FOOKAIYAN33085625

Foo Kai Yan

2024-06-06

Student Name: Foo Kai Yan

Student ID: 33085625

Student Email: kfoo0012@student.monash.edu

Remove/Clean the environment

```
rm(list=ls())
```

Set working directory

```
setwd("C:/Monash/FIT3152")
```

Install and load the libraries used

```
library(tm)
```

```
## Loading required package: NLP
```

```
library(slam)
library(proxy)
```

```
##
## Attaching package: 'proxy'
```

```
## The following objects are masked from 'package:stats':
##
##     as.dist, dist
```

```
## The following object is masked from 'package:base':
##
##     as.matrix
```

```
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(SnowballC)
library(igraphdata)
```

Question 1

I have gathered 15 documents that cover three different topics. The first topic is **Cryptocurrency**, with the materials mainly taken from news articles and different websites devoted to the topic. The second topic explores **Hoyoverse Games**, utilizing data sourced from blogs, game reviews, and the official fandom pages of the games. Finally, the third topic is the **Wonders of the World**, which is examined using information sourced from news articles and websites that provide insight into these amazing locations. Each of the three topics is represented by a set of five documents.

Question 2

```
# Get file path to folder "CorpusAbstracts" where all the 15 documents are located
cname = file.path(".", "CorpusAbstracts")
cname
```

```
## [1] "./CorpusAbstracts"
```

```
dir(cname)
```

```
##  [1] "CC1.txt" "CC2.txt" "CC3.txt" "CC4.txt" "CC5.txt" "GM1.txt" "GM2.txt"
##  [8] "GM3.txt" "GM4.txt" "GM5.txt" "WW1.txt" "WW2.txt" "WW3.txt" "WW4.txt"
## [15] "WW5.txt"
```

```
docs = Corpus(DirSource((cname)))
summary(docs)
```

```
##         Length Class             Mode
## CC1.txt 2      PlainTextDocument list
## CC2.txt 2      PlainTextDocument list
## CC3.txt 2      PlainTextDocument list
## CC4.txt 2      PlainTextDocument list
## CC5.txt 2      PlainTextDocument list
## GM1.txt 2      PlainTextDocument list
## GM2.txt 2      PlainTextDocument list
## GM3.txt 2      PlainTextDocument list
## GM4.txt 2      PlainTextDocument list
## GM5.txt 2      PlainTextDocument list
## WW1.txt 2      PlainTextDocument list
## WW2.txt 2      PlainTextDocument list
## WW3.txt 2      PlainTextDocument list
## WW4.txt 2      PlainTextDocument list
## WW5.txt 2      PlainTextDocument list
```

```
# Function to count the words in each document
countWordsFunc <- function(doc) {
  words <- strsplit(as.character(doc), "\\s+")
  return(length(words[[1]]))
}

# Apply the function to each of the documents
docWordCounts <- lapply(docs, countWordsFunc)

# Convert the list to a dataframe
wordCountsDf <- data.frame(Document_Word_Count = unlist(docWordCounts)) #
data.frame(docName = names(docWordCounts), wordCount = unlist(docWordCounts))
wordCountsDf
```

```
##         Document_Word_Count
## CC1.txt                 265
## CC2.txt                 299
## CC3.txt                 237
## CC4.txt                 279
## CC5.txt                 345
## GM1.txt                 161
## GM2.txt                 164
## GM3.txt                 230
## GM4.txt                 128
## GM5.txt                 210
## WW1.txt                 157
## WW2.txt                 299
## WW3.txt                 305
## WW4.txt                 346
## WW5.txt                 268
```

The dataframe displayed above shows that the number of words in the 15 documents varies, from 128 words in the shortest document to 346 words in the longest one. This variation in content length shows that some documents give brief summaries while others delve deeper into their subject matter. The difference in word count can be explained by the type of sources, as news articles tend to provide extensive coverage while websites may give summarized content.

It is noted that the name of the text file represents which topic the document is under:

- Cryptocurrency is represented by CC

- Hoyoverse Games is represented by GM

- Wonders of the World is represented by WW

For all of the documents that was found, only a portion of the article or websites contents were pasted into a txt file as if the whole content was pasted, the content would be too much, hence, only a selected part was pasted into the txt files. None of the documents were initially in another file format so no additional steps were done to convert the documents into text file format .txt.

Question 3

One of the specific text transformation that was done for the documents is to modify to change 'Hypen' to 'Space' within the documents.

```
# Specific Text Transformation
# Hypen to Space
toSpace = content_transformer(function(x, pattern) gsub(pattern, " ", x))
docs = tm_map(docs, toSpace, "-")

# Tokenization, Stemming
# Remove any potential numbers or numerical within the documents
docs = tm_map(docs, removeNumbers)
# Remove any punctuation from the document
docs = tm_map(docs, removePunctuation)
# Modify the characters within the 15 documents to lowercase for consistency purposes
docs = tm_map(docs, content_transformer(tolower))
# Modify to remove the word 'english'
docs = tm_map(docs, removeWords, stopwords("english"))
# Remove any additional whitespace within the document
```

```r
docs = tm_map(docs, stripWhitespace)
# Stemming to reduces words to their base or root form
docs = tm_map(docs, stemDocument, language = "english")

# Create document term matrix (DTM)
dtm <- DocumentTermMatrix(docs)
dim(dtm)

## [1]   15 1054
```

Without removing any sparse terms (words that lack context but frequently appear in the documents) from the documents, there is 1054 tokens.

```r
dtm_new = removeSparseTerms(dtm, 0.67) # 0.67 because if 0.66 there would be 9 tokens only
dim(dtm_new)

## [1] 15 23
```

