169]  1  1  1  1  1  1  3  2  6  1  3  2  4  2  1  1  1  1  1  1  1  2  1  2
##  [193]  4  1  2  2  1  1  1  1  1  1  1  1  1  1  3 10  5  8  1  1  1  2  2  4
##  [217]  3  1  1  4  3  3  1  1  1  1  2  1  2  2  4  4  1  1  2  1  1  1  1  4
##  [241]  2  1  1  1  1  1  1  2  2  1  1  1  1  2  1  2  1  1  1  1  1  9  9 10
##  [265]  3  3  1  1  1  1  1  2  1  1  1  5  3  3  4  5  4  1  1  2  1  2  2  1
##  [289]  1  2  1  1  1  2  2  4  2  2  2  3  1  2  6  3  3  4  3  1  1  1  1  1
##  [313]  2  2  2  1  1  1  3  2  1  5  4  5  2  5  6  4  2  1  3  1  1  3  2  2
##  [337]  1  1  1  1  2  1  1  1  1  1  8  1  2  1  2  1  2  1  1  8  3  5  2  1
##  [361]  3  1  1  1  1  4  3  6 10  1  3  1  2  2  1  2  1  1  1  2  5  2  1  1
##  [385]  1  2  1  1  1  1  1  1  1  2 10  1  1  1  1  1  3  1  1  1  1  1  2  1
##  [409]  1  1  2  1  1  1  2  2  1  1  2  2  1  2  1  1  1  1  2  2  1  1  1  1
##  [433] 19  7 10  1  5  3 19  3  5  4 16  4  8  5 12  3  4  2  3  7  7  2  2  5
##  [457]  4  5  3  1  1  1  4  3  2  8  3  1  1  2  3  3  1  1  1  3  2  1  2  1
##  [481]  1  2  1  3  3  9  6  3  3  4  3  3  4  2  1  3  2  1  1  1  1  1  1  1
##  [505]  1  1  1  1  1 11 21 18  3  5  2  3  3  1  3  1  3  4  1  4  3  2  1  3
##  [529]  1  2  2  3  2  3  5  3  1  1  1  2  2  2  1  2  1  2  1  4  6 19  2  3
##  [553]  2  1  2  5  5  1  1  1  4  2  2  2  1  1  1  1  1  1 16  8 11  3  9  5
##  [577] 18  3  2  3  4  8  7  7  6  2  2  2  2  2  1  1  1  1  3  2  1  1  2  1
##  [601]  1  2  2  3  3  2  2  2  7  2  1  2  3  3  3  4  1  1  3  1  1  1  2  2
##  [625]  2  2  1  1  1  1  4  4  2  1  7  1  1  1  1  4  3  1  1  1  1  2  7  2
##  [649]  1  1  1  2  1  2  1  2  1  1  3  1  4  1  5  3  3  1  3  3  2  1  1  2
##  [673]  1  1  1  1  1  1  1 46  3 15 26  4  1 10 10  4  5  3  5  5  4  4  2  3
##  [697]  3  5  9  1  1  1  1  1  1  2  1  1  2  1  2  1  2  2  3 19  1  2  2  2
##  [721]  6  1  1  1  1  1  4  1  2  7  3  2  2  1  1 32  8  3 11  3  1 15 23 14
##  [745] 14  9 31  8  6  2  3  4 11  2  5  1  3  7  9  1  4  2  3  2  6  1  7  3
##  [769]  1  1  3  1  3  2  2  1  1  2  2  1  2  2  1  1 12  2  6  3  2  5  8  8
##  [793]  6  1  2  1  1  4  3  1  1  2  1  1  1 11  4  7  3  2  3  2  2  2  3  2
##  [817]  3  4  1  2  1  1  1  1 42  4 11  1 13 14  6  9  5  5 10  1  1  7  4  3
##  [841]  1  2  9  4  6  7  1  1  1  1  1  1  1  1  1  1  3  2  5 10 16 13 17  8
##  [865]  8  9  6  5  1 15 10  6  5  5  1  2  6  1  3  2  1  1  2  1  1  1  2  1
##  [889]  1  1  1  2  1  2  2  1  1 12 15  6 30 23 11  4 12  1  2  1  5  4  5  1
##  [913]  4  2  1  1  7  1  1  3  2  2  1  1  1  3  1  1 14  7  1  6 15  3  3  2
##  [937]  2  1  2  3  1  1  1  1  1  2  1  1  1 21  8  2  3  4  5  2  1  3  1  2
##  [961]  3  1  2  1  6  3  1  1  1  1  1 15  2  6  4  1  1  2  3  3  2  1  1  2
##  [985]  2  1  2  1  1  5  2  7  2  7  9  3  3  3  2  6  5  2  2  3  4  2  4  2
## [1009]  2  2  1  1  5  1  2  2  2  1  1  1  5  1  2  2  2  1  1  1  1  1  1  1
## [1033]  3  1  3  1  1  1  1  1  2  9  2  1  3  1  6  1  9  4  3  2  1  1  1  1
## [1057]  1 17  9  5  3  6  2  2  9  2  2  1  2  1  4  2  6  3  1  4  1  1  1  1
## [1081]  1 14  1 30  3  8 16 11  2  5  3  3 21 11  6  8  6  2  3  2  1  2  1  2
## [1105]  1  1 39  4 10  6  4  3  4  2 36  5  6  3  4  2  2  6  4  2  1  2  5  2
## [1129]  4  2  2  1  5  1  1  3  1  1  1 29 25 20  9 10 12  8 13  3 10  2 20  1
## [1153]  2  3  1  1  3  4  4  5  1  1  2  5  9  2  1  2  1  1  1  1 11  2  2  7
## [1177]  2  1  3  1  1  1  1 14 24  2  4 10  3  3  5  1  2  5  8  1  1  1  1  1
## [1201]  4  1  1  1  1 17  4  4  4  3  1  2  1  1  2  4  2  1  2  1  1  1  2  1
## [1225]  6 47  2  3 13  7  6  6  5  9  3  2  4 13  1  3  1  7  2  1  2  1  1  1
## [1249]  2  1  1  2  2  1  1  2  2  1  4  1  3  1  2  1  3  1  2  1  2  1  1  5
## [1273]  2  1  2  1  1  1  1  2  2  2  2  1  4  1  1  3  2  6  2  3  8  4  1  4
## [1297]  8  4  1  2  1  4  1 27  8  3  3  5  1  4  1  1  2  7  2  3  2  1  3  1
```

