STATISTICAL ANALYSIS OF NETWORK: PROJECT

CRISTINA FARRUKU

6/8/2023

Link: https://networkrepository.com/mammalia-voles-bhp-trapping.php (https://networkrepository.com/mammalia-voles-bhp-trapping.php)

Purpose of the analysis

The primary objective of this analysis is to find social interaction patterns exhibited by mammals, with a specific emphasis on identifying key individuals or groups and exploring the underlying community structure within the mammal network. By studying these social interactions, we aim to gain insights into the dynamics of social relationships among mammals, understand the structure and organization of these interactions that can shed light on communication patterns or social hierarchies.

We start by importing all the useful libraries:

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✔ ggplot2 3.3.5     ✔ purrr   0.3.4
## ✔ tibble  3.1.6     ✔ dplyr   1.0.7
## ✔ tidyr   1.1.4     ✔ stringr 1.4.0
## ✔ readr   2.1.2     ✔ forcats 0.5.1
```

```
## Warning: il pacchetto 'readr' è stato creato con R versione 4.1.2
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(igraph)
```

```
## Warning: il pacchetto 'igraph' è stato creato con R versione 4.1.2
```

```
##
## Caricamento pacchetto: 'igraph'
```

```
## I seguenti oggetti sono mascherati da 'package:dplyr':
##
##     as_data_frame, groups, union
```

```
## I seguenti oggetti sono mascherati da 'package:purrr':
##
##     compose, simplify
```

```
## Il seguente oggetto è mascherato da 'package:tidyr':
##
##     crossing
```

```
## Il seguente oggetto è mascherato da 'package:tibble':
##
##     as_data_frame
```

```
## I seguenti oggetti sono mascherati da 'package:stats':
##
##     decompose, spectrum
```

```
## Il seguente oggetto è mascherato da 'package:base':
##
##     union
```

```
library(igraphdata)
library(sand)
```

```
##
## Statistical Analysis of Network Data with R, 2nd Edition
## Type in C2 (+ENTER) to start with Chapter 2.
```

Next we import the dataset:

```
data<-readLines("/Users/cristinafarruku/Desktop/mammalia-voles-bhp-trapping.edges")

#apply this function since the dataset is a list representation of the graph and convert it into 2 vertex identif
iers
vertex_pairs <- lapply(data, function(x) {
# Split each element by whitespace and extract the first two values
  vertices <- strsplit(x, " ")[[1]][1:2]
  as.numeric(vertices)
})

#create the graph
graph <- graph_from_edgelist(do.call(rbind, vertex_pairs), directed = FALSE)

#create the data frame
df_graph <- data.frame(from = sapply(vertex_pairs, "[[", 1),
                 to = sapply(vertex_pairs, "[[", 2))
head(df_graph)
```

```
##   from to
## 1    1  2
## 2    3  4
## 3    3  5
## 4    3  6
## 5    3  7
## 6    3  8
```

Since this dataset is *weighted*, we use this function that gives us all the weights related to edges.

```
w<-df_graph %>% transmute(id1 = pmin(from, to),id2 = pmax(from,to)) %>%
  group_by(id1, id2) %>% summarize(weight = n())
```

```
## `summarise()` has grouped output by 'id1'. You can override using the `.groups`
## argument.
```

```
w
```

```
## # A tibble: 4,623 × 3
## # Groups:   id1 [1,156]
##      id1   id2 weight
##    <dbl> <dbl>  <int>
##  1     1     2      1
##  2     3     4      1
##  3     3     5      1
##  4     3     6      1
##  5     3     7      1
##  6     3     8      1
##  7     4     5      1
##  8     4     6      1
##  9     4     8      1
## 10     5     6      1
## # … with 4,613 more rows
```

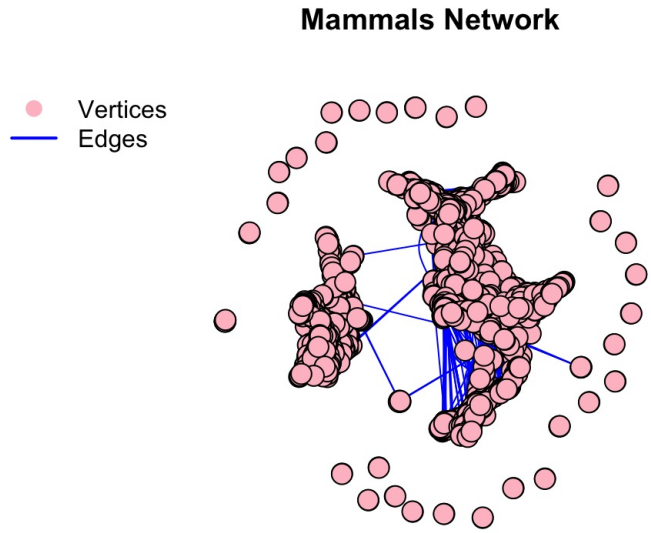
To confirm that the dataset is weighted we use this function that returns False if all the weights are not equal to 1.

```
are_all_ones = w %>%
  pull(weight) %>%
  all(.==1)
are_all_ones
```

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

NETWORK DESCRIPTION

```
# Plot the graph with customized attributes
set.seed(1907)
plot(graph,
     vertex.label = NA,
     vertex.color = "pink",
     vertex.size = 10,
     edge.color = "blue",
     main = "Mammals Network")
legend("topleft", legend = c("Vertices", "Edges"),
       col = c("pink", "blue"),
       pch = c(16, NA),
       pt.cex = 1.5,
       lwd = c(NA, 2),
       bty = "n")
```



From the plot we can notice that there are some mammals that are not connected by edges to the middle of the network, thus being isolated. This may suggest different levels of social interaction or importance within the mammal social network, but to confirm this we have to make further analysis.

Next we proceed with graph decomposition, used to partition a graph into smaller and more manageable subgraph and have a deeper understanding of the graph's structure, patterns or relationships.

