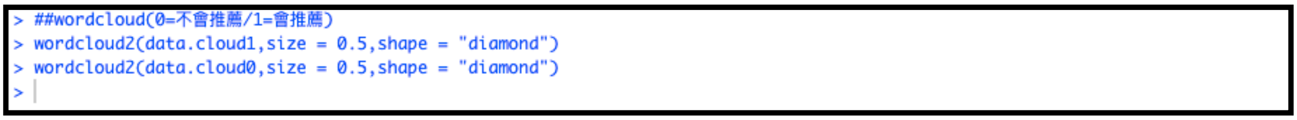
SAS/R商業資料分析作業五

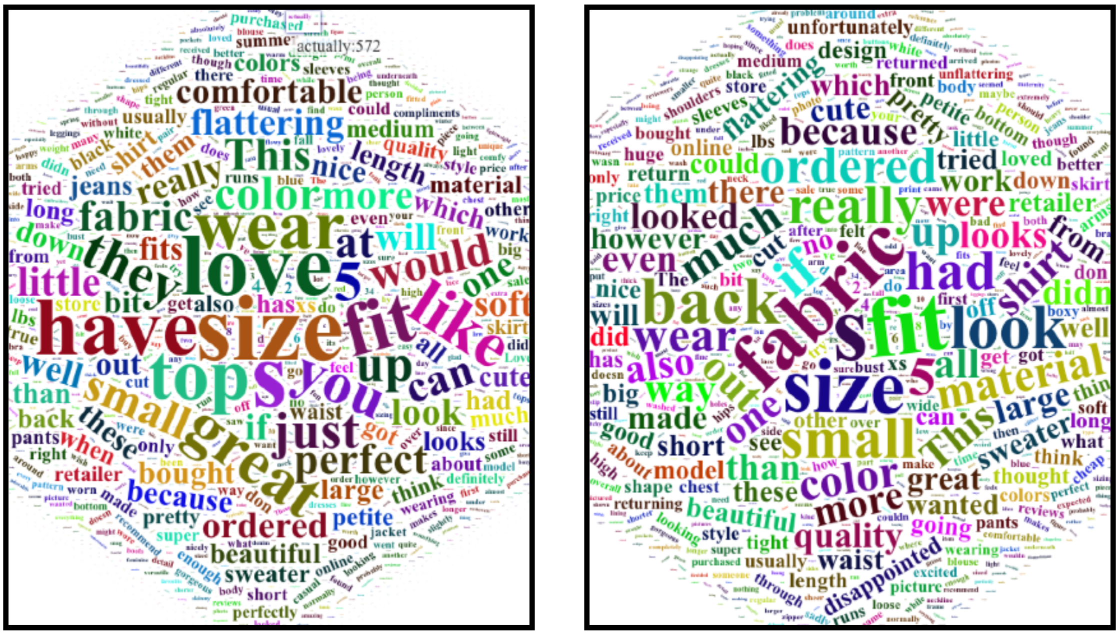
107508006 歐西四 陳葳芃

1. 請用上課的例子review資料集。 Binary variable stating where the customer recommends the product where 1 is recommended, 0 is not recommended.
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。Please compare the review difference between customers who recommended and who not recommended by wordcloud and bar chart.

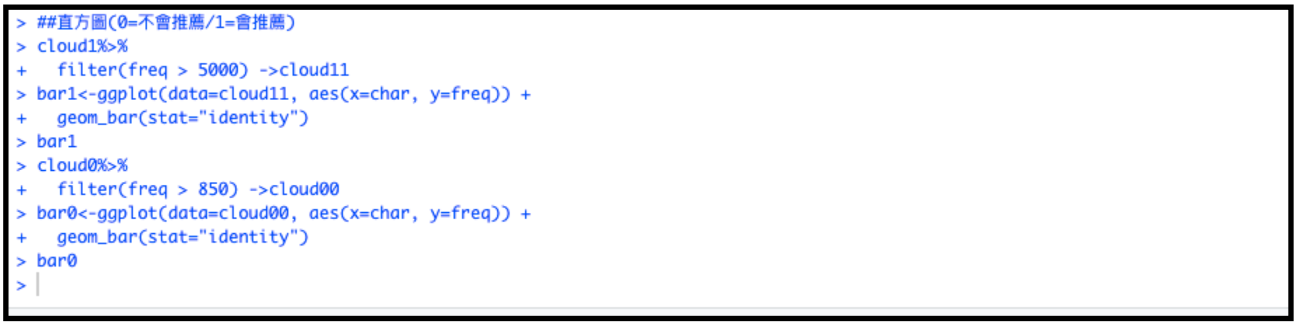
Ans:



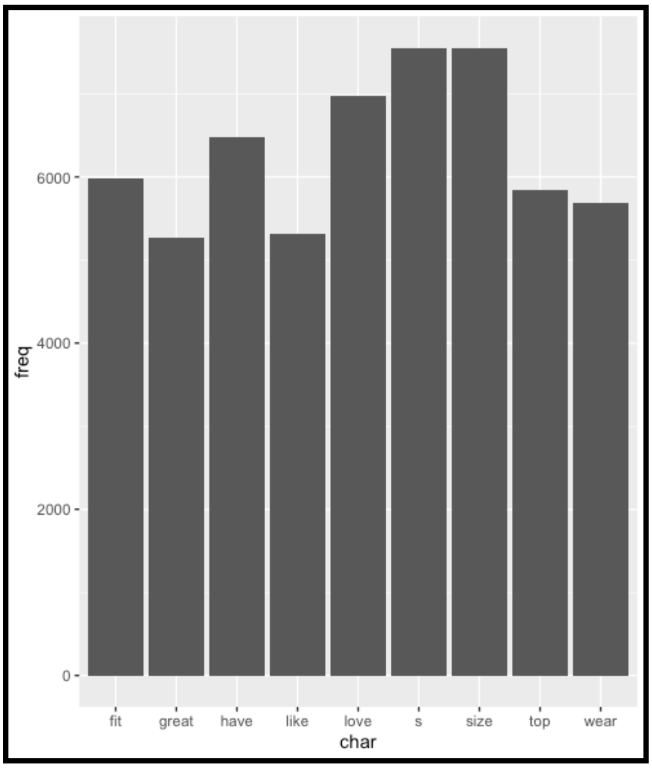
```
> ##wordcloud(0=不會推薦/1=會推薦)
> wordcloud2(data.cloud1,size = 0.5,shape = "diamond")
> wordcloud2(data.cloud0,size = 0.5,shape = "diamond")
> |
```

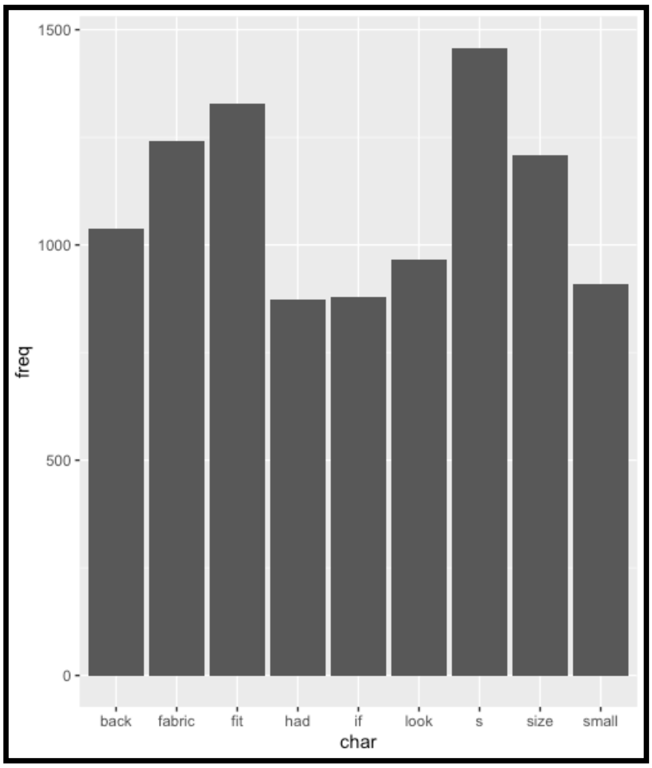


（推薦文字雲/不推薦文字雲）



```
> ##直方圖(0=不會推薦/1=會推薦)
> cloud1%>%
+   filter(freq > 5000) ->cloud11
> bar1<-ggplot(data=cloud11, aes(x=char, y=freq)) +
+   geom_bar(stat="identity")
> bar1
> cloud0%>%
+   filter(freq > 850) ->cloud00
> bar0<-ggplot(data=cloud00, aes(x=char, y=freq)) +
+   geom_bar(stat="identity")
> bar0
> |
```