After removing the sparse terms from the documents, there is 23 tokens. The sparse parameter that is chosen is 0.67 as if it is 0.66 the token is only 9 but when increase in 0.01 there token has increase to 23. The sparse parameter of 0.67 indicates that terms that appear in less than 33% of the documents will be removed.

```r
# Convert to table
dtm_new_table = as.table(dtm_new)

# Convert to dataframe
dtm_new_df = as.data.frame(as.matrix(dtm_new))
dtm_new_df

##         bitcoin can cryptocurr like mani new still system use also although
## CC1.txt       2   2         18    1    1   1     1      1   5    0        0
## CC2.txt       6   2         12    0    0   1     2      1   5    1        1
## CC3.txt       2   0          9    1    0   0     0      0   2    1        1
## CC4.txt       6   0          2    1    0   1     1      0   0    0        1
## CC5.txt       3   1          8    2    1   3     0      2   3    1        0
## GM1.txt       0   0          0    0    0   0     0      0   0    0        0
## GM2.txt       0   2          0    1    1   0     0      0   0    1        0
## GM3.txt       0   0          0    1    0   0     0      1   0    0        0
## GM4.txt       0   0          0    0    0   0     0      0   1    0        0
## GM5.txt       0   2          0    0    0   0     0      0   0    2        0
## WW1.txt       0   0          0    0    1   0     2      0   0    0        0
## WW2.txt       0   0          0    1    0   1     1      0   0    0        0
## WW3.txt       0   0          0    0    1   0     2      0   0    0        2
## WW4.txt       0   0          0    0    1   0     0      1   1    1        1
## WW5.txt       0   0          0    0    0   0     1      0   0    0        0
##         around one peopl list will world high great ancient centuri seven
## CC1.txt      0   0     0    0    0     0    0     0       0       0     0
## CC2.txt      1   1     1    0    0     0    0     0       0       0     0
## CC3.txt      0   0     0    1    2     1    0     0       0       0     0
## CC4.txt      0   1     0    3    1     0    0     0       0       0     0
## CC5.txt      1   0     1    0    0     2    1     0       0       0     0
## GM1.txt      1   0     0    0    0     0    0     0       0       0     0
## GM2.txt      0   0     1    0    1     0    1     0       0       0     0
## GM3.txt      0   2     0    0    0     4    0     0       0       0     0
## GM4.txt      0   1     1    0    0     0    2     0       0       0     0
```

```
## GM5.txt      1   0     0    0    2     1    0     1       0       0     0
## WW1.txt      1   0     0    1    0     3    0     3       6       1     6
## WW2.txt      1   0     0    1    1     8    1     4       5       1     5
## WW3.txt      5   1     1    1    0     2    0     3       4       3     3
## WW4.txt      0   2     0    0    0     1    0     0       1       1     1
## WW5.txt      0   1     0    2    0     1    1     1       1       2     1
##         wonder
## CC1.txt      0
## CC2.txt      0
## CC3.txt      0
## CC4.txt      0
## CC5.txt      0
## GM1.txt      0
## GM2.txt      0
## GM3.txt      0
## GM4.txt      0
## GM5.txt      0
## WW1.txt      6
## WW2.txt     10
## WW3.txt      4
## WW4.txt      1
## WW5.txt      2
```

Transformation done above:

- Specific Text Transformation by modifying hypen (-) to whitespace ( )

- Remove any potential numbers or numerical within the documents

- Remove any punctuation from the document

- Modify the characters within the 15 documents to lowercase for consistency purposes

- Modify to remove the word 'english'

- Remove any additional whitespace within the document

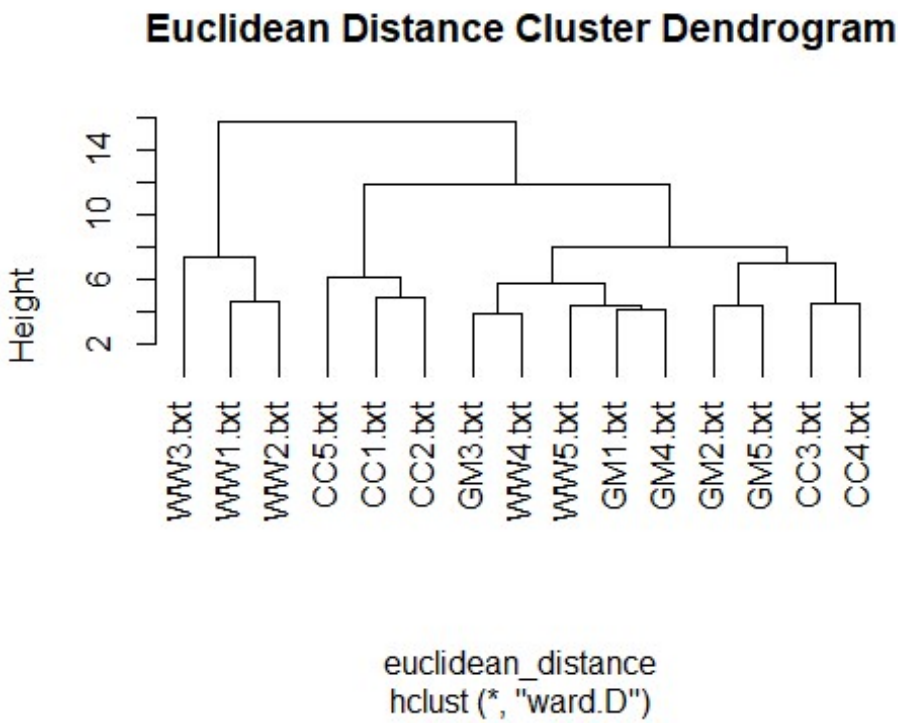
- Stemming to reduces words to their base or root form

Question 4

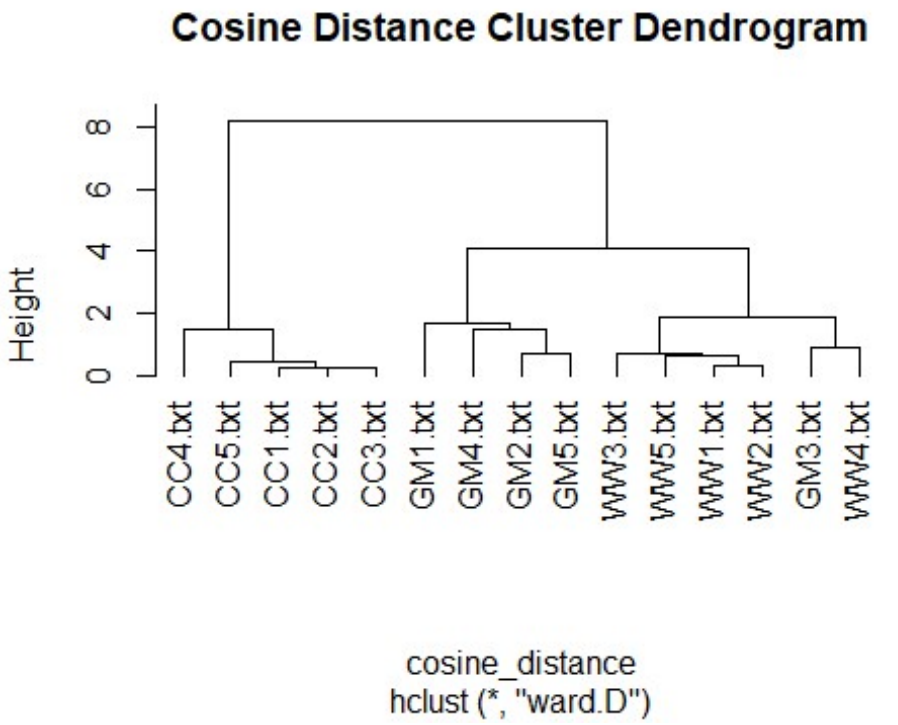
```r
# Euclidean Distance
euclidean_distance = dist(scale(dtm_new))
# Cosine Distance
cosine_distance = dist(crossprod_simple_triplet_matrix(t(dtm_new)) /
sqrt(col_sums(t(dtm_new)^2) %*% t(col_sums(t(dtm_new)^2))))

euclidean_cluster = hclust(euclidean_distance, method = "ward.D")
cosine_cluster = hclust(cosine_distance, method = "ward.D")

# Plot Dendrogram
plot(euclidean_cluster, hang = -1, main = "Euclidean Distance Cluster Dendrogram")
```




```
plot(cosine_cluster, hang = -1, main = "Cosine Distance Cluster Dendrogram")
```