```
#weak=every vertex is reachable from any other vertex regardless the edge directions.
component_list= decompose.graph(graph, mode = "weak")
component_list
```

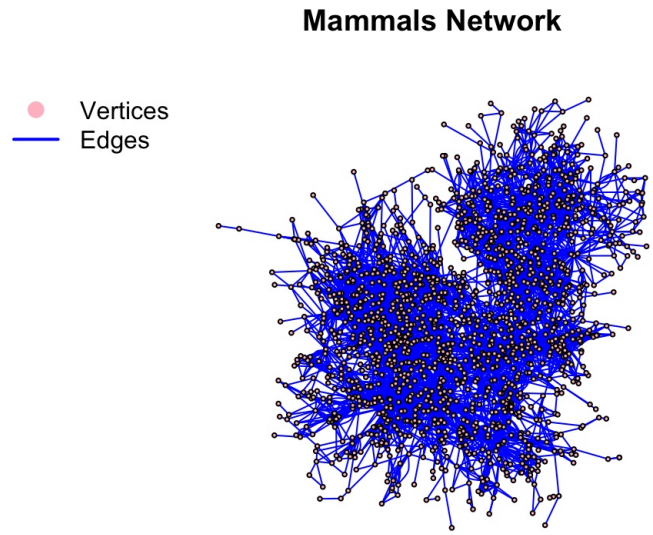
```
## [[1]]
## IGRAPH 8a6cbf5 U--- 2 1 --
## + edge from 8a6cbf5:
## [1] 1--2
##
## [[2]]
## IGRAPH 7a0065a U--- 1613 5261 --
## + edges from 7a0065a:
##  [1]  1-- 2  1-- 3  1-- 4  1-- 5  1-- 6  7-- 8  7-- 9 10--11 10--12 10--13
## [11]  9--10 10--14  9--14 15--16 15--17 15--18  2-- 4  3-- 4  4--11  4-- 6
## [21] 17--18  5--19  2-- 3  2-- 6  3-- 6  8--20 20--21 14--20 21--22 21--23
## [31] 11--14 22--23  3--24 25--26  4--10 10--27 10--11 10--14  4-- 9 15--28
## [41] 15--17 17--29 30--31 16--32 22--31  8--32  3--12 11--14 14--27 11--27
## [51] 24--33  4--24 24--34 24--35 34--36  4--34 25--37 35--36 12--36 15--17
## [61]  4--35 17--38 39--40 30--41 29--42 28--29 29--38 31--43 26--40 38--42
## [71] 37--44 37--45 11--45 45--46 28--47 28--48 18--28 12--49 11--49 12--35
## [81] 13--27 18--48 11--37 42--50 21--41 51--52 51--53 28--51 51--54 51--55
## + ... omitted several edges
##
## [[3]]
## IGRAPH 9c18a27 U--- 2 1 --
## + edge from 9c18a27:
## [1] 1--2
##
```

```
## [[4]]
## IGRAPH 1b0d146 U--- 2 1 --
## + edge from 1b0d146:
## [1] 1--2
##
## [[5]]
## IGRAPH cb675a0 U--- 2 1 --
## + edge from cb675a0:
## [1] 1--2
##
## [[6]]
## IGRAPH 0e79eaf U--- 3 2 --
## + edges from 0e79eaf:
## [1] 1--2 1--3
##
## [[7]]
## IGRAPH e2c0b86 U--- 2 1 --
## + edge from e2c0b86:
## [1] 1--2
##
## [[8]]
## IGRAPH bd6ede8 U--- 2 1 --
## + edge from bd6ede8:
## [1] 1--2
##
## [[9]]
## IGRAPH 329e7e2 U--- 3 4 --
## + edges from 329e7e2:
## [1] 1--2 1--2 1--3 2--3
##
## [[10]]
## IGRAPH 4b22c33 U--- 2 1 --
## + edge from 4b22c33:
## [1] 1--2
##
## [[11]]
## IGRAPH 55bd886 U--- 2 1 --
## + edge from 55bd886:
## [1] 1--2
##
## [[12]]
## IGRAPH 96088e7 U--- 2 1 --
## + edge from 96088e7:
## [1] 1--2
##
## [[13]]
## IGRAPH 4d7927b U--- 2 1 --
## + edge from 4d7927b:
## [1] 1--2
##
## [[14]]
## IGRAPH 912c633 U--- 2 1 --
## + edge from 912c633:
## [1] 1--2
##
## [[15]]
## IGRAPH 9b20875 U--- 2 1 --
## + edge from 9b20875:
## [1] 1--2
##
## [[16]]
## IGRAPH 88d92d9 U--- 2 1 --
## + edge from 88d92d9:
## [1] 1--2
##
## [[17]]
## IGRAPH 9f25acf U--- 3 2 --
## + edges from 9f25acf:
## [1] 1--2 1--3
##
## [[18]]
## IGRAPH be50e4a U--- 2 1 --
## + edge from be50e4a:
## [1] 1--2
##
## [[19]]
## IGRAPH 90b6655 U--- 3 2 --
## + edges from 90b6655:
## [1] 1--2 2--3
```

```
##
## [[20]]
## IGRAPH dd92191 U--- 2 1 --
## + edge from dd92191:
## [1] 1--2
##
## [[21]]
## IGRAPH 3dec32e U--- 2 2 --
## + edges from 3dec32e:
## [1] 1--2 1--2
##
## [[22]]
## IGRAPH 759ffb5 U--- 2 1 --
## + edge from 759ffb5:
## [1] 1--2
##
## [[23]]
## IGRAPH fb301a5 U--- 9 23 --
## + edges from fb301a5:
##  [1] 1--2 1--3 1--4 3--4 1--5 3--5 4--5 1--6 3--6 4--6 5--6 1--7 3--7 4--7 5--7
## [16] 6--7 1--8 3--8 4--8 5--8 6--8 7--8 2--9
##
## [[24]]
## IGRAPH 8d88e27 U--- 3 3 --
## + edges from 8d88e27:
## [1] 1--2 1--3 2--3
##
## [[25]]
## IGRAPH 2d3fce7 U--- 2 1 --
## + edge from 2d3fce7:
## [1] 1--2
##
## [[26]]
## IGRAPH de6fcdc U--- 2 1 --
## + edge from de6fcdc:
## [1] 1--2
##
## [[27]]
## IGRAPH ae6f004 U--- 2 1 --
## + edge from ae6f004:
## [1] 1--2
##
## [[28]]
## IGRAPH 3c9f18c U--- 2 1 --
## + edge from 3c9f18c:
## [1] 1--2
##
## [[29]]
## IGRAPH 0a8aea8 U--- 2 1 --
## + edge from 0a8aea8:
## [1] 1--2
##
## [[30]]
## IGRAPH 39c073b U--- 3 3 --
## + edges from 39c073b:
## [1] 1--2 1--3 2--3
##
## [[31]]
## IGRAPH 390aa13 U--- 2 1 --
## + edge from 390aa13:
## [1] 1--2
```

After the decomposition, we can notice that only the second subgraph is the graph with the most edges and nodes, indeed: - 1613 nodes - 5261 being the bigger. So we are going to analyze only this subgraph.