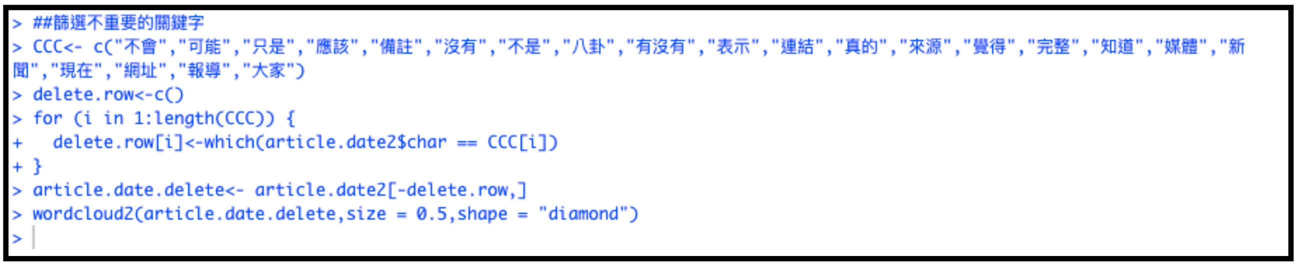


（推薦直方圖/不推薦直方圖）

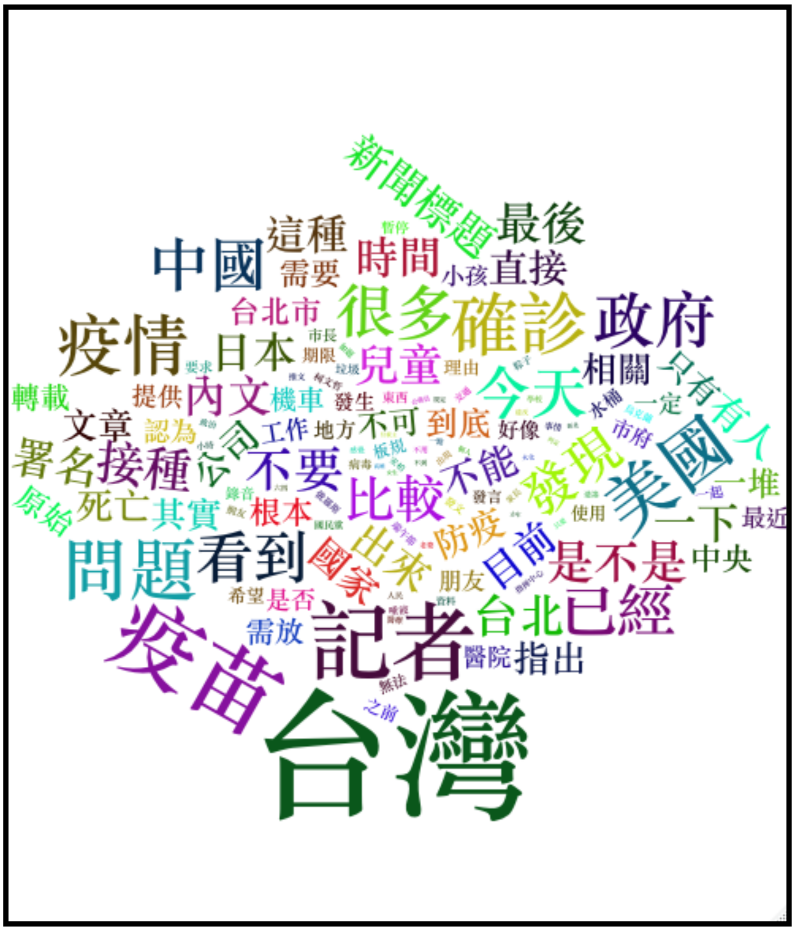
留言分析:我發現如果是正面的評論，明顯字眼會有正面性稱讚(ex:great,perfect)，又可能是以size=s的消費者評論較多；然而負面的評論而言可能會有(fit, look)搭配的問題，或是衣服有material、quaility的問題，同樣是以size=s的消費者評論較多，以及希望問題能獲得改善(if)。