```
## [1321]  1  3  1  1  4  1  1  1  1  1  2  1  1  2  1  1  1  1  1  1  1 12  2  6
## [1345]  2  1  1  2  1  4  4  2  1  7  1  1  1  1  1  1  1
```

```
# Top pairs with highest shared members
top_pairs <- get.data.frame(club_graph, what = "edges")
```

```
## Warning: `get.data.frame()` was deprecated in igraph 2.0.0.
## i Please use `as_data_frame()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
top_pairs <- top_pairs[order(-top_pairs$weight), ]
head(top_pairs, 3)
```

```
##           from                                      to weight
## 1226  Pep Club                          Spanish Club     47
## 680     Debate Forensics (National Forensics League)     46
## 825  Forensics Forensics (National Forensics League)     42
```

c - What is the shared feature, or theme, that brings these clubs together in a cluster?

When analyzing the clusters based off the walktrap algorithm, the majority of the clubs in the clusters share the following common traits:

- Cluster 1
  - Girls 8th grade sports teams
- Cluster 2
  - High school boys sports teams
- Cluster 3
  - Boys 8th grade sports teams
- Cluster 4
  - Majority female academic clubs and girls sports
- Cluster 5
  - Academic organizations and girls sports
- Cluster 6
  - 9th grade girls sports
- Cluster 7
  - Music clubs

I tried several different clustering methods and the walktrap algorithm gave the most coherent clusters.

```
# club_comm <- cluster_louvain(club_graph)
# club_comm <- cluster_edge_betweenness(club_graph)
club_comm <- cluster_walktrap(club_graph)

club_membership <- membership(club_comm)
club_names <- names(club_membership)

