Assignment3

Student ID: 31240291

Student Name: LIANG DIZHEN

```
> setwd("C:/Users/DavidL/OneDrive/CS/FIT3152/A3/A3_Docs")
> rm(list = ls())
> #install.packages("slam")
> library(slam)
> #install.packages("tm")
> library(tm)
> #install.packages("SnowballC")
> library(SnowballC)
```

Q1

```
> #Q1
> #function to create multiples with 100 word from a long string
> word100File <- function(text, name) {
+   #replace all punctuation with white space
+   text <- gsub("[[:punct:]]", " ", text)
+   #replace more whitespaces with 1 whitespace
+   text <- gsub("\\s+", " ", text)
+   #strsplit() function that splits a string into substrings based
on a
+   #specified delimiter. returns a list of substrings. ased on the
delimiter " " (space)
+   #strsplit return list of character vectors, need to use [[1]] to
 get it as a long string
+   words <- strsplit(text, " ")[[1]]
+   n <- length(words)
+   #create every file with at least 100 words
+   word_limit <- 100
+   #determine how many files to create for this string of text
+   #if the remaining words are less than 100, 1 less file is create
d
+
+   num_files <- floor(n/word_limit)
+
+   for (i in 1:num_files) {
+     #adjust the start for every loop of reading the words from tex
t
+     start = (i-1)*word_limit + 1
+     end <- min(i*word_limit, n)
+     #create file with name BYD with id
+     filename <- paste0(name, i, ".txt")
+     #writeLines() function writes one or more lines of text to a f
ile.
+     #paste() function concatenates the strings in the words[start:
end]vector with a space separator.
+     #collapse argument specifies the separator between concatenate
d strings
+     #words[start:end] as one sub-string with 100 words and write t
o the corresponding file
+     writeLines(paste(words[start:end], collapse = " "), filename)
+   }
+ }
> #electric car company - BYD
> #https://www.automotiveworld.com/news-releases/byd-leading-global-
innovation-in-electric-vehicles-for-a-better-life/
> BYD_text = "Rotterdam, the Netherlands – BYD, the world’s leading
manufacturer of New Energy Vehicles (NEV) and power batteries, has b
een at the forefront of battery technology for over 27 years. Since
its formation, BYD’s battery expertise, and pioneering technological
```

```
 innovations have been empowering the transition to electrification
of transportation across all sectors, and inspiring eMobility on a g
lobal level.
+ The popularity of BYD NEV passenger cars has led to record-breakin
g sales for the company, and this together with BYD pure-electric bu
ses for public transport, as well as pure-electric commercial trucks
 and vans, has been hugely influential in BYD becoming the leading g
lobal NEV manufacturer.
+ BYD new energy vehicles are making a valuable contribution to carb
on reduction, helping to protect the environment through zero-emissi
on solutions. BYD eBuses are transforming public transportation givi
ng commuters, shoppers and tourists the ability to travel on non-pol
luting, zero-emission buses with almost zero noise pollution. Simila
rly, BYD eTrucks and NEV passenger cars are rapidly claiming an ever
-increasing market share in their respective sectors. Alongside thi
s, is also a range of BYD pure-electric forklifts for industrial us
e.
+ This success is built on experience. For over two decades, BYD has
 been inspiring eMobility through innovation in battery technology.
From the outset, BYD has focused exclusively on pure-electric batter
y powered vehicles in its commercial range. Taking this a step furth
er, BYD announced in April 2022 that it would be ceasing production
of full combustion engine vehicles to focus on battery electric (BE
V) and plug-in hybrid (PHEV) vehicles. Significantly, BYD is the fir
st OEM in the world to make such a commitment, supporting its vision
 for a sustainable future, driven by electrification, for a better l
ife.
+ BYD pioneering innovation in Iron-Phosphate Battery Technology
+ BYD has made huge strides in the development of battery technology
 over the last 27 years. This unparalleled expertise has served BYD
well in developing some of the most technologically advanced electri
c vehicles. The successful implementation of BYD new energy vehicles
 is an excellent example of how technological innovation is influenc
ing change, demonstrating the reliability and benefits of electrific
ation.
+ Proven BYD Iron-Phosphate Battery Technology developed for safety
and reliability is at the heart of BYD’s NEV product range. BYD is,
in fact, the largest manufacturer of Lithium Iron-Phosphate (LFP) ba
tteries for which industry data shows there is substantially increas
ed demand. As technology continues to advance, LFP batteries are exp
ected to account for more than 60 per cent of the global power batte
ry market by 2024. There is a good reason. LFP batteries are cobalt
free and produced using a material that has excellent thermal stabil
ity compared to other battery alternatives. As such BYD Iron-Phospha
te Battery Technology has passed stringent safety tests, including c
rush tests, heat tests, overcharging tests, which has even exceeded
regulatory requirements. BYD was one of the first companies to use a
 battery thermal management system, to ensure that the battery tempe
rature remains at the optimum level for efficient and reliable opera
tion in all extremes of weather. Such is the energy efficiency, BYD
NEVs in all categories produce some of the industry’s most impressiv
e ranges.
+ BYD Blade Battery revolutionising the industry".

> word100File(BYD_text, "BYD")
> #https://www.scmp.com/business/china-business/article/3221515/tesl
a-offers-china-made-electric-vehicles-sale-canada
> #Telsa
> Tesla_text = "
+ GLP Park, Lingang, Shanghai. Photo: Handout
+ Business
+
+ H&M shuts Beijing Sanlitun district store a year after closing Sha
nghai shop
+
+
+ The H&M store in Sanlitun. Swire Properties says it is finalising
a rental agreement with a new tenant for the site. Photo: VCG
+ Lifestyle
```

```
+ 
+ The best K-dramas of 2022: Extraordinary Attorney Woo, Little Wome
n and more
+ 
+ Kim Go-eun in a still from Little Women, one of our picks for the
top 15 K-dramas of 2022.
+ Lifestyle
+ 
+ Meet Anita Yuen, the Audrey Hepburn of Hong Kong who crossed Jacki
e Chan
+ 
+ 
+ Actress Anita Yuen at an interview with the Post in 1998. At the h
eight of her success, Yuen garnered a reputation for being difficult
 to work with, but for director Peter Chan she was “so good” on scre
en he put aside doubts about casting her.
+ A total of 4,027 of Tesla’s Model Y and Model 3 electric vehicles
await loading at the Nangang port in Shanghai for shipment to the Po
rt of Zeebrugge in Belgium on May 15, 2022. Photo: VCG via Getty Ima
ges.
+ A total of 4,027 of Tesla’s Model Y and Model 3 electric vehicles
await loading at the Nangang port in Shanghai for shipment to the Po
rt of Zeebrugge in Belgium on May 15, 2022. Photo: VCG via Getty Ima
ges.
+ Tesla is listing China-made Model 3 and Model Y models for sale in
 Canada, the company’s website showed on Tuesday, confirming the ele
ctric car maker has completed its first shipments to North America f
rom its Shanghai factory.
+ Tesla’s website showed both rear-wheel drive Model Y vehicles and
the long-range, all-wheel drive version of the Model 3 available for
 immediate delivery in British Columbia, with codes showing they wer
e manufactured at Tesla’s Gigafactory Shanghai.
+ Both models qualify for federal incentives of C$5,000 (US$3,700) i
n Canada, which, unlike the United States, does not link electric-ve
hicle subsidies to the location of the plant that made the car.
+ Tesla representatives in China and at the company’s headquarters i
n the United States did not immediately respond to requests for comm
ent.
+ China’s EV war: BYD, Nio, Xpeng snap at Tesla’s heels with made-fo
r-China models
+ 13 Apr 2023
+ 
+ The company and other electric car manufacturers have a cost advan
tage in China as exports from that market boom. The China-made versi
on of the Model Y was listed for C$61,990 in Canada. That is about 2
2 per cent more than the equivalent vehicle costs in China before in
centives.
+ 
+ Tesla’s move to export to Canada from Shanghai could help it keep
vehicles made at its plants in California and Texas for sale in the
United States, where they qualify for potential tax incentives of up
 to US$7,500 under the Biden administration’s subsidy programme.
+ ‘The advantages are obvious’: how China’s BYD became the world’s N
o 1 EV maker
+ 19 Apr 2023
+ "
> word100File(Tesla_text, "Tesla")
> #visualcapitalist.com/the-top-10-ev-battery-manufacturers-in-2022/
> #Battery
> Bat_text = "ENERGYThe Top 10 EV Battery Manufacturers in 2022Publi
shed 8 months ago on October 5, 2022
+ By Bruno Venditti
+ Graphics/Design:
+ Sabrina Lam
+  Subscribe to the Elements free mailing list for more like this
+ 
+ Top-10-EV-Battery-Manufacturers-by-Market-Share-2022
+ 
+ The Top 10 EV Battery Manufacturers in 2022
```

```
+ This was originally posted on Elements. Sign up to the free mailin
g list to get beautiful visualizations on natural resource megatrend
s in your email every week.
+
+ The global electric vehicle (EV) battery market is expected to gro
w from $17 billion to more than $95 billion between 2019 and 2028.
+
+ With increasing demand to decarbonize the transportation sector, c
ompanies producing the batteries that power EVs have seen substantia
l momentum.
+
+ Here we update our previous graphic of the top 10 EV battery manuf
acturers, bringing you the world's biggest battery manufacturers in
2022.
+
+ Chinese Dominance
+ Despite efforts from the United States and Europe to increase the
domestic production of batteries, the market is still dominated by A
sian suppliers.
+
+ The top 10 producers are all Asian companies.
+
+ Currently, Chinese companies make up 56% of the EV battery market,
 followed by Korean companies (26%) and Japanese manufacturers (1
0%).
+
+ The leading battery supplier, CATL, expanded its market share from
 32% in 2021 to 34% in 2022. One-third of the world's EV batteries c
ome from the Chinese company. CATL provides lithium-ion batteries to
 Tesla, Peugeot, Hyundai, Honda, BMW, Toyota, Volkswagen, and Volvo.
+ Despite facing strict scrutiny after EV battery-fire recalls in th
e United States, LG Energy Solution remains the second-biggest batte
ry manufacturer. In 2021, the South Korean supplier agreed to reimbu
rse General Motors $1.9 billion to cover the 143,000 Chevy Bolt EVs
recalled due to fire risks from faulty batteries.
+
+ BYD took the third spot from Panasonic as it nearly doubled its ma
rket share over the last year. The Warren Buffett-backed company is
the world's third-largest automaker by market cap, but it also produ
ces batteries sold in markets around the world. Recent sales figures
 point to BYD overtaking LG Energy Solution in market share the comi
ng months or years.
+
+ The Age of Battery Power
+ Electric vehicles are here to stay, while internal combustion engi
ne (ICE) vehicles are set to fade away in the coming decades. Recent
ly, General Motors announced that it aims to stop selling ICE vehicl
es by 2035, while Audi plans to stop producing such models by 2033.
+
+ Besides EVs, battery technology is essential for the energy transi
tion, providing storage capacity for intermittent solar and wind gen
eration.
+
+ As battery makers work to supply the EV transition's increasing de
mand and improve energy density in their products, we can expect mor
e interesting developments within this industry.
+ The car company also plans to debut the luxury brand Yangwang this
 year. The first rollout, the U8 sport utility vehicle, comes with t
ech that independently controls each of the four wheels to boost saf
ety and stability.
+ Prices will range from 800,000 yuan to 1.5 million yuan ($116,000
to $218,000). BYD will follow up by releasing an electric supercar.
+ The Chinese automaker typically has targeted the middle market wit
h vehicles priced from 100,000 to 300,000 yuan. The high-end space i
s largely untrodden territory for BYD, but that is exactly where it
needs to be to take on Tesla's Model X SUV.
+ BYD, which entered the automotive industry in 2003, has honed its
technological prowess by learning from foreign manufacturers. The co
```

```
mpany opened a design center in 2019 at its Shenzhen headquarters an
d recruited top talent, such as former Audi designer Wolfgang Egger.
+ However, BYD likely faces three challenges in its expansion, the f
irst being the fate of China's purchase subsidies for new energy veh
icles. Last year, the company booked 10.4 billion yuan in receipts f
rom those subsidies.
+ The automaker's net profit jumped 450% last year to 16.6 billion y
uan, with subsidies contributing 60% of that according to simple ari
thmetic. But China ended the subsidies in December.
+ BYD's sales network is another factor. If the salespeople and main
tenance staff affiliated with the company do not receive enough trai
ning, it could lead to complaints that injure the brand.
+ A Chinese web portal that collects customer complaints shows an ou
tpouring of grievances against automakers across the board, includin
g BYD, on how delivery times are being communicated as well as the p
rocess for booking test drives.
+ "
> word100File(Bat_text, "Bat")
```