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。

Ans:



```
> ##篩選不重要的關鍵字
> CCC<- c("不會","可能","只是","應該","備註","沒有","不是","八卦","有沒有","表示","連結","真的","來源","覺得","完整","知道","媒體","新
聞","現在","網址","報導","大家")
> delete.row<-c()
> for (i in 1:length(CCC)) {
+   delete.row[i]<-which(article.date2$char == CCC[i])
+ }
> article.date.delete<- article.date2[-delete.row,]
> wordcloud2(article.date.delete,size = 0.5,shape = "diamond")
> |
```



(ptt爬蟲wordcloud)

我使用的方法是ptt爬蟲，我自己則是有跑了300頁的資料，最後分類出來是6月1-4的文章，先用向量刪除了一些(23個)不重要的關鍵字，我們可以發現近期八卦版討論主題是以許多以疫情討論（防疫、疫苗、唾液快篩）議題為大宗，另外有關的可能是中國美國台灣等國家議題。

附錄: R 程式碼

```
#HW5

##1.請用上課的例子review資料集。變數Recommended IND表示客戶是否推薦購買。
#(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。)

setwd("~/Downloads/1102 R/HW/hw 5")
library(readr)
library(tidyverse)
library(devtools)
library(jiebaR)
library(tm)
library(tmcn)
library(jsonlite)
library(wordcloud2)

data <- read.csv("reviews.csv")
str(data)

data$X <- c(1:23486)

##會推薦的data
data1 <- data[which(data$Recommended.IND==1),]
text1 <- as.character(data1$Review.Text)
cc1 <-worker(stop_word = "stop1.txt")
cc1[text1]

count1 <-freq(cc1[text1])  #can also use table(cc[text])
count1

str(count1)

cloud1 = data.frame(count1)
head(cloud1[order(cloud1$freq,decreasing = TRUE),],20)
data.cloud1 = cloud1[order(cloud1$freq,decreasing = TRUE),] #存下排序

##不會推薦的data
data0 <- data[which(data$Recommended.IND==0),]
text0=as.character(data0$Review.Text)
cc0 <-worker(stop_word = "stop0.txt")
cc0[text0]

count0 <-freq(cc0[text0])  #can also use table(cc[text])
count0

str(count0)
```

```
cloud0 = data.frame(count0)
head(cloud0[order(cloud0$freq,decreasing = TRUE),],20)
data.cloud0 = cloud0[order(cloud0$freq,decreasing = TRUE),] #存下排
序

##wordcloud(0=不會推薦/1=會推薦)
wordcloud2(data.cloud1,size = 0.5,shape = "diamond")
wordcloud2(data.cloud0,size = 0.5,shape = "diamond")

##直方圖(0=不會推薦/1=會推薦)
cloud1%>%
  filter(freq > 5000) ->cloud11
bar1<-ggplot(data=cloud11, aes(x=char, y=freq)) +
  geom_bar(stat="identity")
bar1

cloud0%>%
  filter(freq > 850) ->cloud00
bar0<-ggplot(data=cloud00, aes(x=char, y=freq)) +
  geom_bar(stat="identity")
bar0


##########2.利用網路爬蟲方式，任選一筆資料整理，做出wordcloud。
library(tidyverse)
library(rvest)
library(stringr)
library(jiebaR)
library(tmcn)
library(wordcloud2)
jieba.worker <- worker()

ptt.url <- "https://www.ptt.cc"
gossiping.url <- paste(ptt.url,"/bbs/Gossiping",sep = "")
gossiping.url

gossiping.session <- html_session(url = gossiping.url)
gossiping.session

#表單認證
gossiping.form <- gossiping.session %>%
  html_node("form") %>%
  html_form()
gossiping.form

gossiping <- submit_form(
  session = gossiping.session,
  form = gossiping.form,
  submit = "yes")
```

```
gossiping

##開始爬蟲頁碼
page.latest <- gossiping %>%
  html_nodes("a") %>%
  html_attr("href") %>%
  str_subset("index[0-9]{2,}\\.html") %>%
  str_extract("[0-9]+") %>%
  as.numeric()
page.latest

links.article <- NULL
page.length <- 300
for (page.index in page.latest:(page.latest - page.length)) {
  link <- str_c(gossiping.url, "/index", page.index, ".html")
  print(link)
  links.article <- c(
    links.article,
    gossiping %>%
      jump_to(link) %>%
      html_nodes("a") %>%
      html_attr("href") %>%
      str_subset("[A-z]\\.[0-9]+\\.[A-z]\\.[A-z0-9]+\\.html")
  )
}

###連結整理
links.article <- unique(links.article)
head(links.article,20)

##爬蟲
article.table <- tibble() # 建立文章儲存空間
j=0
for (temp.link in links.article) {
