中文斷詞與詞語統計

Sung-Chien Lin

2018 年 9 月 20 日

課程簡介

課程內容

- 以聯合報歷史新聞的生活新聞為例
- 利用 jiebaR 套件進行中文斷詞
- 進行簡單的文件分析與視覺化
- 根據出現次數與出現的文件數判斷重要的詞語
- 以長條圖與文字雲進行視覺化

學習目標

- 能夠說明斷詞的目的與意義
- 能夠使用 jiebaR 套件進行中文斷詞
- 能夠指出 TF-IDF 的意義並說明做法
- 能夠利用長條圖與文字雲將斷詞後的關鍵詞語視覺化

斷詞的目的

- 詞是語言最小的意義單位
- 自然語言處理通常以詞語為處理單位
- 中文句子的詞語之間沒有明顯界限
- 進行中文自然語言處理前需要先斷詞

文字資料處理常用的單位

- 字：書寫的基本單位。
- 詞：多個文字所組成、具有意義的單位。
- 句子：一組連續的詞構成的序列。
- 文件：由多個句子構成的組織。
- 文件集合：文件所成的集合，有時會用語料庫。

jiebaR 套件

- CRAN 中的一個中文斷詞套件
- 可將輸入的中文字串(character)，斷詞成為一組詞的向量
- 第一次使用套件前，需要安裝
  - 在 Console 上輸入

```
install.packages("jiebaR")
```

文字資料處理常用的分析格式

- 文件-詞語矩陣(document-term matrix, dtm)
- 將文件集合的各文件視為紀錄，表示為矩陣的 rows
- 將出現在文件集合內的詞語視為各文件紀錄的屬性，表示為矩陣的 columns
- 矩陣上的值為各文件分配在各詞語上的重要性(以 row 來看)，各詞語在各文件上的重要性(以 column 來看)
- 例如：下圖中的文件-詞語矩陣表示，文件集合中共有 M 個文件，分別表示為 d_1、d_2、d_3、……、d_M，文件集合內共曾出現 N 個詞語，分別是 t_1、t_2、t_3、……、t_N。
- 在下圖的文件-詞語矩陣中，w_{ij} 表示詞語 t_N 在文件 d_i 上的重要性(以 column 來看)。

判斷詞語的重要性

- 詞語出現次數愈多愈重要？
- 描述詞語出現次數與其排名次序之間的關係：Zipf's Law
- 在多數文件上都會出現的詞語並不重要

斷詞結果的視覺化

- 可以利用長條圖比較詞語的重要性與排序
- 一般常用文字雲(word cloud)快速瀏覽文件中的重要詞語

wordcloud 套件

- 文字雲套件
- 可將輸入的一組文字與其在圖形上的大小，畫出文字雲
- 第一次使用套件前，需要安裝
  - 在 Console 上輸入

```
install.packages("wordcloud")
```

預備工作

準備工作目錄與檔案

- 在 rCourse 下，建立工作目錄 07
- 將 06 的生活新聞資料檔案複製到 07 下

設定工作目錄

- 首先開啟新的 Script
- 在 Script 上，設定工作目錄

```
setwd("rCourse/07")
```

載入 tidyverse 套件

- 在 Script 上輸入

```
library(tidyverse)
```

載入 jiebaR 套件

- 在 Script 上輸入

```
library(jiebaR)
```

載入 wordcloud 套件

- 在 Script 上輸入

```
library(wordcloud)
```

讀取與處理生活新聞

讀取生活新聞 csv 檔案

- 在 Script 上輸入

```
tdf <- data.frame()

for (i in 1:15) {
  file <- sprintf("udn_2018_09_%02d.csv", i)
  df <- read.csv(file, fileEncoding="UTF-8", stringsAsFactors = FALSE)
  df$date <- sprintf("2018/09/%02d", i)
  tdf <- rbind(tdf, df)
}
```

為便利處理將每則新聞進行編號(id)

- row_number()依照紀錄順序產生流水編號
- 在本次課程，將每則新聞內容視為一筆文件，其中可能包含一到多個句子。

```
tdf <- tdf %>%
  mutate(id = row_number())
```

- 以 View(tdf)檢視 tdf 的結果

	link	text	date	id
分餐廳暫停營業	/news/story/7270/3342674	板橋麥當勞10點30分真的打烊了，門口擺滿了準備補貨的麵...	2018/09/01	1
路 網友感嘆「看了好想哭」	/news/story/7266/3343358	民眾搭計程車時，司機直接拿出一本厚厚的地圖找路，令她...	2018/09/01	2
狗一樓 睡覺做惡夢	/news/story/7270/3342739	麥當勞今天上午10時30分起，大部分餐廳暫停營業。記者王...	2018/09/01	3
3種警訊更應該當心	/news/story/7266/3343148	近期網路傳出「聞不到花生醬，就是得失智症」訊息，國健...	2018/09/01	4
塞車「員工苦難日」	/news/story/7270/3342651	昨麥當勞推出大麥克買一送一，民眾排隊到門外。記者鄭清...	2018/09/01	5
優惠「資、姿」不分	/news/story/7266/3342622	詐騙臉書頁面把球后戴資穎寫成戴「姿」穎。圖／擷取自臉...	2018/09/01	6
強颱風 下周二起襲日	/news/story/7266/3342639	強烈颱風燕子明日至下周一接近日本南方海域。圖／翻攝自...	2018/09/01	7

查看最短的 10 則新聞長度

```
tdf %>%
  mutate(msg.len=nchar(text)) %>% #計算新聞長度
  arrange(msg.len) %>% #依照新聞長度進行排序
  slice(1:10) %>% #選出最短的10 則新聞
  select(msg.len)
```

-
```
# A tibble: 10 x 1
   msg.len
     <int>
 1      60
 2      60
 3      60
 4      64
 5      64
 6      68
 7      69
 8      69
 9      69
10      72
```

對新聞內容進行斷詞

設定斷詞器模式

- 使用 jiebaR 斷詞，需要先設定斷詞器模式
- jiebaR 斷詞器為 worker()
- type="mix"：選擇混合(mix)模式(預設)
- symbol=TRUE：設定輸出標點符號及特殊符號
- bylines=TRUE：設定逐一紀錄進行斷詞

```
word.seg <- worker(type="mix", symbol=TRUE, bylines=TRUE)
```

- 注意右上的 Environment 上增加了 word.seg 的 Environment

對新聞中的文句進行斷詞

- segment(text, word.seg)：以 word.seg 設定的斷詞器，對 title 與 text 上的文字進行斷詞
  - paste(title, text, sep="。")：將 title 與 text 連接起來，中間以"。"間隔
- word 欄位中的資料為 vector 形式的斷詞結果

```
ws <- tdf %>%
  mutate(word=segment(paste(title, text, sep="。"), word.seg))
```