Referring to the Dendrograms in the image above, each branch represents a cluster in the dendrogram. The y-axis measures the Euclidean distance where when the value is higher, then there is a larger degree of difference.

```
cut_euclidean_tree = cutree(euclidean_cluster, k = 3) # 3 Topics
tbl_results_euclidean = table(TopicNames = c("Cryptocurrency", "Cryptocurrency",
"Cryptocurrency", "Cryptocurrency", "Cryptocurrency",
```

```
                                          "Hoyoverse Games", "Hoyoverse Games", "Hoyoverse
Games", "Hoyoverse Games", "Hoyoverse Games",
                                          "Wonders of the World", "Wonders of the World",
"Wonders of the World", "Wonders of the World",
                                          "Wonders of the World"),
                           Clusters = cut_euclidean_tree)
tbl_results_euclidean
```

```
##                        Clusters
## TopicNames              1 2 3
##   Cryptocurrency        3 2 0
##   Hoyoverse Games       0 5 0
##   Wonders of the World  0 2 3
```

```
accuracy_euclidean = (tbl_results_euclidean[1, 1] + tbl_results_euclidean[2, 2] +
tbl_results_euclidean[3, 3]) / (tbl_results_euclidean[1, 1] + tbl_results_euclidean[1, 2]
+ tbl_results_euclidean[1, 3] + tbl_results_euclidean[2, 1] + tbl_results_euclidean[2, 2]
+ tbl_results_euclidean[2, 3] + tbl_results_euclidean[3, 1] + tbl_results_euclidean[3, 2]
+ tbl_results_euclidean[3, 3])
accuracy_euclidean
```

```
## [1] 0.7333333
```

The table above stored within the `tbl_results_euclidean` contains the distribution of three different groups across three clusters, which is the result of hierarchical clustering. Each number in the table indicates how many items from each topic group were assigned to each cluster. There is a total of 3 clusters. For the topic **Cryptocurrency**, 3 documents were assigned to Cluster 1, 2 documents to Cluster 2, and none to Cluster 3. For the topic **Hoyoverse Games**, 5 documents were assigned to Cluster 2 which indicates that there exists a strong similarity within this group. Finally, no documents were assigned to Cluster 1, 2 documents to Cluster 2, and 3 documents to Cluster 3 for the topic **Wonders of the World**. This suggets that **Cryptocurrency** is mostly in Cluster 1, **Hoyoverse Games** is exclusively in Cluster 2, and **Wonders of the World** is divided between Clusters 2 and 3, suggesting different patterns of similarity within each documents of the topic. It is calculated that there is an accuracy of approximately 0.7333333 which is 73.33%. The accuracy shows that the euclidean clustering model correctly predict the documents to clusters only about 73.33% of the time.

```
cut_cosine_tree = cutree(cosine_cluster, k = 3) # 3 Topics
tbl_results_cosine = table(TopicNames = c("Cryptocurrency", "Cryptocurrency",
"Cryptocurrency", "Cryptocurrency", "Cryptocurrency",
                                          "Hoyoverse Games", "Hoyoverse Games", "Hoyoverse
Games", "Hoyoverse Games", "Hoyoverse Games",
                                          "Wonders of the World", "Wonders of the World",
"Wonders of the World", "Wonders of the World",
                                          "Wonders of the World"),
                           Clusters = cut_cosine_tree)
tbl_results_cosine
```

```
##                        Clusters
## TopicNames              1 2 3
##   Cryptocurrency        5 0 0
##   Hoyoverse Games       0 4 1
##   Wonders of the World  0 0 5
```

```
accuracy_cosine = (tbl_results_cosine[1, 1] + tbl_results_cosine[2, 2] +
tbl_results_cosine[3, 3]) / (tbl_results_cosine[1, 1] + tbl_results_cosine[1, 2] +
tbl_results_cosine[1, 3] + tbl_results_cosine[2, 1] + tbl_results_cosine[2, 2] +
```

```
tbl_results_cosine[2, 3] + tbl_results_cosine[3, 1] + tbl_results_cosine[3, 2] +
tbl_results_cosine[3, 3])
accuracy_cosine

## [1] 0.9333333
```

The table above stored within the `tbl_results_cosine` contains the distribution of three different groups across three clusters, which is the result of hierarchical clustering. Each number in the table indicates how many items from each topic group were assigned to each cluster. There is a total of 3 clusters. For the topic **Cryptocurrency**, 5 documents were assigned to Cluster 1 which indicates that there exists a strong similarity within this group. For the topic **Hoyoverse Games**, 4 documents were assigned to Cluster 2 and 1 document to Cluster 3. Finally, for the topic **Wonders of the World**, all 5 documents were assigned to Cluster 3. This suggests that **Cryptocurrency** is exclusively in Cluster 1, **Hoyoverse Games** 4 documents were correctly grouped into Cluster 2, and 1 document was most likely incorrectly assigned into Cluster 3, and **Wonders of the World** exclusively in Cluster 3. It is calculated that there is an accuracy of approximately 0.9333333 which is 93.33%. The accuracy shows that the cosine clustering model correctly predict the documents to clusters only about 93.33% of the time.

By observing the results of the dendrogram and the measure of quality of each of the clustering done, both cosine and euclidean clustering shows that there is high clustering accuracy of more than 70% but cosine clustering express a better clustering result than euclidean clustering as cosine clustering only incorrectly cluster 1 **Hoyoverse Games** but euclidean clustering incorrectly cluster 2 **Cryptocurrency** and **Wonders of the World** to the same cluster as **Hoyoverse Games**. Euclidean clustering did generally correctly classify the documents to their topics but cosine clustering did a better clustering that euclidean clustering. Moreover, euclidean clustering has a lower clustering accuracy than cosine clustering.