In this assignment, the main area of investigation is electric cars. Three website articles have been chosen for this purpose: one analyzing Tesla (Visual Capitalist, 2022), one analyzing BYD (Automotive World, 2022), and one discussing electric car batteries (Visual Capitalist, 2022). To ensure a certain level of correlation between the final documents, the content of each sub-topic from all three articles is evenly distributed (each has 100 words) into 16 final text files. This approach increases the correlation while maintaining a reasonable level of difference due to the focus on different sub-topics.

Q2

```
> #Q2
> #return back to parent directory to read all files of  A3_Docs dir
ectory as corpus
> setwd("C:/Users/DavidL/OneDrive/CS/FIT3152/A3")
> cname = file.path(".", "A3_Docs") #get the folder path
> #print(dir(cname)) #print all the file names under this directory/
folder
> #get multiple documents from the directory source
> docs = Corpus(DirSource(cname)) #Corpus for multiple documents
```

Instead of converting each document into a text format, the online content is handled by the Word100File function. Word100File is a user-defined function designed to convert the long string of content from different websites into multiple text files for later analysis. Before final conversion, pre-processing is applied to the long string, which includes replacing punctuation with white space and extra white spaces with single white space using the gsub function. The cleaned string is then returned as a list of character vectors, each containing individual words from the long string using strsplit.

After that, 16 text files are created one by one using the paste0 function with 100 words written in order from the clean long string via writeLines + paste function. All these files are stored in a directory named “A3_Docs”. Eventually, file.path is used to get the directory path of all created documents and Corpus(DirSource()) function is used to convert all documents into one corpus.

