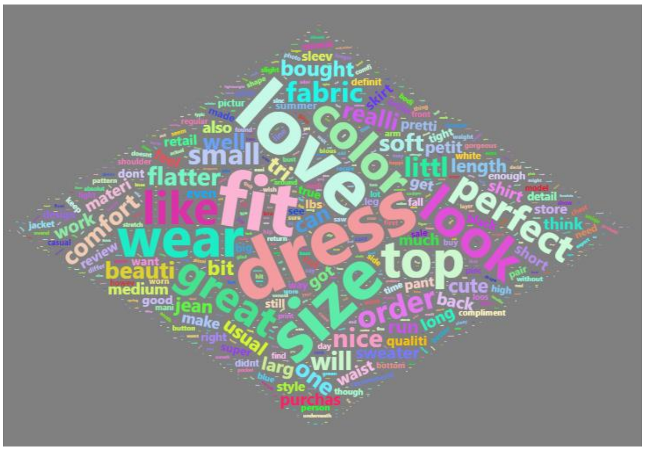
商業分析：SAS / R HW5

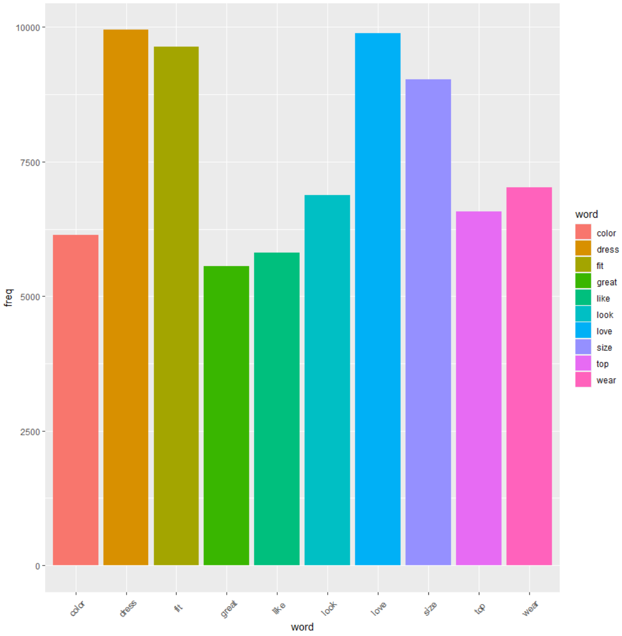
108208004 經濟三 白植允

1. 請用上課的例子 review 資料集。

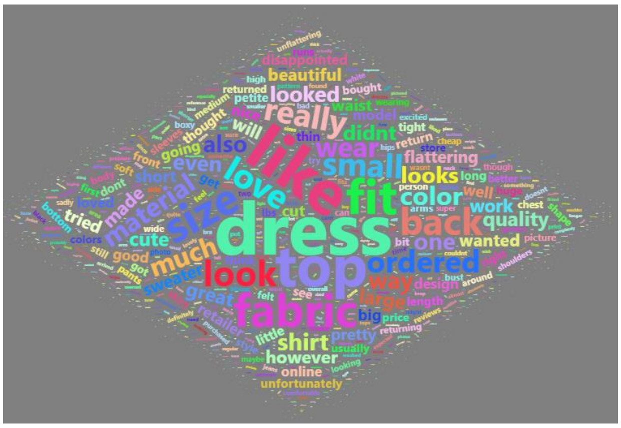
   請將資料分成會推薦及不會推薦來比較，分別做 wordcloud 及直方圖，分析這兩種顧客的留言差異。

