國立東華大學應用數學系

111 學年度第二學期專題

指導教授：曹振海 博士

# 紅樓夢人物分析

***The Analysis of Characters in Dream of the Red Chamber***


學生：許堯智 李世勛 黃定綸 撰

中 華 民 國 一 一 二 年 六 月

# 目錄

# 第一章 緒論

## 1.1 研究背景與動機

中國古典小說是中國文學中不可或缺的重要部份，在歷史上有許多值得進行探勘的名著。其中《紅樓夢》[1] 被譽爲中國古典小說的代表作之一，並且具有很高的探究性，小說中人物的命運和許多事件間有著密切而複雜的關係，需要經過深入的研究才能理解其眞正含義。

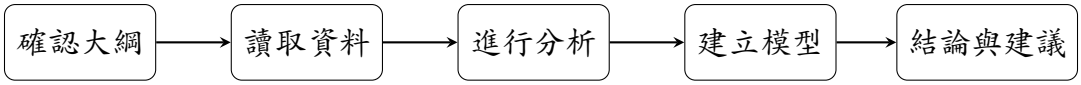
在探究過程中，我們參考了人物關係圖《紅樓夢 | 人物關係表 | 四大家族 | 寶玉、黛玉、寶釵血緣關係 | 5 分鐘弄懂紅樓夢人物關係 | 淺談紅樓夢 | 2023》[3]。在關係圖中可以見到人物關係錯綜複雜，最主要的四大家族即使有顏色標示，也難以明白其關係。但這已是研究者經過四天四夜建構出的關係圖，研究者也承認很難將人物關係完全弄懂。

雖然同樣存在著人物關係相對單純的小說，但正因《紅樓夢》值得深掘的研究價值，在文學界也誕生了新名詞「紅學」。在本研究中，《紅樓夢》被選爲分析對象有其特殊的原因。首先，《紅樓夢》是一部長篇白話小說，使用的是比較容易理解的白話文，相對於使用大量文言文的古典名著，更易於進行文字分析和處理。其次，《紅樓夢》具有固定的主角群，自開頭到結尾都以同一群人推進劇情，這種連貫性使得人物之間的關係更爲明確。

隨著生活節奏的加快，伴隨著大量的外部訊息和刺激，人們的專注力正逐漸下降。因此，爲了吸引讀者的注意力，懶人包儼然成爲一種新型流行趨勢，不僅使得訊息的呈現變得簡潔易懂，同時還具有足夠的視覺吸引力，並以一種簡單、直觀的方式呈現訊息，貼近了現代人快節奏的生活步調。本研究旨在希望能透過分析《紅樓夢》中人物出現的次數、頻率，並以簡單、直觀的方式呈現人物間的關係和生活圈，使讀者能夠藉由本研究了解《紅樓夢》主要角色的關係。作爲一部白話章回小說，《紅樓夢》提供了明確的人物關係和通俗的文字敘述，爲研究和分析提供了有價值的案例。本研究除了以能讓讀者更深入地了解小說的人物關係爲目的，同時也以滿足現代社會對於簡潔、易懂和吸引人的訊息呈現的需求作爲目標。

## 1.2 研究流程

本次研究流程與架構分爲以下五步驟：




圖一 研究流程架構圖

研究流程說明：

**確認大綱**：

從本身所接觸到的各種領域及應用中，選擇文字探勘技術作爲分析工具，並藉由文獻探討現有的相關研究方向，進一步決定研究主題。最後，將數據資料以清晰、直觀的方式呈現。

**讀取資料**：

使用 R 語言將《紅樓夢》文本轉換成可進行分析的格式，例如進行分詞、刪除標點符號，並將人物及其別名等資料抓取出來，以及對資料做初步的處理。

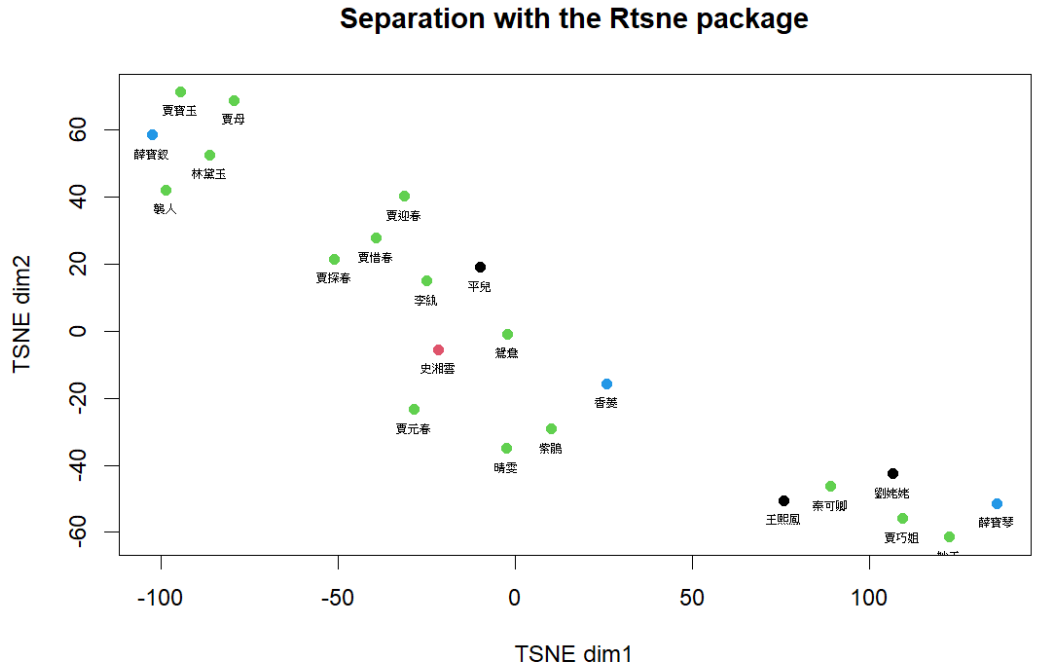
```
# 自定義檔名函數
file_name_function = function(i){
  if (i < 10)
    file_name <- paste0(" 紅樓夢 _ 第", 0, 0, i, " 回")
  else if (i < 100)
    file_name <- paste0(" 紅樓夢 _ 第", 0, i, " 回")
  else
    file_name <- paste0(" 紅樓夢 _ 第", i, " 回")
  return(file_name)
}
# 字數統計函數
stat_words <- function(file_name){
  returnValue(sum(nchar(gsub("[，。!?:「」；《》『』——　【】1234567890]",
  ↪  "", gsub("\\[", " ", gsub("\\]", " ", gsub("-", " ", gsub("\\
  ↪  “\\” \\ ‵\\″ ", " ", file_name)))))))
}
```

**進行分析：**

透過對資料的分析繪製可視化圖表，如次數長條圖、相關係數矩陣以及角色之間的相關性，並針對資料與圖表進行初步分析，以提供作爲後續計算模型依據。

**建立模型：**

利用 networkD3、Rtsne 套件，最終做出的人物關係進行可互動的視覺化網路圖，可將研究成果放在網路上供比對與參考。




圖二　t-SNE 分群

**結論與建議：**

不同的模型具有不同的可解釋性，本研究會將結果與原先預測來做比對，並加以整理之後作爲本研究的結論。不僅可以獲得分類結果，還可以獲取人物關係圖等重要訊息，從而更進一步的分析和解釋，獲得更多有用的資訊。