選取編號(id)及訊息斷詞產生的詞語(word)

- 每一筆記錄：每一則新聞(文件)以及它上面的詞語

```
ws <- ws %>%
  select(id, word)
```

- 以 View(ws)觀察 ws 目前的資料型態與內容，word 欄位為 character vector，內容為各訊息的詞語

id	word
1	c("麥當勞", "之亂", " ", "10", ":", "30", "之後", "大部分", "餐廳",...
2	c("老", "司機", "拿出", "超厚自", "製", "地圖", "找路", " ", "網友...
3	c("麥當勞", "之亂", "！", " ", "員工", "：", "累", "得", "跟", "狗",...
4	c("聞", "花生醬", "測失", "智症", "？", " ", "出現", "這", "3", "種...
5	c("麥當勞", "之亂", "？", "插隊", "、", "違", "停", "、", "塞車", "...
6	c("小心", "受騙", "！", "搭", "大", "麥克", "熱潮", " ", "假", "傻...
7	c("吳德榮", "：", "燕子", "增強", "為", "今年", "最", "強腦風", " ...
8	c("每天", "工作", "逾", "12", "小時", " ", "月", "收不到", "3", "萬...

將斷詞產生的詞語，設為觀察值，展開 data frame

- unnest(ws, word)將 ws 的 word 欄位中的每一個詞展開
- 每一筆記錄：一則新聞以及一個出現在這則新聞上的詞語

```
ws <- ws %>%
  unnest(word)
```

id	word
1	麥當勞
1	之亂
1	
1	10
1	:
1	30
1	之後
1	大部分

利用 Regular expresseion 篩選出中文詞語

- grepl：第一個參數表達的 Regular expression 是否出現在第二個參數內
- 中文字的 Regular expression：\\p{Han}
- 至少出現一個中文字的 Regular expression (\\p{Han})+

```
ws <- ws %>%
  filter(grepl("(\\p{Han})+", word, perl=TRUE))
```

id	word
1	麥當勞
1	之亂
1	之後
1	大部分
1	餐廳
1	暫停營業
1	板橋
1	麥當勞

練習：將上述處理的過程運用 pipe(%>%)寫成一個段落

新聞的詞語重要性分析

依據詞語出現情形的重要性

- 最直觀判斷每個詞語重要性的方法是依據這些詞語出現在文件集合的次數。如果某一個詞語在文件集合內出現多次，表示這個詞語代表的概念經常被集合內的文件涉及。例如在這份講義中，你一定注意到「詞語」這個詞語出現了很多次。如果某一個文件集合中，有不少文件都提到「詞語」相當多次，表示「詞語」對這個文件集合而言，可能很重要。所以是否我們可以用詞語出現在文件集合內的次數，以及從次數延伸計算得到的頻率(詞語出現次數/所有詞語出現次數總和)，代表詞語的重要性？
- Zipf 曾經觀察到一個有趣的現象，不論是哪一種文件集合，只要裡面的文件有某種程度的關連，詞語依據出現頻率的排序結果與出現頻率存在某種特別的數學關係， 後來這種現象被稱為 Zipf's Law。以下我們將以視覺化的方式，檢視我們蒐集的生活新聞是否有這種現象。
- 最後，我們以長條圖檢視生活新聞中出現頻率最高的詞語。

每個詞語出現的次數與頻率

- 我們將首先統計新聞中每個詞語出現的總次數。
- 然後計算每個詞語的出現頻率。
- 我們並且將根據出現次數畫成直方圖(histogram)，觀察生活新聞中出現頻率最高的詞語。

統計新聞中每個詞語出現的總次數

```
word.df <- ws %>%
  count(word, sort=TRUE) #統計詞語在新聞中的出現總次數，並依據次數高低排序
```

- word.df：詞語以及其在文件集合中的出現總次數

計算每個詞語的出現頻率

- 對所有詞語出現次數進行總和，此即所有新聞上所有詞語的次數總和
- 某一詞語的頻率定義為詞語出現次數除以所有詞語的次數總和

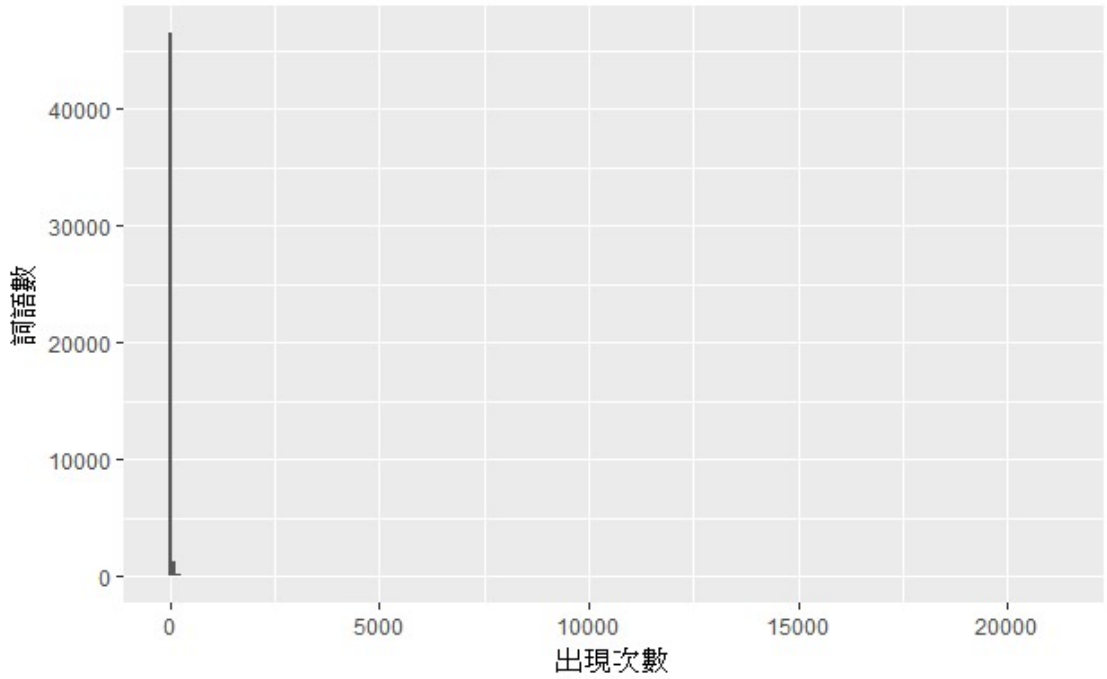
```
word.df <- word.df %>%
  mutate(frequency=n/sum(n)) #計算詞語出現頻率
```

word	n	frequency
的	21194	0.038418104
在	4616	0.008367367
有	3930	0.007123863
是	3607	0.006538365
也	3439	0.006233833
分享	3398	0.006159513
與	2781	0.005041085
為	2673	0.004845314