```
graph_decomp<- component_list[[2]]
plot(graph_decomp,
     layout = layout_with_fr(graph_decomp),
     vertex.label = NA,
     vertex.size = 2,
     edge.label = NA,
     vertex.color = "pink",
     edge.color = "blue",
     main = "Mammals Network")
legend("topleft", legend = c("Vertices", "Edges"),
       col = c("pink", "blue"),
       pch = c(16, NA),
       pt.cex = 1.5,
       lwd = c(NA, 2),
       bty = "n"
)
```



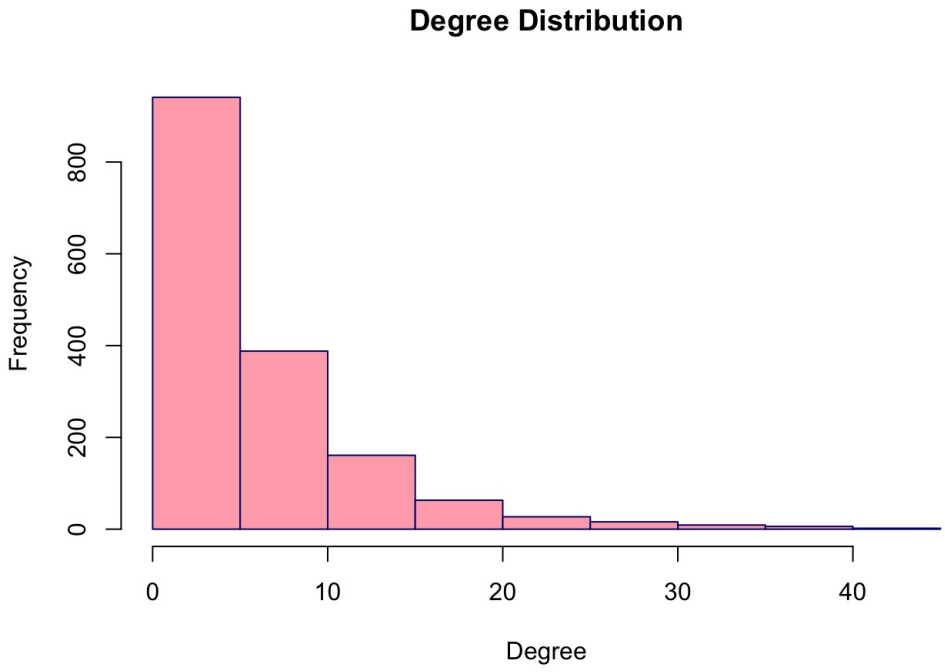

```
is_simple(graph_decomp) ###MULTI-GRAPH <- meaning that has no edges for which both ends connect to a single verte
x(LOOPS) and no pairs of vertices with more than one edge between them (multi-edges).
```

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

NETWORK ANALYSIS

Since we are dealing with social interaction network among mammals, we can adopt the “principal actor” that refers to identifying individuals or nodes within the network that play a central or influential role in the network dynamics. There are several measures we can use to determine the principal actors. First measure of importance: *DEGREE*, which is the number of edges incident to each node.

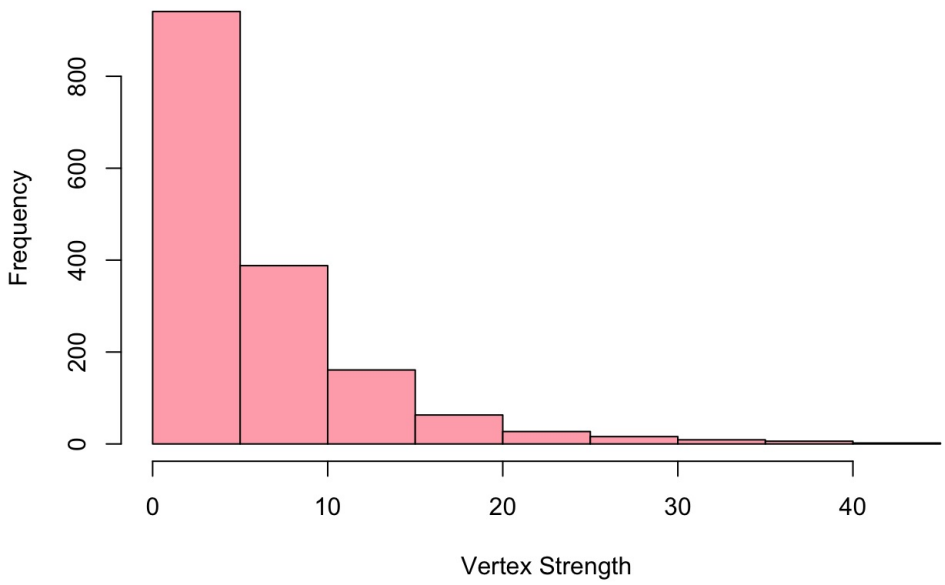
```
mam_degree<-degree(graph_decomp)
hist(mam_degree,
        main = "Degree Distribution",
        xlab = "Degree",
        ylab = "Frequency",
        col = "lightpink1",
        border = "navyblue",
)
```



From the histogram, we can notice the a few mammals have a relatively higher number of connections, while the most mammals have lower number of social connections. This could indicate a hierarchical social structure where a small number of mammals have a greater influence to social resources.

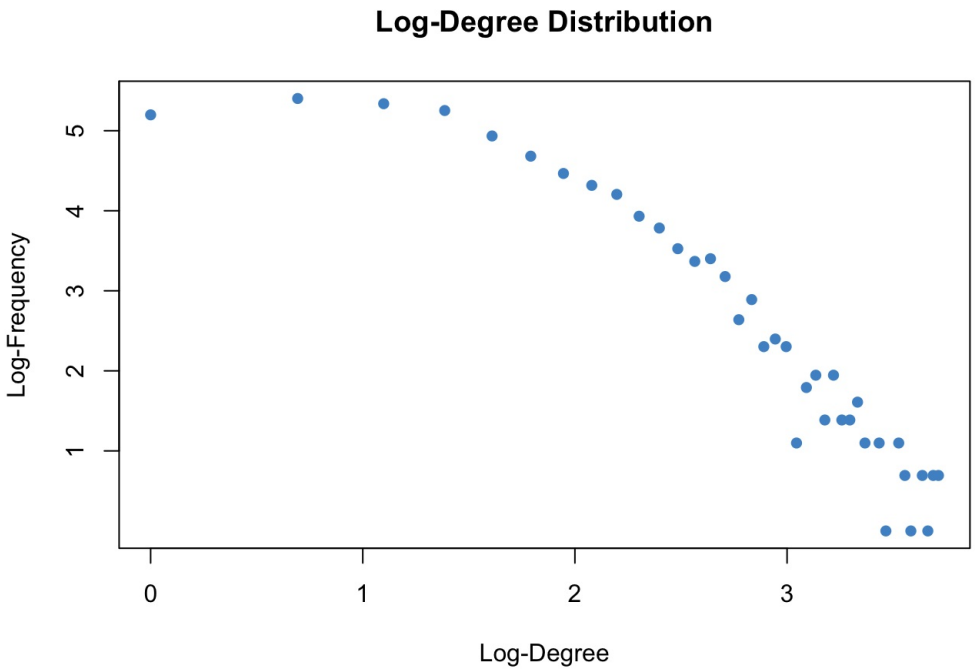
Since it is a weighted graph, a useful generalization of degree is vertex strength, obtained by summing up the weights of edges incident to a given vertex.