Question 5

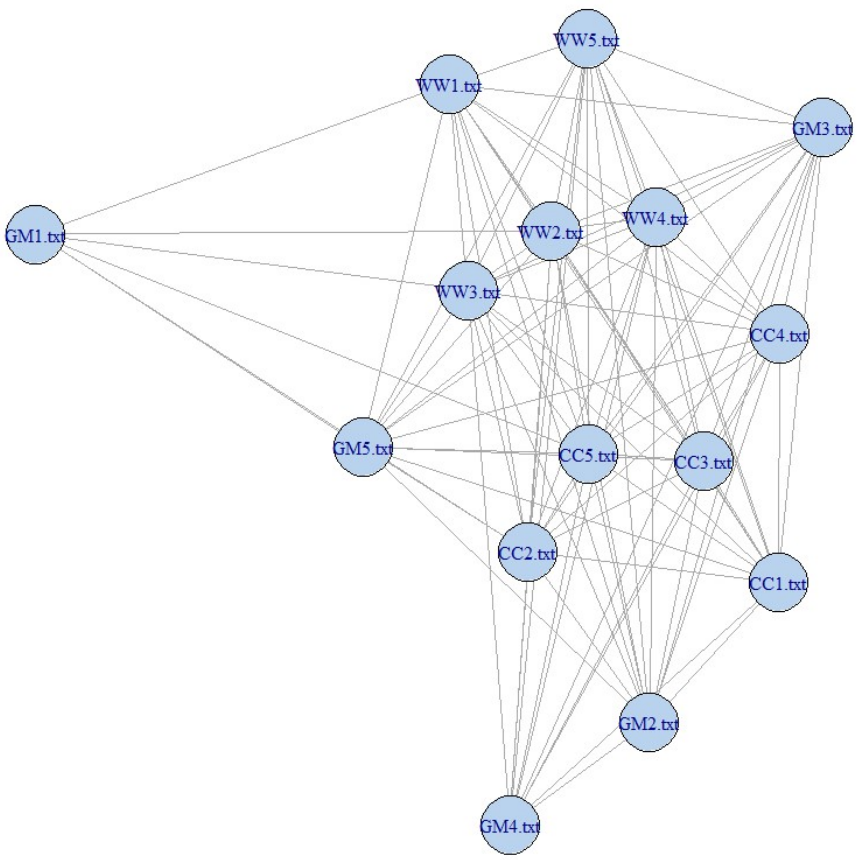
```
# Convert to Matrix
dtm_matrix = as.matrix(dtm_new)

# Convert to Binary Matrix
dtm_matrix_binary = as.matrix((dtm_matrix > 0) + 0)

# Transpose the Binary Matrix
dtm_matrix_binary_tranpose = dtm_matrix_binary %*% t(dtm_matrix_binary)

# Make Leading Diagonal Zero
diag(dtm_matrix_binary_tranpose) = 0

# Create Graph Object
grph_obj = graph_from_adjacency_matrix(dtm_matrix_binary_tranpose, mode = "undirected",
weighted = TRUE)
plot(grph_obj, vertex.color = "slategray2", main = "Relationship of 15 documents from 3
topics")
```



Each nodes in the network graph above is represented by the 15 documents. The lines connecting each nodes represents the relationships or degree of similarity between each nodes. The nodes positioned more middle of the network graph indicates that the said node possess more connection to other nodes hence is positioned more central to the network graph.

```
degree = as.table(degree(grph_obj))
betweenness = as.table(betweenness(grph_obj))
closeness = as.table(closeness(grph_obj))
eig = as.table(evcent(grph_obj)$vector)
```

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

```
averagePath = average.path.length(grph_obj)
```

```
## 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.
```

```
averagePath

## [1] 2.161905

diameter = diameter(grph_obj)
diameter

## [1] 4

tabularised_data_df = as.data.frame(rbind(degree, betweenness, closeness, eig))
tabularised_data_tbl = t(tabularised_data_df)
tabularised_data_tbl

##         degree betweenness  closeness       eig
## CC1.txt     13   3.0666667 0.03571429 0.6542463
## CC2.txt     14   0.2000000 0.02857143 0.8231657
## CC3.txt     13   0.0000000 0.02777778 0.6893646
## CC4.txt     13   0.5333333 0.03333333 0.6770217
## CC5.txt     14   0.2000000 0.02777778 0.8942904
## GM1.txt      6  23.0190476 0.04000000 0.1026816
## GM2.txt     13   4.9500000 0.03448276 0.4892002
## GM3.txt     13   5.2857143 0.03846154 0.3799503
## GM4.txt     11  15.1785714 0.04000000 0.3027036
## GM5.txt     13  12.1500000 0.04000000 0.4929622
## WW1.txt     13   4.3357143 0.03846154 0.8409133
## WW2.txt     14   2.7261905 0.03448276 0.9755577
## WW3.txt     14   0.0000000 0.03125000 1.0000000
## WW4.txt     13   0.0000000 0.02439024 0.8412389
## WW5.txt     13   3.2595238 0.03225806 0.8148967
```

Using the data on the graph as displayed as a table above, we can know on the degree, betweenness, closeness and Eigenvector Centrality (EIG) as well as the diameter and average path of the graph. The degree indicates the number of connections a node has and nodes with higher degrees like CC2.txt, CC5.txt, WW2.txt and WW3.txt have a degree of 14 are more connected within the network as it connects to more nodes within the network. It is noted from the table that there are 10 nodes with a degree of 13 too. Betweenness measures a node's centrality in the network graph based on the shortest paths that pass through it. A high betweenness is seen with GM1.txt which have a betweenness value of 23.0190476 suggests the node may be used to act as a bridge within the network. Closeness reflects how close a node is to all other nodes. The highest closeness is noted at 0.04000000 for GM1.txt, GM4.txt, and GM5.txt which indicates that these 3 nodes can quickly interact with each others in the network graph. EIG indicates a node's influence based on the connectivity of its neighbors. A high EIG is noted in WW3.txt with a value of 1.0000000 shows that the node may be connected to many well-connected nodes within the graph. Average path of the network graph have a length of 2.161905 which strongly suggests that, on average, it takes about 2.16 steps to get from one node to another within the network graph. This indicates that there is a relatively high level of connectivity within the nodes of the network graph as nodes can be reached quickly from one another within the network graph. Diameter shows how spread out the nodes is within the network graph and in this network graph, there is a diameter of 4 means that the farthest distance between any two nodes in the network is four steps. It can be observed that the highest betweenness is also the highest closeness. No significant further improvement is done as the network graph is already readable but if an improvement is really required, the code chunk below shows an improvement done whereby the network lines between the nodes is scaled in proportion to the betweenness of the nodes.