將詞語出現次數的分布情形化成直方圖

- 將直方圖的組界寬度(bin width)設為 100

```
word.df %>%
  ggplot(aes(n)) +
  geom_histogram(binwidth=100) +
  labs(x="出現次數", y="詞語數")
```



-
- 絕大多數的詞語出現次數小於 100 次

詞語排序與出現頻率的關係|Zipf's Law

- 請參考維基百科上有關 Zipf's Law 的說明
- 絕大多數的詞語僅出現很少的次數
- 出現次數較多的詞語只佔有相當少數

詞語排序與出現頻率的關係

- 根據出現頻率，將詞語由大到小設定順序
- 將詞語排序與出現頻率的關係畫成折線圖
- 劃出輔助線，驗證**詞語出現的頻率與其排序成反比**

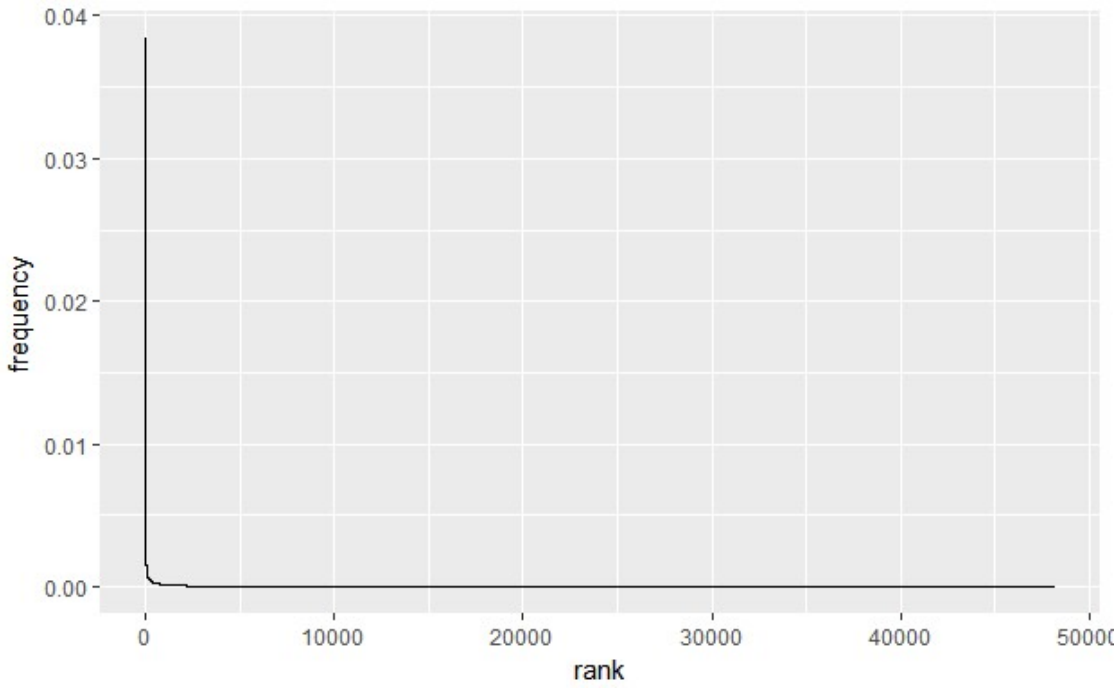
根據出現頻率，將詞語由大到小設定順序

```
word.df <- word.df %>%
  arrange(desc(frequency)) %>% #由大到小排定順序
  mutate(rank=row_number()) #依照紀錄順序設定流水編號
```

將詞語排序與出現頻率的關係畫成折線圖

- aes(x=rank, y=frequency)
  - 以詞語的排序為 x 軸座標
  - 以詞語的出現頻率為 y 軸座標

```
word.df %>%
  ggplot(aes(x=rank, y=frequency)) +
  geom_line()
```

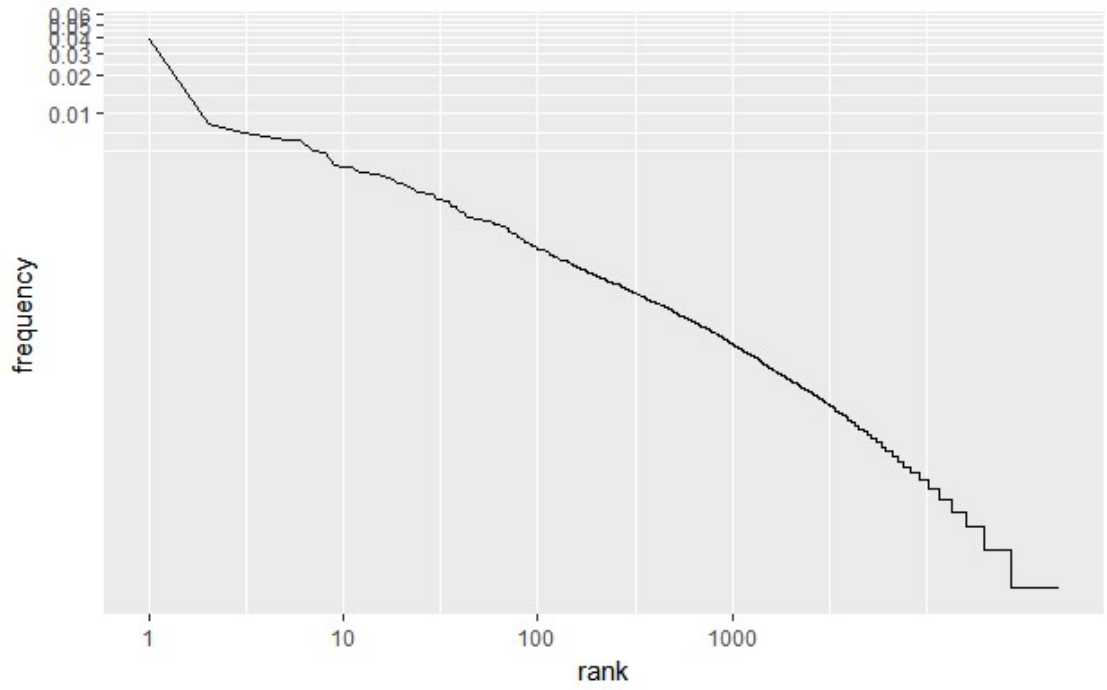


- 只有極少數的詞語有較高出現頻率，絕大多數詞語的出現頻率都很低(很多詞語只出現一次)