除此之外，也會將研究過程中本應還可再進行深入的部份，一同建議研究方法，希望能讓下一位研究者獲得靈感。

# 第二章 文獻研究

## 2.1 分析方式簡介

以下爲本研究中使用到的分析方法：

**correlation**：

相關係數使用數值表示，是用來描述兩個變數之間相關程度的指標，表示兩個變數相互關聯的程度，其取值範圍介於 -1 到 +1 之間，其中正的相關係數表示兩者呈現正相關，負的相關係數表示兩者呈現負相關，而接近零的相關係數表示兩者之間沒有關係。相關係數可以幫助我們了解兩個變數之間的關聯性，並發現資料數據的發展趨勢。

**wordcloud**：

文字雲是一種用於文字資料視覺化的方法。這種方法可以將大量文字資料轉換成一個具有可視性的圖形，將高出現頻率的詞彙以較大的字型呈現，並集中在中央；低出現頻率的詞彙以較小的字型呈現，會散佈於外圍，是資料的一維呈現方法。

**MDS**：

MDS 全名爲 Multi-dimensional Scaling ，是一種可將高維度的資料降至二維或三維平面上，以更容易地理解和分析數據的方式。通常用於探索數據之間的相似性或差異性，並以圖形化的方式展示數據的結構。在《紅樓夢》人物關係的研究中，MDS 可把人物之間的相似度轉換爲距離單位，然後將這些距離單位映射到二維或三維空間中，並將每個人物表示爲一個點，人物之間的距離表示他們之間的相似程度。

**Jaccard**：

Jaccard 用於評估兩個文本之間的相似程度，這可以通過將兩個文本中共同出現的單詞數量除以兩個文本中所有不同單詞的總數來完成。簡單來說分子是交集，分母則是聯集。在 R 語言中使用該函數，可以更精確計算兩個文本之間的相似度，進而進行文本分析和比較。

**PCA**：

PCA 把高維度的資料以盡可能保留其特性的方式降到低維度，爲了將一組相關變量轉換爲一組新的不相關變量，稱爲主成分 (PC) 。主成分是原始變量的線性組合併且彼此正交。第一個 PC 捕獲數據中的最大變異數，每個後續 PC 捕獲最大剩餘變異數，但前提是它與之前的 PC 正交。其降維方法主要利用線性的方式，也就是降維生成的每一個 PC 都是原本變數（x1, x2, x3 ... ）的線性組合。

**t-SNE**：

t-SNE 將點之間的相似度轉化爲條件機率，原始空間點的相似度由常態分佈表示，嵌入空間中點的相似度由 t 分佈表示。t-SNE 的主要優勢在於解決了降維後的擁擠問題，使得相似的樣本能夠聚集在一起，而差異大的樣本能夠有效地分開，避免了其他降維方法各個點分佈擁擠、邊界不明顯的缺點

## 2.2 論文與相關研究參考

爲了尋找適合的視覺化結果，我們在碩博士論文網站中找到了《金庸小說互動式視覺化文字探勘》論文[2]，論文提供的方法是文字雲：取出需要的人物名稱並合併別名後，可發現到當人物出現次數越多，則該人物字體越大，鮮豔又淺顯易懂，符合作爲懶人包的必要條件。只可惜文字雲侷限在單一人物的出現頻率，無法知道一對一或多對一之間的人物關聯。

接著，在知乎網站發掘到能顯示人物關聯性的網路圖[4]。參考該網站後，發現該圖以人物爲節點，關聯性作爲線段粗細的網路圖，若各節點線段愈粗，便代表兩節點關係愈深，節點大小代表的是人物出現的頻率，顏色則代表的是人物所代表的家族，比起文字雲多了角色跟角色之間的關聯。

即使網路圖解決了人物與人物的關聯性問題，依舊還侷限在單一人物的對應關係，難以明確知道多人對應多人的關係，因此我們透過其他視覺化方式，來分析呈現資料特性。

爲此，從人物關係的網路圖轉而進行分群方法：MDS 、PCA 跟 t-SNE 。其中，分群之後的結果仍然較爲鬆散，爲了改進這方面的問題，在原本的分群套件上再加上名爲 Jaccard[7] 的篩選方式。

# 第三章 研究方法

## 3.1 使用套件

corrplot：描述不同變數之間相互關聯性的關係矩陣。

tm：在 R 中進行文本分析、處理、建模等任務變得更加方便，支持生成文字雲、頻率圖、主題分布圖等。

wordcloud：根據詞彙的頻率或權重，創建具有不同大小和顏色的文字雲圖形。

jiebaR：用於 R 語言中的中文分詞套件，將中文文本進行分詞處理，也是進行中文分詞最常見的方式。

networkD3：創建互動性網路圖可視化。

readxl：用於讀取和解析 Excel 文件。

tidyverse：包括 ggplot2、dplyr、tidyr 等等，涵蓋了分組統計、數據可視化、轉換等各方面的套件。

igraph：網絡圖的操作和轉換，這次用於創建邊數據。

magrittr：使得連續的函數調用和數據處理更加易讀和易寫，例如%>% 。

dplyr：用於對資料進行快速、直觀和一致的操作。

ggpubr：在 ggplot2 的基礎上構建的擴展套件，可視化功能和統計分析工具。

MASS：提供了線性模型、分類分析等的數據分析和建模。

vegan：用於處理和分析生態學數據，這次是使用 Jaccard 進行修正。

Rtsne：將高維數據降低到二維或三維，以便更好地理解數據的結構和關係。

## 3.2 文字探勘

文字探勘是從大量文字中提取感興趣資訊或挖掘有用知識，並透過電腦的運算能力，過濾及轉化大量的文字內容，找出隱含且有用的資訊。此處結合語言處理、統計分析等方法，於文本中找尋趨勢和關聯性。

由於《紅樓夢》擁有許多個性鮮明的角色，在此使用 R 語言將主要的角色與其別名進行整理，找出他們在文本中的出現頻率，以便後續分析不同角色間的互動模式，提高後續資料分析和模型建立的準確性和可靠性。

```
# 主要人物名稱
name_main <- c(" 賈寶玉", " 林黛玉", " 薛寶釵", " 賈元春", " 賈探春", " 史湘雲", " 妙玉", " 賈迎春", " 賈惜春", " 王熙鳳", " 賈巧姐", " 李紈", " 秦可卿")
# 別名
nickname_ 賈寶玉 <- c(" 寶玉", " 此石", " 寶二爺", " 怡紅公子", " 絳洞花王")
nickname_ 林黛玉 <- c(" 黛玉", " 顰顰", " 顰兒", " 林姑娘", " 林丫頭", " 瀟湘妃子")
nickname_ 薛寶釵 <- c(" 寶釵", " 蘅蕪君", " 寶姐姐", " 寶丫頭")
nickname_ 賈元春 <- c(" 元春", " 賈妃", " 元妃", " 貴妃", " 大姑娘")
nickname_ 賈探春 <- c(" 探春", " 蕉下客", " 三姑娘")
nickname_ 史湘雲 <- c(" 湘雲", " 枕霞舊友")
nickname_ 妙玉 <- c(" 妙玉", " 檻外人")
nickname_ 賈迎春 <- c(" 迎春", " 二木頭", " 二姑娘")
nickname_ 賈惜春 <- c(" 惜春", " 藕榭", " 四姑娘")
nickname_ 王熙鳳 <- c(" 熙鳳", " 鳳辣子", " 璉二奶奶")
nickname_ 賈巧姐 <- c(" 巧姐", " 妞妞", " 大姐兒")
nickname_ 李紈 <- c(" 李紈", " 宮裁", " 稻香老農")
nickname_ 秦可卿 <- c(" 可卿", " 蓉大奶奶", " 兼美"," 秦氏")
# 其餘人物名稱
```

```r
other_name <- c(" 賈母", " 劉姥姥", " 香菱", " 平兒", " 晴雯", " 襲人", " 紫鵑", " 鴛鴦", " 薛寶琴")
```

name_matrix 爲串聯所有章節中出現人物的名稱，其作用爲方便做文字探勘，以方便後續資料處理。

```r
# 定義要搜尋的人物名稱
name_matrix <- " 小熊維尼" # 給定預設值 (後面不會用到)
for (i in 1:length(name_main)){
  person_names <- paste0("nickname_", name_main[i]) # 人物別名指定
  name_matrix <- c(name_matrix, get(person_names)) # 合併至 name_matrix
}
name_matrix <- c(name_matrix[-1], other_name) # 刪除" 小熊維尼" 並合併其餘人物
```

