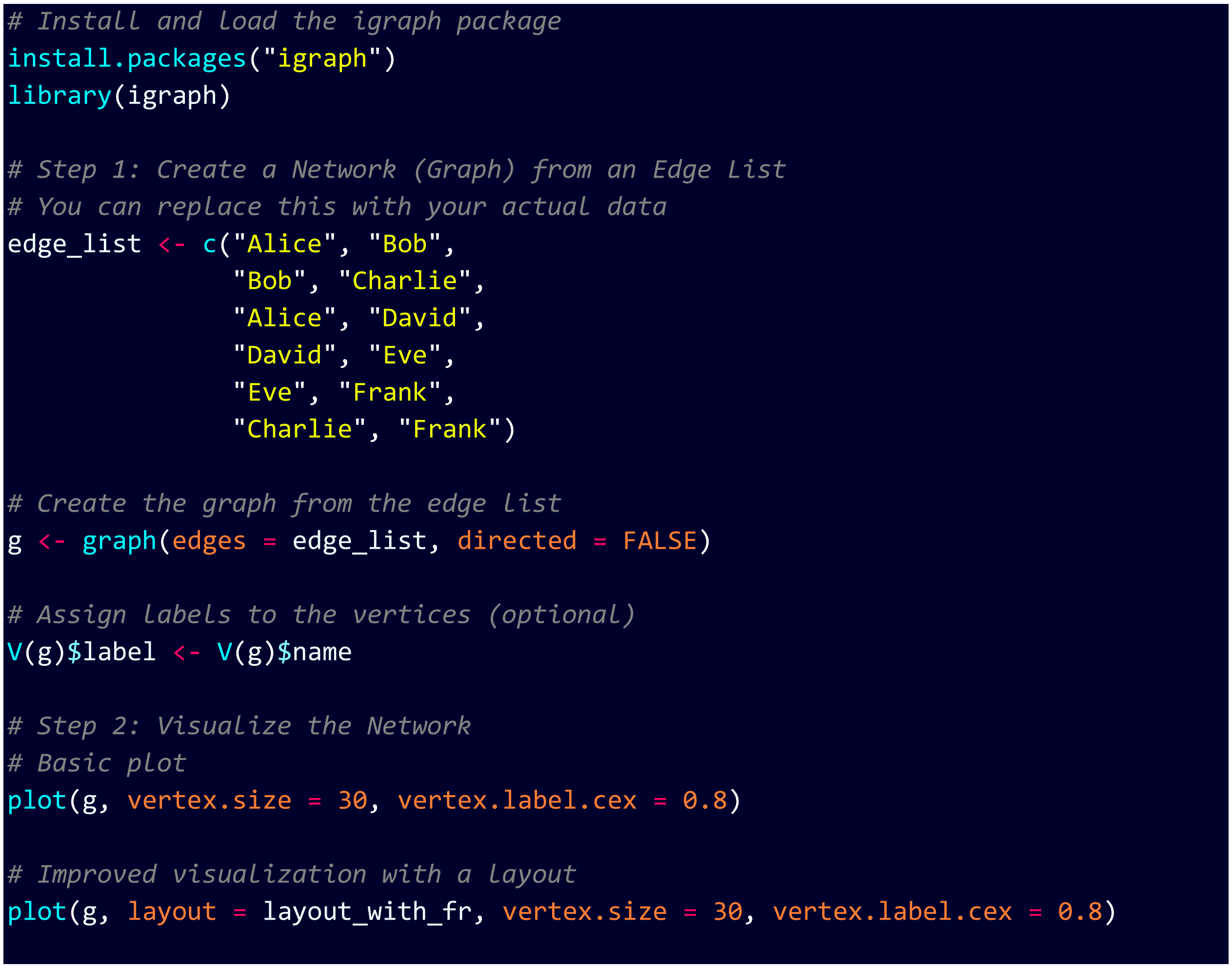
VIT Vidyalankar Institute of Technology Accredited A+ by NAAC	**DEPARTMENT OF COMPUTER ENGINEERING**

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)
```

```r
# 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)
```

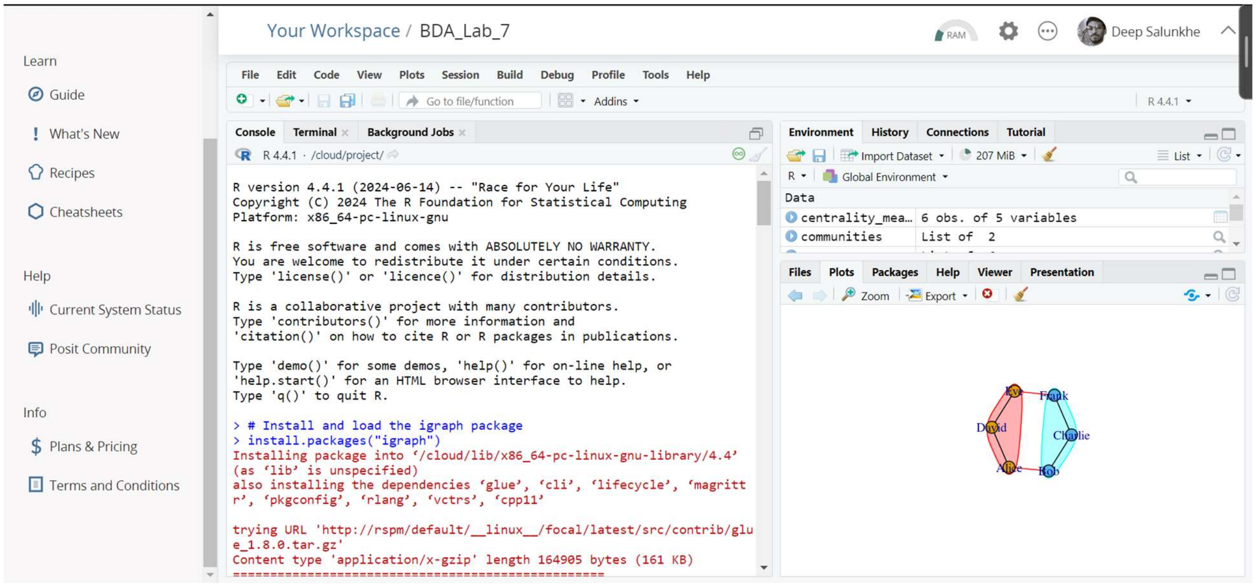
```
==================================================
downloaded 121 KB


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)
```



```
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’, ‘magritt
r’, ‘pkgconfig’, ‘rlang’, ‘vctrs’, ‘cpp11’

trying URL 'http://rspm/default/__linux__/focal/latest/src/contrib/glu
e_1.8.0.tar.gz'
Content type 'application/x-gzip' length 164905 bytes (161 KB)
==================================================
```

```
Data
centrality_mea…   6 obs. of 5 variables
communities       List of  2
```