

任選 2 部電影，利用文字探勘方式探討評論，畫出兩部電影評論的文字雲，比較不同滿意指標或不同分數的評論文字情緒。

```
# 載入要用到的套件
library(tm)
# install.packages("proxy")
library(proxy)
# install.packages("tidytext")
library(tidytext)
library(wordcloud2)
library(syuzhet)

# 載入檔案
setwd("C:\\Users\\ASUS\\Desktop\\五 234 R\\HW5") #放你的路徑
data <- read.csv("IMDb_Feature Film_2022_review_data.csv")
data <- na.omit(data)
```

選兩部分數、評論數較為相似的电影：**Scream & The Adam Project**

```
# 第一部電影 Scream
data_s <- subset(data, Title=="Scream")

# 創建 Corpus
x <- Corpus(VectorSource(data_s$Review))

# 清理文本
x <- tm_map(x, content_transformer(tolower))
x <- tm_map(x, removePunctuation)
# 針對 Scream 刪掉一些評論中常出現但無含意的字詞
myStopWords <- c(stopwords("english"), "just", "even", "will", "much",
  "also", "still", "one", "can", "really") #remove words
x <- tm_map(x, removeWords, myStopWords)

# 建立 Term-Document Matrix
x_tdm <- TermDocumentMatrix(x)

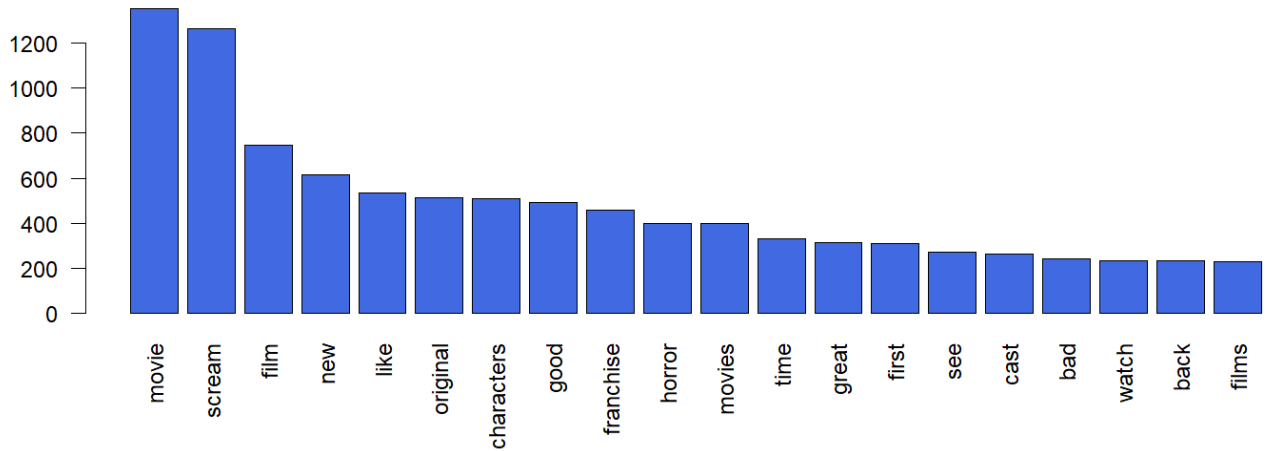
# 轉換為矩陣
review_mx <- as.matrix(x_tdm)

# Sum rows and frequency data frame
freq_dfx <- rowSums(review_mx)

# Sort term_frequency in descending order
freq_dfx <- sort(freq_dfx, decreasing = T)
```

```
# View the top 20 most common words
```