給定一空矩陣，透過 stat_name 這一個自定義的文文字探勘函數計算各個人物在各回的出現次數，並將人物名稱與出現次數合併到 name 矩陣。

```r
# 定義回數 120 回
num.hei <- 120

# 統計出現次數函數
name <- 0 # 宣告有 name 的存在
name <- stat_name(name_matrix, num.hei, name)
name_all <- name # 將所有名字存入 name_all 方便以後分析
```

以下程式碼爲 stat_name 函數的運作方式，此處作法部分參考自銘傳大學論文[2]，並加以修改成符合本研究所需的樣式。

```r
# 統計出現幾次特定文字函數
stat_name <- function(name_matrix, num.hei, name){ # name_matrix <- 欲統計的字詞, num.hei <- 總回數, name <- 儲存的矩陣

  # 名稱的替代 (方便處理)
  name_matrix_change <- paste("person_", 1:length(name_matrix), "~", sep = "")
```

```
  name_matrix_change2 <- paste("person_", 1:length(name_matrix), sep = "")

  # 創建矩陣存放各章節人物出現次數
  name <- matrix(0, nrow = length(name_matrix), ncol = num.hei, dimnames = list(name_matrix, c(1:num.hei)))

  for (chapter in 1:num.hei){
    for (i in 1:length(name_matrix)){
      temp_store_name <- gsub(name_matrix[i], name_matrix_change[i], get(file_name_function(chapter)), fixed = TRUE) # 替換人物名稱並存入 temp_store_name
      temp_store_name1 <- strsplit(temp_store_name, "~") # 切割字串並以"~" 分割
      name[i, chapter] <- length(grep(name_matrix_change2[i], temp_store_name1[[1]])) # 計算字串被切割幾次並儲存到 name
    }
  }
  return(name)
}
```

將 name 矩陣中屬於同一人物的名字與出現次數合併，統整爲人物各個章回出現次數矩陣，以利進行後續的各項統計與分析。

```
# 合併所有重複人物
total <- 1
temp_1 <- 0
name <- matrix(0, nrow = length(c(name_main,other_name)), ncol = num.hei, dimnames = list(c(name_main,other_name), c(1:num.hei)))
for (i in 1:length(name_main)){
  temp <- name_all[total:(total + length(get(paste0("nickname_", name_main[i]))) - 1), ]
  name[i,] <- apply(temp,2,sum)
  total <- total + length(get(paste0("nickname_", name_main[i])))
}
```

```
name[(i+1):length(c(name_main,other_name)), ] <-
  name_all[(length(name_matrix) - length(other_name) +
  1):(length(name_matrix)), ]
```

經由以上文字探勘與簡化的過程，我們得到以下人物總出現次數矩陣，此矩陣可運用在文字雲的呈現，並運用數據資料進行分析。

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 賈寶玉 | 4 | 2 | 32 | 0 | 61 | 14 | 29 | 85 | 35 | 5 | 10 | 1 | 11 | 16 | 57 | 25 | 48 | 41 | 117 | 56 | 49 | 43 | 54 |
| 林黛玉 | 0 | 1 | 83 | 3 | 10 | 0 | 9 | 25 | 4 | 0 | 0 | 3 | 2 | 1 | 0 | 8 | 0 | 18 | 33 | 24 | 14 | 29 | 18 |
| 薛寶釵 | 0 | 0 | 0 | 2 | 5 | 0 | 12 | 20 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 10 | 2 | 21 | 8 | 24 | 2 |
| 賈元春 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 5 | 44 | 3 | 2 | 0 | 1 | 3 |
| 賈探春 | 0 | 1 | 4 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 1 | 0 | 3 | 4 |
| 史湘雲 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 15 | 16 | 0 |
| 妙玉 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 |
| 賈迎春 | 0 | 1 | 3 | 1 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 2 | 0 | 2 | 0 | 5 | 2 |
| 賈惜春 | 0 | 1 | 1 | 0 | 1 | 0 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 2 | 3 |
| 王熙鳳 | 0 | 0 | 11 | 0 | 0 | 2 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 賈巧姐 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 李紈 | 0 | 0 | 0 | 3 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 5 | 0 | 0 | 1 | 3 | 0 |
| 秦可卿 | 0 | 0 | 0 | 0 | 16 | 1 | 13 | 2 | 0 | 8 | 18 | 0 | 14 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 賈母 | 0 | 0 | 33 | 3 | 6 | 2 | 3 | 15 | 7 | 0 | 6 | 4 | 4 | 0 | 3 | 10 | 3 | 27 | 3 | 4 | 3 | 42 | 7 |
| 劉姥姥 | 0 | 0 | 0 | 0 | 0 | 61 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 香菱 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 平兒 | 0 | 0 | 0 | 0 | 0 | 12 | 5 | 0 | 0 | 0 | 5 | 1 | 3 | 2 | 0 | 9 | 0 | 0 | 0 | 0 | 28 | 0 | 0 |
| 晴雯 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 6 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 6 | 0 | 0 | 0 |
| 襲人 | 0 | 0 | 9 | 0 | 4 | 11 | 0 | 6 | 5 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 55 | 14 | 24 | 9 | 7 |
| 紫鵑 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 |
| 鴛鴦 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 薛寶琴 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

表一　name 矩陣/人物總出現次數矩陣

爲了能夠從資料裡提取到人物間的關係，此處我們將 name 矩陣再度簡化，僅表示該章回中人物是否有出現，並將結果儲存至 person_exist 矩陣，稱之爲人物出現矩陣。

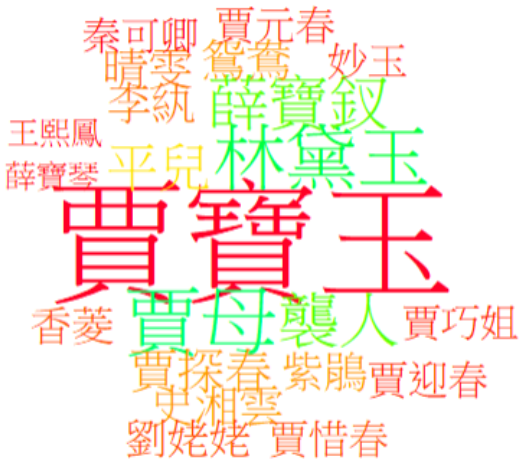
```
# 人物有出現在哪一回矩陣
person_exist <- 1 * (name & TRUE) # 將人物是否出現轉爲 {0,1} 元素
```

