|  |  |
| --- | --- |
|  | **DEPARTMENT OF COMPUTER ENGINEERING** |

**Experiment No. 07**

|  |  |
| --- | --- |
| Semester | B.E. Semester VII – Computer Engineering |
| Subject | Big Data Analysis |
| Subject Professor In-charge | Prof. Pankaj Vanvari |
| Lab Professor In-charge | Dr. Umesh Kulkarni |
| Academic Year | 2024-25 |

|  |  |
| --- | --- |
| Student Name | Deep Salunkhe |
| Roll Number | 21102A0014 |

**Title:** Social Network Analysis

*# Install and load the igraph package*

install.packages("igraph")

library(igraph)

*# Step 1: Create a Network (Graph) from an Edge List*

*# You can replace this with your actual data*

edge\_list <- c("Alice", "Bob",

               "Bob", "Charlie",

               "Alice", "David",

               "David", "Eve",

               "Eve", "Frank",

               "Charlie", "Frank")

*# Create the graph from the edge list*

g <- graph(edges = edge\_list, directed = FALSE)

*# Assign labels to the vertices (optional)*

V(g)$label <- V(g)$name

*# Step 2: Visualize the Network*

*# Basic plot*

plot(g, vertex.size = 30, vertex.label.cex = 0.8)

*# Improved visualization with a layout*

plot(g, layout = layout\_with\_fr, vertex.size = 30, vertex.label.cex = 0.8)

*# Step 3: Calculate Centrality Measures*

*# Degree Centrality*

degree\_centrality <- degree(g)

print("Degree Centrality:")

print(degree\_centrality)

*# Betweenness Centrality*

betweenness\_centrality <- betweenness(g, normalized = TRUE)

print("Betweenness Centrality:")

print(betweenness\_centrality)

*# Closeness Centrality*

closeness\_centrality <- closeness(g, normalized = TRUE)

print("Closeness Centrality:")

print(closeness\_centrality)

*# Step 4: Analyze Network Properties*

*# Network Density*

network\_density <- edge\_density(g)

print("Network Density:")

print(network\_density)

*# Network Diameter*

network\_diameter <- diameter(g)

print("Network Diameter:")

print(network\_diameter)

*# Clustering Coefficient*

avg\_clustering\_coef <- transitivity(g, type = "average")

print("Average Clustering Coefficient:")

print(avg\_clustering\_coef)

*# Step 5: Detect Communities in the Network*

*# Apply the edge betweenness community detection algorithm*

communities <- cluster\_edge\_betweenness(g)

*# Print community membership for each node*

membership <- membership(communities)

print("Community Membership:")

print(membership)

*# Plot the communities with different colors*

plot(communities, g, vertex.size = 30, vertex.label.cex = 0.8)

*# Step 6: Additional Analysis*

*# Shortest Paths*

shortest\_path <- shortest\_paths(g, from = "Alice", to = "Frank")

print("Shortest Path from Alice to Frank:")

print(shortest\_path$vpath)

*# Assortativity (Degree Assortativity)*

assortativity\_degree <- assortativity\_degree(g)

print("Degree Assortativity:")

print(assortativity\_degree)

*# Step 7: Save and Export Results*

*# Create a data frame of centrality measures and community membership*

centrality\_measures <- data.frame(

  Node = V(g)$name,

  Degree = degree\_centrality,

  Betweenness = betweenness\_centrality,

  Closeness = closeness\_centrality,

  Community = membership

)

*# Save the data frame to a CSV file*

write.csv(centrality\_measures, "centrality\_measures.csv", row.names = FALSE)