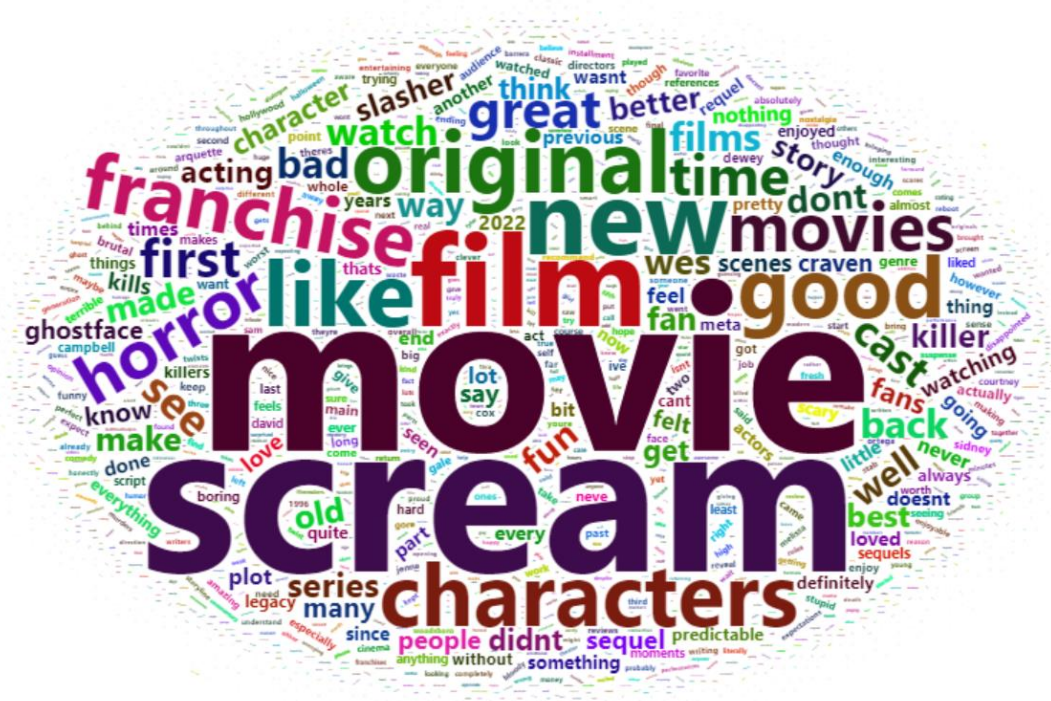
```
barplot(freq_dfx[1:20], col = "royalblue", las = 2)
```



```
freq_dfx <- data.frame(word = names(freq_dfx), num = freq_dfx)
```

```
# 生成文字雲
```

```
wordcloud2(freq_dfx, size = 0.9)
```



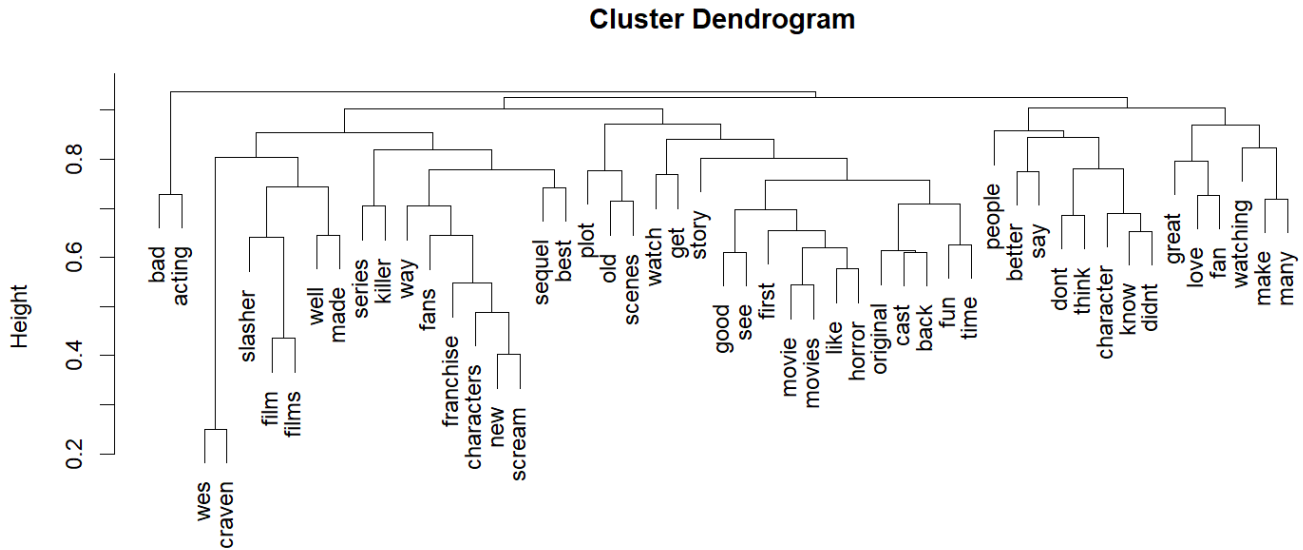
```
# 計算相似度
```

```
x_tdm2 <- removeSparseTerms(x_tdm, sparse = 0.9)
```

```
mydata_x <- as.data.frame(as.matrix(x_tdm2))
```

```
hc_x <- hclust(d=dist(mydata_x, method="cosine"), method="complete")
```

```
plot(hc_x)
```



```
dist(mydata_x, method = "cosine")
hclust (*, "complete")
```