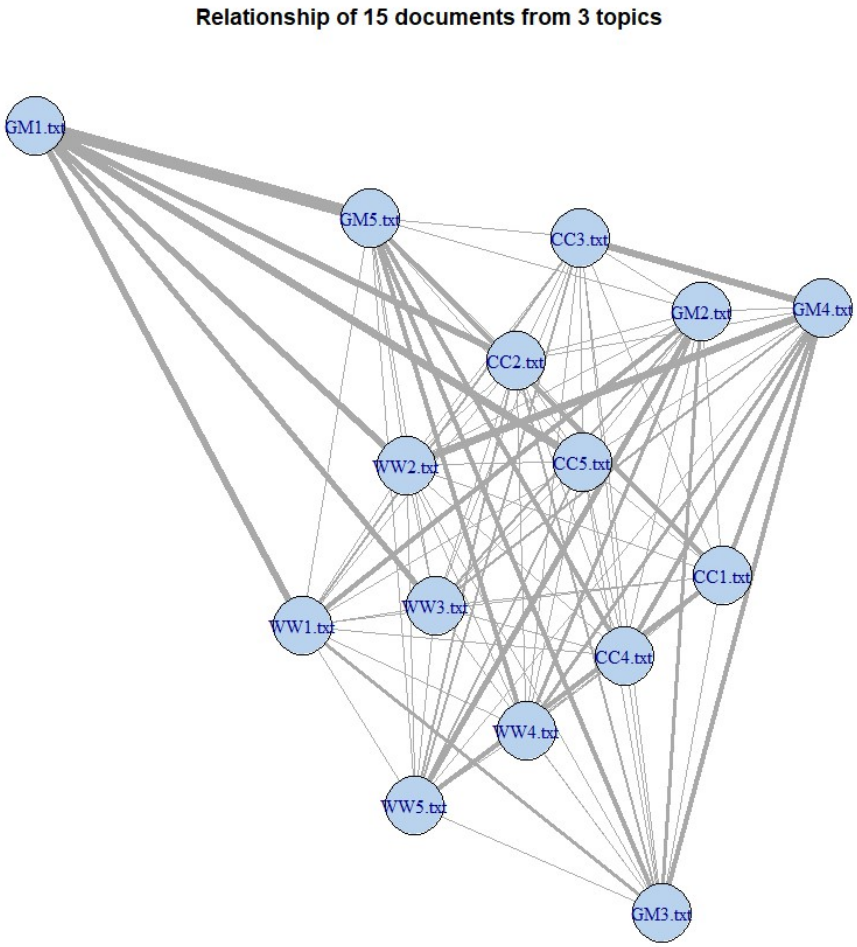
```
grph_obj_betweeness_improvement <-
graph_from_adjacency_matrix(dtm_matrix_binary_tranpose, mode = "undirected", weighted =
```

```
TRUE)

# Get the Betweenness of Edges in graph
edge_betweenness <- edge_betweenness(grph_obj_betweeness_improvement)

# Normalize the Betweenness of Edges in graph
max_betweenness <- max(edge_betweenness)
min_betweenness <- min(edge_betweenness)
normalized_edge_betweenness <- (edge_betweenness - min_betweenness) / (max_betweenness -
min_betweenness) * 10 + 1

# Include into the graph
plot(grph_obj_betweeness_improvement, vertex.color = "slategray2", main = "Relationship
of 15 documents from 3 topics",
     edge.width = normalized_edge_betweenness)
```



The edges connecting the nodes indicates the relationships or connections between the 15 documents. It is noted that these edges are undirected, so they don't have a specific direction. After the improvement, the thicker edges means stronger relationships between each of the connected documents. The original network graph without the improvement is also shown above for comparison purposes.

Question 6

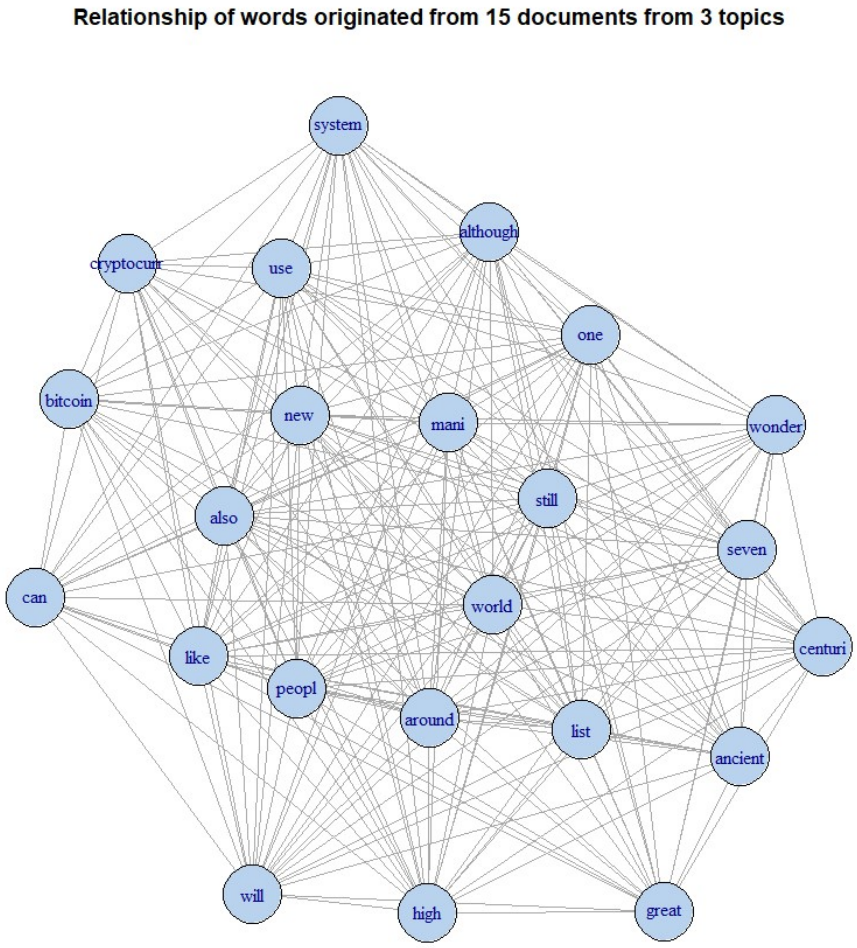
```
# Convert to Matrix
dtm_matrix = as.matrix(dtm_new)

# Convert to Binary Matrix
dtm_matrix_binary = as.matrix((dtm_matrix > 0) + 0)

# Transpose the Binary Matrix
dtm_matrix_binary_tranpose_bi = t(dtm_matrix_binary) %*% dtm_matrix_binary

# Make Leading Diagonal Zero
diag(dtm_matrix_binary_tranpose_bi) = 0

# Create Graph Object
grph_obj_bi = graph_from_adjacency_matrix(dtm_matrix_binary_tranpose_bi, mode =
"undirected", weighted = TRUE)
plot(grph_obj_bi, vertex.color = "slategray2", main = "Relationship of words originated
from 15 documents from 3 topics")
```



Each nodes in the network graph above is represented by the terms found within 15 documents. The terms might not appear in all the 15 documents. According to the network graph above, it is shown that