# Combine club names and cluster numbers
club_clusters <- data.frame(
  Club = club_names,
  Cluster = club_membership
)

# View the clubs in each cluster
```

```r
split(club_clusters$Club, club_clusters$Cluster)
```

```
## $`1`
## [1] "Basketball, girls 8th"    "Cross Country, girls 8th"
## [3] "Track, girls 8th"         "Volleyball, 8th"
##
## $`2`
##  [1] "Band, Jazz"                  "Band, Marching (Symphonic)"
##  [3] "Baseball, JV (10th)"         "Baseball, V"
##  [5] "Basketball, boys 9th"        "Basketball, boys JV"
##  [7] "Basketball, boys V"          "Cross Country, boys V"
##  [9] "Football, 9th"               "Football, V"
## [11] "Golf, boys V"                "Hispanic Club"
## [13] "Orchestra, Symphonic"        "Soccer, V"
## [15] "Swim & Dive Team, boys"      "Tennis, boys V"
## [17] "Track, boys V"               "Wrestling, V"
##
## $`3`
## [1] "Band, 8th"                "Basketball, boys 8th"
## [3] "Cross Country, boys 8th" "Football, 8th"
## [5] "Track, boys 8th"          "Wrestling, 8th"
##
## $`4`
##  [1] "Academic decathalon"
##  [2] "Art Club"
##  [3] "Cheerleaders, JV"
##  [4] "Cheerleaders, Spirit Squad"
##  [5] "Cheerleaders, V"
##  [6] "Choir, women's ensemble"
##  [7] "Debate"
##  [8] "Drill Team"
##  [9] "Drunk Driving"
## [10] "Forensics"
## [11] "Forensics (National Forensics League)"
## [12] "German Club"
## [13] "Key Club"
## [14] "Latin Club"
## [15] "PEER"
## [16] "Pep Club"
## [17] "Pep Club Officers"
## [18] "Softball, JV (10th)"
## [19] "Softball, V"
## [20] "STUCO"
## [21] "Teachers of Tomorrow"
## [22] "Tennis girls V"
## [23] "Track, girls V"
## [24] "Volleyball, JV"
## [25] "Yearbook Contributors"
##
## $`5`
##  [1] "Asian Club"                        "Basketball, girls JV"
##  [3] "Basketball, girls V"               "Chess Club"
##  [5] "Choir, a capella"                  "Choir, barbershop quartet (4 men)"
##  [7] "Choir, chamber singers"            "Choir, vocal ensemble (4 women)"
```

```
##  [9] "Close-up"                          "Cross Country, girls V"
## [11] "Drunk Driving Officers"            "French Club (high)"
## [13] "French NHS"                        "Full IB Diploma Students (12th)"
## [15] "German NHS"                        "Internships"
## [17] "Junior Class Board"                "Newspaper Staff"
## [19] "NHS"                               "Quiz-Bowl (all)"
## [21] "Science Olympiad"                  "Spanish Club (high)"
## [23] "Spanish NHS"                       "Theatre Productions"
## [25] "Thespian Society (ITS)"            "Volleyball, V"
## [27] "Yearbook Editors"
##
## $`6`
## [1] "Basketball, girls 9th"   "Cheerleaders, 9th"
## [3] "Choir, concert"          "French Club (low)"
## [5] "Orchestra, Full Concert" "Spanish Club"
## [7] "Swim & Dive Team, girls" "Volleyball, 9th"
##
## $`7`
## [1] "Cheerleaders, 8th" "Choir, treble"     "Orchestra, 8th"
```

2 - 1997 Dataset

a - What is the order, size, and density of G?

```
G <- graph_from_incidence_matrix(as.matrix(affiliations97))

order_G <- gorder(G)
size_G <- gsize(G)
density_G <- edge_density(G)

cat("Order (number of vertices):", order_G, "\n")
```

```
## Order (number of vertices): 1386
```

```
cat("Size (number of edges):", size_G, "\n")
```

```
## Size (number of edges): 2641
```

```
cat("Density:", density_G, "\n")
```

```
## Density: 0.002751601
```

b - Is the network G connected? If not, what fraction of vertices belong to the largest connected component? If the network is not connected, consider only the largest component H for the remaining questions.

```
is_connected(G)
```

```
## [1] FALSE
```

```
cat("The graph is NOT connected\n\n")
```

```
## The graph is NOT connected
```

```
comp <- components(G)
```

```r
# Size of largest component
largest_comp_size <- max(comp$csize)
fraction_largest <- largest_comp_size / gorder(G)

# Create subgraph H
H <- induced_subgraph(G, which(comp$membership == which.max(comp$csize)))

cat("Percent of vertices belonging to largest component: ", fraction_largest, "\n")
```

```
## Percent of vertices belonging to largest component:  0.6789322
```

c - What is the average path length of H?