將 x 軸與 y 軸改為對數

- scale_x_log10()與 scale_y_log10()

```
word.df %>%
  ggplot(aes(x=rank, y=frequency)) +
  geom_line() +
  scale_x_log10(breaks=c(1, 10, 100, 1000)) +
  scale_y_log10(breaks=seq(0, 0.1, 0.01))
```



-
- 中央有部分很接近直線
- 這便是 Zipf's Law 的現象

在圖形上加入預測線，以線性迴歸的方式計算預測線的截距與斜率

- 選出圖形中接近直線的一段
- 利用線性模型估計可能的直線 lm(y ~ x)
  - 以 x 預測 y，y=ax+b
  - 在此，x 是 log10(rank)，y 是 log10(frequency)
  - 計算直線的 a(斜率, slope)與 b(截距, intercept)

```
rank_subset <- word.df %>%
  filter(rank>10, rank<1000)


lm(log10(frequency) ~ log10(rank), data = rank_subset)
```

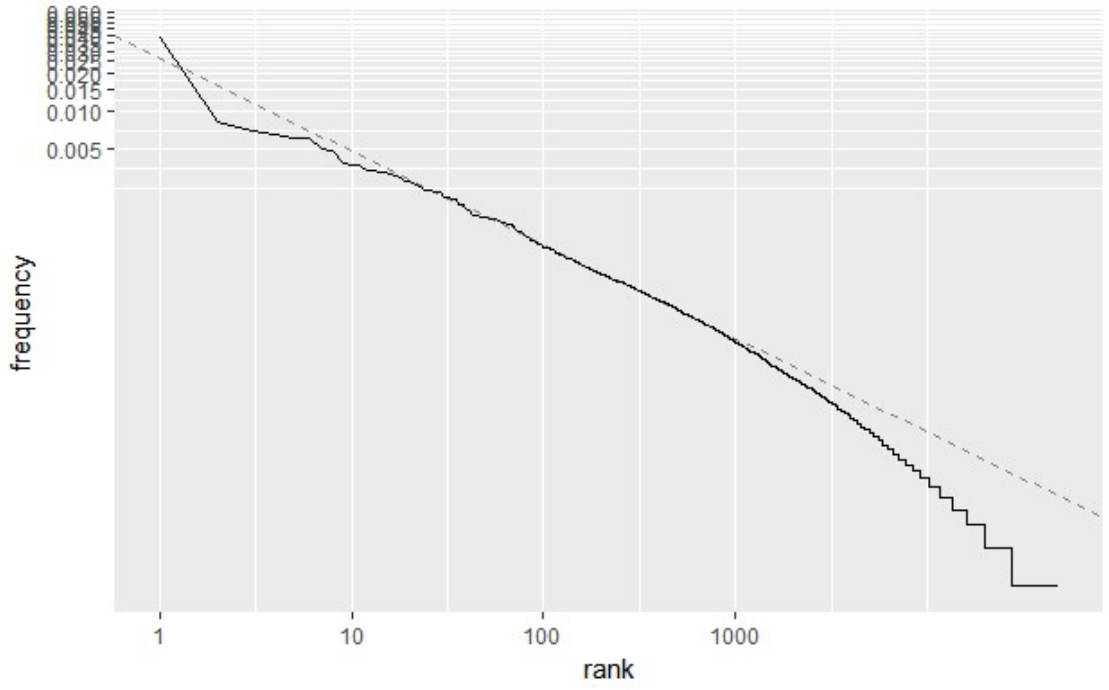
Call:
lm(formula = log10(frequency) ~ log10(rank), data = rank_subset)

Coefficients:
(Intercept) log10(rank)
  -1.5724   -0.7408

加上預測線

- 利用上面得到的斜率與截距在圖形上畫出預測線
- geom_abline(intercept, slope, color, linetype)在圖形上畫出截距為 intercept、斜率為 slope 的直線
  - 設定預測線的顏色為 gray50，形狀為線段虛線

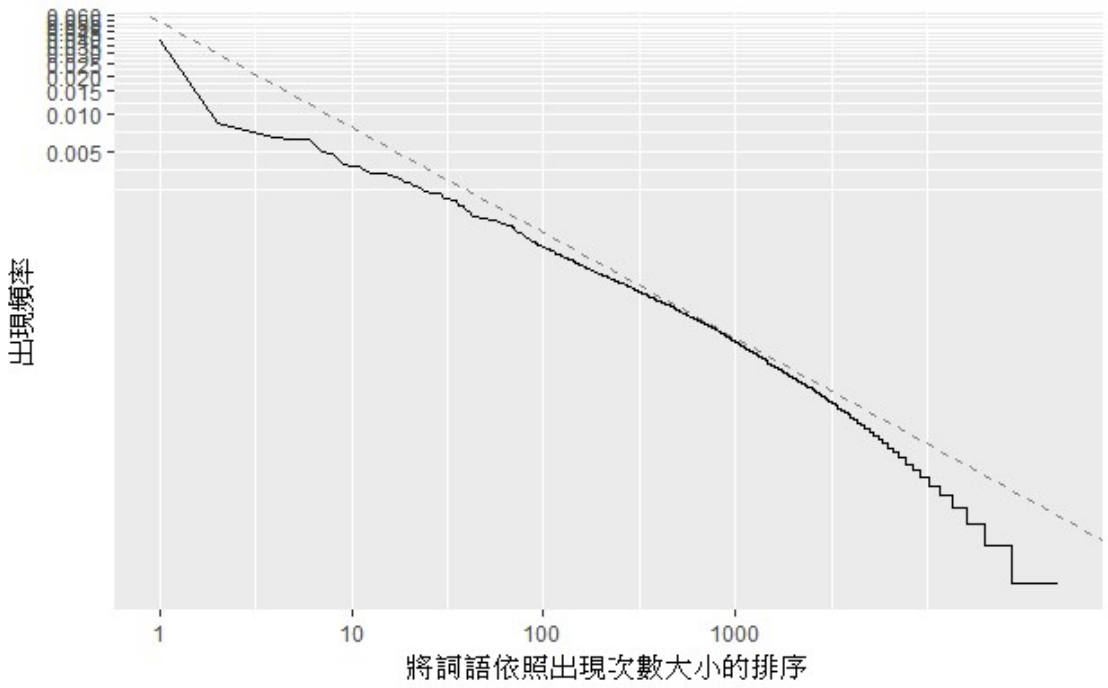
```
word.df %>%
  ggplot(aes(x=rank, y=frequency)) +
  geom_abline(intercept = -1.57, slope = -0.74, color = "gray50", linet
ype = 2) +
  geom_line() +
  scale_x_log10(breaks=c(1, 10, 100, 1000)) +
  scale_y_log10(breaks=seq(0, 0.1, 0.005))
```