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 賈寶玉 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 林黛玉 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 薛寶釵 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 賈元春 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 賈探春 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 史湘雲 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 妙玉 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 賈迎春 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 賈惜春 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 王熙鳳 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 賈巧姐 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 李紈 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 秦可卿 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 賈母 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 劉姥姥 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 香菱 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 平兒 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 晴雯 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 襲人 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 紫鵑 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 鴛鴦 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 薛寶琴 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

表二 人物出現矩陣

## 3.3 資料視覺化

資料視覺化是將數據以圖形或圖表的形式呈現，幫助人們將繁雜的數據簡化成易懂好吸收的內容。在 R 語言中，有多個套件可用於資料視覺化，包含 wordcloud 及 networkD3 等等。wordcloud 能讓讀者在不閱讀所有文章的前提下，快速了解並聚焦大批文章中主要的議題；而 networkD3 則可以透過顏色、節點大小或者線段粗細等等方式來展示資料的不同屬性，其可互動性也高於其他二維圖形。




圖三 使用 wordcloud 後的視覺化結果

# 第四章 研究探討與分析

## 4.1 數據清理

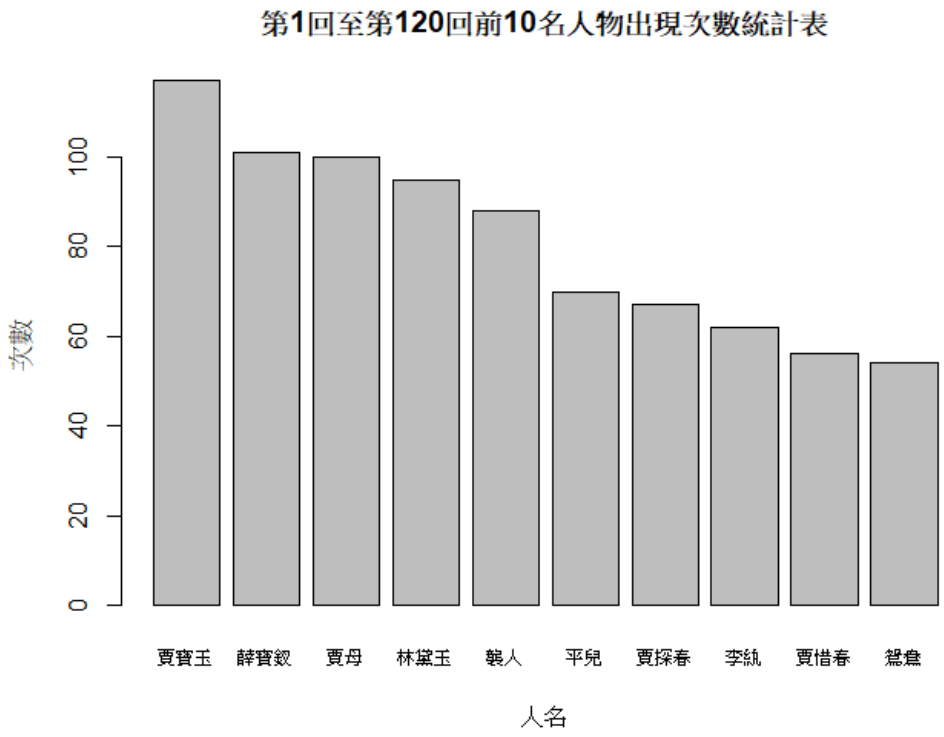
將文本讀入編譯器後，透過先前自製的篩選函數，蒐集《紅樓夢》各個篇章內容，計算每個人物在各章節總出現次數。我們可以篩選出各個人物的名稱，接著建立一個矩陣，用來紀錄每個人物是否出現在各回中，分別利用 1 跟 0 去判斷人物是否出現在該章回裡，這個矩陣將成爲後續所有視覺化結果的資料來源。

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 賈寶玉 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 林黛玉 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 薛寶釵 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 賈元春 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 賈探春 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 史湘雲 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 妙玉 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 賈迎春 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 賈惜春 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 王熙鳳 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 賈巧姐 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 李紈 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 秦可卿 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 賈母 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 劉姥姥 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 香菱 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 平兒 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 晴雯 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 襲人 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 紫鵑 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 鴛鴦 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 薛寶琴 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

表三 人物出現矩陣

接著使用 R 語言中的相關套件來製作長條圖，根據出現次數從多到少進行排序，顯示各個人物包含別名在《紅樓夢》中的出現次數。賈寶玉、薛寶釵、賈母和林黛玉等人將在這張圖中佔據較前面的位置，表示他們是故事中重要的角色。

這張長條圖爲我們提供一個直觀的方式，能更簡單地理解《紅樓夢》中各個人物的重要性和出現頻率。

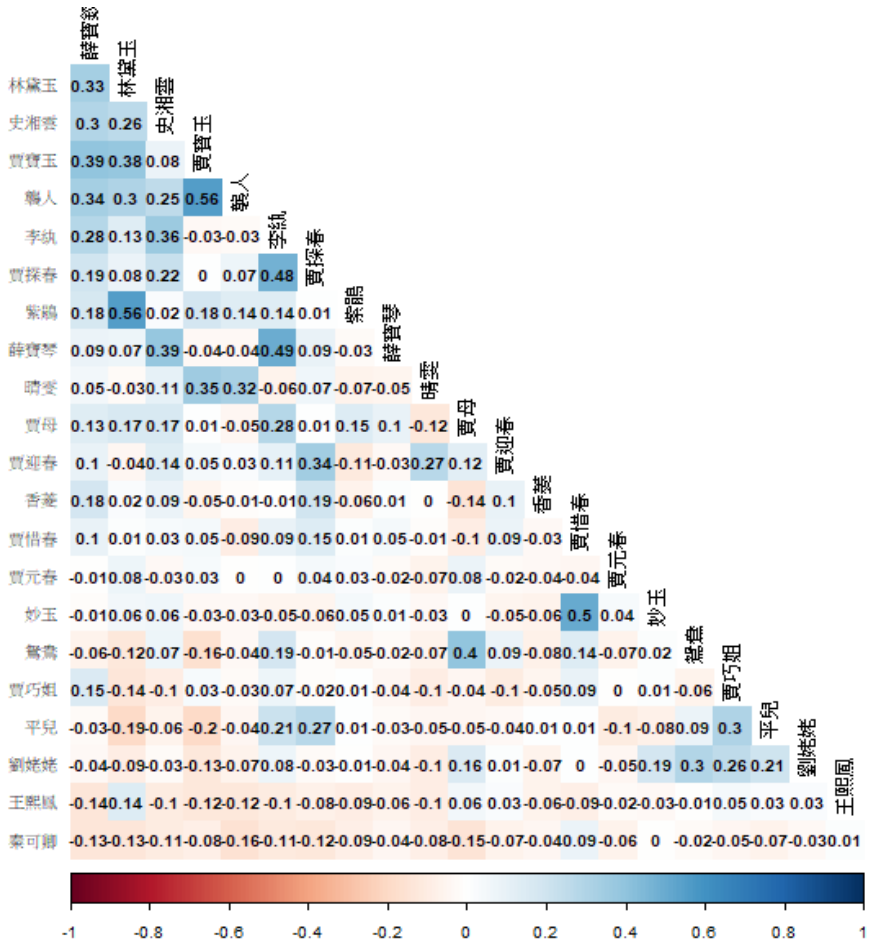



圖四 人物出現次數統計長條圖

利用上述人物出現矩陣進行初步視覺化，使用 corrplot 進行矩陣內所有角色的相關分析，其中角色之間的相關性將以數值和顏色的方式呈現。如果相關係數的絕對值愈接近 1，顏色會越深，表示相關性大；相反，相關係數愈接近 0，顏色則會越淺，表示相關性小。