Q3

```
> #Q3
> #Tokenisation
> #inspect(docs[[5]])
> docs <- tm_map(docs, removeNumbers)
> docs <- tm_map(docs, removePunctuation)
> docs <- tm_map(docs, content_transformer(tolower))
> #function to change target pattern into space
> toSpace <- content_transformer(function(x, pattern) gsub(pattern,
" ", x))
> # Hyphen to space
> #pattern = "-", this is replaced with space
> docs <- tm_map(docs, toSpace, "-")
> #Filter words
> #Remove stop words and white space
> docs <- tm_map(docs, removeWords, stopwords("english"))
> #strip extra white space and leave with pure word
> docs <- tm_map(docs, stripWhitespace)
> #Stem, change each word back to their sterm
> docs <- tm_map(docs, stemDocument, language = "english")
> #Create DTM
> dtm <- DocumentTermMatrix(docs)
> #remove the words that have less than 90% present rate of the docu
ments
> dtms <- removeSparseTerms(dtm, 0.9)
> dim(as.matrix(dtms))
[1]  16 129
> #select top20 frequency tokens
> token_dtm = dtms
> freq <- colSums(as.matrix(token_dtm))
> #order the word by their frequencies
> ord = order(freq)
> #get top20 frequency tokens
> top20_tokens = freq[tail(ord, 20)]
> token_names = names(top20_tokens)
> #DTM to keep the top20 tokens columns
> top20DTM <- dtms[, token_names]
> top20DTM <- as.matrix(top20DTM)
> #export dtms as a csv file since it is now a matrix
> write.csv(top20DTM, "A3_DTM.csv")
> dim(top20DTM)
[1] 16 20
```

	new	subsidi	nev	world	car	shanghai	top	energi	year	china	tesla	technolog	manufactu	market	model	electr	compani	vehicl	byd	batteri
Bat1.txt	0	0	0	0	0	0	3	0	0	0	0	0	3	2	0	1	0	1	0	4
Bat2.txt	0	0	0	1	0	0	2	0	0	0	0	0	3	2	0	0	4	0	0	6
Bat3.txt	0	0	0	1	0	0	0	1	0	0	1	0	1	1	0	0	1	0	1	5
Bat4.txt	0	0	0	2	0	0	0	1	2	0	0	0	0	4	0	1	1	3	1	2
Bat5.txt	0	0	0	0	1	0	0	2	1	0	0	1	0	0	1	0	1	1	0	2
Bat6.txt	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1	3	0
Bat7.txt	1	4	0	0	0	0	0	1	2	2	0	0	0	0	0	0	2	1	2	0
BYD1.txt	1	0	2	1	1	0	0	1	1	0	0	2	1	0	0	2	1	1	4	3
BYD2.txt	1	0	2	0	1	0	0	1	0	0	0	0	1	1	0	1	0	1	6	0
BYD3.txt	0	0	0	1	0	0	0	0	0	0	0	2	0	0	0	2	0	3	5	4
BYD4.txt	1	0	1	0	0	0	0	1	1	0	0	5	1	0	0	1	0	2	5	3
BYD5.txt	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	2	7
Tesla1.txt	1	0	0	0	0	2	1	0	1	0	0	0	0	0	0	0	0	0	0	0
Tesla2.txt	0	0	0	0	0	1	0	0	0	0	2	0	0	0	4	2	0	2	0	0
Tesla3.txt	0	0	0	0	1	2	0	0	0	1	3	0	1	0	5	1	1	1	0	0
Tesla4.txt	0	1	0	0	2	1	0	0	0	5	2	0	1	1	2	2	2	1	1	0

Table 1 DTM Table

As usual, to obtain useful tokens from all documents, all numbers, punctuation and stop words are removed and all words are transformed into lower case. The hyphen is transformed into whitespace and all extra white spaces are stripped. As a result, all words delimited by single white space are returned to be transformed into their own stem.

After that, the corpus is converted into a DocumentTermMatrix and words with less than 90% present rate are removed. Although the number of tokens is dramatically reduced, there are still 129 words left over. Therefore, the DTM matrix is sliced to keep the 20 tokens with highest frequencies after ordering all tokens. The final top20DTM matrix is then saved to an A3-DTM.csv file.

Q4

```
> #Q4
> #Euclidean Distance(similarity)
> elu_matrix = dist(scale(top20DTM))
> fit = hclust(elu_matrix, method = "ward.D")
> #3 topic so 3 clusters
> cutfit = cutree(fit, k = 3)
> plot(fit,hang=-1)

> sort(cutfit)
  Bat1.txt   Bat2.txt   Bat3.txt   Bat4.txt   Bat5.txt   Bat6.txt 
Bat7.txt 
         1          1          1          1          1          1 
       1 
  BYD5.txt Tesla1.txt   BYD1.txt   BYD2.txt   BYD3.txt   BYD4.txt Te
sla2.txt 
         1          1          2          2          2          2 
       3 
Tesla3.txt Tesla4.txt 
         3          3 
> #Consince distance(similarity)
> library(proxy)
> # Create a DocumentTermMatrix
> # Calculate cosine distance
> cos_Matrix <- dist(scale(top20DTM), method = "cosine")
> fit = hclust(cos_Matrix, method = "ward.D")
> cos_cutfit = cutree(fit, k = 3)
> plot(fit, hang = -1)
> sort(cos_cutfit)
  Bat1.txt   Bat2.txt   Bat3.txt   Bat4.txt   Bat5.txt   BYD5.txt 
Bat6.txt 
         1          1          1          1          1          1 
       2 
  BYD1.txt   BYD2.txt   BYD3.txt   BYD4.txt   Bat7.txt Tesla1.txt Te
sla2.txt 
         2          2          2          2          3          3 
       3 
Tesla3.txt Tesla4.txt 
         3          3 
```

