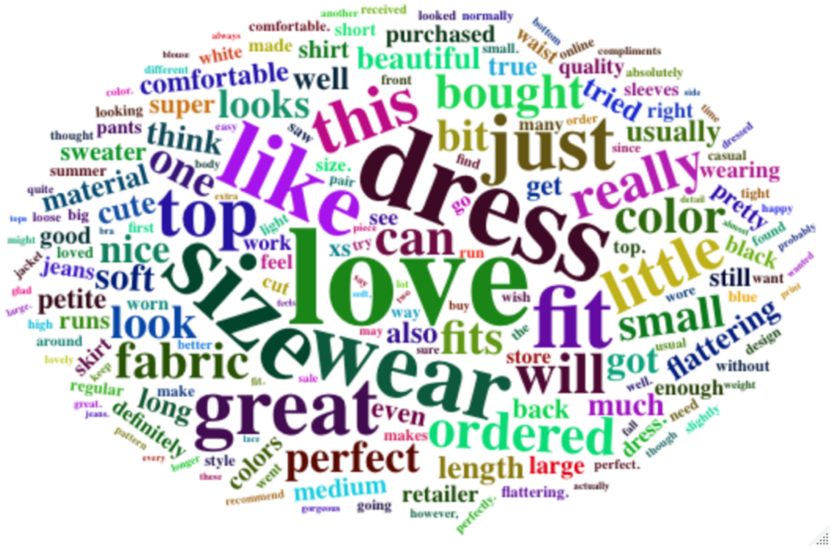
商業分析 HW5

105305072 企管四 許惠甄

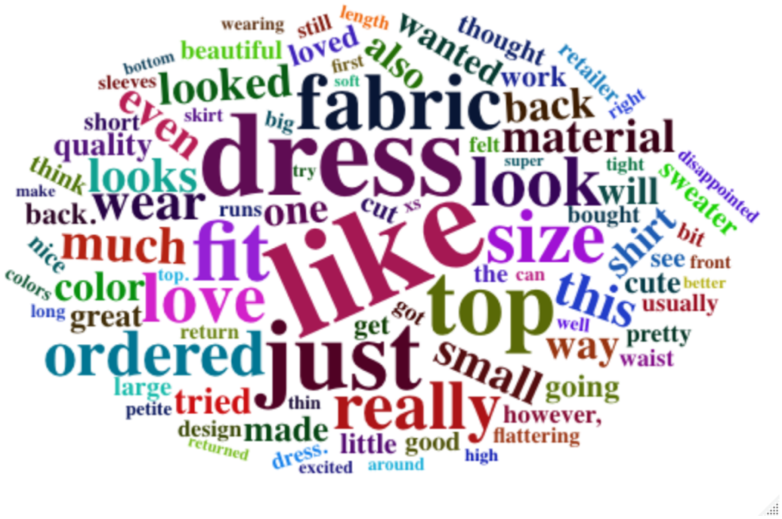
1. 將資料分成會推薦及不會推薦來比較

   a. 做成 Wordcloud

      i. 會推薦



      ii. 不會推薦

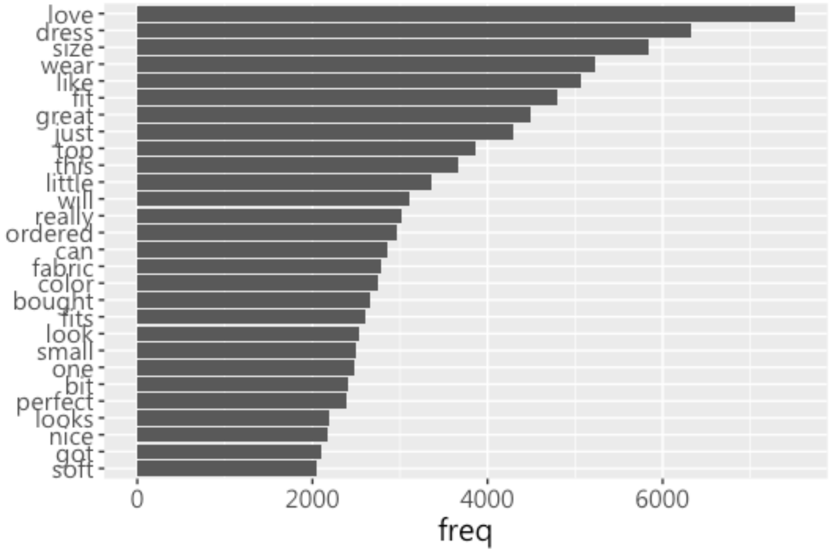


- 會推薦的顧客會留下 Love，但不會推薦的顧客只會留下 like，而在不會推薦的顧客文字雲當中可看見 Fabric 跟 Material 佔了蠻大部分，