以賈寶玉和林黛玉爲例，他們之間的相關係數爲 0.38 ，表示兩者之間相較於其他角色，存在密切的關係。而劉姥姥和王熙鳳的相關係數接近 0 ，表示他們與其他角色之間可能沒有太大關聯性。

此相關矩陣圖不僅能夠簡單分析各個人物之間的關聯性，還可以作爲接下來 network、MDS、PCA 模型的對照。透過比較這些視覺化方法的結果，可以了解它們在展示和解釋數據方面的差異。

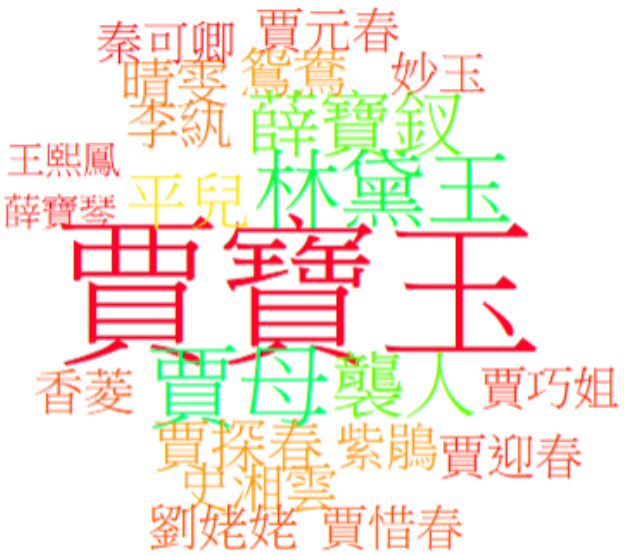



圖五 相關係數矩陣圖

## 4.2 文字雲

首先做的是人物總出現次數文字雲，除了參考上述論文[2] 的文字雲模型，並以此爲基礎進行修改，透過字體的大小來去呈現出此關鍵字的重要性，更可以一眼看出各個關鍵字的熱度。

其中人物出現次數愈多，則該人物字體愈大，且出現次數愈接近顏色也愈接近。但文字雲僅能透過字體的大小來去呈現出此關鍵字的重要性，無法藉此了解人物間彼此的關係，且當出現同樣文字大小的關鍵字，會因爲關鍵字的字元長度不同，可能讓人產生誤判的狀況。



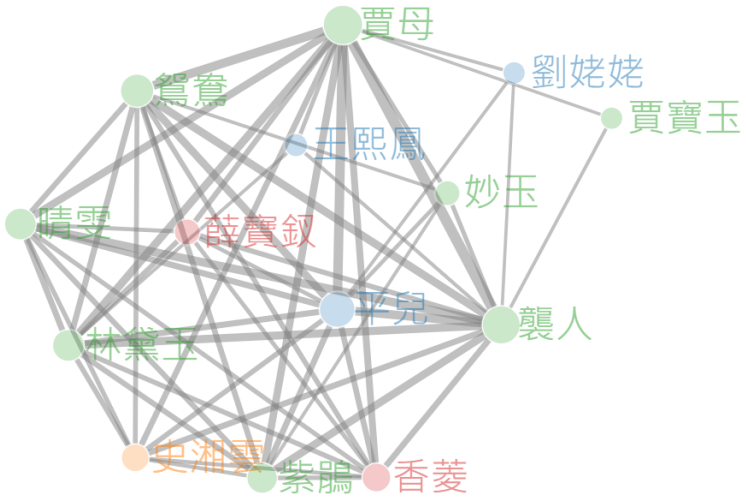

圖六 人物總出現次數文字雲

## 4.3 網路圖

由於文字雲難以顯示出人物關聯性的狀態，透過《R 語言基於共現提取《雪中悍刀行》人物關係並畫網絡圖》[4] 這篇網路文章，發現此程式碼生成的網路圖擁有高互動性，並且相對直觀的了解該人物之間的密切關係。以下將以此篇文章所介紹的程式碼進行繪製網路圖。

首先，將先載入主要的套件「networkd3 」來繪製網路圖，便能將資料透過文章中提供的程式碼來呈現。

其中，該網站的程式碼是使用一種分詞器，一種名爲「jiebaR」的套件來進行分詞。這個 R 語言的內建套件支援四種不同的分詞模式，同時還支援姓名、簡體中文、正體中文和關鍵字等功能。



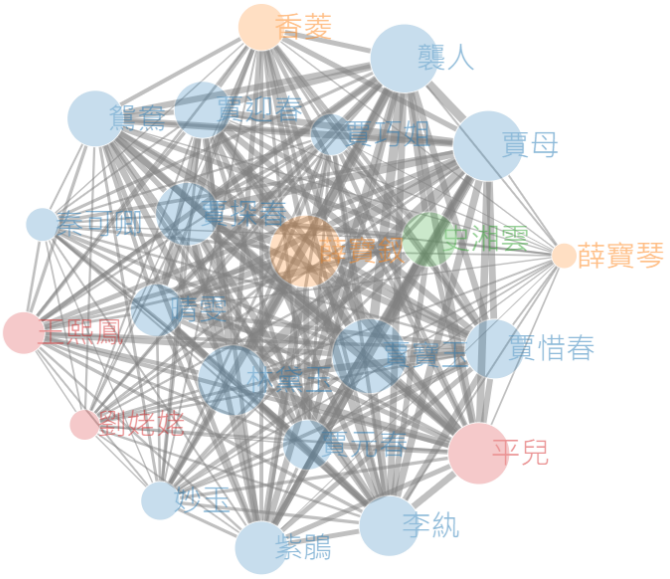

圖七 網站程式碼所構出的網路圖

在此網路圖中，節點所代表的是該人物出現的頻率，節點間的線段代表人物在同一章回中共同出現的次數，其中線段愈粗者，表示人物間一起出現次數愈高，而節點顏色所代表的則是個人物所代表的家族。

然而，在網路文章中提供的程式碼執行後，發現結果和最初預期有相當大的區別，並且與人物出現次數統計長條圖、相關係數矩陣圖以及人物總出現次數文字雲這三張先前所製作出的圖形有資料上的落差。例如，該程式碼將「賈母」這個人物判定爲《紅樓夢》中出現頻率最高的角色，但根據之前收集的資料和前面所繪製出的文字雲，出現頻率前三名依序應該是「賈寶玉」、「賈母」和「林黛玉」才對。