```
#weighted degree distribution
hist(strength(graph_decomp),
     col="lightpink1",
     xlab="Vertex Strength",
     ylab="Frequency",
     main="")
```



We can see that the distribution of degrees associated with the edges among the the vertices is quite heterogeneous. The majority of nodes have weaker interactions with other nodes in terms of weight, while there are a few nodes with high-weighted connections, indicating that they play a significative role in the network. Given the nature of the decay in the distribution, a log-scale can be more effective in summarizing the degree information.

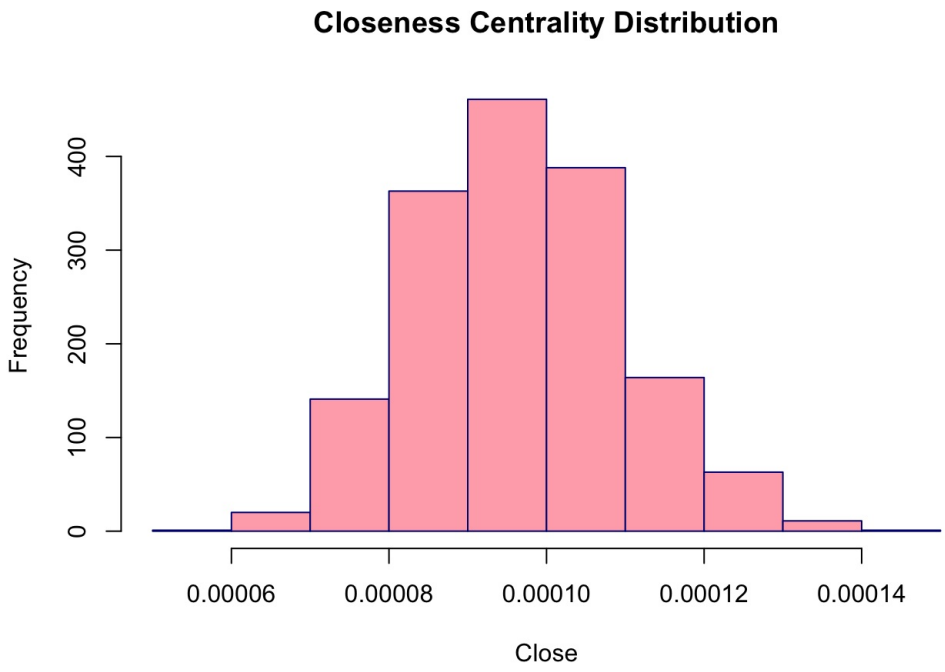
```
degree_freq<- table(mam_degree)
log_degree<-log(as.numeric(names(degree_freq)))
log_freq<-log(degree_freq)