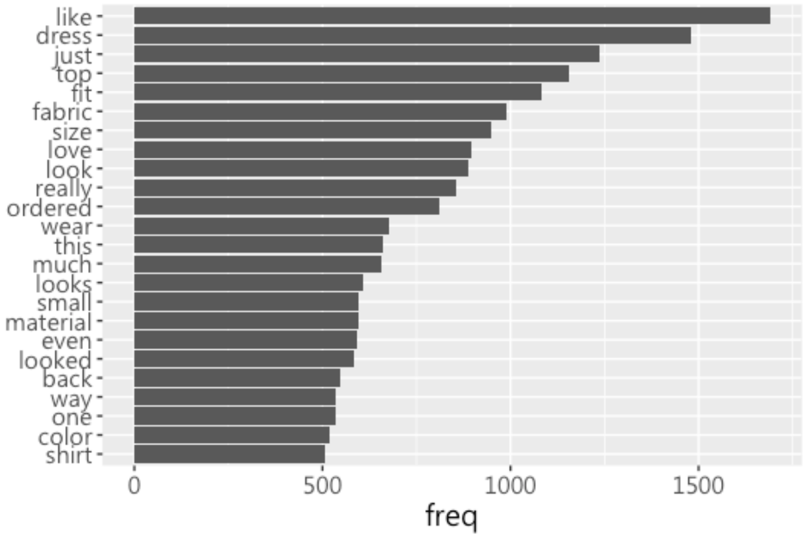
可以後續追蹤是否改善衣服的材質。

b. 做成直方圖

i. 會推薦



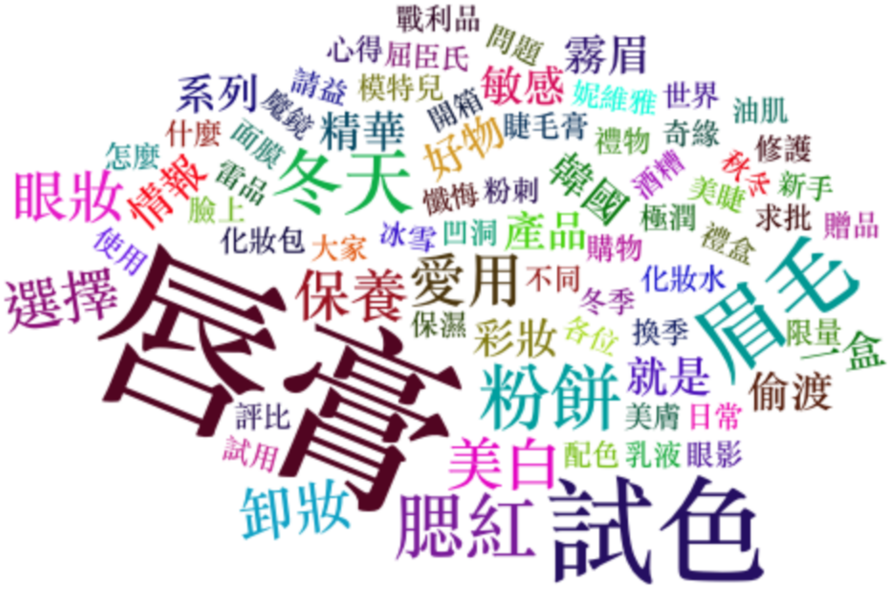
ii. 不會推薦



iii. 會推薦的顧客的留言比較多情緒性字眼，例如：love、perfact、nice 等，但不會推薦的顧客比較針對衣服本身的資訊留言，例如：fabric、size、material 等，店家可以從不會推薦的顧客評論