是否觀察到預測線與實際結果的相似(重疊)情形

加上標題

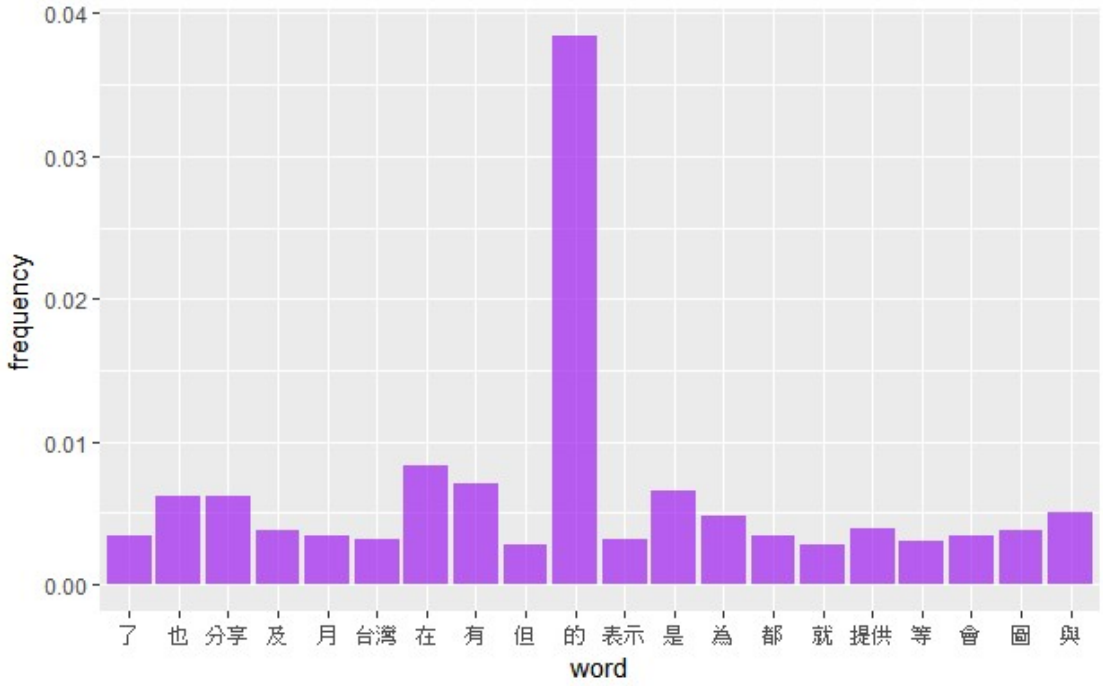
```
word.df %>%
  ggplot(aes(x=rank, y=frequency)) +
  geom_abline(intercept = -1.26, slope = -0.84, color = "gray50", linetype = 2) +
  geom_line() +
  scale_x_log10(breaks=c(1, 10, 100, 1000)) +
  scale_y_log10(breaks=seq(0, 0.1, 0.005)) +
  labs(x="將詞語依照出現次數大小的排序", y="出現頻率")
```



新聞中出現頻率最高的詞語

- top_n(word.df, 20, frequency)：選出出現頻率最高的 20 個詞語
- 將其出現頻率畫成長條圖
  - aes(x=word, y=frequency)：20 個詞語為 x 軸(以詞語的字碼順序排序)，其出現頻率為 y 軸

```
word.df %>%
  top_n(20, frequency) %>% #取出前二十名
  ggplot(aes(x=word, y=frequency)) +
  geom_col(fill = alpha("purple", 0.7))
```