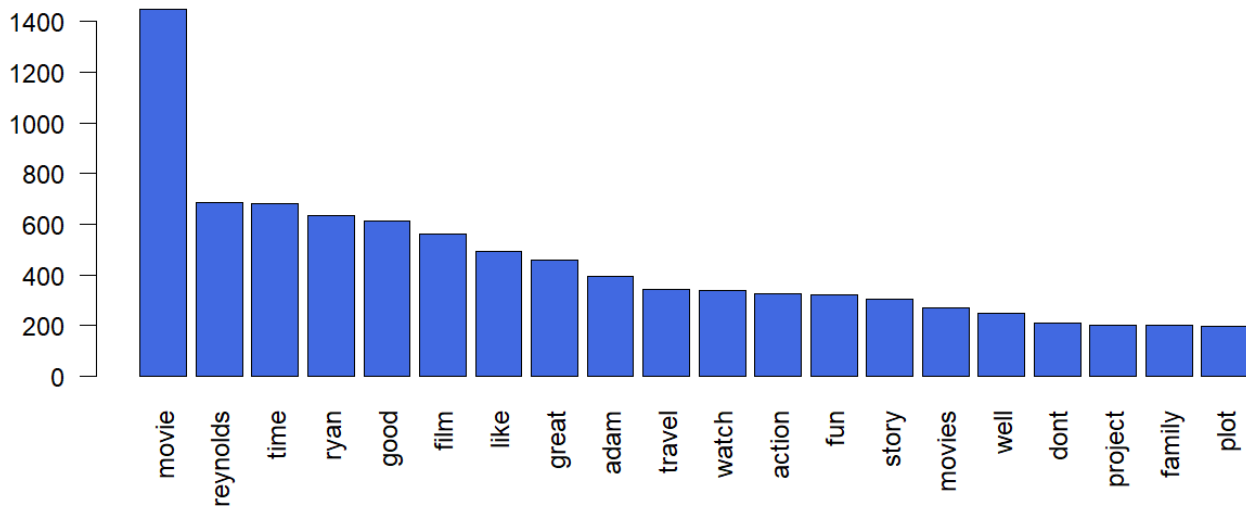
```
#sentiments
get_sentiments("bing")
bing_word_countsx <- freq_dfx %>%
  inner_join(get_sentiments("bing"))
bing_word_countsx
table(bing_word_countsx$sentiment)
bing_word_countsx %>%
  filter(sentiment == "positive") %>%
  select(word,num)%>%
  wordcloud2()
```



```
# 第二部電影 The Adam Project

data_a <- subset(data, Title=="The Adam Project")
y <- Corpus(VectorSource(data_a$Review))
y <- tm_map(y, content_transformer(tolower))
y <- tm_map(y, removePunctuation)
myStopWords <- c(stopwords("english"), "just", "can", "really", "will",
  "still", "one", "every", "even", "much" )
y <- tm_map(y, removeWords, myStopWords)
y_tdm <- TermDocumentMatrix(y)
review_my <- as.matrix(y_tdm)
freq_dfy <- rowSums(review_my)
freq_dfy <- sort(freq_dfy, decreasing = T)
```

```
barplot(freq_dfy[1:20], col = "royalblue", las = 2)
```