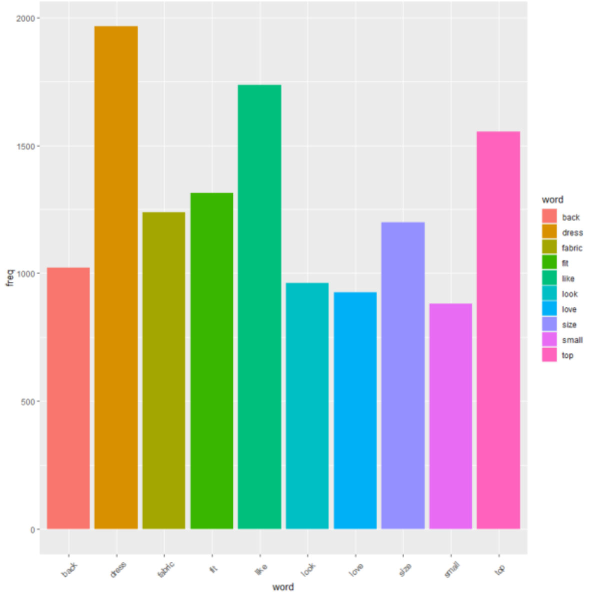
Recommend-



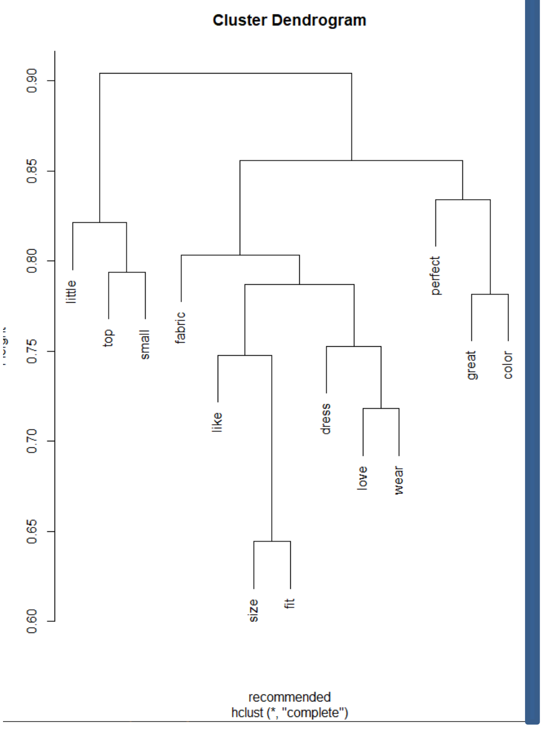


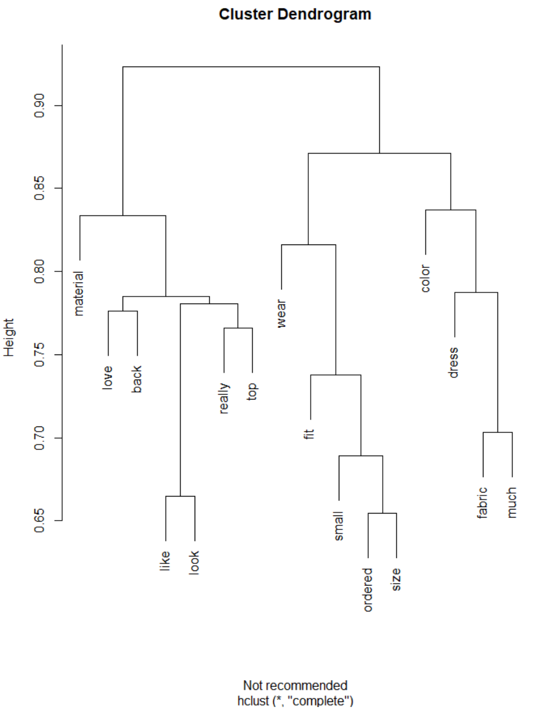
Not recommend –



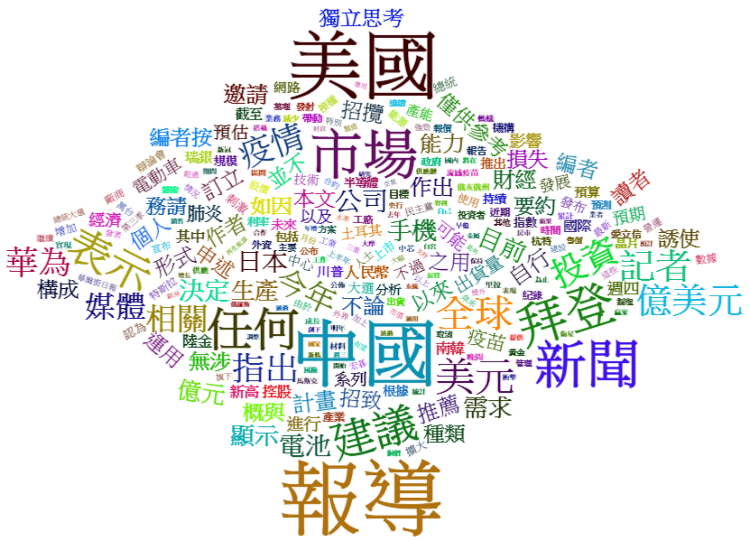


=>最多出現的字都是 dress，可見洋裝是客人最重視的商品，另外，從推薦者的 worldcloud 可推論顧客喜歡的點是 size 合適(size,fit)、顏色好看(look,color,great)，而不推薦的人不喜歡的原因推論可能是材質不好(fabric)，size 不合(size,small)