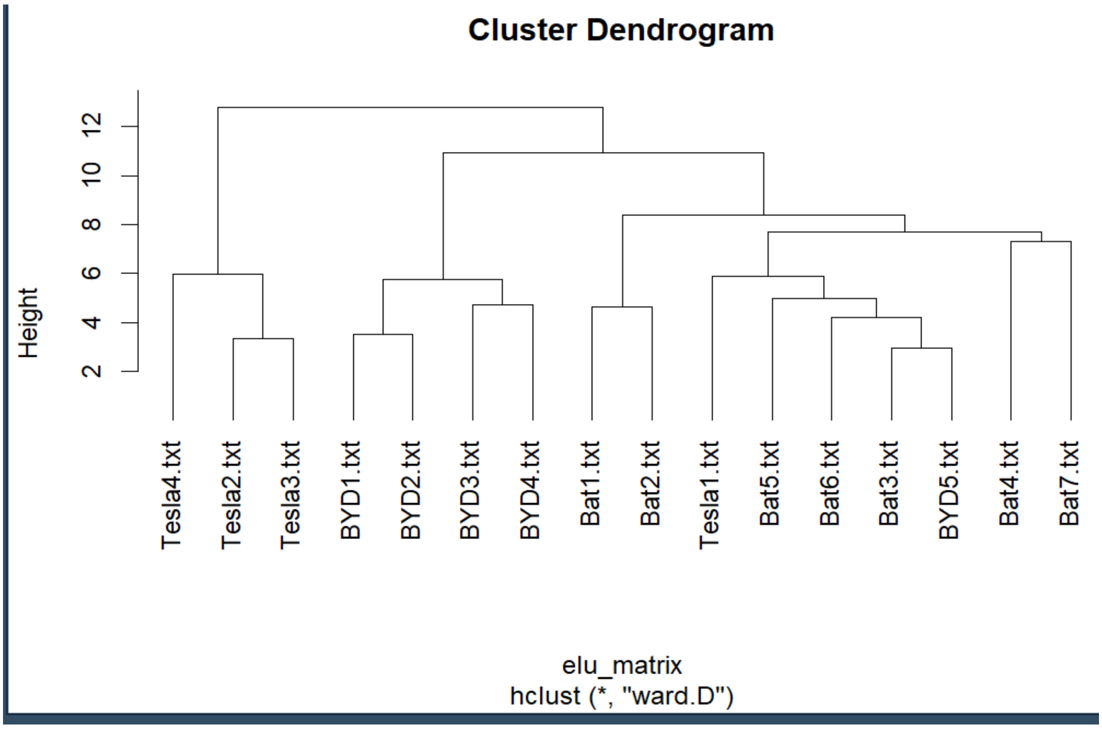


Table 4.1 Cluster Dendrogram with Euclidean Distancing

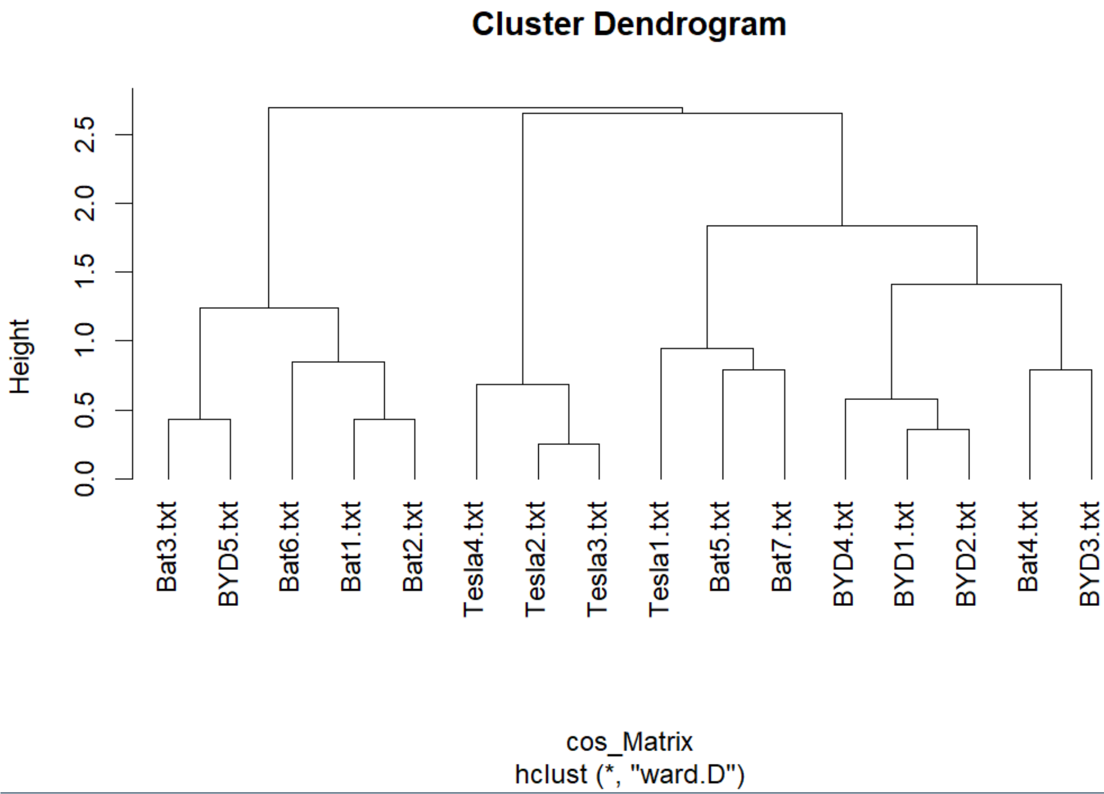


Table 4.1 Cluster Dendrogram with Cosine Distancing

Since the DTM matrix is matrix that have row as documents and column as tokens, with value as element, the DTM matrix can be used for two clustering methods: Euclidean Distancing and Cosine Distancing. Given the similarity between documents are measured in dimensional-space by treating each row of the DTM as a vector of n-dimensions. The values of the vector would be normalised for better machine learning performing, before used to calculate the Euclidean distance and the degree of the Cosine Distancing. Based on those values, the similarity can be calculated, as shorter the Euclidean Distance and lower degree meaning more similar these two documents. Therefore, the clusters can be formed.

Two cluster dendrograms are produced which show difference of clustering among documents. From two plots, hierarchical cluster with Cosine Distancing is relatively flattened. Under consideration of correlations between those documents, hierarchical clustering with Cosine Distancing is relatively more accurate which means under this circumstance Cosine distancing is more accurate than Euclidean Distancing. This fact is also proven by sorted and sliced clusters produced from cutting hierarchical cluster then sorted in-order of cluster ID. Documents with similar names are more clustered into same cluster. Clustering with Cosine Distancing accurately clusters documents while still showing distinctive variety of documents.