**Output:**

**R version 4.4.1 (2024-06-14) -- "Race for Your Life"**

**Copyright (C) 2024 The R Foundation for Statistical Computing**

**Platform: x86\_64-pc-linux-gnu**

**R is free software and comes with ABSOLUTELY NO WARRANTY.**

**You are welcome to redistribute it under certain conditions.**

**Type 'license()' or 'licence()' for distribution details.**

**R is a collaborative project with many contributors.**

**Type 'contributors()' for more information and**

**'citation()' on how to cite R or R packages in publications.**

**Type 'demo()' for some demos, 'help()' for on-line help, or**

**'help.start()' for an HTML browser interface to help.**

**Type 'q()' to quit R.**

**> # Install and load the igraph package**

**> install.packages("igraph")**

**Installing package into ‘/cloud/lib/x86\_64-pc-linux-gnu-library/4.4’**

**(as ‘lib’ is unspecified)**

**also installing the dependencies ‘glue’, ‘cli’, ‘lifecycle’, ‘magrittr’, ‘pkgconfig’, ‘rlang’, ‘vctrs’, ‘cpp11’**

**trying URL 'http://rspm/default/\_\_linux\_\_/focal/latest/src/contrib/glue\_1.8.0.tar.gz'**

**Content type 'application/x-gzip' length 164905 bytes (161 KB)**

**==================================================**

**downloaded 161 KB**

**trying URL 'http://rspm/default/\_\_linux\_\_/focal/latest/src/contrib/cli\_3.6.3.tar.gz'**

**Content type 'application/x-gzip' length 1267179 bytes (1.2 MB)**

**==================================================**

**downloaded 1.2 MB**

**trying URL 'http://rspm/default/\_\_linux\_\_/focal/latest/src/contrib/lifecycle\_1.0.4.tar.gz'**

**Content type 'application/x-gzip' length 124181 bytes (121 KB)**

**==================================================**

**downloaded 121 KB**

**trying URL 'http://rspm/default/\_\_linux\_\_/focal/latest/src/contrib/magrittr\_2.0.3.tar.gz'**

**Content type 'application/x-gzip' length 221550 bytes (216 KB)**

**==================================================**

**downloaded 216 KB**

**trying URL 'http://rspm/default/\_\_linux\_\_/focal/latest/src/contrib/pkgconfig\_2.0.3.tar.gz'**

**Content type 'application/x-gzip' length 17998 bytes (17 KB)**

**==================================================**

**downloaded 17 KB**

**trying URL 'http://rspm/default/\_\_linux\_\_/focal/latest/src/contrib/rlang\_1.1.4.tar.gz'**

**Content type 'application/x-gzip' length 1561854 bytes (1.5 MB)**

**==================================================**

**downloaded 1.5 MB**

**trying URL 'http://rspm/default/\_\_linux\_\_/focal/latest/src/contrib/vctrs\_0.6.5.tar.gz'**

**Content type 'application/x-gzip' length 1284679 bytes (1.2 MB)**

**==================================================**

**downloaded 1.2 MB**

**trying URL 'http://rspm/default/\_\_linux\_\_/focal/latest/src/contrib/cpp11\_0.5.0.tar.gz'**