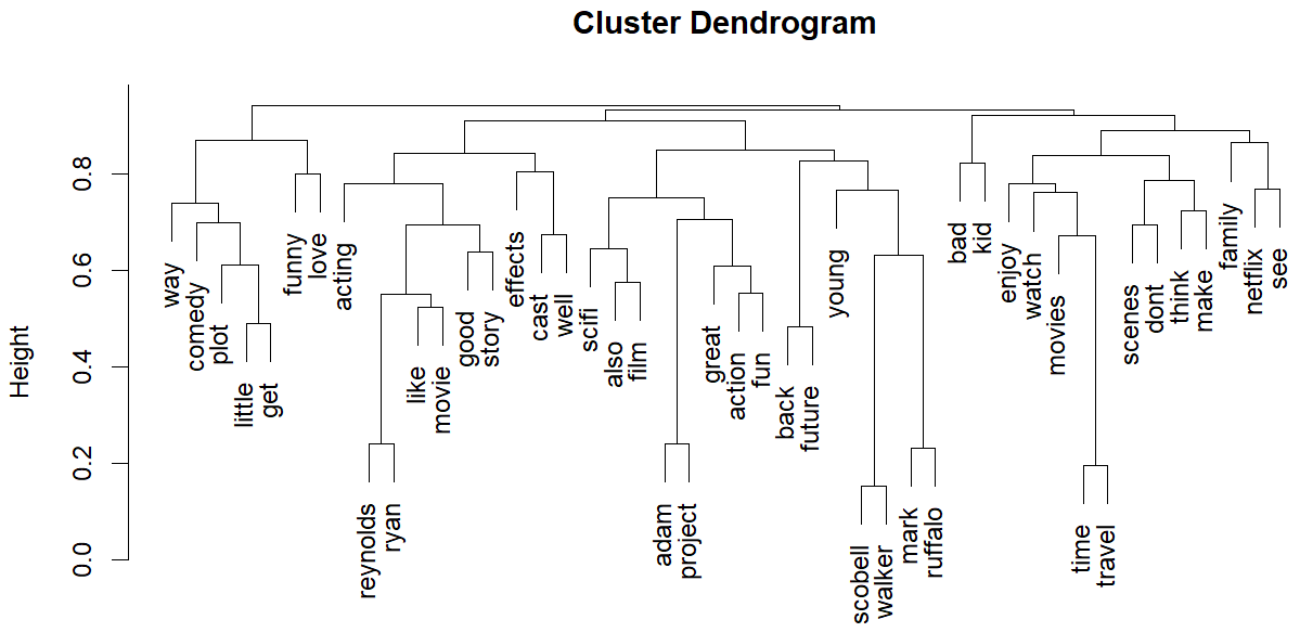


```
freq_dfy <- data.frame(word = names(freq_dfy), num = freq_dfy)
wordcloud2(freq_dfy, size = 0.9)
```



```
y_tdm2 <- removeSparseTerms(y_tdm, sparse = 0.9)
mydata_y <- as.data.frame(as.matrix(y_tdm2))
hc_y <- hclust(d = dist(mydata_y, method = "cosine"), method =
"complete")
```

```
plot(hc_y)
```



```
dist(mydata_y, method = "cosine")
hclust (*, "complete")
```

```
bing_word_countsy <- freq_dfy %>%
  inner_join(get_sentiments("bing"))
bing_word_countsy
table(bing_word_countsy$sentiment)
bing_word_countsy %>%
  filter(sentiment == "positive") %>%
  select(word,num)%>%
  wordcloud2()
```