中找到商品可能的問題點。

2. Dcard 美妝版爬蟲



```r
library(devtools)
library("jiebaR")
library(tm)
library(tmcn)
library('wordcloud2')

data <- read.csv("women_clothes.csv")

#將資料分成兩組
Recommend <- data[data$Recommended.IND == 1,]
Unrecommend <- data[data$Recommended.IND == 0,]

#擷取心得欄
Recommend_words <- Recommend$Review.Text
Unrecommend_words <- Unrecommend$Review.Text


x <- VectorSource(Recommend_words)
x <- VCorpus(x)

myStopWords <- c(stopwords()) #remove some words
x <- tm_map(x, removeWords, myStopWords)
head(myStopWords)

tdm <- TermDocumentMatrix(x, control =list(wordLengths = c(2, Inf)))

m1 <- as.matrix(tdm) #轉Matrix
v <- sort(rowSums(m1), decreasing = TRUE)
d <- data.frame(word = names(v), freq = v) #count freq
new_d <- d[d$freq > 500,]
head(new_d)

wordcloud2(new_d,size=0.5)

extract_d <- d[d$freq > 2000,]
extract_d %>%
  filter(freq > 6) %>%
  mutate(word = reorder(word, freq)) %>%
  ggplot(aes(word,freq))+
  theme(text=element_text(family="微軟正黑體", size=14))+
  geom_col() +
  xlab(NULL) +
  coord_flip()



y <- VectorSource(Unrecommend_words)
y <- VCorpus(y)

myStopWords <- c(stopwords()) #remove some words
y <- tm_map(y, removeWords, myStopWords)
head(myStopWords)

tdm2 <- TermDocumentMatrix(y, control =list(wordLengths = c(2, Inf)))

m2 <- as.matrix(tdm2) #轉Matrix
v2 <- sort(rowSums(m2), decreasing = TRUE)
```

```
d2 <- data.frame(word = names(v2), freq = v2) #count freq
new_d2 <- d2[d2$freq > 200,]
head(new_d2)

wordcloud2(new_d2,size=0.5)

extract_d2 <- d2[d2$freq > 500,]
extract_d2 %>%
  filter(freq > 6) %>%
  mutate(word = reorder(word, freq)) %>%
  ggplot(aes(word,freq))+
  theme(text=element_text(family="微軟正黑體", size=14))+
  geom_col() +
  xlab(NULL) +
  coord_flip()
```

```
library(httr)
library(jsonlite)
library(tidyverse)

options(stringsAsFactors = FALSE)
options(encoding = "UTF-8")

dcardurl <- "https://www.dcard.tw/_api/forums/"
board <- 'makeup'

#把url跟看板融合成一個網址，將抓的順序設定成熱門排序，所以用true
mainurl <- paste0(dcardurl,board,'/posts?popular=true')

# 抽出json，把他存入resdata這個data.frame裡面
resdata <- fromJSON(content(GET(mainurl), "text"))

#先查看前兩個Column
head(resdata[,c(1,2)])

#假設要抓200篇文章
n <- 200

# 因為不改limit值，所以他預設會每次抓20篇回來，我們把要抓的文章/20便是我們要抓的次數
# 還要再減一，因為我們一開始就先抓了前20筆
page <- (200/20)-1

#抓到的最後一篇文章id
end <- resdata$id[length(resdata$id)]

#寫一個loop，重複做page次
for(i in 1:page){
  # 從「目前抓到的最後一篇文章id」往前抓20篇
  url <- paste0(mainurl,"&before=",end)
  # 測試時可以把url印出來檢查有沒有抓對
  print(url)
  # 把抓到存入暫存的tmpres，這只是暫存
  tmpres <- fromJSON(content(GET(url), "text"))

  # 從tmpres裡更新「最後一篇文章的id」
  end <- tmpres$id[length(tmpres$id)]

  # 然後把我們新抓到的tmpres和之前已經有的resdata合併
  resdata <- bind_rows(resdata[,c(1:12)],tmpres[,c(1:12)])
}

#省記憶體
rm(tmpres)

#查看前幾筆
head(resdata)

cc<-worker()
count <-table(cc[resdata[,2]])

newd = data.frame(count)
```

```
head(newd[order(newd$Freq,decreasing = TRUE),],20)
newdd = newd[order(newd$Freq,decreasing = TRUE),]
wordcloud2(newdd)

word <- cc[resdata[,2]]
newd = data.frame(table(word))

newd %>%
  filter(!str_detect(word, "[a-zA-Z0-9]+")) %>%  #去掉english and number
  filter(nchar(as.character(word)) > 1) %>% #一個字的去掉
  filter( Freq > 1) ->temp #可留下頻率>某數字

wordcloud2(temp)
```