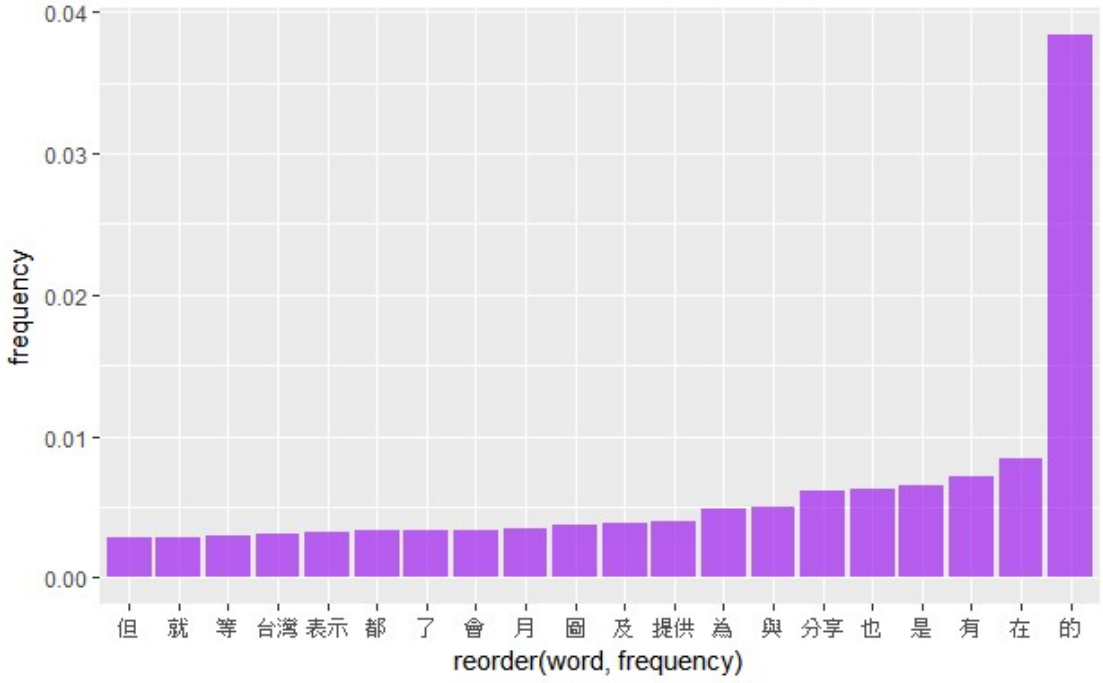


- 沒有按照出現頻率排序，不容易比較

按照詞語的出現頻率排序

- aes(x=reorder(word, frequency), y=frequency)
  - 以 frequency 排序後的詞語做為 x 軸

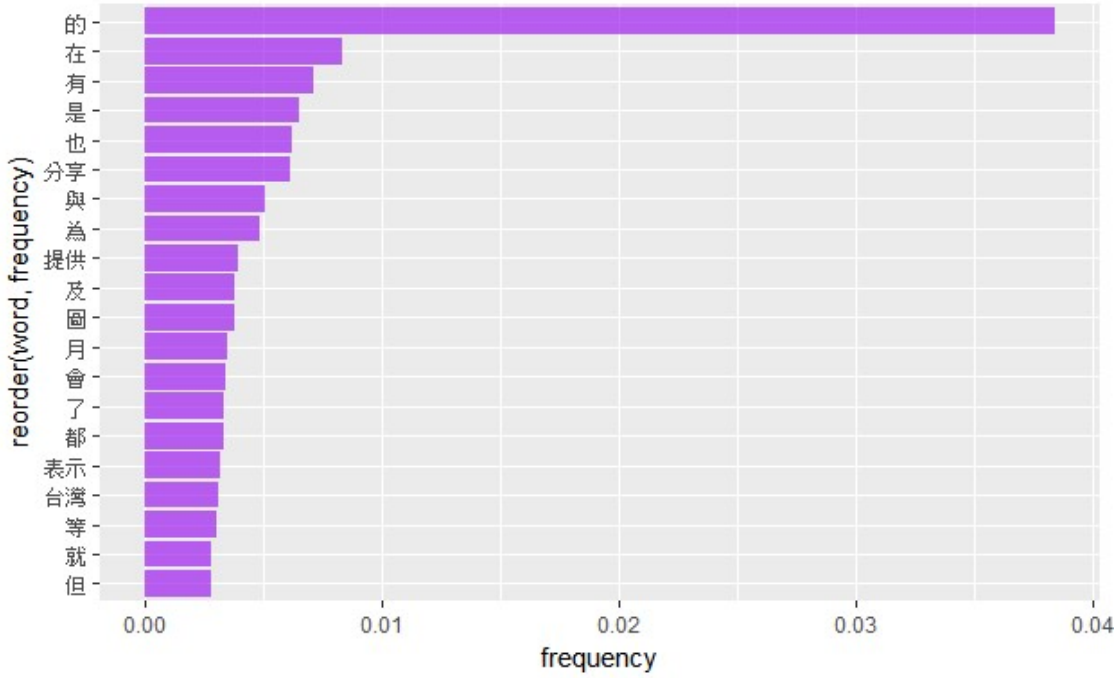
```
word.df %>%
  top_n(20, frequency) %>% #取出前二十名
  # 使長條圖按照將詞語的出現頻率排列
  ggplot(aes(x=reorder(word, frequency), y=frequency)) +
  geom_col(fill = alpha("purple", 0.7))
```



翻轉座標

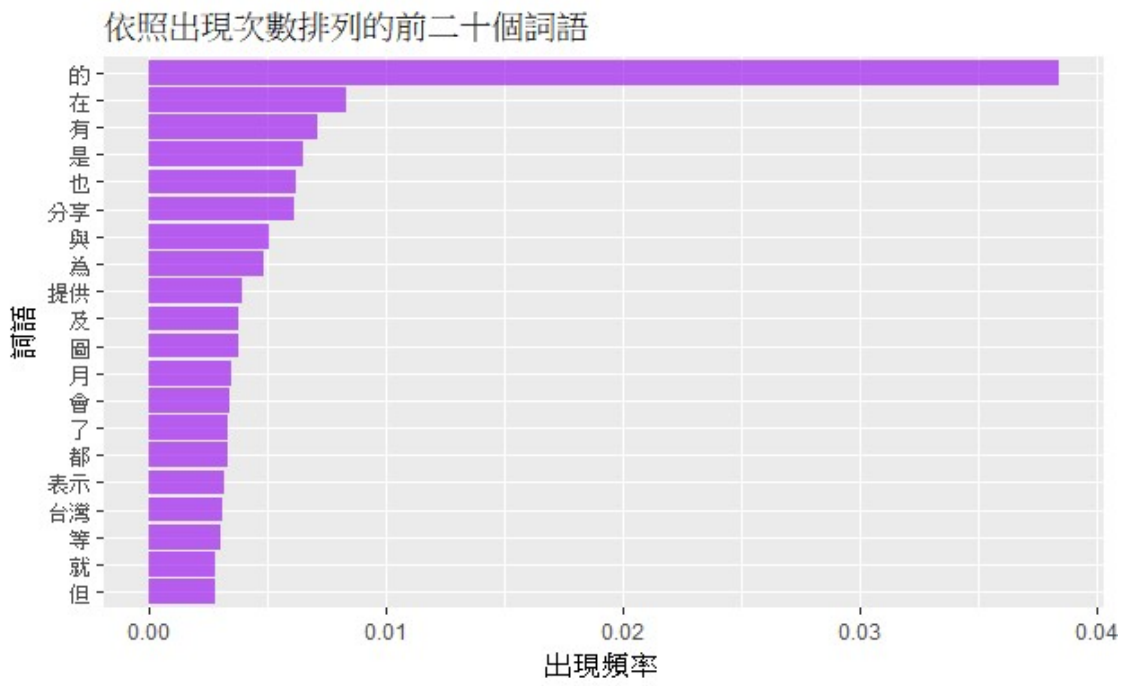
- coord_flip()將 x 軸與 y 軸翻轉

```
word.df %>%
  top_n(20, frequency) %>% #取出前二十名
  # 使長條圖按照將詞語的出現頻率排列
  ggplot(aes(x=reorder(word, frequency), y=frequency)) +
  geom_col(fill = alpha("purple", 0.7)) +
  coord_flip() #將詞語置於y 軸，出現次數置於x 軸
```



加上標題

```
word.df %>%
  top_n(20, frequency) %>% #取出前二十名
  # 使長條圖按照將詞語的出現頻率排列
  ggplot(aes(x=reorder(word, frequency), y=frequency)) +
  geom_col(fill = alpha("purple", 0.7)) +
  labs(title="依照出現次數排列的前二十個詞語", x="詞語", y="出現頻率") + #
加上標題
  coord_flip() #將詞語置於 y 軸，出現次數置於 x 軸
```



-

以出現頻率決定詞語重要性的討論

- 許多詞語都是不具意義的停用詞(stop words)
- 這些詞語出現在許多新聞內

IDF | inverse document frequency

IDF

- 出現在愈多訊息的詞語，愈不重要
- 根據每個詞語出現的新聞數，判斷它們的重要性
- 某個詞語的 idf 定義為 log(D/di)：D 是新聞總數，di 是這個詞語出現的新聞數

計算 idf

- mutate(total.doc=n_distinct(id))：計算新聞總數，也就是上文的 D
- group_by(word, total.doc)：對詞語進行分群
  - 為了在未來計算中使用 total.doc，所以保留在 group_by 內
- summarise(doc.freq=n_distinct(id)): 統計每個詞語出現的新聞數，即上文的 di
  - doc.freq=n_distinct(id)：詞語出現的新聞數
- mutate(idf=log(total.doc/doc.freq))：計算 idf
- select(word, idf)：選擇詞語及其 idf

```
word.idf <- ws %>%
  mutate(total.doc=n_distinct(id)) %>% #新聞總數
  group_by(word, total.doc) %>% #依據每個詞語分群
  summarise(doc.freq=n_distinct(id)) %>% #統計各詞語出現的新聞數
  mutate(idf=log(total.doc/doc.freq)) %>%
  select(word, idf)
```