plot(log_degree,
     log_freq,
     type="p",
     pch=16,
     main = "Log-Degree Distribution",
     xlab = "Log-Degree",
     ylab = "Log-Frequency",
     col = "steelblue3"
)
```



From the plot we can notice that the distribution is heavy-tailed indicating degree heterogeneity within the network and also suggesting the existence of social hierarchies within the mammal community.

Second measure: *CLOSENESS*, for a given node, it takes the sum of the inverses of the length of the shortest paths to every node. It provides insights into the accessibility and efficiency of information flow within the mammal community.

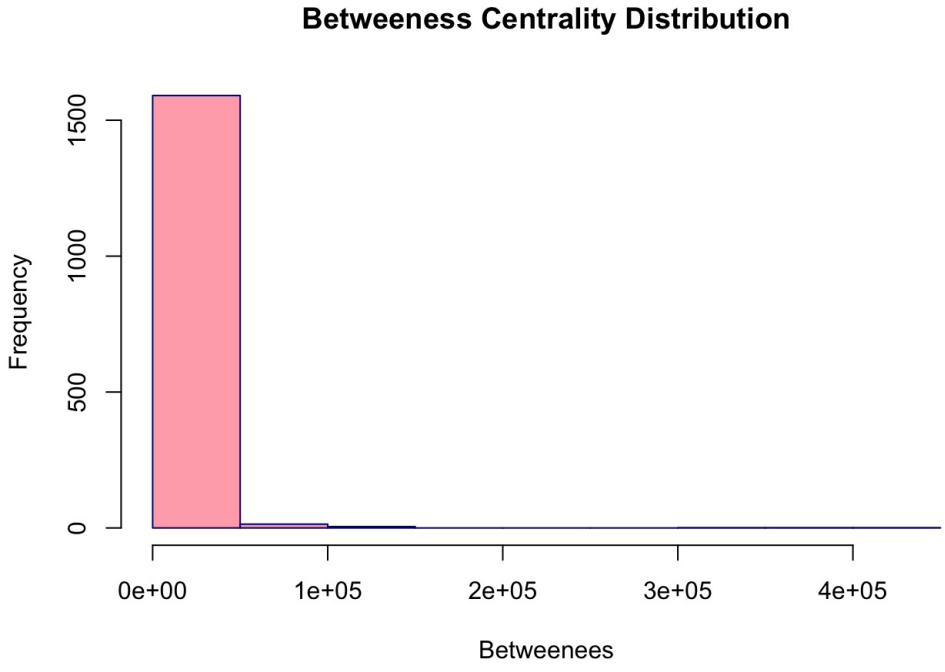
```
mam_close<- closeness(graph_decomp, mode = "total")
hist(mam_close,
        main = "Closeness Centrality Distribution",
        xlab = "Close",
        ylab = "Frequency",
        col = "lightpink1",
        border = "navyblue",
)
```



From the histogram above, we can see a relatively symmetric closeness distribution, but with very low values suggesting that most mammals might have limited access to social information.

Third measure: *betweeness* wich counts how many shortest paths each node is involved in. In particular here, we measure the mammal or groups that serve as a bridge or intermediary between other individuals in the network.

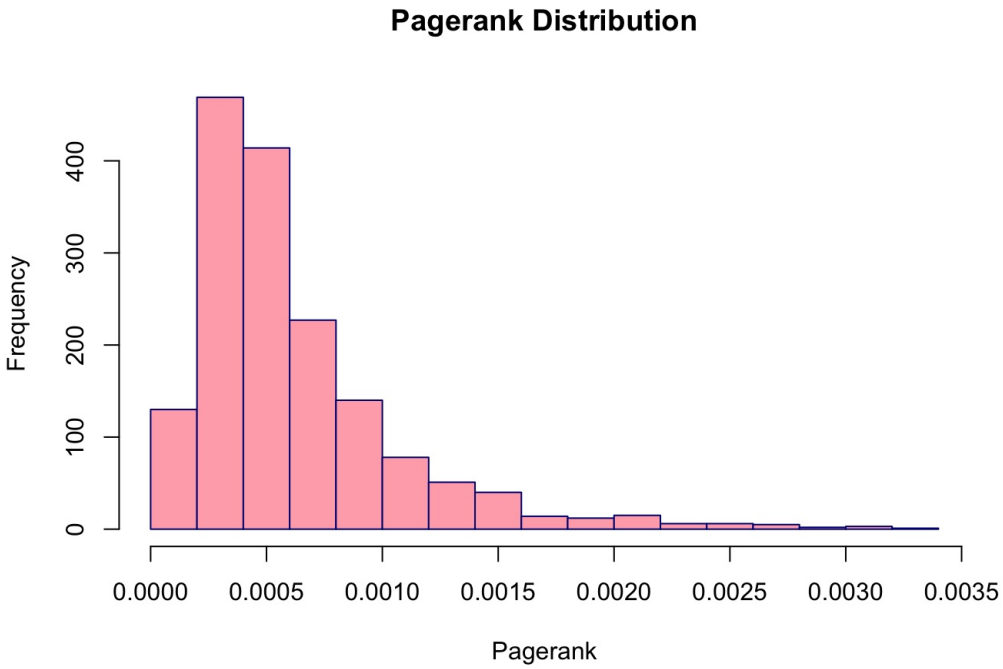
```
mam_between<- betweenness(graph_decomp)
hist(mam_between,
        main = "Betweeness Centrality Distribution",
        xlab = "Betweenees",
        ylab = "Frequency",
        col = "lightpink1",
        border = "navyblue",
)
```



The histogram suggests that there are a few key mammals or nodes that act as important intermediaries or bridge between nodes, while the majority low value of betweeness.

Another measure is *PAGERANK* that measures nodes as being important if they are connected to other nodes that are more important.