Q5

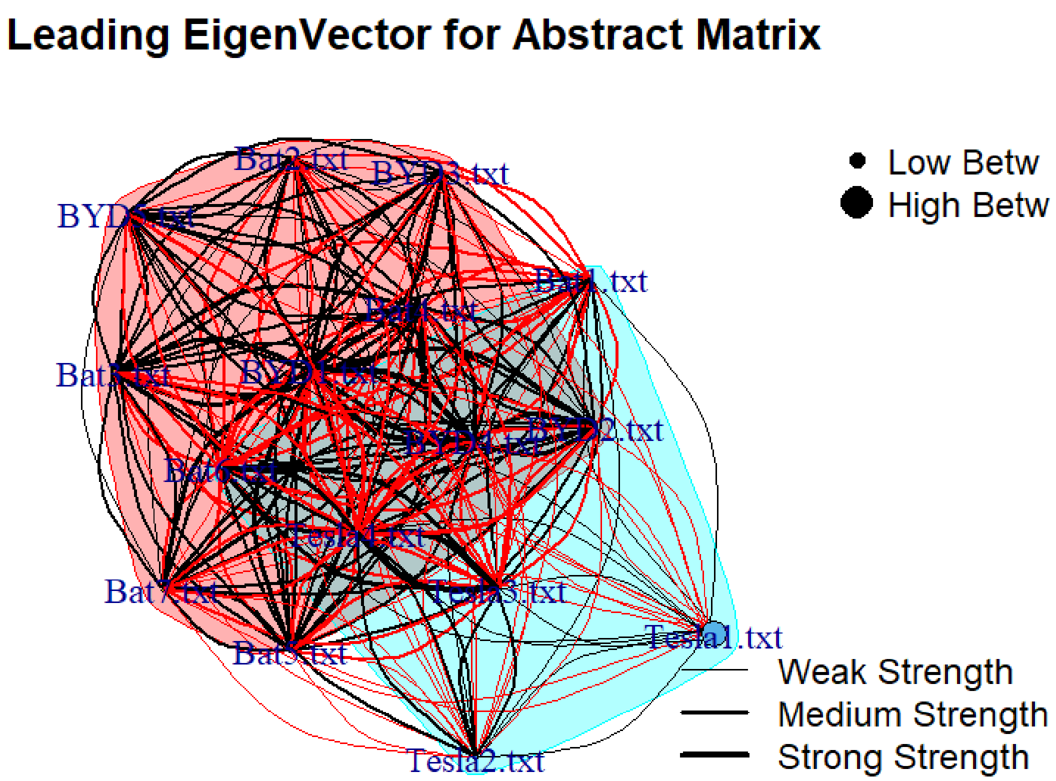
```
> #Q5
> #start with original document-terms matrix
> #convert to binary matrix
> dtms_bin_mat = as.matrix((top20DTM>0)+0)
> #mutliple binary matrix by its transpose
> ByAbsMatrix=dtms_bin_mat%*%t(dtms_bin_mat)
> #head(ByAbsMatrix)
>
> # make leading diagonal zero - remove loop from itself
> #since closeness between one and itself must be closest
> diag(ByAbsMatrix) = 0
>
> par(mfrow=c(1,1))
>
> #create graph object
> library(igraph)
> library(igraphdata)
>
>
> #ByAbs = graph_from_adjacency_matrix(ByAbsMatrix, mode = "undirect
ed", weighted = TRUE)
> #plot(ByAbs)
>
> #show strengh of connection/edge
> #convert to contingency table then dataframe to use $ to get weigh
t of edge
> edges_df = as.data.frame(as.table(ByAbsMatrix))
> colnames(edges_df) = c("u", "v", "weight")
> edges_df <- edges_df[edges_df$weight > 0,]
> #remove loop to itself or useless edges (zero-weight)
>
> #create graph
> Abs_nw = graph_from_data_frame(edges_df, directed = FALSE)
>
> #network statistic
> d = as.table(degree(Abs_nw))
```

```
> #from a list of vertices with their own degree, then convert to a
table
> b = as.table(betweenness(Abs_nw))
> c = as.table(closeness(Abs_nw))
> #Eigeencentrality
> e = as.table(evcent(Abs_nw)$vector)
>
> #bind all those rows
> #4 matrics in the row for each vertex listed in column
> stats = as.data.frame(rbind(d,b,c,e))
> #stats
> #t - transpose to turn row into column
> stats = as.data.frame(t(stats))
> colnames(stats) = c("degree", "betweenness","closeness", "eigenvec
tor")
> #sort and explore key nodes
> #head(stats)
> #node has most hub potential
> #stats[order(-stats$betweenness),][1,]
> stats[order(-stats$betweenness),]
           degree betweenness  closeness eigenvector
Tesla1.txt     24  65.6493490 0.04545455   0.1892375
Tesla2.txt     26  19.6594524 0.03571429   0.3963970
Tesla3.txt     30   8.9934066 0.03125000   0.6703313
Bat2.txt       28   3.9639580 0.03225806   0.4885557
Bat7.txt       30   3.7525253 0.02857143   0.6027139
Bat1.txt       30   1.8208438 0.03125000   0.5703348
BYD5.txt       26   0.1052632 0.02439024   0.4883048
Bat3.txt       28   0.0000000 0.02380952   0.6897891
Bat4.txt       30   0.0000000 0.02777778   0.8277421
Bat5.txt       30   0.0000000 0.02857143   0.6551038
Bat6.txt       30   0.0000000 0.02777778   0.8269798
BYD1.txt       30   0.0000000 0.02173913   1.0000000
BYD2.txt       30   0.0000000 0.02941176   0.7626558
BYD3.txt       28   0.0000000 0.02222222   0.5889215
BYD4.txt       30   0.0000000 0.02272727   0.8280733
Tesla4.txt     30   0.0000000 0.02857143   0.8484793
> #Tesla1 has highest betweeness - 73.8, closeness - 0.04000000, so
the most important
> #Tesla2.txt     28  15.9627106 closeness - 0.02857143
> #BYD5.txt       30  12.4163370 closeness - 0.03125000
>
> stats[order(-stats$closeness),]
           degree betweenness  closeness eigenvector
Tesla1.txt     24  65.6493490 0.04545455   0.1892375
Tesla2.txt     26  19.6594524 0.03571429   0.3963970
Bat2.txt       28   3.9639580 0.03225806   0.4885557
Tesla3.txt     30   8.9934066 0.03125000   0.6703313
Bat1.txt       30   1.8208438 0.03125000   0.5703348
BYD2.txt       30   0.0000000 0.02941176   0.7626558
Bat5.txt       30   0.0000000 0.02857143   0.6551038
Bat7.txt       30   3.7525253 0.02857143   0.6027139
Tesla4.txt     30   0.0000000 0.02857143   0.8484793
Bat4.txt       30   0.0000000 0.02777778   0.8277421
Bat6.txt       30   0.0000000 0.02777778   0.8269798
BYD5.txt       26   0.1052632 0.02439024   0.4883048
Bat3.txt       28   0.0000000 0.02380952   0.6897891
BYD4.txt       30   0.0000000 0.02272727   0.8280733
BYD3.txt       28   0.0000000 0.02222222   0.5889215
BYD1.txt       30   0.0000000 0.02173913   1.0000000
> stats[order(-stats$eigenvector),]
           degree betweenness  closeness eigenvector
BYD1.txt       30   0.0000000 0.02173913   1.0000000
Tesla4.txt     30   0.0000000 0.02857143   0.8484793
BYD4.txt       30   0.0000000 0.02272727   0.8280733
Bat4.txt       30   0.0000000 0.02777778   0.8277421
Bat6.txt       30   0.0000000 0.02777778   0.8269798
BYD2.txt       30   0.0000000 0.02941176   0.7626558
Bat3.txt       28   0.0000000 0.02380952   0.6897891
```

```
Tesla3.txt     30   8.9934066 0.03125000   0.6703313
Bat5.txt       30   0.0000000 0.02857143   0.6551038
Bat7.txt       30   3.7525253 0.02857143   0.6027139
BYD3.txt       28   0.0000000 0.02222222   0.5889215
Bat1.txt       30   1.8208438 0.03125000   0.5703348
Bat2.txt       28   3.9639580 0.03225806   0.4885557
BYD5.txt       26   0.1052632 0.02439024   0.4883048
Tesla2.txt     26  19.6594524 0.03571429   0.3963970
Tesla1.txt     24  65.6493490 0.04545455   0.1892375
> #vector importance, numebr of connection
> stats[order(-stats$degree),]
           degree betweenness  closeness eigenvector
Bat4.txt       30   0.0000000 0.02777778   0.8277421
Bat5.txt       30   0.0000000 0.02857143   0.6551038
Bat6.txt       30   0.0000000 0.02777778   0.8269798
Bat7.txt       30   3.7525253 0.02857143   0.6027139
BYD1.txt       30   0.0000000 0.02173913   1.0000000
BYD2.txt       30   0.0000000 0.02941176   0.7626558
BYD4.txt       30   0.0000000 0.02272727   0.8280733
Tesla3.txt     30   8.9934066 0.03125000   0.6703313
Tesla4.txt     30   0.0000000 0.02857143   0.8484793
Bat1.txt       30   1.8208438 0.03125000   0.5703348
Bat2.txt       28   3.9639580 0.03225806   0.4885557
Bat3.txt       28   0.0000000 0.02380952   0.6897891
BYD3.txt       28   0.0000000 0.02222222   0.5889215
BYD5.txt       26   0.1052632 0.02439024   0.4883048
Tesla2.txt     26  19.6594524 0.03571429   0.3963970
Tesla1.txt     24  65.6493490 0.04545455   0.1892375
>
>
> #create network
> #thicker the connection,
> #plot(Abs_nw, edge.width = E(Abs_nw)$weight)
>
> #clear groups/cluster among documents by community detection
> #create adjacency matrix
> #create community groupings
> #ceb = cluster_edge_betweenness(Abs_nw)
> #cluster_fast_greedy only work on single-edge network
> #cfb = cluster_fast_greedy(Abs_nw)
> #clp = cluster_label_prop(Abs_nw)
> cle = cluster_leading_eigen(Abs_nw)
>
> #plot network
> #scaling
> #install.packages("scales")
> library(scales)
> # Rescale the edge weights to the range [1, 3] to avoid negative w
eighted edge
> E(Abs_nw)$weight <- rescale(E(Abs_nw)$weight, to = c(1, 3))

> #create community in the network
> #plot(ceb, Abs_nw,vertex.label=V(Abs_nw)$role,main="Edge Betweenne
ss")
> #cluster/communities with fast greedy can not work on mutltiple ed
ges
> #plot(cfb, Abs_nw,vertex.label=V(Abs_nw)$role,main="Fast Greedy")
> #plot(clp, Abs_nw,vertex.label=V(Abs_nw)$role,main="Label Propogat
ion")
> plot(cle, Abs_nw,vertex.label=V(Abs_nw)$role,vertex.size = between
ness(Abs_nw)
+      ,edge.width=E(Abs_nw)$weight, main="Leading EigenVector for A
bstract Matrix")
> # Add legend to network
> #node  pt.cex = point size (vertex size), pch = different plotting
 character
> legend("topright",legend = c("Low Betw", "High Betw"),pch = 16
+        ,pt.cex = c(1, 2),bty = "n")
> #strength of connection/edge weight
```

```
> #led to set line width, bty to set the box around plot
> legend("bottomright",legend = c("Weak Strength","Medium Strength",
"Strong Strength")
+         ,lwd = c(1,2,3),bty = "n")
```