TF | Term frequency

詞語在各則新聞的出現次數與頻率

- group_by(id, word)：以訊息和詞語進行分群
- summarise(count=n())：計算詞語在各則新聞中的出現次數
- mutate(tf=count/sum(count))：以詞語出現次數除以新聞內所有詞語出現次數，計算詞語在新聞上的出現頻率

```
word.msg <- ws %>%
  group_by(id, word) %>%
  summarise(count=n()) %>% #詞語在新聞中的出現次數
  mutate(tf=count/sum(count)) %>% #詞語在每一則新聞內的頻率
  ungroup()
```

- 每個詞語在各個文件(新聞)內的出現次數與頻率
- 在這裡，word.msg 便是一般在文字資料處理上常說的文件-詞語矩陣(document-term matrix, dtm)的 long data format

文件-詞語矩陣的圖示

	一世	一起	一個	一場	一片	一樣	了		t_N
1	0.0105	0.0105	0	0	0	0	0.0316		
2	0	0	0.0137	0.0137	0	0	0.0137		
3	0	0	0.0071	0.0036	0.0036	0.0036	0.0036		
d_M									

•

word.msg 的圖示

2		
2		
3		
3		
3		
3	了	

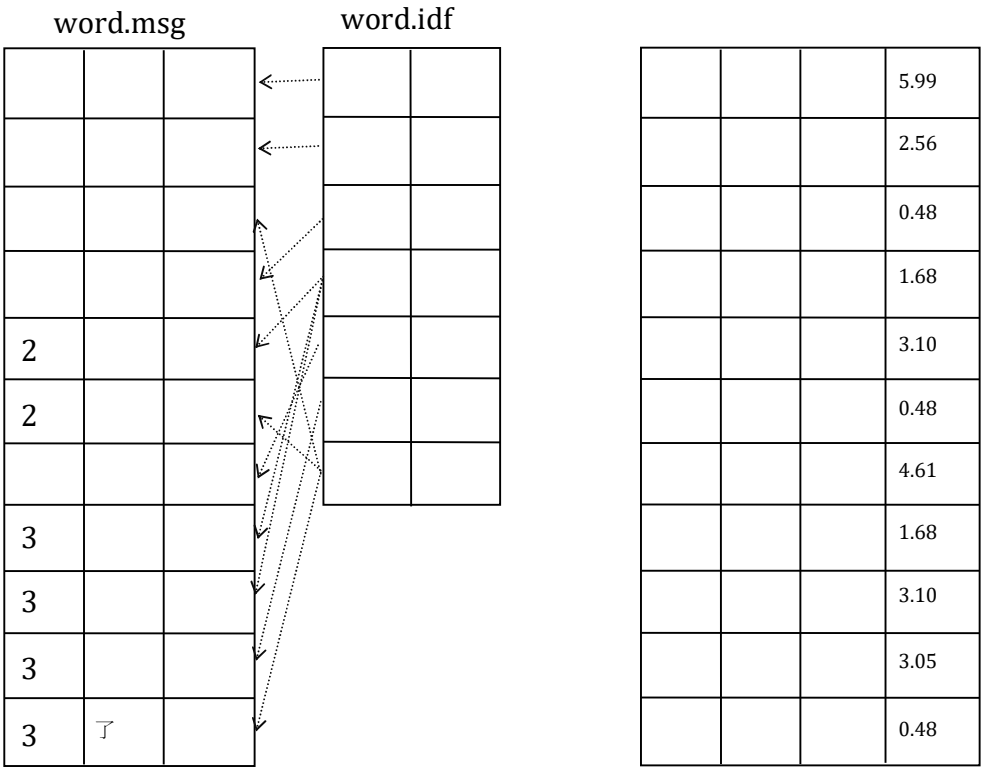
TF*IDF

計算詞語在各則新聞的 tf*idf 值

- 將各則新聞的詞語及其出現次數連結上 idf
- left_join(word.msg, word.idf, by=c("word"="word"))以 word.msg 為主，合併 word.idf 的欄位
  - 合併的條件是兩個 word 資料必須相等

```
word.msg <- word.msg %>%
  left_join(word.idf, by=c("word"="word")) %>%
  mutate(tfidf=tf*idf)
```

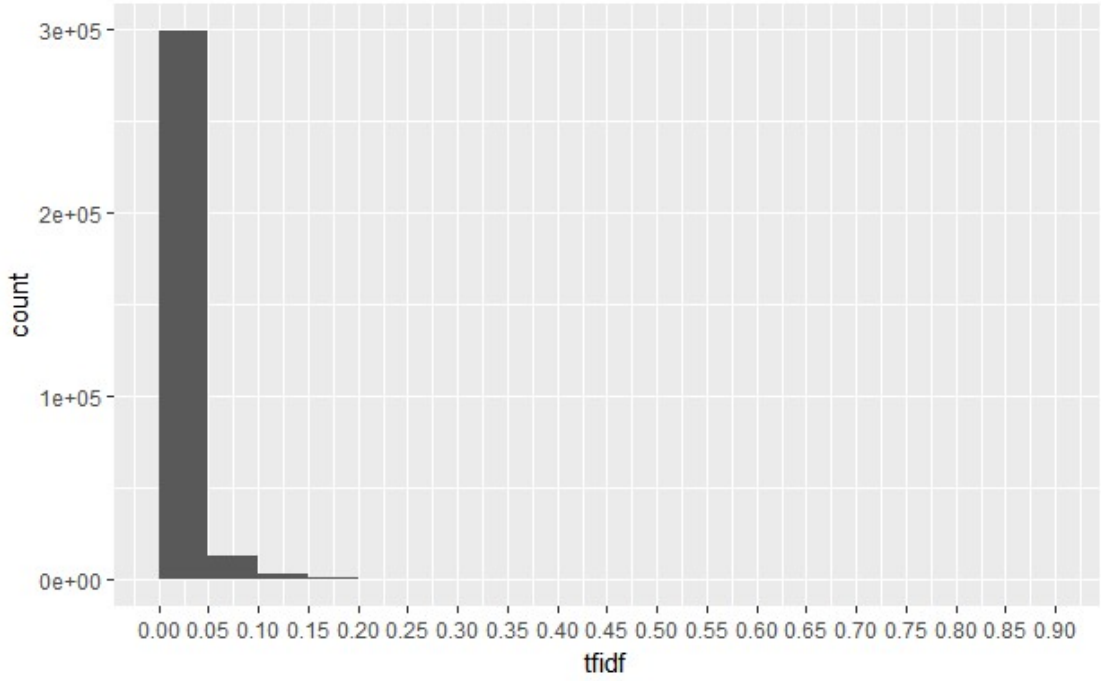
以圖表示 left_join(word.msg, word.idf, by=c("word"="word"))