再進一步分析 Review Sentiment 和 Satisfaction Index 的關係

第一部電影

```
# 執行情感分析，將情感分數添加到 movie_data
```

```
data$s$Sentiment Score <- get sentiment(data $Review, method = "afinn")
```

第二部電影

```
data_a$Sentiment_Score <- get_sentiment(data_a$Review, method = "afinn")
```


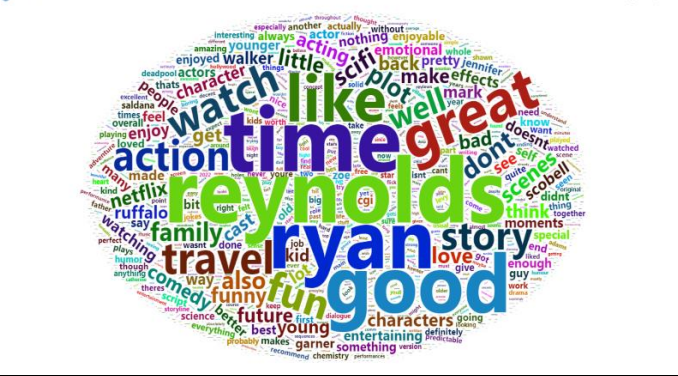
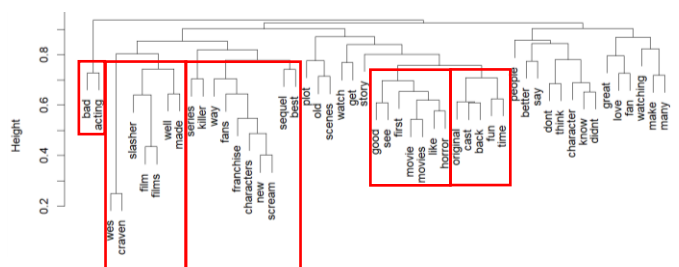
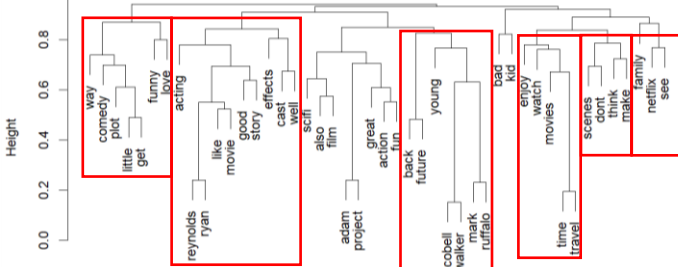


```
# 繪製 Review 和 Satisfaction_Index 的散點圖
```

```
plot(data_s$Sentiment_Score, data_s$Satisfaction_Index,  
      xlab = "Sentiment Score", ylab = "Satisfaction Index",  
      main = "Review Sentiment vs. Satisfaction Index",  
      xlim = c(-30, 80), ylim = c(-6, 4))  
points(data_a$Sentiment_Score, data_a$Satisfaction_Index, col = "red")
```