2. 利用上課或 TA 課(或其他你會的)網路爬蟲方式，任選一筆資料整理，做出 wordcloud。



=>中國跟美國對財經市場影響很大，疫情也有影響，華為最近可能有新動態，大選似乎影響不大

附錄:R 程式碼

```
library(tm)
library(tmcn)
library(devtools)
library(jiebaR)
library(tidyverse)
library(wordcloud2)
library(proxy)
library(rvest)
library(stringr)
library(httr)
library(jsonlite)
library(devtools)
install.packages("RSelenium")
library(RSelenium)
#1
data = read.csv("reviews.csv")
recommend <- data %>%
  filter(Recommended.IND == 1) %>%
  select(Review.Text)
not_recommend <- data %>%
  filter(Recommended.IND == 0) %>%
  select(Review.Text)

x = Corpus(VectorSource(recommend$Review.Text))
x = tm_map(x,removeNumbers)
x = tm_map(x,removePunctuation)
x = tm_map(x,removeWords,c(stopwords("english"),"just"))
x = tm_map(x,tolower)
x = tm_map(x,removeWords,c(stopwords("english"),"just"))
x = tm_map(x,stripWhitespace)

x_tdm <- TermDocumentMatrix(x)
inspect(x_tdm)
x_matrix <- as.matrix(x_tdm)
x_v <- sort(rowSums(x_matrix), decreasing = TRUE)
x_d <- data.frame(word = names(x_v), freq = x_v)

wordcloud2(x_d,size=0.5,color = "random-light",
backgroundColor = "grey",shape = 'diamond')

ggplot(aes(x = word,y = freq,fill = word),data = x_d[1:10,])+
  geom_bar(stat = "identity")+
  theme(axis.text.x = element_text(angle = 45, hjust = 0.5,
vjust = 0.5))


y = Corpus(VectorSource(not_recommend$Review.Text))
y = tm_map(y,removeNumbers)
y = tm_map(y,removePunctuation)
y = tm_map(y,removeWords,c(stopwords("english"),"just"))
y = tm_map(y,tolower)
y = tm_map(y,removeWords,c(stopwords("english"),"just"))
y = tm_map(y,stripWhitespace)

y_tdm <- TermDocumentMatrix(y)
inspect(y_tdm)
y_matrix <- as.matrix(y_tdm)
y_v <- sort(rowSums(y_matrix), decreasing = TRUE)
y_d <- data.frame(word = names(y_v), freq = y_v)

wordcloud2(y_d,size=0.5,color = "random-light",
backgroundColor = "grey",shape = 'diamond')

ggplot(aes(x = word,fill = word),data = y_d[1:10,])+
  geom_bar()+
  theme(axis.text.x = element_text(angle = 45, hjust = 0.5,
vjust = 0.5))

x_tdm2 <- removeSparseTerms(x_tdm, sparse = 0.85)
x_mydata <- as.data.frame(as.matrix(x_tdm2))
xhc <- hclust(d = dist(x_mydata, method = "cosine"), method =
"complete")
plot(xhc,xlab = 'recommended')

y_tdm2 <- removeSparseTerms(y_tdm, sparse = 0.85)
y_mydata <- as.data.frame(as.matrix(y_tdm2))
yhc <- hclust(d = dist(y_mydata, method = "cosine"), method =
"complete")
plot(yhc,xlab = 'Not recommended')

#2
```

```
url <- "http://blog.moneydj.com/news/"
doc <- read_html(url, encoding = "UTF-8")

article.all <- c()
df.all <- data.frame()

for(i in 1:4) {
  url <- paste0("http://blog.moneydj.com/news/", "page/", i)
  doc <- read_html(url, encoding = "UTF-8")
  header <- doc %>%
    html_nodes(".entry-title.mh-loop-title") %>%
    html_nodes("a") %>%
    html_text()
  href <- doc %>%
    html_nodes(".entry-title.mh-loop-title") %>%
    html_nodes("a") %>%
    html_attr("href")
  article.page <- c()
  for(i in 1:length(href)) {
    doc.a <- read_html(href[i])
    article <- doc.a %>%
      html_nodes("div.entry-content.mh-clearfix") %>%
      html_nodes("p") %>%
      html_text() %>%
      str_c(collapse = "")
    article <- ifelse(str_length(article) < 10 ||
rlang:::is_empty(article), NA, article)
    article.all <- append(article.all, article)
  }
  df <- data.frame(title = header, content = article.all) %>%
    na.omit() %>%
    mutate(title = as.character(title),
           content = as.character(content))
}
cc <- worker()
word <-cc[df[,2]]
word_df <- as.data.frame(table(word))
word_df %>%
  filter(!str_detect(word, "[a-zA-Z0-9]+")) %>%
  filter(nchar(as.character(word)) > 1) %>%
  filter( Freq > 10) ->temp
wordcloud2(temp,size = 0.5,color = "random-dark",
backgroundColor = "white",shape = 'diamond')
```