there is a connection between each of the terms even if the terms originated from different documents. The lines connecting each nodes represents the relationships or degree of similarity between each nodes. The nodes positioned more middle of the network graph indicates that the said node possess more connection to other nodes hence is positioned more central to the network graph. From the graph, it can be seen that there is no clear groups or clusters on the terms.

```
degree_2 = as.table(degree(grph_obj_bi))
betweenness_2 = as.table(betweenness(grph_obj_bi))
closeness_2 = as.table(closeness(grph_obj_bi))
eig_2 = as.table(evcent(grph_obj_bi)$vector)

averagePath_2 = average.path.length(grph_obj_bi)
averagePath_2
```

```
## [1] 2.162055
```

```
diameter_2 = diameter(grph_obj_bi)
diameter_2
```

```
## [1] 4
```

```
tabularised_data_df_2 = as.data.frame(rbind(degree_2, betweenness_2, closeness_2, eig_2))
tabularised_data_tbl_2 = t(tabularised_data_df_2)
tabularised_data_tbl_2
```

```
##            degree_2 betweenness_2 closeness_2     eig_2
## bitcoin          17    1.06666667  0.01785714 0.5746799
## can              17    4.81501832  0.01923077 0.5108676
## cryptocurr       17    1.06666667  0.01785714 0.5746799
## like             22    4.13909774  0.02040816 0.6989261
## mani             22    0.00000000  0.01960784 0.7206341
## new              22    6.15228456  0.02325581 0.6295523
## still            22    0.00000000  0.01851852 0.8723767
## system           19    7.23830228  0.02222222 0.5526171
## use              21    4.94285714  0.02272727 0.6349594
## also             22   14.98731203  0.02325581 0.6351226
## although         21    1.95739348  0.02040816 0.6259698
## around           22    0.08695652  0.01694915 0.7763313
## one              22    2.82505176  0.02173913 0.7162443
## peopl            22   16.55238787  0.02564103 0.5462013
## list             20    2.24761905  0.02083333 0.7629773
## will             21   25.21925466  0.02702703 0.5004169
## world            22    0.00000000  0.01515152 1.0000000
## high             21   14.82695870  0.02222222 0.5368546
## great            18   13.73920608  0.02272727 0.6415497
## ancient          19    7.01636522  0.02380952 0.7033986
## centuri          19    7.01636522  0.02380952 0.7033986
## seven            19    7.01636522  0.02380952 0.7033986
## wonder           19    7.01636522  0.02380952 0.7033986
```

Using the data on the graph as displayed as a table above, we can know on the degree, betweenness, closeness and EIG as well as the diameter and average path of the graph. The degree indicates the number of connections a node has and nodes with higher degrees. The highest number of degree recorded is 22 with the least is 17, there is only a difference of 5 for the degree. Betweenness measures a node's centrality in the network graph based on the shortest paths that pass through it. A high betweenness is seen with the word `will` which have a betweenness value of 25.21925466 suggests the

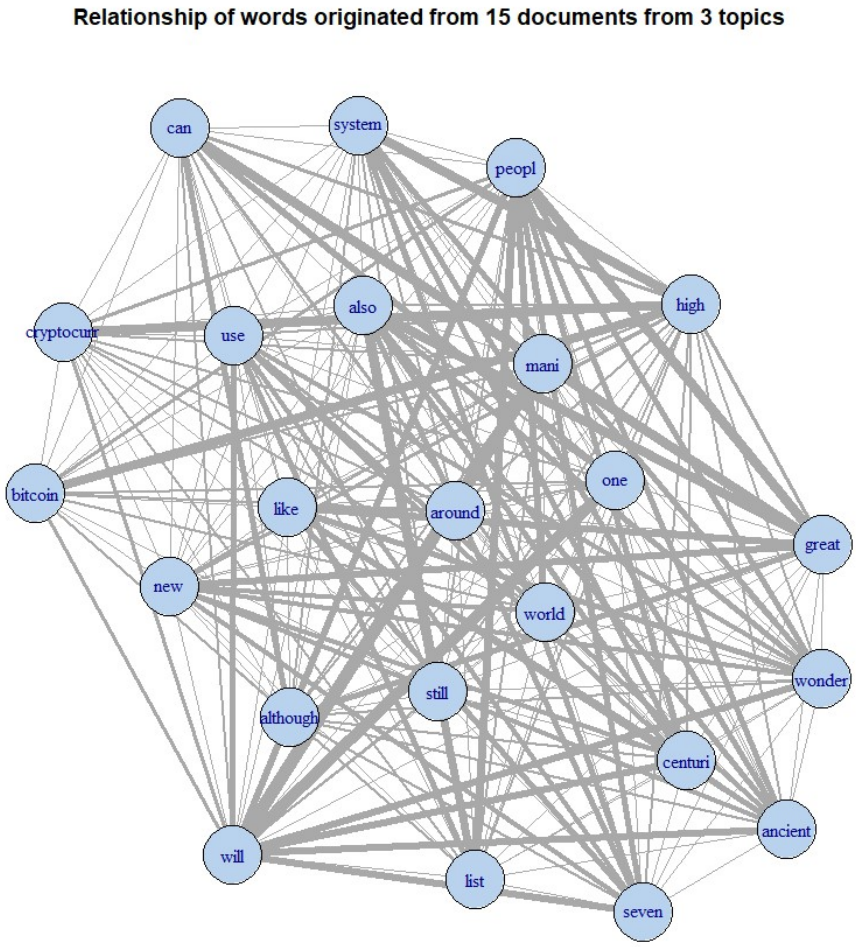
node may be used to act as a bridge within the network graph. Closeness reflects how close a node is to all other nodes and the highest closeness is noted at 0.02702703 for the word `will` which indicates that the `will` nodes can quickly interact with others nodes within the network graph. EIG indicates a node's influence based on the connectivity of its neighbors. A high EIG is noted in the word `world` with a value of 1.0000000 shows that the node may be connected to many well-connected nodes within the graph. Average path of the network graph have a length of 2.162055 which strongly suggests that, on average, it takes about 2.16 steps to get from one node to another within the network graph. This indicates that there is a relatively high level of connectivity within the nodes of the network graph as nodes can be reached quickly from one another within the network graph. Diameter shows how spread out the nodes is within the network graph and in this network graph, there is a diameter of 4 means that the farthest distance between any two nodes in the network is four steps. It can be observed that the highest betweenness is also the highest closeness. No significant further improvement is done as the network graph is already readable as it is but if an improvement is really required, the code chunk below shows an improvement done whereby the network lines between the nodes is scaled in proportion to the betweenness of the nodes. The stronger the relationship between the words from the documents, the thicker the network graph line is.

```
grph_obj_bi_betweeness_improvement <-
graph_from_adjacency_matrix(dtm_matrix_binary_tranpose_bi, mode = "undirected", weighted
= TRUE)

# Get the Betweenness of Edges in graph
edge_betweenness <- edge_betweenness(grph_obj_bi_betweeness_improvement)

# Normalize the Betweenness of Edges in graph
max_betweenness <- max(edge_betweenness)
min_betweenness <- min(edge_betweenness)
normalized_edge_betweenness <- (edge_betweenness - min_betweenness) / (max_betweenness -
min_betweenness) * 10 + 1

# Include into the graph
plot(grph_obj_bi_betweeness_improvement, vertex.color = "slategray2", main =
"Relationship of words originated from 15 documents from 3 topics",
     edge.width = normalized_edge_betweenness)
```



The edges connecting the nodes indicates the relationships or connections between the words from the 15 documents. It is noted that these edges are undirected, so they don't have a specific direction. After the improvement, the thicker edges means stronger relationships between each of the connected documents. The original network graph without the improvement is also shown above for comparison purposes.