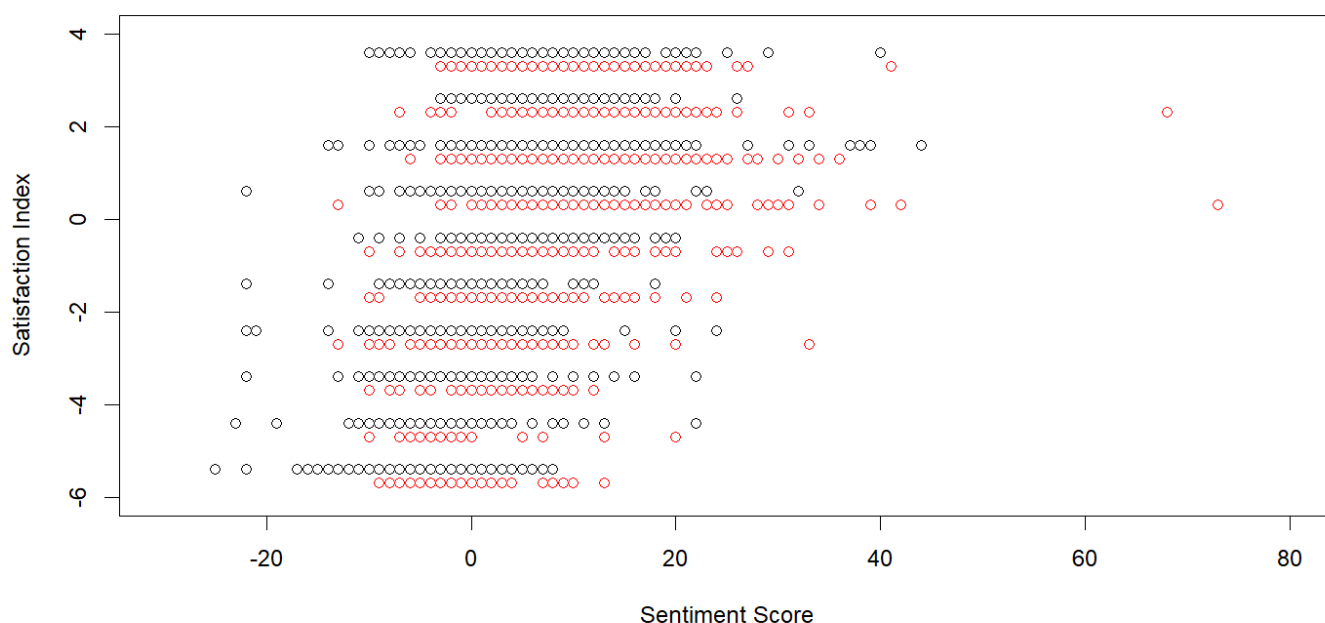
針對兩部電影文字雲的部分，將"movie"和"film"等代表電影的字彙刪除（經常出現在評論中，但不具正負面意涵），並將電影名稱中包含的字彙刪除（Scream、Adam、Project），再重新生成一次文字雲（只做文字雲的部分，不做後續分群及情緒分析），以便做進一步分析

Scream	The Adam Project
	
<p>是一個系列電影（franchise、series），但具有其原創性（new、original）且是一部有殺手和鬼臉出現的恐怖電影（killer、ghostface、horror）</p>	<p>應該有萊恩雷洛斯參演（ryan、reynolds），並且是一部關於穿越時空的科幻電影（travel、time、future、scifi），能在網飛（netflix）上看到</p>
<p>兩部電影皆為正評居多（like、good、great、fun、enjoy、love、well）但仍帶有些負評（bad），與其電影分數的狀況相符（Scream 6.4、The Adam Project 6.7），此外還可看出對評論者來說電影卡司（character）很重要，在兩部影評中都大量被提及。</p>	
	
<ol style="list-style-type: none"> 1. 部份的人認為演技不好 2. 衛斯克萊文製作這部跟鐮刀有關的電影受好評 3. 被認為是 Scream 系列電影最好的續集 4. 是留言者看或喜歡的第一部恐怖電影 5. 前面系列的演員有回歸演出 	<ol style="list-style-type: none"> 1. 觀影者喜歡這部有趣的喜劇電影 2. 喜歡這部電影的人是因為萊恩雷洛斯的演出及好故事 3. Walker Scobell 和 Mark Ruffalo 出演的角色與時間旅行（back、future）有關 4. 喜歡這部時空旅行的電影 5. 覺得不合理 6. 可以和家人在 netflix 上觀看



評分相當的兩部電影，文字情緒也相當，甚至常出現的字彙也差度多（good、like、great、well、love、fun、instersting、awesome）

Review Sentiment vs. Satisfaction Index



從散布圖可以看出 Satisfaction 越低的評論 Sentiment 越低，兩者間大致呈現正相關。再分析兩部電影的結果，明顯第一部電影的 Sentiment score 偏左，散佈在-30~50 之間。而第二部電影的 Sentiment score 較偏向右邊，散佈範圍在-20~80 之間，右偏極值較多，Satisfaction 低的評論者也明顯較少。