**Content type 'application/x-gzip' length 272044 bytes (265 KB)**

**==================================================**

**downloaded 265 KB**

**trying URL 'http://rspm/default/\_\_linux\_\_/focal/latest/src/contrib/igraph\_2.0.3.tar.gz'**

**Content type 'application/x-gzip' length 5461951 bytes (5.2 MB)**

**==================================================**

**downloaded 5.2 MB**

**\* installing \*binary\* package ‘glue’ ...**

**\* DONE (glue)**

**\* installing \*binary\* package ‘cli’ ...**

**\* DONE (cli)**

**\* installing \*binary\* package ‘magrittr’ ...**

**\* DONE (magrittr)**

**\* installing \*binary\* package ‘pkgconfig’ ...**

**\* DONE (pkgconfig)**

**\* installing \*binary\* package ‘rlang’ ...**

**\* DONE (rlang)**

**\* installing \*binary\* package ‘cpp11’ ...**

**\* DONE (cpp11)**

**\* installing \*binary\* package ‘lifecycle’ ...**

**\* DONE (lifecycle)**

**\* installing \*binary\* package ‘vctrs’ ...**

**\* DONE (vctrs)**

**\* installing \*binary\* package ‘igraph’ ...**

**\* DONE (igraph)**

**The downloaded source packages are in**

**‘/tmp/Rtmp9eF4d9/downloaded\_packages’**

**> library(igraph)**

**Attaching package: ‘igraph’**

**The following objects are masked from ‘package:stats’:**

**decompose, spectrum**

**The following object is masked from ‘package:base’:**

**union**

**>**

**> # Step 1: Create a Network (Graph) from an Edge List**

**> # You can replace this with your actual data**

**> edge\_list <- c("Alice", "Bob",**

**+ "Bob", "Charlie",**

**+ "Alice", "David",**

**+ "David", "Eve",**

**+ "Eve", "Frank",**

**+ "Charlie", "Frank")**

**>**

**> # Create the graph from the edge list**

**> g <- graph(edges = edge\_list, directed = FALSE)**

**>**

**> # Assign labels to the vertices (optional)**

**> V(g)$label <- V(g)$name**

**>**

**> # Step 2: Visualize the Network**

**> # Basic plot**

**> plot(g, vertex.size = 30, vertex.label.cex = 0.8)**

**>**

**> # Improved visualization with a layout**

**> plot(g, layout = layout\_with\_fr, vertex.size = 30, vertex.label.cex = 0.8)**

**>**

**> # Step 3: Calculate Centrality Measures**

**> # Degree Centrality**

**> degree\_centrality <- degree(g)**

**> print("Degree Centrality:")**

**[1] "Degree Centrality:"**

**> print(degree\_centrality)**

**Alice Bob Charlie David Eve Frank**

**2 2 2 2 2 2**

**>**

**> # Betweenness Centrality**

**> betweenness\_centrality <- betweenness(g, normalized = TRUE)**

**> print("Betweenness Centrality:")**

**[1] "Betweenness Centrality:"**

**> print(betweenness\_centrality)**

**Alice Bob Charlie David Eve Frank**

**0.2 0.2 0.2 0.2 0.2 0.2**

**>**

**> # Closeness Centrality**

**> closeness\_centrality <- closeness(g, normalized = TRUE)**

**> print("Closeness