```r
average_path_length_H <- average.path.length(H, directed = FALSE)
```

```
## Warning: `average.path.length()` was deprecated in igraph 2.0.0.
## i Please use `mean_distance()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
cat("Average path length: ", average_path_length_H)
```

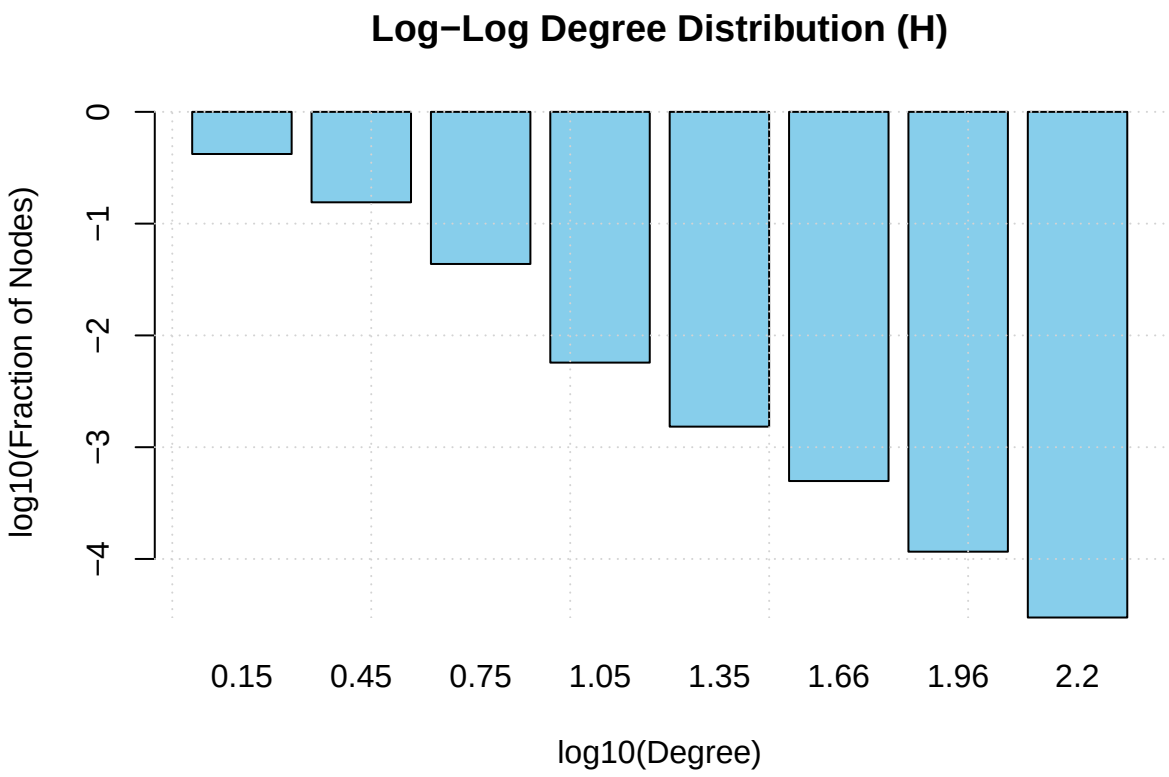
```
## Average path length:  3.738472
```

d - Is H scale-free? Provide statistical evidence (e.g., by examining the degree distribution and fitting a power-law distribution)

Yes, the subgraph H appears to be scale-free because the degree distribution follows a power-law for degrees >= 4. The scaling exponent is 2.46, and the high bootstrap p-value of 0.53 suggests a good fit to the power-law model.

```r
degrees_H <- degree(H)

## Logarithmic binning and log-log plot
log_bins <- function(degrees, alpha = 2) {
  min_deg <- min(degrees[degrees > 0])
  max_deg <- max(degrees)
  bin_edges <- c()
  current_edge <- min_deg
  while (current_edge < max_deg) {
    bin_edges <- c(bin_edges, current_edge)
    current_edge <- current_edge * alpha
  }
  bin_edges <- c(bin_edges, max_deg)
  return(bin_edges)
}

# Generate log bins and bin the degrees
bin_edges <- log_bins(degrees_H, alpha = 2)
binned_degrees <- cut(degrees_H, breaks = bin_edges, include.lowest = TRUE)
binned_counts <- table(binned_degrees)
bin_widths <- diff(bin_edges)
normalized_counts <- as.numeric(binned_counts) / bin_widths
bin_centers <- sqrt(bin_edges[-length(bin_edges)] * bin_edges[-1])
```

```r
# Log-log plot
log_bin_centers <- log10(bin_centers)
log_normalized_counts <- log10(normalized_counts / length(degrees_H))

# Plot
barplot(log_normalized_counts,
        names.arg = round(log_bin_centers, 2),
        xlab = "log10(Degree)",
        ylab = "log10(Fraction of Nodes)",
        main = "Log-Log Degree Distribution (H)",
        col = "skyblue",
        border = "black")
grid()
```




```r
## Power-law fitting using poweRlaw
m_pl <- displ$new(degrees_H)
est_xmin <- estimate_xmin(m_pl)
m_pl$setXmin(est_xmin)
est_pars <- estimate_pars(m_pl)
m_pl$setPars(est_pars)

# Extract alpha (scaling exponent) and k_min
alpha <- m_pl$pars
k_min <- m_pl$xmin
cat("Estimated scaling exponent (alpha):", alpha, "\n")
```

```
## Estimated scaling exponent (alpha): 2.458653
```

```r
cat("Estimated minimum degree (k_min):", k_min, "\n")
```

```
## Estimated minimum degree (k_min): 4
```

```r
## Goodness-of-fit via bootstrapping
set.seed(123)  # for reproducibility
bs <- bootstrap(m_pl, no_of_sims = 100)
```