```
mam_pr<-page.rank(graph_decomp)$vector
hist(mam_pr,
     main = "Pagerank Distribution",
     xlab = "Pagerank",
     ylab = "Frequency",
     col = "lightpink1",
     border = "navyblue",
)
```



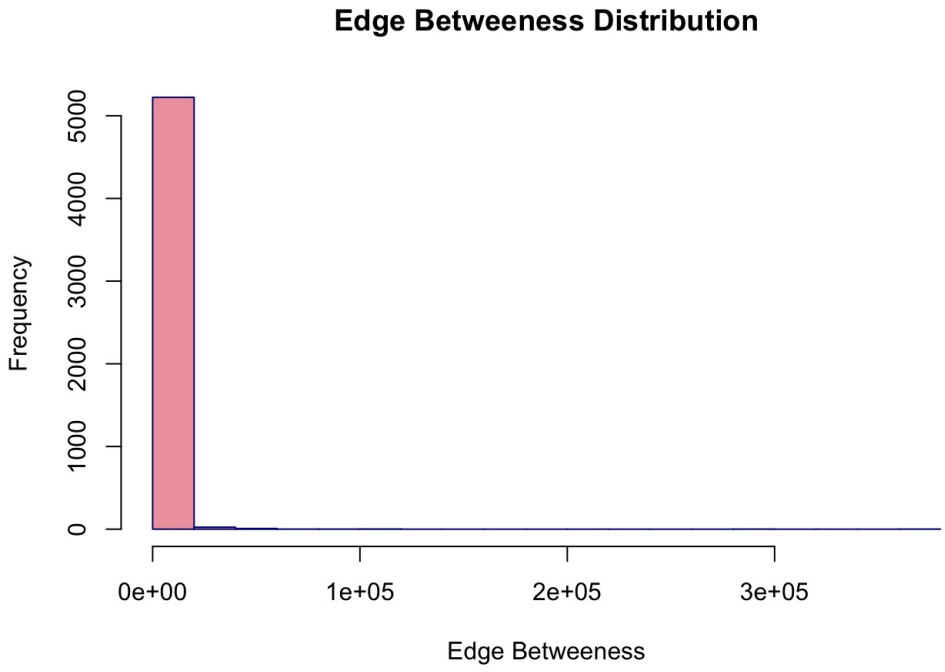
Also from here, we can see how the plot is left-skewed implying that there are specific mammals with high social prestige who have significant impact on the overall social interactions. They may play leadership roles or have a strong influence that shape the social interactions.

Next we have *EDGE BETWEENESS* which counts how many shortest paths go through a given edge. Measures the importance of individual edges in the network.

```
mam_eb<-edge_betweenness(graph_decomp)
tibble(edge=attr(E(graph_decomp), "vnames"), mam_eb) %>% arrange(-mam_eb)
```

```
## # A tibble: 5,261 × 1
##     mam_eb
##      <dbl>
##  1 377106.
##  2 287590.
##  3 117019.
##  4 101379.
##  5  94328.
##  6  68013.
##  7  50038.
##  8  48201.
##  9  45837.
## 10  45753.
## # … with 5,251 more rows
```

```
hist(mam_eb,
     main = "Edge Betweeness Distribution",
     xlab = "Edge Betweeness",
     ylab = "Frequency",
     col = "lightpink2",
     border = "navyblue",
)
```



By looking at the histogram, we notice that it is left-skewed suggesting the presence of specific edges that act as important connections within the mammal social network.

TRANSITIVITY

Next we are going to measure *Transitivity*, that measures the probability that the neighbours of a node are also connected to each other.

```
transitivity(graph_decomp)
```

```
## [1] 0.2657443
```

A value of around 0.26 means that there is a tendency for mammals to form local clusters or groups in their social networks. This moderate level of transitivity indicates that mammals tend to have interactions with specific communities that may represent social hierarchies.

After completing all the above measures and noticing that there may be a hierarchical structure in the network, we proceed the analysis by doing community detection, that typically seeks subsets of vertices that are well-connected among themselves and relatively-separated from others. This refers to the process of identifying communities or groups of closely connected nodes within a network.

ERGM

In order to study social interaction networks in mammals, ERGM can be very useful for analyzing the structure and dynamics of the network.

```
library(ergm)
```

```
## Caricamento del pacchetto richiesto: network
```

```
## Warning: il pacchetto 'network' è stato creato con R versione 4.1.2
```

```
##
## 'network' 1.18.1 (2023-01-24), part of the Statnet Project
## * 'news(package="network")' for changes since last version
## * 'citation("network")' for citation information
## * 'https://statnet.org' for help, support, and other information
```

```
##
## Caricamento pacchetto: 'network'
```

```
## I seguenti oggetti sono mascherati da 'package:igraph':
##
##     %c%, %s%, add.edges, add.vertices, delete.edges, delete.vertices,
##     get.edge.attribute, get.edges, get.vertex.attribute, is.bipartite,
##     is.directed, list.edge.attributes, list.vertex.attributes,
##     set.edge.attribute, set.vertex.attribute
```

```
##
## 'ergm' 4.5.0 (2023-05-27), part of the Statnet Project
## * 'news(package="ergm")' for changes since last version
## * 'citation("ergm")' for citation information
## * 'https://statnet.org' for help, support, and other information
```

```
## 'ergm' 4 is a major update that introduces some backwards-incompatible
## changes. Please type 'news(package="ergm")' for a list of major
## changes.
```

```
am_mam<- get.adjacency(graph_decomp, sparse = FALSE)
g_mam<- as.network(am_mam, directed= FALSE)