Centrality:")**

**[1] "Closeness Centrality:"**

**> print(closeness\_centrality)**

**Alice Bob Charlie David Eve Frank**

**0.5555556 0.5555556 0.5555556 0.5555556 0.5555556 0.5555556**

**>**

**> # Step 4: Analyze Network Properties**

**> # Network Density**

**> network\_density <- edge\_density(g)**

**> print("Network Density:")**

**[1] "Network Density:"**

**> print(network\_density)**

**[1] 0.4**

**>**

**> # Network Diameter**

**> network\_diameter <- diameter(g)**

**> print("Network Diameter:")**

**[1] "Network Diameter:"**

**> print(network\_diameter)**

**[1] 3**

**>**

**> # Clustering Coefficient**

**> avg\_clustering\_coef <- transitivity(g, type = "average")**

**> print("Average Clustering Coefficient:")**

**[1] "Average Clustering Coefficient:"**

**> print(avg\_clustering\_coef)**

**[1] 0**

**>**

**> # Step 5: Detect Communities in the Network**

**> # Apply the edge betweenness community detection algorithm**

**> communities <- cluster\_edge\_betweenness(g)**

**>**

**> # Print community membership for each node**

**> membership <- membership(communities)**

**> print("Community Membership:")**

**[1] "Community Membership:"**

**> print(membership)**

**Alice Bob Charlie David Eve Frank**

**1 2 2 1 1 2**

**>**

**> # Plot the communities with different colors**

**> plot(communities, g, vertex.size = 30, vertex.label.cex = 0.8)**

**>**

**> # Step 6: Additional Analysis**

**> # Shortest Paths**

**> shortest\_path <- shortest\_paths(g, from = "Alice", to = "Frank")**

**> print("Shortest Path from Alice to Frank:")**

**[1] "Shortest Path from Alice to Frank:"**

**> print(shortest\_path$vpath)**

**[[1]]**

**+ 4/6 vertices, named, from ab0b759:**

**[1] Alice Bob Charlie Frank**

**>**

**> # Assortativity (Degree Assortativity)**

**> assortativity\_degree <- assortativity\_degree(g)**

**> print("Degree Assortativity:")**

**[1] "Degree Assortativity:"**

**> print(assortativity\_degree)**

**[1] NaN**

**>**

**> # Step 7: Save and Export Results**

**> # Create a data frame of centrality measures and community membership**

**> centrality\_measures <- data.frame(**

**+ Node = V(g)$name,**

**+ Degree = degree\_centrality,**

**+ Betweenness = betweenness\_centrality,**

**+ Closeness = closeness\_centrality,**

**+ Community = membership**

**+ )**

**>**

**> # Save the data frame to a CSV file**

**> write.csv(centrality\_measures, "centrality\_measures.csv", row.names = FALSE)**

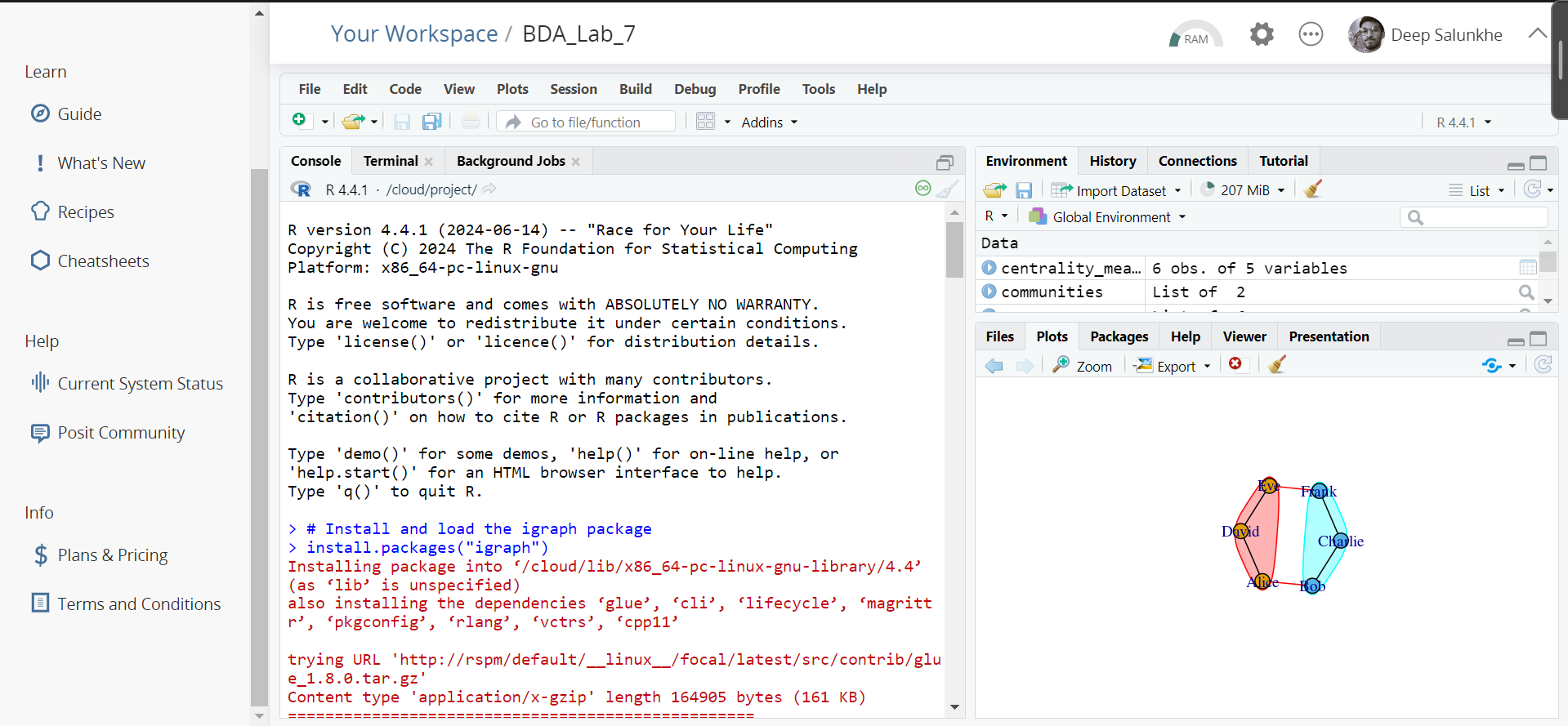
**>**

**> # Step 8: Load Network Data from a File (Optional)**

**> # If you want to load an edge list from a CSV file instead**

**> # edge\_data <- read.csv("edge\_list.csv")**

**> # g <- graph\_from\_data\_frame(edge\_data, directed = FALSE)**

****