```
## Expected total run time for 100 sims, using 1 threads is 5.65 seconds.
```

```r
gof_obs <- bs$gof

# Extract GOF values from bootstrapped samples
gof_bootstrap <- bs$bootstraps$gof

# Calculate p-value: fraction of bootstrap GOFs > observed GOF
p_val <- mean(gof_bootstrap > gof_obs)

# Display
cat("Bootstrap p-value:", p_val, "\n")
```

```
## Bootstrap p-value: 0.56
```

```r
if (p_val > 0.1) {
  cat("The power-law distribution is a good fit (p > 0.1)\n")
} else {
  cat("The power-law distribution is NOT a good fit (p <= 0.1)\n")
}
```

```
## The power-law distribution is a good fit (p > 0.1)
```

e - What is the fraction of edges that are attached to the top 10% of high-degree vertices?

Fraction of edges attached to the top 10% high-degree vertices: 0.9613783

```r
deg_H <- degree(H)
top_10_percent_count <- ceiling(0.10 * length(deg_H))

top_vertices <- names(sort(deg_H, decreasing = TRUE))[1:top_10_percent_count]

# Count total number of edges in H
total_edges <- gsize(H)

top_edges <- unique(unlist(incident_edges(H, top_vertices)))
fraction_top_edges <- length(top_edges) / total_edges

cat("Fraction of edges attached to the top 10% high-degree vertices:", fraction_top_edges, "\n")
```