Graph 5.1, Single-Mode Network With Leading EigenVector for Abstract Matrix

To create a Single-Mode Network to see the correlations or connections between documents, a Binary Abstract Matrix for documents (ByAbsMat) is created by transforming frequency > 0 in matrix to be 1 (present) and frequency = 0 to be (absent) and performing matrix multiplication between original matrix with transposed matrix. Since loop to itself is not necessary as correlation between one document and itself is 100% correlated or 0 weight of edge/connection distance between itself in network, diagonal value in matrix is set to be 0. Afterward, matrix is converted into a table then a dataframe which contains 3 columns: start vertex/document (U), end vertex/document (V) and weight of edge (weight). Notably, the edge weight is actually the frequency from one document to a token; matrix multiplication sums it up to obtain the frequency between two documents by passing related tokens (since there is either connection between one document to another document via a token (1) or no (0)). Finally, the network is created with clean data frame by removing useless edge (edge.weight = 0). The weight of edge will be later used for variety of edge's width in network for strength of connection.

Since the network is created, all four matrices: degree, betweenness, closeness and Eigen centrality are used to evaluate each vertex (document). Degree is commonly used to ensure relative importance of vertex by measuring how many incoming connections from other vertices to it; higher degree of vertex meaning document is relatively more important. Rest of 3 matrices are for evaluation of centrality of vertex. To better visualize from network, all four corresponding attributes for vertices are bound into one data frame and sorted according to each one of four matrices. As we can see from above 4 tables, degrees are almost same; betweenness instead used to find out central documents as which used to indicate degree of vertex is between other vertices hence Tesla1.txt has 65.65 and Tesla2.txt has 19.65 which substantially higher than others. Moreover, in testing closeness (total distance from one vertex to all other vertices), these two documents also have highest values which former has 0.045 and later has 0.036 but distinctive to other by much less amount.

Therefore, betweenness chosen for variety of vertices' size in network for relative importance. Since communities required to be obtained, among all four different clustering method trials, one with Leading Eigenvector performs best at finding communities from network that vertices have correlation to each other (best clustering method for clustering documents highly correlated to each other). To better visualize, width of edge rescaled to 3 levels. Like mentioned above, betweenness determines size of vertices and frequency determines width of edge and corresponding legends added for reference. From final network there two main communities overlapping with each other (red and blue region); it makes sense with content of documents as battery documents indeed most common to both BYD and Tesla electric car companies.

