

ANEXO

Código usado en RStudio

```
installed.packages("igraph")
library(igraph)
installed.packages("tidygraph")
library(tidygraph)
install.packages("tidyverse")
library(tidyverse)
installed.packages("ggraph")
library(ggraph)

#load data
data <- read.csv("19952.csv", header=T)
y <- data.frame(data$Palabra1, data$Palabra2, data$Grados)

# Cargar la librería igraph si no está cargada
if (!requireNamespace("igraph", quietly = TRUE)) {install.packages("igraph")}
library(igraph)

# Crear una red de ejemplo
edges <- c(1, 2, 1, 3, 2, 3) # Ejemplo de lista de aristas
net <- graph(edges)

# Establecer márgenes más pequeños
par(mar = c(3, 3, 1, 1))

# Crear el histograma del grado de los nodos
hist(degree(net),
col = 'yellow',
main = 'Histogram of Node Degree',
ylab = 'Number of nodes',
xlab = 'Degree of vertices')

# Cargar la librería igraph si no está cargada
if (!requireNamespace("igraph", quietly = TRUE)) {install.packages("igraph")}
install.packages("igraph")
library(igraph)

# Crear una red de ejemplo
edges <- c(1, 2, 1, 3, 2, 3) # Ejemplo de lista de aristas
net <- graph(edges)

# Calcular los grados de los nodos en la red
degrees <- degree(net)
```

```

# Establecer márgenes más pequeños
par(mar = c(3, 3, 1, 1))

# Crear el histograma del grado de los nodos
hist(degrees,
col = 'yellow',
main = 'Histogram of Node Degree',
ylab = 'Number of nodes',
xlab = 'Degree of vertices')
net <- graph.data.frame(y, directed = T)
V(net)
V(net)$label <- V(net)$name
V(net)$degree <- degree(net)
hist(V(net)$degree,
col = 'yellow',
main = 'Histogram of Node Degree',
ylab = 'Number of nodes',
xlab = 'Degree of vertices')

#Diagrama de red solo muestra conexiones
set.seed(222)
plot(net,
vertex.color = 'white',
vertex.size = 2,
edge.arrow.size = 0.2,
vertex.label.cex = 0.8)
is.finite(V(net)$degree)

plot(net,
vertex.color = rainbow(52),
vertex.size = V(net)$degree * 0.4,
edge.arrow.size = 0.1,
layout = layout.fruchterman.reingold)

plot(net,
vertex.color = rainbow(52),
vertex.size = V(net)$degree * 0.4, # Corregido de dgree a degree
edge.arrow.size = 0.1,
layout = layout.fruchterman.reingold)

#Cargue el paquete igraph si aún no está cargado
if (!requireNamespace("igraph", quietly = TRUE)) {
  install.packages("igraph")
}
library(igraph)

#Calcular el grado de centralidad
degree_centrality <- degree(net)

```

```

#Calcular el grado de intermediación
betweenness centrality <- betweenness(net)

#Calcular el grado de centralidad
closeness centrality <- closeness(net)

#Cargue el paquete igraph si aún no está cargado
if (!requireNamespace("igraph", quietly = TRUE)) {
  install.packages("igraph")
}
library(igraph)

#Número de nodos
num_nodes <- vcount(net)

#Número de aristas
num_edges <- ecount(net)

# Calcular comunidades utilizando multilevel.community
communities <- multilevel.community(net)

# Imprimir las comunidades
print(communities)

# Porcentaje de grado
avg_degree <- mean(degree(net))

# Diámetro
diameter <- diameter(net)

# Densidad
density <- graph.density(net)

# Radio
radius <- radius(net)

# Número de triángulos
num_triangles <- sum(triangles(net) / 3)

# Calcular comunidades utilizando edge.betweenness.community
communities <- edge.betweenness.community(net)

# Calcular la modularidad basada en las comunidades detectadas
modularity_value <- modularity(communities)

# Imprimir la modularidad
print(modularity_value)

# Centralidad de intermediación
betweenness centrality <- betweenness(net)

```

```
# Centralidad de cercanía
closeness_centrality <- closeness(net)

# Coeficiente de agrupamiento
clustering_coefficient <- transitivity(net, type = "local")

# Imprima o utilice las medidas calculadas según sea necesario
print(num_nodes)
print(num_edges)
print(communities)
print(avg_degree)
print(diameter)
print(density)
print(radius)
print(num_triangles)
print(modularity)
print(betweenness_centrality)
print(closeness_centrality)
print(clustering_coefficient)
```