```
## Fraction of edges attached to the top 10% high-degree vertices: 0.9613783
```

f - What distributions do the following centrality measures follow:

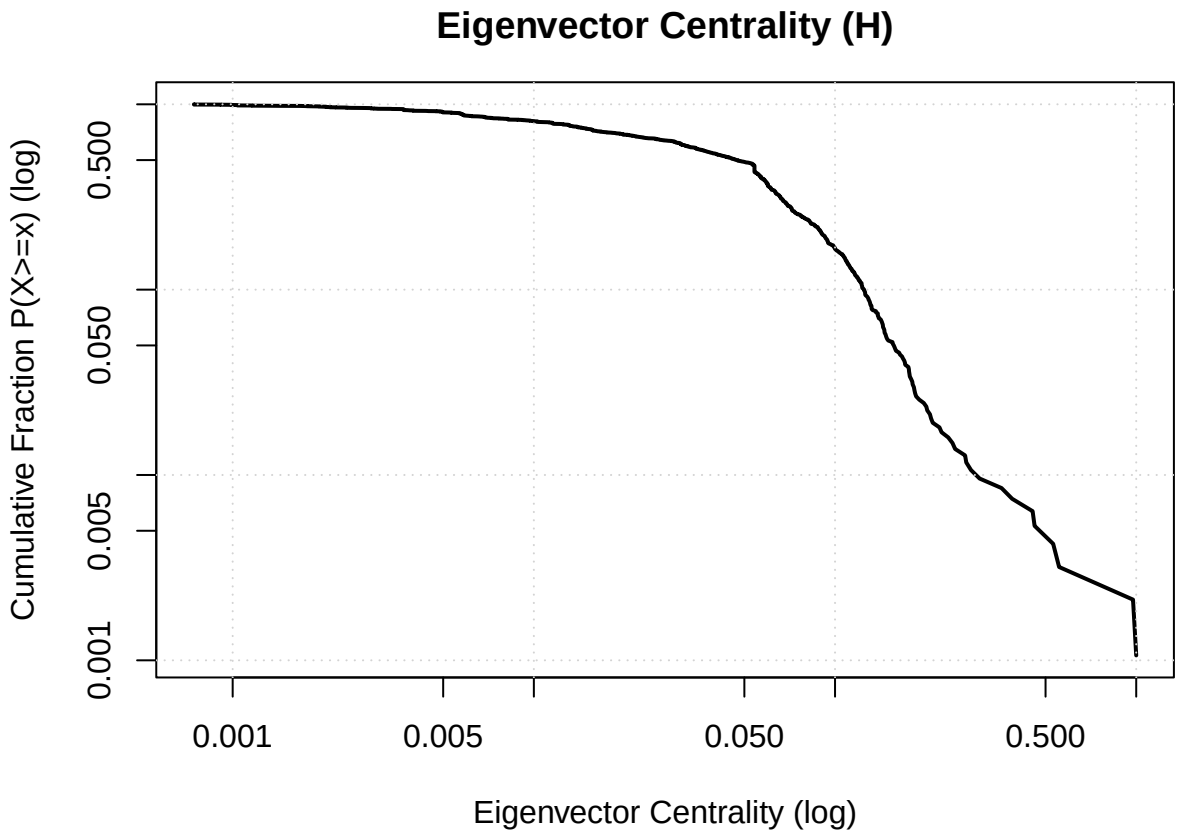
Eigenvector and Betweenness

When analyzed on a log-log scale, both measure show a similar pattern of gradually curving downwards and then decreasing at a somewhat similar rate. This type of trend is indicative of a powerlaw distribution.

Closeness

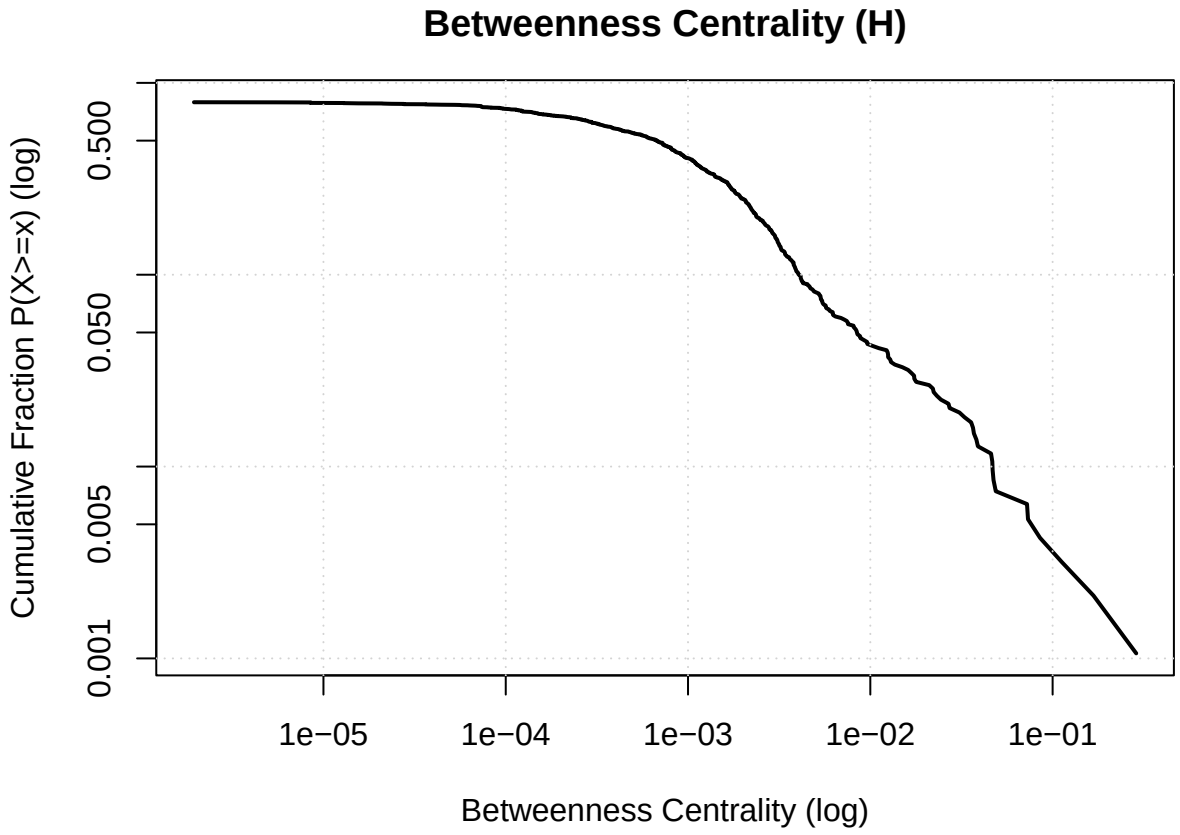
The closeness measure does not have a long tail in its distribution and it is not monotonically decreasing. Therfore, this centrality measure does not show a power-law distribution.

```
# Calculate centrality measures
eigen_H <- eigen_centrality(H)$vector
betweenness_H <- betweenness(H, normalized = TRUE)
closeness_H <- closeness(H)

### ---- 1. Eigenvector Centrality ---- ###
# Sort and compute cumulative fraction
sorted_eigen <- sort(eigen_H, decreasing = TRUE)
cumulative_eigen_frac <- (1:length(sorted_eigen)) / length(sorted_eigen)