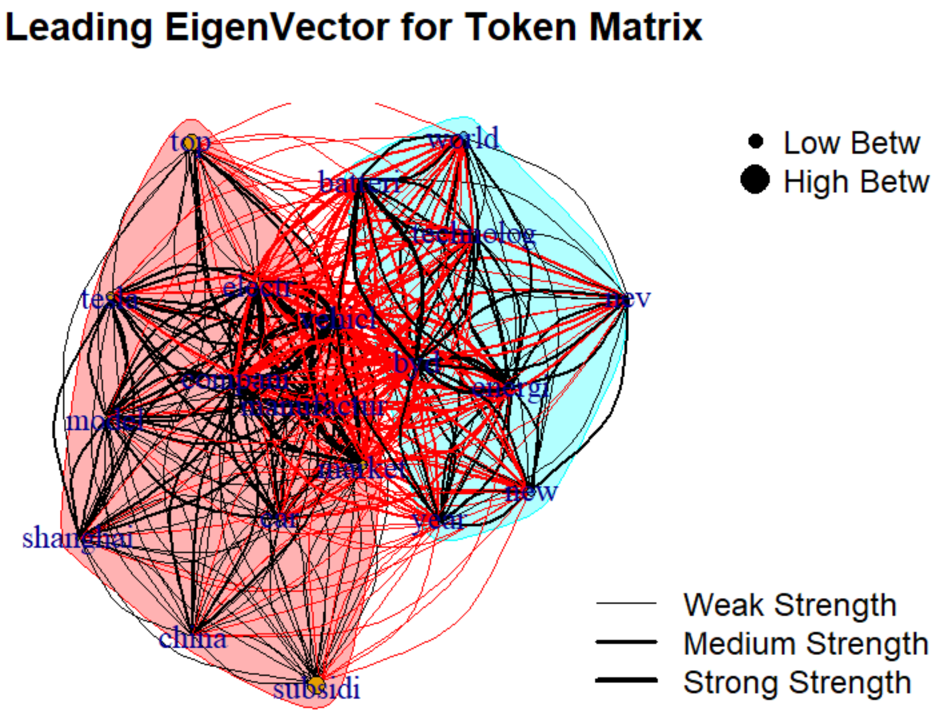
Q6

```
> #Q6
> #convert to binary matrix
> dtms_bin_mat = as.matrix((top20DTM>0)+0)
>
> #Token matrix
> tokenMat = t(dtms_bin_mat)%*%(dtms_bin_mat)
> diag(tokenMat) = 0
>
> #ByAbs = graph_from_adjacency_matrix(ByAbsMatrix, mode = "undirect
ed", weighted = TRUE)
> #plot(ByAbs)
>
> #show strengh of connection/edge
> #convert to contingency table then dataframe to use $ to get weigh
t of edge
> edges_df = as.data.frame(as.table(tokenMat))
> colnames(edges_df) = c("u", "v", "weight")
> edges_df <- edges_df[edges_df$weight > 0,]
> #edges_df
> #remove loop to itself or useless edges (zero-weight)
> #edges_df
>
> #create graph
> token_nw = graph_from_data_frame(edges_df, directed = FALSE)
>
> #network statistic
> d = as.table(degree(token_nw))
> #from a list of vertices with their own degree, then convert to a
table
> b = as.table(betweenness(token_nw))
> c = as.table(closeness(token_nw))
> #Eigeencentrality
> e = as.table(evcent(token_nw)$vector)
```

```
>
> #bind all those rows
> #4 matrics in the row for each vertex listed in column
> stats = as.data.frame(rbind(d,b,c,e))
> #stats
> #t - transpose to turn row into column
> stats = as.data.frame(t(stats))
>
> colnames(stats) = c("degree", "betweenness","closeness", "eigenvec
tor")
> #sort and explore key nodes
> #head(stats)
> #node has most hub potential
> #stats[order(-stats$betweenness),][1,]
> #dim(stats)
> stats[order(-stats$betweenness),]
            degree betweenness  closeness eigenvector
subsidi         28  51.0200855 0.03571429   0.2017103
top             28  24.8854701 0.03225806   0.2646673
tesla           32  13.5642735 0.02941176   0.4455461
market          38   9.1058608 0.02941176   0.6707280
world           30   7.5820513 0.02777778   0.4626428
model           32   7.4822222 0.02777778   0.4416924
nev             26   6.7658730 0.02439024   0.3566401
shanghai        28   6.6848718 0.02941176   0.2784918
new             34   5.5335653 0.03030303   0.4464622
year            36   4.0812698 0.02941176   0.5288263
energi          34   2.2055556 0.02631579   0.6783912
china           28   1.9521368 0.02631579   0.2884073
car             36   0.9076923 0.02564103   0.5226313
batteri         32   0.3611111 0.02222222   0.7363327
technolog       32   0.0000000 0.02439024   0.5593183
manufactur      38   0.0000000 0.02222222   0.8201519
electr          38   0.0000000 0.02272727   0.8799967
compani         38   0.0000000 0.02000000   0.8533453
vehicl          38   0.0000000 0.01754386   1.0000000
byd             38   0.0000000 0.02272727   0.8979633
> stats[order(-stats$closeness),]
            degree betweenness  closeness eigenvector
subsidi         28  51.0200855 0.03571429   0.2017103
top             28  24.8854701 0.03225806   0.2646673
new             34   5.5335653 0.03030303   0.4464622
shanghai        28   6.6848718 0.02941176   0.2784918
year            36   4.0812698 0.02941176   0.5288263
market          38   9.1058608 0.02941176   0.6707280
tesla           32  13.5642735 0.02941176   0.4455461
world           30   7.5820513 0.02777778   0.4626428
model           32   7.4822222 0.02777778   0.4416924
energi          34   2.2055556 0.02631579   0.6783912
china           28   1.9521368 0.02631579   0.2884073
car             36   0.9076923 0.02564103   0.5226313
nev             26   6.7658730 0.02439024   0.3566401
technolog       32   0.0000000 0.02439024   0.5593183
electr          38   0.0000000 0.02272727   0.8799967
byd             38   0.0000000 0.02272727   0.8979633
manufactur      38   0.0000000 0.02222222   0.8201519
batteri         32   0.3611111 0.02222222   0.7363327
compani         38   0.0000000 0.02000000   0.8533453
vehicl          38   0.0000000 0.01754386   1.0000000
> stats[order(-stats$eigenvector),]
            degree betweenness  closeness eigenvector
vehicl          38   0.0000000 0.01754386   1.0000000
byd             38   0.0000000 0.02272727   0.8979633
electr          38   0.0000000 0.02272727   0.8799967
compani         38   0.0000000 0.02000000   0.8533453
manufactur      38   0.0000000 0.02222222   0.8201519
batteri         32   0.3611111 0.02222222   0.7363327
energi          34   2.2055556 0.02631579   0.6783912
market          38   9.1058608 0.02941176   0.6707280
```

```
technolog      32    0.0000000 0.02439024   0.5593183
year           36    4.0812698 0.02941176   0.5288263
car            36    0.9076923 0.02564103   0.5226313
world          30    7.5820513 0.02777778   0.4626428
new            34    5.5335653 0.03030303   0.4464622
tesla          32   13.5642735 0.02941176   0.4455461
model          32    7.4822222 0.02777778   0.4416924
nev            26    6.7658730 0.02439024   0.3566401
china          28    1.9521368 0.02631579   0.2884073
shanghai       28    6.6848718 0.02941176   0.2784918
top            28   24.8854701 0.03225806   0.2646673
subsidi        28   51.0200855 0.03571429   0.2017103
> #vector importance, numebr of connection
> stats[order(-stats$degree),]
           degree betweenness  closeness eigenvector
manufactur     38    0.0000000 0.02222222   0.8201519
market         38    9.1058608 0.02941176   0.6707280
electr         38    0.0000000 0.02272727   0.8799967
compani        38    0.0000000 0.02000000   0.8533453
vehicl         38    0.0000000 0.01754386   1.0000000
byd            38    0.0000000 0.02272727   0.8979633
car            36    0.9076923 0.02564103   0.5226313
year           36    4.0812698 0.02941176   0.5288263
energi         34    2.2055556 0.02631579   0.6783912
new            34    5.5335653 0.03030303   0.4464622
technolog      32    0.0000000 0.02439024   0.5593183
batteri        32    0.3611111 0.02222222   0.7363327
tesla          32   13.5642735 0.02941176   0.4455461
model          32    7.4822222 0.02777778   0.4416924
world          30    7.5820513 0.02777778   0.4626428
subsidi        28   51.0200855 0.03571429   0.2017103
shanghai       28    6.6848718 0.02941176   0.2784918
top            28   24.8854701 0.03225806   0.2646673
china          28    1.9521368 0.02631579   0.2884073
nev            26    6.7658730 0.02439024   0.3566401
> #manufactur    238    158.2729 0.005025126   0.7708475
>
> #create network
> #thicker the connection,
> #plot(Abs_nw, edge.width = E(Abs_nw)$weight)
>
> #clear groups/cluster among documents by community detection
> #create adjacency matrix
> #create community groupings
> #ceb = cluster_edge_betweenness(token_nw)
> cle = cluster_leading_eigen(token_nw)
>
>
> #plot network
> #scaling
> # Rescale the edge weights to the range [1, 3] to avoid negative w
eighted edge
> E(token_nw)$weight <- rescale(E(token_nw)$weight, to = c(1, 3))
> #create community in the network
> #plot(ceb, token_nw,vertex.label=V(token_nw)$role,vertex.size = be
tweenness(token_nw)
> #      ,edge.width=E(token_nw)$weight,main="Edge Betweenness")
> plot(cle, token_nw,vertex.label=V(token_nw)$role,vertex.size = bet
weenness(token_nw)
+      ,edge.width=E(token_nw)$weight, main="Leading EigenVector for
 Token Matrix")
>
> # Add legend to network
> #node  pt.cex = point size (vertex size), pch = different plotting
 character
> legend("topright",legend = c("Low Betw", "High Betw"),pch = 16
+        ,pt.cex = c(1, 2),bty = "n")
>
> #strength of connection/edge weight
```

```
> #led to set line width, bty to set the box around plot
> legend("bottomright",legend = c("Weak Strength","Medium Strength",
"Strong Strength")
+          ,lwd = c(1,2,3),bty = "n")
```



Graph 6.1 Cluster with Leading Eigenvector for Token Matrix

The approach to question 6 is very similar to approach of question 5. Instead of documents, tokens analyzed. This analysis reveals which words highly correlated or coupled to each other considering content from given documents.