畫出分布圖

```
x_breaks = seq(0, max(word.msg$tfidf)+0.05, 0.05)


word.msg %>%
  ggplot(aes(tfidf)) +
  geom_histogram(breaks=x_breaks) +
  scale_x_continuous(breaks=x_breaks)
```



-
- 極大多數詞語的 tf*idf 小於等於 0.05

捨棄各則新聞中的 tf*idf 值過小的詞語

```
word.msg <- word.msg %>%
  filter(tfidf>0.05)
```

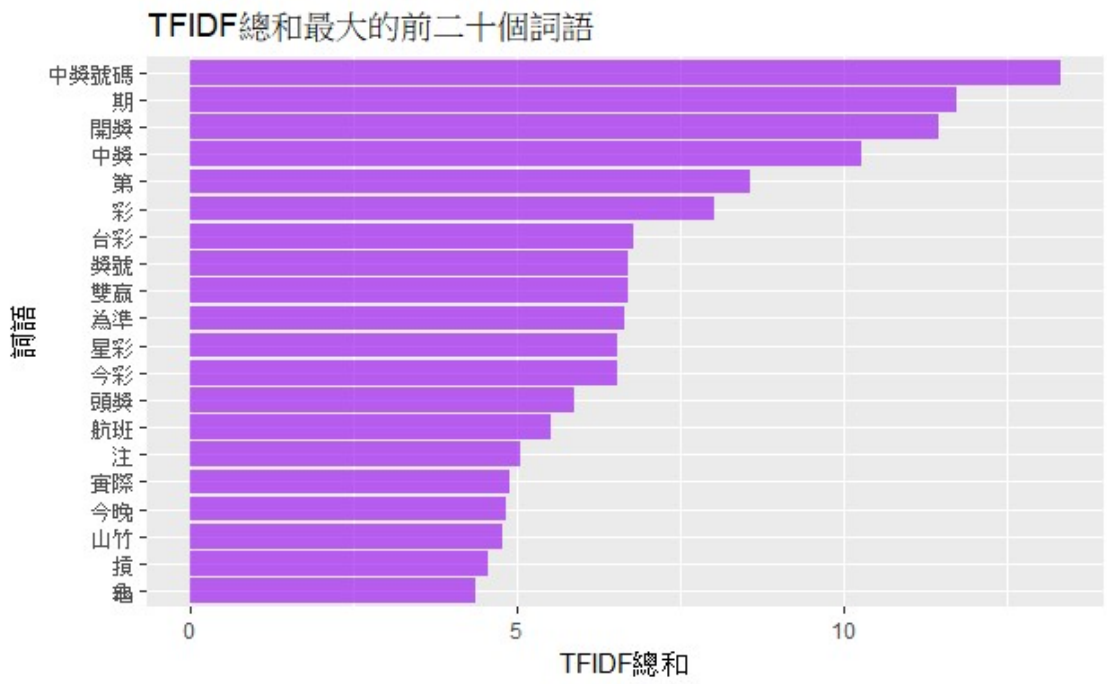
統計詞語在所有新聞上的 tf*idf 總和

```
word.scr <- word.msg %>%
  group_by(word) %>%
  summarise(sum.tfidf=sum(tfidf)) %>%
  arrange(desc(sum.tfidf))
```

- word.src：詞語與其重要性(詞語在所有新聞上的 tfidf 總和)

繪製所有新聞上的 tf*idf 總和最高的前二十個詞語的長條圖

```
word.scr %>%
  slice(1:20) %>%
  ggplot(aes(x=reorder(word, sum.tfidf), y=sum.tfidf)) +
  geom_col(fill = alpha("purple", 0.7)) +
  labs(title="TFIDF 總和最大的前二十個詞語", x="詞語", y="TFIDF 總和") +
  coord_flip()
```

- 

繪製所有新聞上的 tfidf 總和最高的前五十個詞語的文字雲

- pal <- brewer.pal(8, “Dark2”)：從“Dark2”顏色配置中挑選 8 種顏色
- wordcloud 繪製文字雲
  - 前兩個參數是詞語與其在圖形上的大小(詞語的重要程度)
  - scale=c(3, 0.3)：第一個數值是最大的尺寸，第二個數值代表每一級字的差距
  - random.order=FALSE：較重要的詞語不僅用較大的字體，而且按照詞語的重要程度繪製，較重要的詞語會在圖形中心
  - random.color=TRUE：隨機選擇顏色
  - rot.per=0：垂直方向的詞語比例為 0
  - colors：字體採用的顏色配置

```
pal <- brewer.pal(8, "Dark2")
wordcloud(word.scr$word[1:50], word.scr$sum.tfidf[1:50],
          scale=c(4, 0.1), random.order=FALSE, random.color=TRUE,
          rot.per=0, colors=pal)
```

-

本次課程小結

小結

- 中文文字資料在分析前需要先經過斷詞
- 斷詞需要仰賴詞典，也就是預先編輯好的詞語集合
- 即便詞典收錄的詞語再齊全，仍有許多文字資料會包含若干詞典未收錄的新詞，稱為未知詞(unknown words)
- 未知詞通常與分析的文字資料所屬領域有極大的關係
- 在未來的課程，我們將討論如何根據文字資料的分析結果，偵測可能的未知詞，以便擴增詞典，提升斷詞效能

小結

- 文字資料中，詞語出現的次數差異懸殊，少數詞語出現次數遠大於其他詞語(Zipf's Law)
- 文字資料中出現較多次的詞語通常是介詞、連接詞、句末補語等停用詞
- 因此若是只根據出現次數判斷詞語的重要性，可能找出的詞語中會有許多是停用詞

小結

- 停用詞通常廣泛地出現在各個文件上
- 一般關鍵詞語則集中在少數文件上大量出現
- 所以文字資料處理通常會利用 IDF 做為判斷停用詞的參考資訊

小結

- 在相同大小的圖形下，相較於長條圖，文字雲可以呈現更多的詞語
- 文字雲可以提供分析者對於文字資料內容的速寫
- 但長條圖可以提供更細節而正確的重要性比較與排序

延伸思考

1. 本次課程利用斷詞、關鍵詞語擷取與資訊視覺化，請你想想看是否可以將這些方法融入質性研究的內容分析當中。
2. 從本次課程，我們可以看到在一般的情形下，文件中會有許多沒有收錄到詞典當中的未知詞，有沒有方法可以自動發現這些未知詞。