# Open new plot
par(mar = c(4, 4, 3, 2))

plot(sorted_eigen, cumulative_eigen_frac, log = "xy", type = "l", lwd = 2,
     xlab = "Eigenvector Centrality (log)", ylab = "Cumulative Fraction P(X>=x) (log)",
     main = "Eigenvector Centrality (H)")
grid()
```

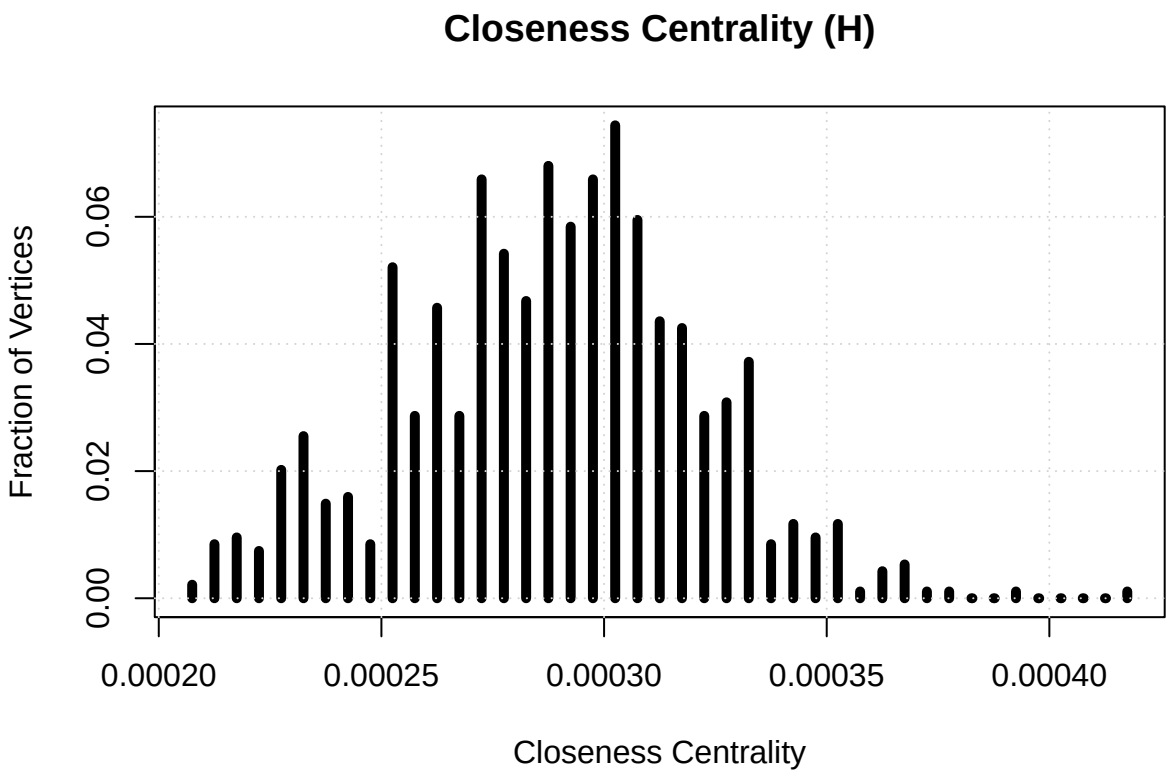



```
### ---- 2. Betweenness Centrality ---- ###
sorted_betweenness <- sort(betweenness_H, decreasing = TRUE)
valid_betweenness_indices <- which(sorted_betweenness > 0)
sorted_betweenness_nonzero <- sorted_betweenness[valid_betweenness_indices]
cumulative_betweenness_frac_nonzero <- valid_betweenness_indices / length(betweenness_H)
```

```r
par(mar = c(4, 4, 3, 2))

plot(sorted_betweenness_nonzero, cumulative_betweenness_frac_nonzero,
     log = "xy", type = "l", lwd = 2,
     xlab = "Betweenness Centrality (log)", ylab = "Cumulative Fraction P(X>=x) (log)",
     main = "Betweenness Centrality (H)")
grid()
```




```r
### ---- 3. Closeness Centrality ---- ###
hist_data_closeness <- hist(closeness_H, breaks = 30, plot = FALSE)
fraction_vertices_closeness <- hist_data_closeness$counts / length(closeness_H)