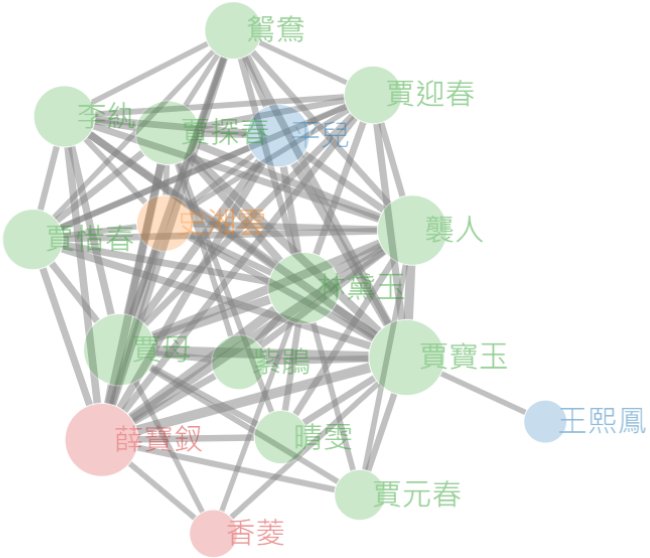
經過對程式碼的逐一排查，發現使用「jiebaR 」套件的分詞器可能對於《紅樓夢》這種接近白話文但仍帶有部分文言文的作品分詞依然不夠精確，例如林黛玉的分詞「黛玉」可能會被變成諸如「和黛玉」、「比黛玉」、「黛玉同」等字詞，使得在繪製網路圖時，網路圖容易發生人物關係不正確的狀況。

由於上述所發生的問題，必須對分詞的資料進行了修正，於是以下使用之前建立的「人物出現矩陣」進行了篩選，並將兩個人物在同一回內出現的關係設置爲 1 ，否則設置爲 0 。再根據這些關係計算頻率和相關資料重新繪製出如圖八的網路圖。




圖八 經調整後的網路圖

經過調整之後的網路圖如上所示。從圖中可以看出，這張圖已經接近最初所收集的資料，相較於 R 語言內建的分詞工具，其相對展示正常的人物關係，更貼近原本《紅樓夢》的故事發展。儘管已將有關的人物關係合併並使用粗線條表示，但發現該圖雖然仍具有互動性，但由於線條關係眾多造成閱讀障礙，因此我們將兩個人物之間的關係頻率小於 25 的線段刪去，並得到以下網路圖。



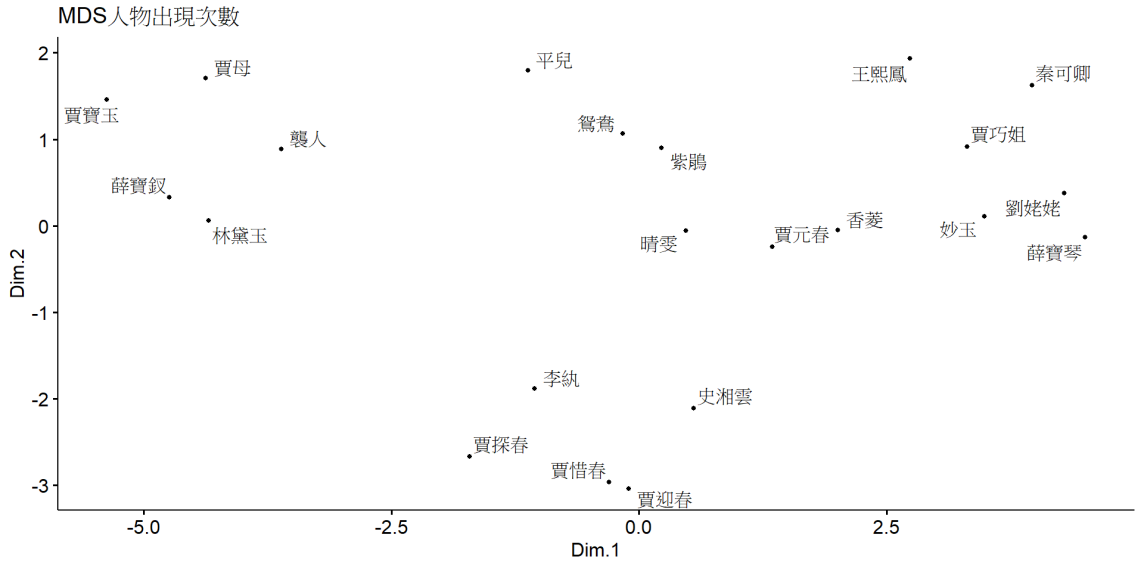

圖九 經刪減後的網路圖

從上圖中，可以看到只繪製了主要互動的人物關係，並能輕鬆地透過節點大小與線段粗細分別看出人物的出現頻率和關係。然而，圖中人物之間的距離沒有具體的含義。爲了使人物之間的距離具有意義並更好地進行分群，於是採用了多元尺度分析（MDS）方法。

## 4.4 MDS

MDS 全名爲 Multi-dimensional Scaling ，是一種多變量分析技術，也是一種降維方法，可以將高維度的資料映射到低維度，同時保留原始資料的相對關係。其運算方法是計算兩筆資料之間的歐氏距離，然後透過矩陣運算獲得物體的位置並繪製出來。

以下是使用 MDS 繪製的人物相對位置圖。從圖中可以看出，雖然資料可以大致分爲左上、右上、中下三個群組，但各群組之間似乎仍然比較鬆散。爲了能夠更好的將各群區分出來，此處引入了一種相似度的算法。




圖十 MDS 分群效果

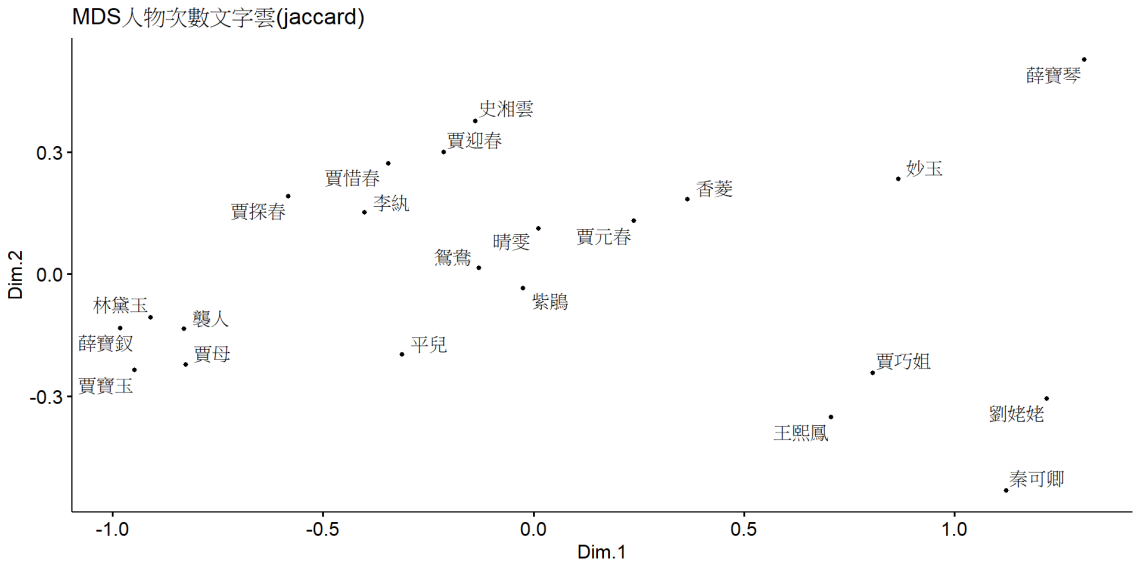
## 4.5 Jaccard

Jaccard 是一種衡量兩個物體相似程度的方法，其計算方式如式一，是爲「兩個物體的聯集除以兩個物體的交集」。將 Jaccard 方法應用於上述 MDS 的資料中，便能顯示兩個人物在同一回中是否出現。

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

式一　Jaccard 計算方式[5][8][9]