Question 7

```
# Clone to another variable
dtm_newdf = as.data.frame(as.matrix(dtm_new))
# Add row names to dataframe
dtm_newdf$ABS = rownames(dtm_newdf)

dtm_newdf_b = data.frame()
for (i in 1:nrow(dtm_newdf)) {
  for (j in 1:(ncol(dtm_newdf) - 1)) {
    touse = cbind(dtm_newdf[i, j], dtm_newdf[i, ncol(dtm_newdf)], colnames(dtm_newdf[j]))
    dtm_newdf_b = rbind(dtm_newdf_b, touse)
  }
}
colnames(dtm_newdf_b) = c("weight", "abs", "token")

# Delete 0 Weights
```

```r
dtm_newdf_c = dtm_newdf_b[dtm_newdf_b$weight != 0, ]

# Order the columns in this order: abs, token, weight
dtm_newdf_c = dtm_newdf_c[, c(2, 3, 1)]

# Create Graph Object and Declare Bipartite
bipartite_graph = graph.data.frame(dtm_newdf_c, directed = FALSE)
```

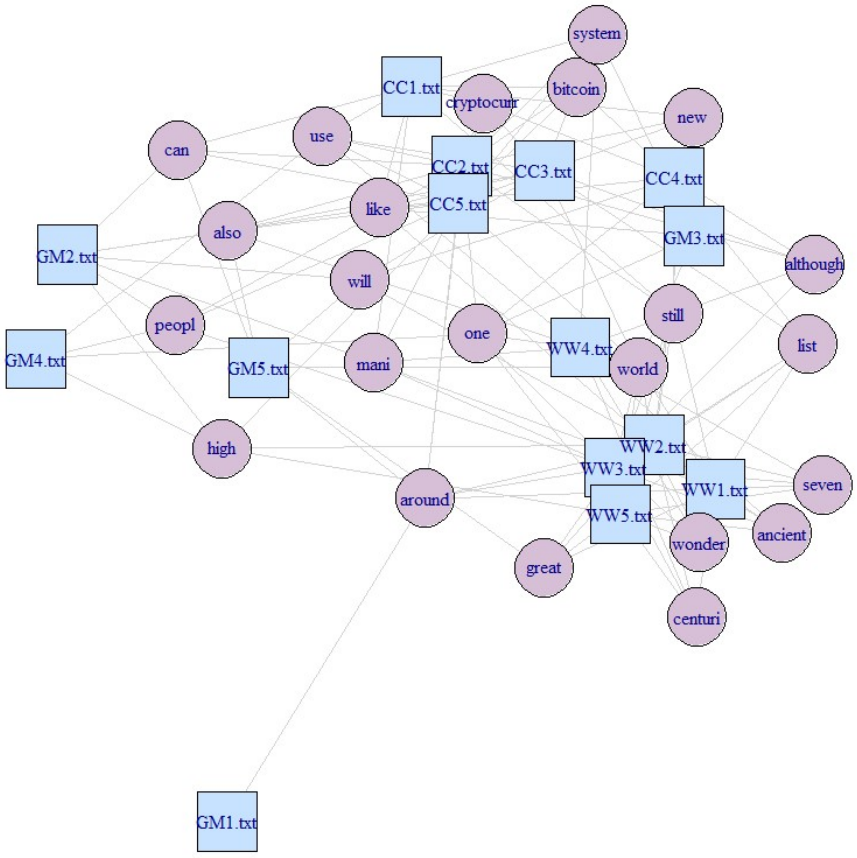
```
## Warning: `graph.data.frame()` was deprecated in igraph 2.0.0.
## ℹ Please use `graph_from_data_frame()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
bipartite.mapping(bipartite_graph)
```

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

## $res
## [1] TRUE
##
## $type
##    CC1.txt    CC2.txt    CC3.txt    CC4.txt    CC5.txt    GM1.txt    GM2.txt
##      FALSE      FALSE      FALSE      FALSE      FALSE      FALSE      FALSE
##    GM3.txt    GM4.txt    GM5.txt    WW1.txt    WW2.txt    WW3.txt    WW4.txt
##      FALSE      FALSE      FALSE      FALSE      FALSE      FALSE      FALSE
##    WW5.txt    bitcoin        can cryptocurr       like       mani        new
##      FALSE       TRUE       TRUE       TRUE       TRUE       TRUE       TRUE
##      still     system        use       also   although     around        one
##       TRUE       TRUE       TRUE       TRUE       TRUE       TRUE       TRUE
##      peopl       list       will      world       high      great    ancient
##       TRUE       TRUE       TRUE       TRUE       TRUE       TRUE       TRUE
##    centuri      seven     wonder
##       TRUE       TRUE       TRUE
```

```r
# Bipartite Network Graph Plot
V(bipartite_graph)$type = bipartite_mapping(bipartite_graph)$type
V(bipartite_graph)$color = ifelse(V(bipartite_graph)$type, "thistle", "slategray1")
V(bipartite_graph)$shape = ifelse(V(bipartite_graph)$type, "circle", "square")
E(bipartite_graph)$color = "lightgray"
plot(bipartite_graph)
```



Each pink nodes in the bipartite network graph above is represented by the terms found within 15 documents whereas the blue nodes represents the 15 documents. The terms might not appear in all the 15 documents. According to the network graph above, it is shown that there is a connection between each of the terms and documents even if the terms originated from different documents. The lines connecting each nodes represents the relationships or degree of similarity between each nodes. The nodes positioned more middle of the network graph indicates that the said node possess more connection to other nodes hence is positioned more central to the network graph. From the graph, it can be seen that there is no clear groups or clusters on the terms.

```
degree_bipartite_graph = as.table(degree(bipartite_graph))
betweenness_bipartite_graph = as.table(betweenness(bipartite_graph))
closeness_bipartite_graph = as.table(closeness(bipartite_graph))
eig_bipartite_graph = as.table(evcent(bipartite_graph)$vector)

averagePath_bipartite_graph = average.path.length(bipartite_graph)
averagePath_bipartite_graph
```

```
## [1] 2.641536
```

```
diameter_bipartite_graph = diameter(bipartite_graph)
diameter_bipartite_graph
```

```
## [1] 6
```

```
tabularised_data_bipartite_graph_df = as.data.frame(rbind(degree_bipartite_graph,
betweenness_bipartite_graph, closeness_bipartite_graph, eig_bipartite_graph))
tabularised_data_bipartite_graph_tbl = t(tabularised_data_bipartite_graph_df)
tabularised_data_bipartite_graph_tbl
```