plot(hist_data_closeness$mids, fraction_vertices_closeness, type = "h", lwd = 5,
     xlab = "Closeness Centrality", ylab = "Fraction of Vertices",
     main = "Closeness Centrality (H)")
grid()
```

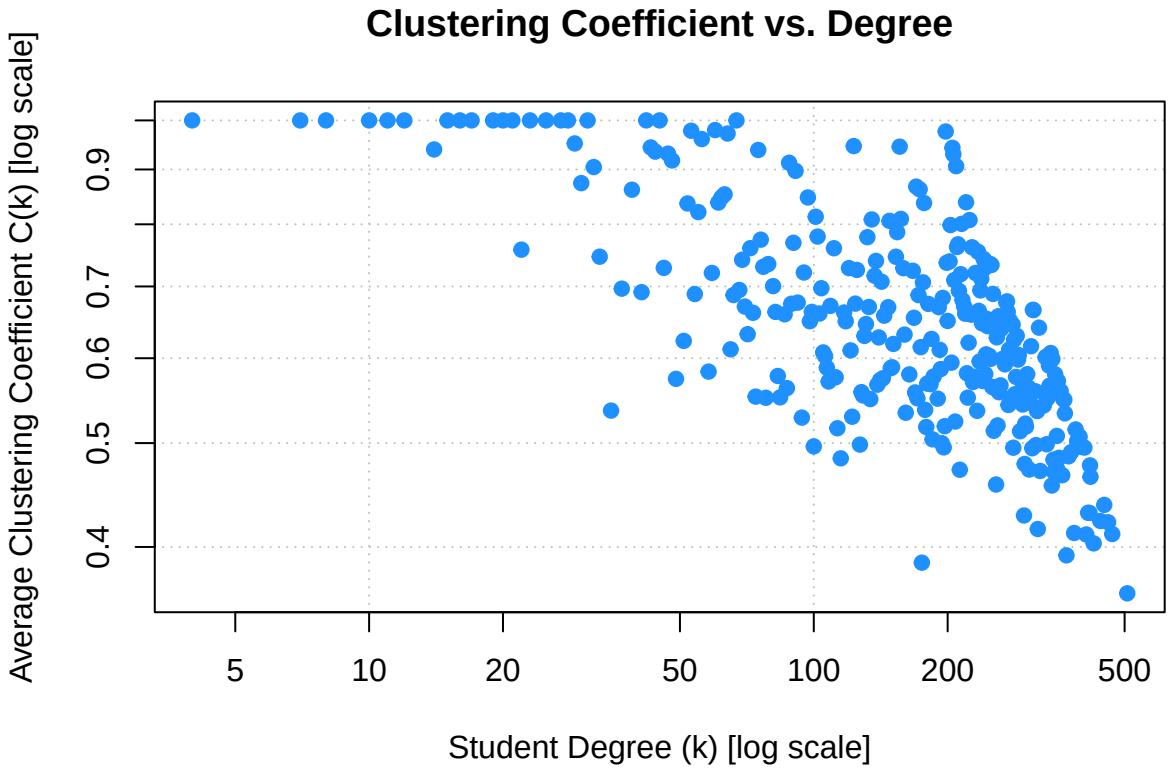


g - How does the clustering coefficient of vertices change with vertex degrees?

The vertices with low degrees have high clustering coefficients. This makes sense as some clubs are very tight knit. We would expect that every student in the barbershop quartet should know each other. Similarly for extremely large clubs, it is unlikely that all the students know each other.

```
V(H)$type <- V(G)$type[match(V(H)$name, V(G)$name)] # Assign type from original G
projections_H <- bipartite_projection(H)
student_graph_H <- projections_H$proj1 # Graph where nodes are students (type FALSE)

# Calculate degrees for vertices in the student network
deg_student_H <- degree(student_graph_H)
local_cc_student_H <- transitivity(student_graph_H, type = "local")

df_student_cc <- data.frame(
  vertex_id = V(student_graph_H)$name,
  degree = deg_student_H,
  local_cc = local_cc_student_H
)

df_student_cc_valid <- df_student_cc[!is.na(df_student_cc$local_cc), ]

avg_cc_by_degree_student <- df_student_cc_valid %>%
  group_by(degree) %>%
  summarise(
    avg_cc = mean(local_cc, na.rm = TRUE), # Should not be needed if NAs are filtered
    .groups = 'drop'
  ) %>%
  filter(degree > 0) # Ensure degree is positive
```

```r
plot_data_student_loglog <- avg_cc_by_degree_student %>%
  filter(avg_cc > 0) # Only need to filter avg_cc > 0 now

plot(plot_data_student_loglog$degree, plot_data_student_loglog$avg_cc, log = "xy",
     xlab = "Student Degree (k) [log scale]",
     ylab = "Average Clustering Coefficient C(k) [log scale]",
     main = "Clustering Coefficient vs. Degree",
     pch = 19,
     col = "dodgerblue",
     panel.first = grid(col = "gray", lty = "dotted")
     )
```



h - Does H exhibit assortative mixing in terms of vertex degrees? Provide the assortativity coefficient and interpret its value.

H does not exhibit assortative mixing. The negative coefficient (-0.3829303) indicates that nodes tend to connect to nodes with dissimilar degrees

```r
assortativity_coeff_H_student <- assortativity_degree(H, directed = FALSE)

cat("Assortativity coefficient (degree) for H:", assortativity_coeff_H_student, "\n")
```

```
## Assortativity coefficient (degree) for H: -0.3829303
```