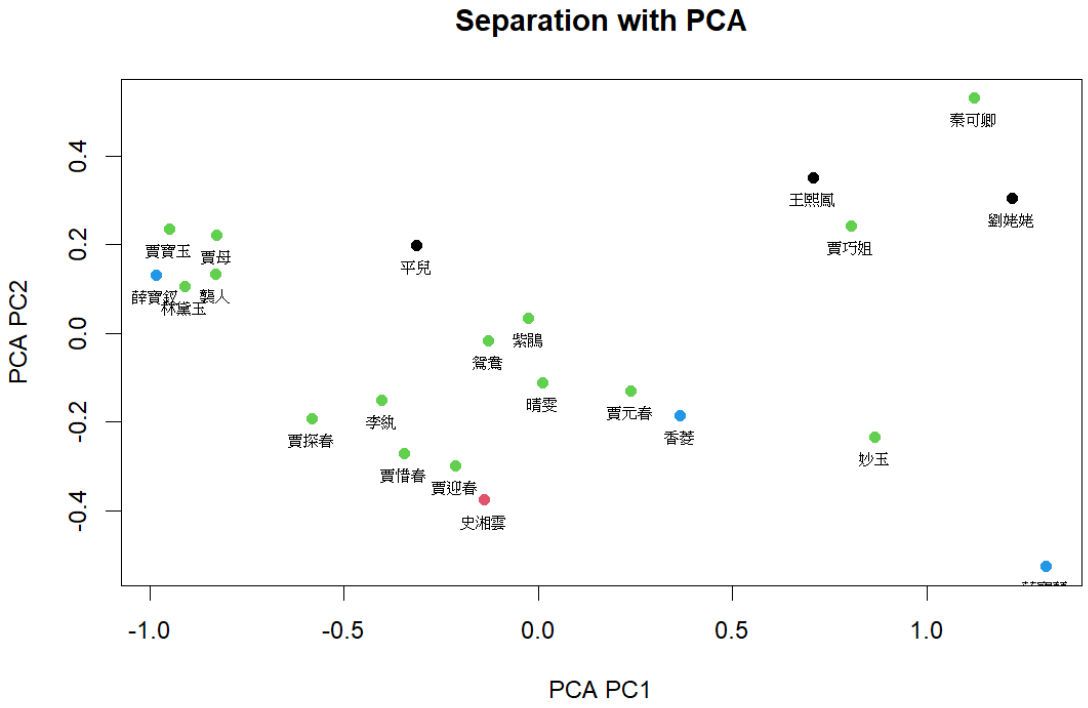
以下是添加了 Jaccard 後生成的 MDS 文字雲。從圖中可以清楚地看到群組中的點與點之間更加緊密，並且可以更好地觀察到人物之間的生活圈，以更好地判讀人物之間的關係。此外，藉由數據的相似度轉換，也可以有效將出場次數較少的人物分開，如 MDS 分群結果中妙玉看似和六到七位關係親密，但透過 Jaccard 轉換後，該人物即明顯與原先其他人物有一定距離，而小說故事狀況也是如此。




圖十一　MDS 經由 Jaccard 方法後的分群效果

## 4.6 PCA

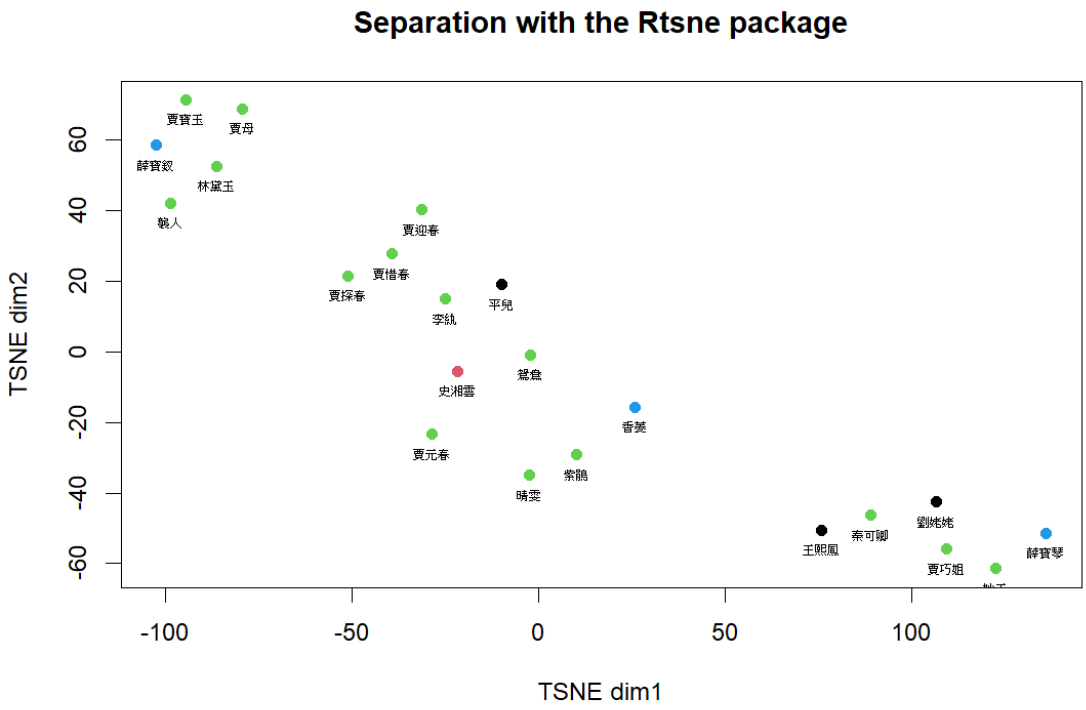
同樣先將數據做 Jaccard 後使用 PCA 方法作圖後，發現其結果與 MDS 結果相似，且具有鏡射關係。依據圖中的遠近關係，概略分爲左中右三大群，如賈母、賈寶玉等人爲一群，賈元春、賈迎春等人爲一群，剩餘將右側六位人物概括爲一群，如賈巧姐、妙玉等。值得注意的是此次分群仍有些疑慮，如圖中右側的人物間仍具有一定距離，較難判斷人物之間是否有關係。因此我們考慮使用 t-SNE，嘗試使用另一種降維方法來檢視此方法對於人物分群是否更明顯。