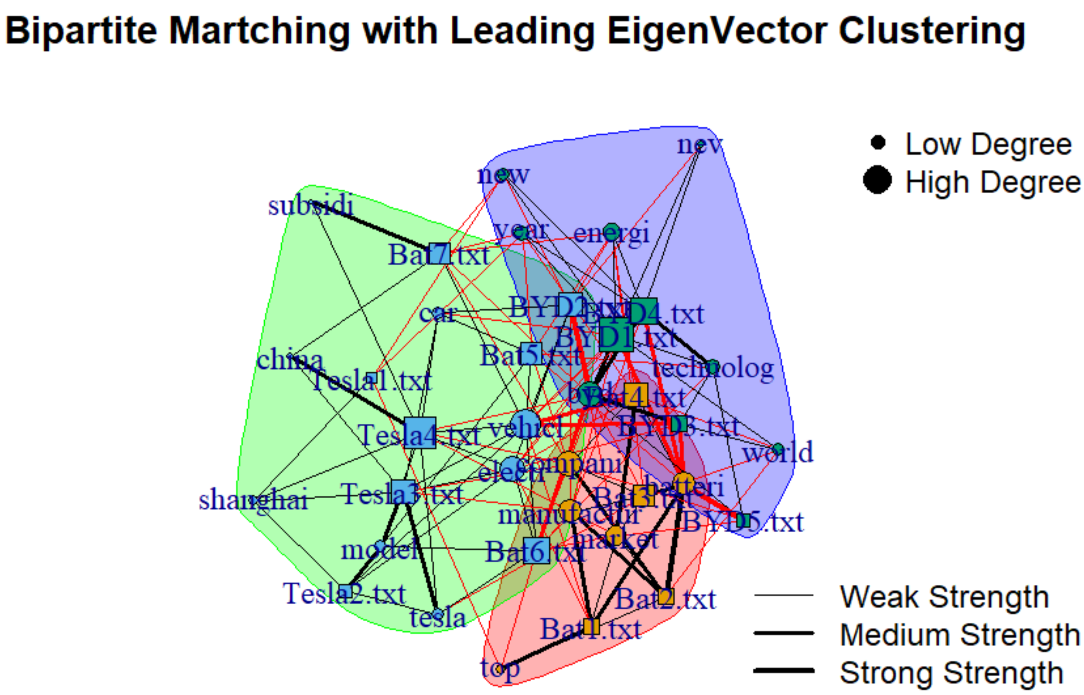
Q7

```
> #Q7
> # start with document term matrix dtms
> #head(top20DTM)
> top20DTM_bipar = as.data.frame(top20DTM)
> #row names to Abs column
> top20DTM_bipar$Abs = rownames(top20DTM_bipar)
> dim(top20DTM_bipar)
[1] 16 21
> dtmsb = data.frame()
>
> for (i in 1:nrow(top20DTM_bipar)){
+   for (j in 1:(ncol(top20DTM_bipar)-1)){
+     #cbind used to bind value like column
+     #bind value with corresponding document name and token in a ro
w
+     touse = cbind(top20DTM_bipar[i,j],top20DTM_bipar[i,ncol(top20D
TM_bipar)]
+                      ,colnames(top20DTM_bipar[j]))
+     #bind as row to a dataset
+     dtmsb = rbind(dtmsb,touse)
+   }
+ }
>
>
> colnames(dtmsb) = c("weight", "abs", "token")
> dtmsc = dtmsb[dtmsb$weight != 0,] # delete 0 weights
> #switch the column to correct position
> dtmsc = dtmsc[,c(2,3,1)]
>
> dtmsc$weight = as.numeric(dtmsc$weight)
> dtmsc$weight = rescale(dtmsc$weight, to = c(1,3))
> dtmsc$weight = format(dtmsc$weight, digits = 0)
> #create bipartite network
> bipar <- graph.data.frame(dtmsc, directed=FALSE)
>
> #network statistic
> d = as.table(degree(bipar))
> #from a list of vertices with their own degree, then convert to a
table
> b = as.table(betweenness(bipar))
> c = as.table(closeness(bipar))
> #Eigeencentrality
> e = as.table(evcent(bipar)$vector)
>
> #bind all those rows
> #4 matrics in the row for each vertex listed in column
> stats = as.data.frame(rbind(d,b,c,e))
> stats = as.data.frame(t(stats))
> colnames(stats) = c("degree", "betweenness","closeness", "eigenvec
tor")
>
> cle = cluster_leading_eigen(bipar)
>
>
> #map bipartite with the graph
> bipartite.mapping(bipar)
$res
[1] TRUE

$type
 [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FAL
SE FALSE
[13] FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TR
UE  TRUE
```

```
[25]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TR
UE  TRUE

>
> #two type, one is token, one is document
> V(bipar)$type <- bipartite_mapping(bipar)$type
> V(bipar)$color <- ifelse(V(bipar)$type, "lightblue", "salmon")
> V(bipar)$shape <- ifelse(V(bipar)$type, "circle", "square")
> E(bipar)$color <- "lightgray"
>
> #plot network
> plot(bipar,edge.width=E(bipar)$weight)
> #scaling
> # Rescale the edge weights to the range [1, 3] to avoid negative w
eighted edge
> #E(bipar)$weight <- rescale(E(bipar)$weight, to = c(1, 3))
> #create community in the network
> plot(cle, bipar,vertex.label=V(bipar)$role,vertex.size = degree(bi
par)
+       ,edge.width=E(bipar)$weight, main="Bipartite Martching with L
eading EigenVector Clustering")
>
> # Add legend to network
> #node  pt.cex = point size (vertex size), pch = different plotting
 character
> legend("topright",legend = c("Low Degree", "High Degree"),pch = 16
+         ,pt.cex = c(1, 2),bty = "n")
>
> #strength of connection/edge weight
> #led to set line width, bty to set the box around plot
> legend("bottomright",legend = c("Weak Strength","Medium Strength",
"Strong Strength")
+         ,lwd = c(1,2,3),bty = "n")
```



Graph 7.1 Bipartite Matching with Leading Eigenvector Clustering

The approach to Question 7 is very similar to that of Questions 6 and 5. Rather than using either document to document or token to token, the document to token with weight of connection in between is considered (frequency). Therefore, the matrix is formed with columns “weight”, “abs” and “token”. Since the original DTM already has all this information, the only modification needed is to rearrange them into the correct format. To achieve this, a new column of original matrix is created to store row names of original matrix (names of documents). For each document and specific token and frequency are horizontally bound with cbind() function. After that, this whole row is added to new blank data frame named “dtmsb” then repeat this process for all frequencies in original matrix. Every processing same with single-mode network afterward except specific bipartite matching function used (bipartite.mapping()) and setting document vertices to be blue square and token vertices to be red circle for differentiation.

In conclusion, by comparing the network and the cluster, the cluster is more suitable for grouping similar documents. However, the network is better to visualise the connection strength between the documents or tokens, and the relative importance of the documents or tokens (central documents/tokens). Trying to find communities/clusters on network is much more difficult than on pure clusters.

Reference

Automotive World. (2022, May 31). BYD - leading global innovation in electric vehicles for a better life. Retrieved from https://www.automotiveworld.com/news-releases/byd-leading-global-innovation-in-electric-vehicles-for-a-better-life/

South China Morning Post. (n.d.). Tesla offers China-made electric vehicles for sale in Canada. Retrieved from https://www.scmp.com/business/china-business/article/3221515/tesla-offers-china-made-electric-vehicles-sale-canada

Visual Capitalist. (2022). The Top 10 EV Battery Manufacturers in 2022. Retrieved from https://www.visualcapitalist.com/the-top-10-ev-battery-manufacturers-in-2022/