  j <- j+1
  print(c(j,length(links.article)))
  article.url <- str_c(ptt.url, temp.link) # 文章網址
  temp.html <- gossiping %>% jump_to(article.url) # 連結至文章網址
  article.header <- try(temp.html %>%
                          html_nodes("span.article-meta-value")
%>% # 開頭部分元素
                          html_text())
  article.author <- article.header[1] %>% str_extract("^[A-z0-9_]
+") # 作者
  article.title <- article.header[3] # 標題
  article.datetime <- article.header[4] # 時間
  article.content <- try(temp.html %>%
                           html_nodes( # 內文部分
                             xpath = '//div[@id="main-content"]/
node()[not(self::div|self::span[@class="f2"])]'
                           ) %>%
                           html_text(trim = TRUE) %>%
                           str_c(collapse = ""))
  article.table <- article.table %>% # 合併文章資料
    bind_rows(
      tibble(
        datetime = article.datetime,
        title = article.title,
        author = article.author,
        content = article.content,
        url = article.url
      )
    )
}
article.table <- article.table %>% # 格式整理清除 NA
  mutate(
    datetime = str_sub(datetime, 5) %>% parse_datetime("%b %d %H:
%M:%S %Y"),
    month = format(datetime, "%m"),
    day = format(datetime, "%d")
  ) %>%
  filter_all(
    all_vars(!is.na(.))
  )

jieba.worker <- worker()

new_user_word(jieba.worker, c("柯文哲","蔡英文","發大財"))

article.date <- article.table %>%
  group_by(day,month) %>% # 以每日做分組
  do((function(input) {
    freq(segment(as.character(input$content), jieba.worker)) %>% #
斷詞後計算詞頻
      filter(
        !(char %in% toTrad(stopwordsCN())), # 過濾 stopword
        !str_detect(char, "[A-z0-9]"), # 過濾英文數字
        nchar(char) > 1 # 過濾單個字
      ) %>%
      arrange(desc(freq)) %>% # 以詞頻排序
      slice(1:100) %>% # 取前 100
      return})(.)) %>%
  ungroup
article.date.words <- freq(article.date$char) %>%
  rename(freq.all = freq)
article.date
article.everyday <- article.date %>%
  left_join( # 比對全部詞
    article.date.words,
    by = 'char'
  ) %>%
  group_by(day,month) %>% # 以每日做分組
  arrange(freq.all) %>% # 每組的詞頻做排序由小到大
  slice(1:5) %>% # 取每組前 5
  summarise( # 合併詞並對詞頻加總
    char = str_c(char, collapse = ", "),
    freq = sum(freq)
  ) %>%
  ungroup
article.everyday

article.everyday %>%  as.data.frame() %>%
  mutate( # 計算月日和頻率排名
    #    month = str_c(format(date, "%m"), "月"),
    #    day = format(date, "%d") %>% parse_number(),
    freq.rank = rank(freq)
  )  %>%
  ggplot() +
  geom_text(
    aes(x = 1,
        y = as.numeric(day),
        label = char,
        color = freq.rank
    ),
    hjust = 1,
    size = 3,
    family="黑體-繁 細體"
  ) +
  geom_text(
    aes(
      x = 0,
      y = as.numeric(day),
      label = as.numeric(day)#format(date, "%d")
    ),
    hjust = 0,
    size = 3,
    alpha = 0.4,
    family="黑體-繁 細體"
  ) +
  scale_color_continuous(low = "#03A9F4", high = "#EF5350") +
  scale_y_reverse() +
  facet_grid( ~ as.numeric(month)) +
  theme_void()
```

```
article.date2 =  article.date %>% group_by(char) %>%
summarise(freq=sum(freq))

##篩選不重要的關鍵字
CCC<-c("不會","可能","只是","應該","備註","沒有","不是","八卦","有沒有","
表示","連結","真的","來源","覺得","完整","知道","媒體","新聞","現在","網址
","報導","大家")

delete.row<-c()

for (i in 1:length(CCC)) {
  delete.row[i]<-which(article.date2$char == CCC[i])
}

article.date.delete<- article.date2[-delete.row,]

wordcloud2(article.date.delete,size = 0.5,shape = "diamond")
```