ergm_fit<- ergm(g_mam~edges)
```

```
## Starting maximum pseudolikelihood estimation (MPLE):
```

```
## Obtaining the responsible dyads.
```

```
## Evaluating the predictor and response matrix.
```

```
## Maximizing the pseudolikelihood.
```

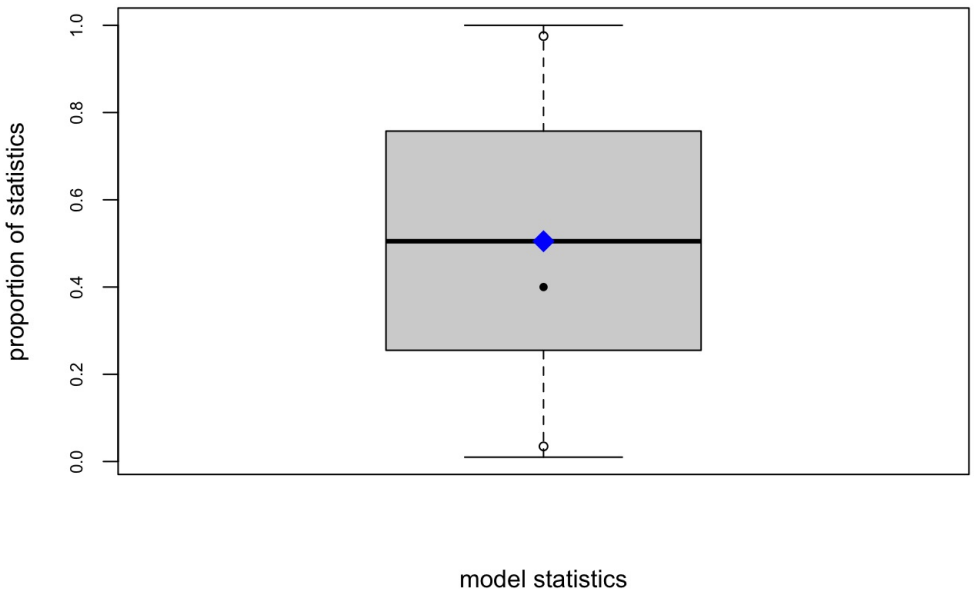
```
## Finished MPLE.
```

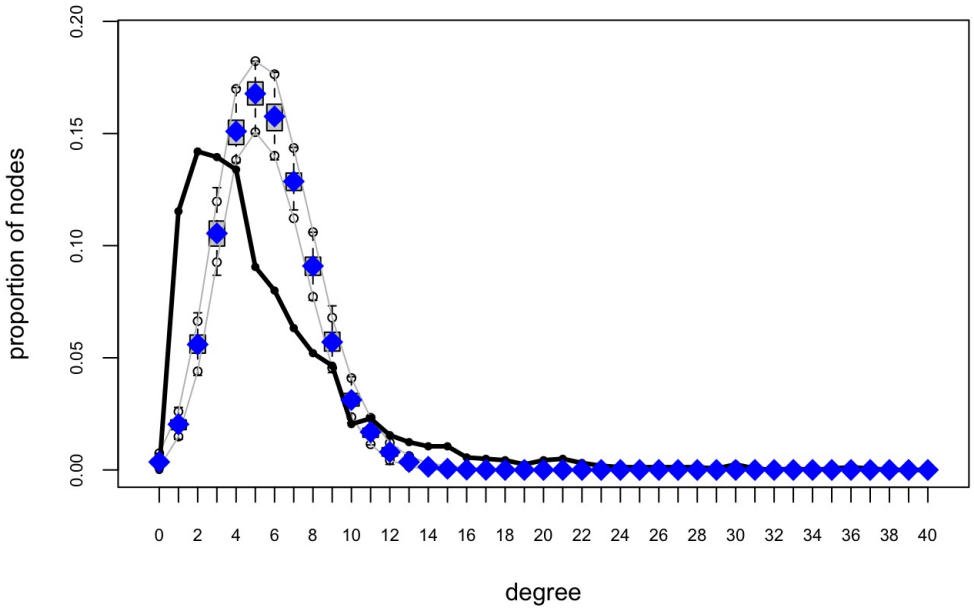
```
## Evaluating log-likelihood at the estimate.
```

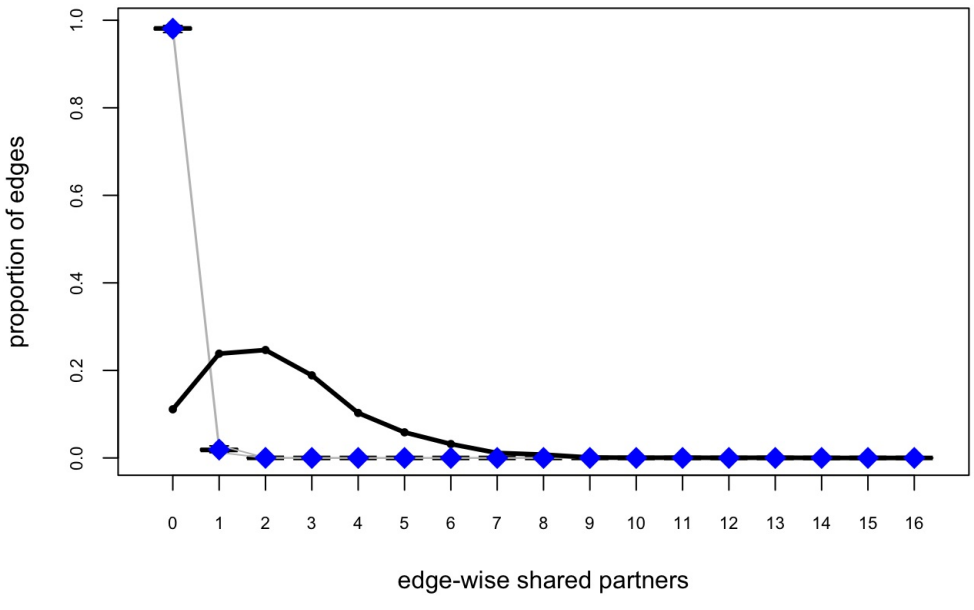
```
ergm_fit  %>% summary()
```

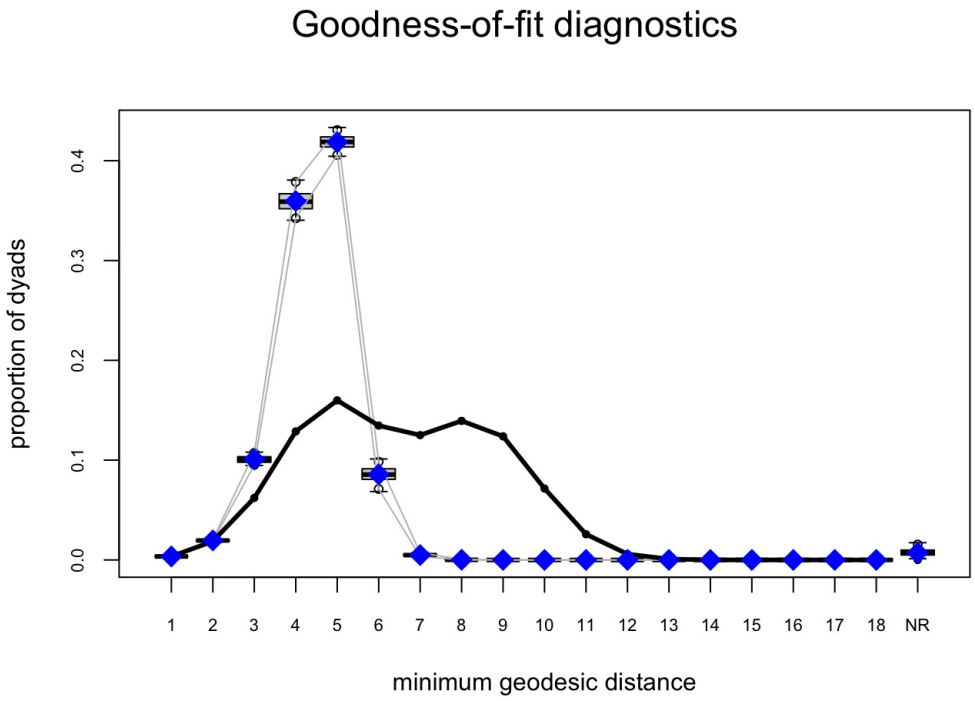
```
## Call:
## ergm(formula = g_mam ~ edges)
##
## Maximum Likelihood Results:
##
##       Estimate Std. Error MCMC % z value Pr(>|z|)
## edges -5.64890    0.01483      0  -380.9   <1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 1802291  on 1300078  degrees of freedom
##  Residual Deviance:   60681  on 1300077  degrees of freedom
##
## AIC: 60683  BIC: 60695  (Smaller is better. MC Std. Err. = 0)
```