```
##           degree_bipartite_graph betweenness_bipartite_graph
## CC1.txt                        9                  33.72359552
## CC2.txt                       12                  58.27775558
## CC3.txt                        9                  38.49938166
## CC4.txt                        9                  59.97413110
## CC5.txt                       13                  47.03016983
## GM1.txt                        1                   0.00000000
## GM2.txt                        7                  29.16410779
## GM3.txt                        4                   1.80682455
## GM4.txt                        4                   9.37145910
## GM5.txt                        6                  16.12570763
## WW1.txt                       10                  13.10864968
## WW2.txt                       13                  60.13993196
## WW3.txt                       13                  14.05189533
## WW4.txt                       11                 123.02613481
## WW5.txt                       10                  67.17543861
## bitcoin                        5                   0.00000000
## can                            5                   0.07692308
## cryptocurr                     5                   0.00000000
## like                           7                  41.43062216
## mani                           6                  44.93558549
## new                            5                  13.00172605
## still                          7                  24.11541204
## system                         5                  24.33465864
## use                            6                   7.86713564
## also                           6                  27.70780772
## although                       5                  24.18647627
## around                         7                  63.57234432
## one                            7                  41.15628897
## peopl                          5                  21.35260295
## list                           6                  13.44772450
## will                           5                   8.94478022
## world                          9                  26.00333719
## high                           5                  19.60709176
## great                          5                   2.92692308
## ancient                        5                   2.40039683
## centuri                        5                  14.00694942
## seven                          5                   2.40039683
## wonder                         5                   0.00000000
##           closeness_bipartite_graph eig_bipartite_graph
## CC1.txt                0.010869565         0.754582060
## CC2.txt                0.012048193         0.584720385
## CC3.txt                0.011363636         0.384028241
## CC4.txt                0.010989011         0.153060015
## CC5.txt                0.011627907         0.395801794
## GM1.txt                0.007936508         0.001776643
## GM2.txt                0.011235955         0.018297859
## GM3.txt                0.009433962         0.019122112
## GM4.txt                0.009615385         0.015476437
```

```
## GM5.txt                0.010000000         0.020183666
## WW1.txt                0.010309278         0.044003085
## WW2.txt                0.011904762         0.066338036
## WW3.txt                0.010869565         0.046698319
## WW4.txt                0.012500000         0.028850473
## WW5.txt                0.011627907         0.017282653
## bitcoin                0.006944444         0.286665726
## can                    0.008928571         0.114479076
## cryptocurr             0.006134969         1.000000000
## like                   0.011363636         0.079447800
## mani                   0.012048193         0.046797362
## new                    0.010416667         0.099757150
## still                  0.011627907         0.085081153
## system                 0.010869565         0.079151617
## use                    0.009433962         0.315908506
## also                   0.011904762         0.052748867
## although               0.011494253         0.045192524
## around                 0.011111111         0.048907280
## one                    0.011904762         0.033172986
## peopl                  0.010416667         0.038542511
## list                   0.010638298         0.037591413
## will                   0.010309278         0.038002148
## world                  0.011363636         0.075361596
## high                   0.011111111         0.019204989
## great                  0.008928571         0.020885082
## ancient                0.009433962         0.030101634
## centuri                0.010638298         0.011401224
## seven                  0.009433962         0.028405235
## wonder                 0.008928571         0.042778689
```

Using the data on the bipartite network graph as displayed as a table above, we can know on the degree, betweenness, closeness and EIG as well as the diameter and average path of the graph. The degree indicates the number of connections a node has and nodes with higher degrees. The highest number of degree recorded is 13 with the least is 1. Betweenness measures a node's centrality in the network graph based on the shortest paths that pass through it. A high betweenness is seen with the document `WW4.txt` which have a betweenness value of 123.02613481 suggests the node may be used to act as a bridge within the different nodes in the network graph. Closeness reflects how close a node is to all other nodes and the highest closeness is noted at 0.012500000 for the document `WW4.txt` which indicates that the document `WW4.txt` nodes can quickly interact with others nodes within the network graph. EIG indicates a node's influence based on the connectivity of its neighbors. A high EIG is noted in the word `cryptocurr` with a value of 1.0000000 shows that the node may be connected to many well-connected nodes within the graph. Average path of the network graph have a length of 2.641536 which strongly suggests that, on average, it takes about 2.64 steps to get from one node to another within the network graph. This indicates that there is a relatively high level of connectivity within the nodes of the network graph as nodes can be reached quickly from one another within the network graph. Diameter shows how spread out the nodes is within the network graph and in this network graph, there is a diameter of 6 means that the farthest distance between any two nodes in the network is four steps. It can be observed that the highest betweenness is also the highest closeness. No improvement is done as the network graph is already readable.

References:

Cryptocurrency

- State University of New York. (n.d.). The Basics about Cryptocurrency | CTS. Www.oswego.edu. https://www.oswego.edu/cts/basics-about-cryptocurrency#:~:text=A%20cryptocurrency%20is%20a%20digital
- What Is Cryptocurrency and How Does It work? (2022). Kaspersky. https://www.kaspersky.com/resource-center/definitions/what-is-cryptocurrency
- Ashford, K. (2023, February 16). What Is Cryptocurrency? Forbes Advisor. https://www.forbes.com/advisor/investing/cryptocurrency/what-is-cryptocurrency/
- Cryptocurrency Market News: Ether Fumbles After ETF Nod, Bitcoin Briefly Slides Below $68K. (n.d.). Investopedia. Retrieved May 30, 2024, from https://www.investopedia.com/cryptocurrency-market-news-ether-fumbles-after-etf-nod-pop-bitcoin-slides-below-usd68k-8654452
- PwC. (2023). Making sense of bitcoin and blockchain. PwC. https://www.pwc.com/us/en/industries/financial-services/fintech/bitcoin-blockchain-cryptocurrency.html

Hoyoverse Games

- Oct. 14, T. N. P., 2020, & A.m, 6:55. (2020, October 13). Genshin Impact Review. IGN Southeast Asia. https://sea.ign.com/review/164919/genshin-impact-review
- Best Aventurine build Honkai Star Rail – 5-star domination | ONE Esports. (2024, April 12). Www.oneesports.gg. https://www.oneesports.gg/honkai-star-rail/best-aventurine-build-hsr-light/
- Aeon. (n.d.). Honkai: Star Rail Wiki. https://honkai-star-rail.fandom.com/wiki/Aeon
- Luocha. (n.d.). Honkai: Star Rail Wiki. Retrieved May 30, 2024, from https://honkai-star-rail.fandom.com/wiki/Luocha
- Aventurine. (n.d.). Honkai: Star Rail Wiki. Retrieved May 30, 2024, from https://honkai-star-rail.fandom.com/wiki/Aventurine

Wonders of the World

- The Editors of Encyclopaedia Britannica. (2019). lighthouse of Alexandria | History, Location, & Facts. In Encyclopædia Britannica. https://www.britannica.com/topic/lighthouse-of-Alexandria
- The Editors of Encyclopedia Britannica. (2019). Hanging Gardens of Babylon | History & Pictures. In Encyclopædia Britannica. https://www.britannica.com/place/Hanging-Gardens-of-Babylon
- National Geographic Society. (2022, June 2). Seven Wonders of the Ancient World. Education.nationalgeographic.org; National Geographic Society. https://education.nationalgeographic.org/resource/seven-wonders-ancient-world/
- manishsiq. (2023, January 2). Seven Wonders of the World Names, List, Details 2023. https://www.studyiq.com/articles/seven-wonders-of-the-world/
- 7 Wonders of the World, Name, Location, Pictures, Features. (2024, May 3). https://www.studyiq.com/articles/seven-wonders-of-the-world/#:~:text=The%20world