```
gof_mam<- gof(ergm_fit)
plot(gof_mam)
```


proportion of statistics
0.0
0.2
0.4
0.6
0.8
1.0
model statistics


proportion of nodes
0.00
0.05
0.10
0.15
0.20
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40
degree


proportion of edges
0.0
0.2
0.4
0.6
0.8
1.0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
edge-wise shared partners



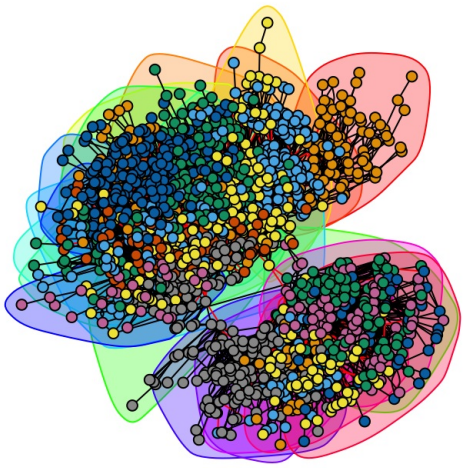
From the plots we can see that the observed network with the network generated by the fitted ergm model exhibit similar network properties, in particular looking at the degree distribution.

CLUSTERING

Clustering according to edge betweenn ess.

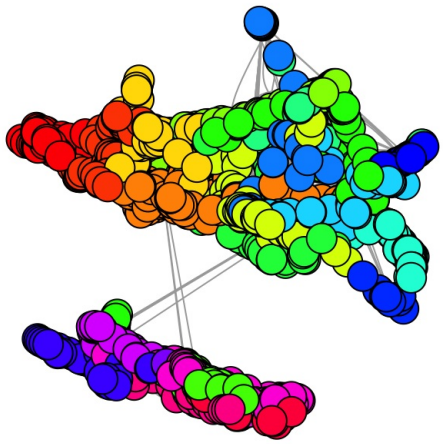
Since we are dealing with a graph with multi-edges, algorithms as frast_greedy are not useful. So, we proceed by looking for groups within our mammal network by *clustering* node groups according to their edge betweenness, that focuses on the edges of a network graph rather than the individual nodes.

```
mam_clus<-cluster_edge_betweenness(graph_decomp)
plot(mam_clus,
     graph_decomp,
     vertex.size = 5,
     vertex.label = NA,
     layout= layout.kamada.kawai
)
```



From the plot we can identify two main clusters, also quite defined and separated, indicating a well-defined community structure. After identifying the clusters, we can plot the membership of the clusters based on colors depending by the membership.

```
cluster_membership<-membership(mam_clus)
num_clusters <- max(cluster_membership)
color_palette <- rainbow(num_clusters)
V(graph_decomp)$color <- color_palette[cluster_membership]
plot(graph_decomp,
     vertex.size = 15,
     vertex.label = NA)
```



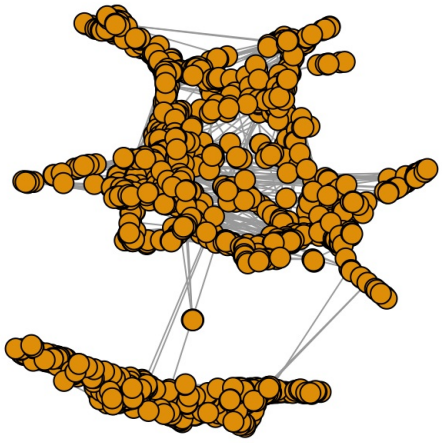
SPECTRAL CLUSTERING

Another way of clustering is by doing *Spectral Clustering*, very useful in cases where datapoints are not easily separable, since in this case we have multi-edges. In particular, it is based on the idea that the eigenvector of the matrix derived from the graph, contain useful information about the underlying structure of the network.

```
# Create a graph object from the adjacency matrix
adjacency_matrix<-as_adjacency_matrix(graph_decomp, sparse = FALSE)
spec_graph <- graph.adjacency(adjacency_matrix, mode = "undirected", weighted = TRUE)

# Calculate the Laplacian matrix
laplacian <- laplacian_matrix(graph)

# Compute the eigenvectors of the Laplacian matrix
eigenvals <- eigen(laplacian)$values
eigenvecs <- eigen(laplacian)$vectors

# Perform k-means clustering on the eigenvectors
k <- 3  # Number of clusters
cluster_labels <- kmeans(eigenvecs[, 2:k], centers = k)$cluster

#table that counts num of mammals in each cluster
table(cluster_labels)
```

```
## cluster_labels
##    1    2    3
## 1684    1    1
```

```
# Visualize the clusters
plot(spec_graph,
     vertex.color = cluster_labels,
     vertex.size = 10,
     vertex.label = NA)
```



Also here, after plotting, we can see two distinct clusters in the mammal social interaction network. This suggests the presence of the existence of two distinct social groups or subpopulations within the community. This may indicate distinct social behaviors, habitat preferences or ecological factors that influence their social interactions.