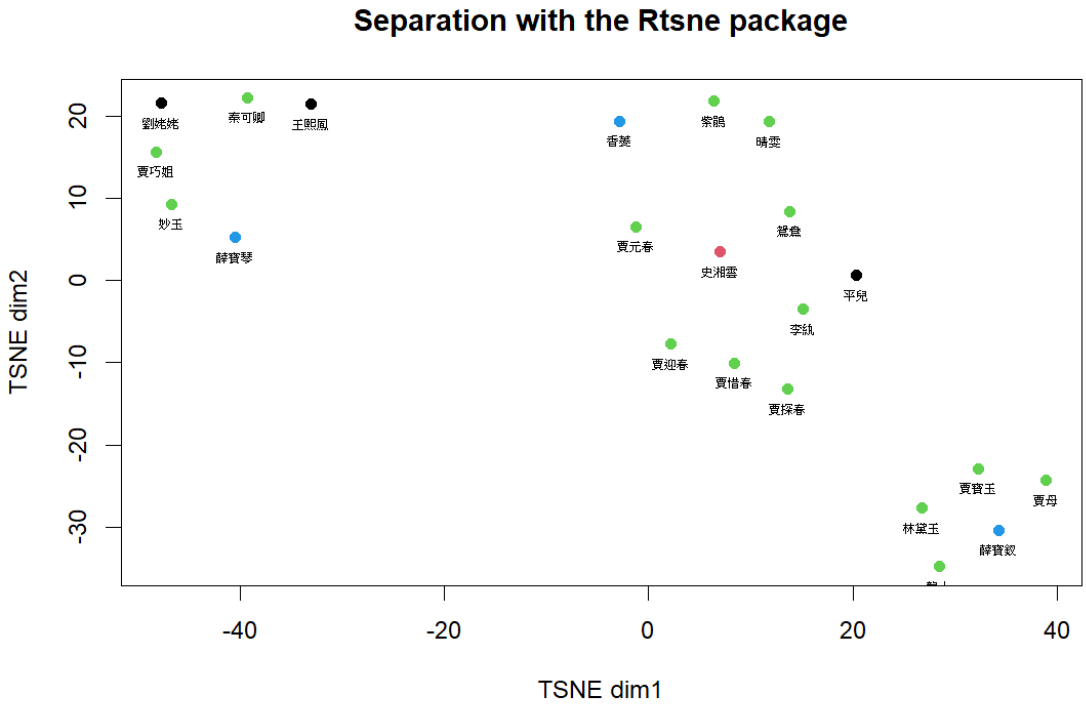

圖十二 PCA 分群效果

## 4.7 t-SNE

將數據藉由 t-SNE 方法降維，當困惑度（perplexity）設定於 5 後，發現相較於前兩個降維方法發生分群不明顯的狀況獲得改善，可以明確將圖中人物分爲三大群，推測我們的數據較適合使用非線性降維方法，使得我們的模型得到更好解釋。值得注意的是，由於 t-SNE 降維後會產生不同的點分佈，故每次執行結果皆不相同，因此需要重複執行多遍。以下兩張圖是分群較爲滿意的結果：




圖十三　t-SNE 分群結果




圖十四　t-SNE 分群結果

# 第五章 結論及建議

在決定撰寫以《紅樓夢》爲主題的文字探勘時，考慮到該作品以白話文爲主，且人物之間的關係複雜而密切，因此我們認爲這是一個理想的文本進行探勘和分析的對象。

由於分析的數據屬於二進位制非對稱類型，我們將 Jaccard 方法應用於上述的數據中，以更精確地衡量關鍵字之間的相似性。在研究中，我們分別選擇線性降維方法 (如 MDS 和 PCA ) 以及非線性降維方法 (如 t-SNE ) 進行操作，將高維度資料映射至二維度的空間，使得圖形更加容易理解，同時也盡可能保留更多的訊息。結果顯示，t-SNE 應用在人物出現次數的數據時，是三者中最能夠有效區分不同群體的方法。

本研究的方法也可以應用於網路媒體和文本挖掘領域，與最近著名的 Chat-GPT 相似，都是致力於從文本數據中獲取有價值的訊息。另外，對於文學領域而言，本研究提供了一個相對客觀的工具，能夠自動化分析文本並提取重要資訊，例如人名、虛字等等。即使對文本內容一無所知或了解有限，研究結果也能讓人了解關鍵字之間的關聯性，不論讀者對該文本的內容是否具有深入了解，這個研究成果都能爲他們提供一份有價值的資源，類似於一份「懶人包」。

研究過程中，我們還創建了人物總出現次數的數據，也就是《紅樓夢》中各人物在各回中的總出現次數。未來的相關研究可以借鑒我們的分析方式，比較和原模型差異之處，並建議探索其他降維方法，以尋找更合理的解釋模型。另外在蒐集資料時，也有找到其他研究者做過植物出現頻率的研究，因此未來可以針對其他主題進行關聯分析，找尋各項變數間彼此的關係。而在視覺化圖形結果方面也呈現出初步的懶人包效果，我們認爲相較於單單的數據呈現，圖形化的結果對於觀眾而言會是較簡便的呈現方式。在未來我們希望能夠透過 Shiny 套件或其他更好的方法實現將資料結果以互動式視覺化網頁的方式呈現，讓所有的資料都能夠與使用者互動，從而增進互動上的便利性，更有利服務人群。

# 參考文獻


[1] 曹雪芹、高鶚（約爲乾隆初年）。*紅樓夢*。（程偉元編）。維基文庫。取自https://zh.wikisource.org/zh-hant/

[2] 顏守玄（2021）。*互動式資料視覺化之文字探勘以金庸三部小說爲例 Interactive Data Visualization of Text Mining to Jin Yong Three Novels*。臺灣博碩士論文知識加值系統。取自https://hdl.handle.net/11296/9a2p75

[3] 喵喵（2020）。| 讀 | *CN* | 紅樓夢 | 人物關係表 | 四大家族 | 寶玉、黛玉、寶釵血緣關係 | *5* 分鐘弄懂紅樓夢人物關係 | 淺談紅樓夢 | *2023*。NightelfMeowMeow。取自https://nightelfmeowmeow.com/378/1105014-reading-cn-novel-dream-of-the-red-chamber-family-tree/

[4] 重明論（2021）。*R* 語言基於共現提取《雪中悍刀行》人物關係並畫網絡圖。知乎。取自https://zhuanlan.zhihu.com/p/388637831

[5] Alan H. Lipkus. (1999, October). *A Proof of the Triangle Inequality for the Tanimoto Distance*. Journal of Mathematical Chemistry. 1999, 26 (1-3): 263—265 https://link.springer.com/article/10.1023/A:1019154432472

[6] Flodel. (2012). *How to Create an Edge List from a Matrix in R?* Stack Overflow. https://stackoverflow.com/questions/13204046/how-to-create-an-edge-list-from-a-matrix-in-r

[7] Finnstats. (2021, November). *How to Calculate Jaccard Similarity in R*. R-Bloggers. https://www.r-bloggers.com/2021/11/how-to-calculate-jaccard-similarity-in-r/

[8] Michael Levandowsky, & David Winter. (1971). *Distance between Sets*. Nature. 1971, 234 (5): 34—35, https://www.nature.com/articles/234034a0

[9] Sven kosub. (2016, December). *A Note on the Triangle Inequality for the Jaccard Distance*. https://arxiv.org/pdf/1612.02696.pdf

[10] *Plotting PCA (Principal Component Analysis)*. https://cran.r-project.org/web/packages/ggfortify/vignettes/plot_pca.html

[11] *R: The R Project for Statistical Computing*. https://www.r-project.org/

[12] *RStudio*. Posit. https://www.rstudio.com

[13] *TSNE test*. https://www.bioinformatics.babraham.ac.uk/tsne/

[14] *Wordlayout: Word Layout. In wordcloud: Word Clouds.* (2019). Rdrr.Io. https://rdrr.io/rforge/wordcloud/man/wordlayout.html

# 附錄

## R 語言程式碼

本研究之 R 語言程式碼，使用 R 與 RStudio 製作。

```
# 紅樓夢人物分析

# 安裝相關係數可視化套件
install.packages("corrplot")

# 安裝文字雲所需套件
install.packages("tm")
install.packages("wordcloud")

# 安裝 MDS 所需套件
install.packages("magrittr")
install.packages("dplyr")
install.packages("ggpubr")
install.packages("MASS")

# 安裝 jaccard 套件
install.packages("vegan")

# 安裝人物關係相關套件
install.packages("networkD3")
install.packages("readxl")
install.packages("tidyverse")
install.packages("jiebaR")

# 安裝 edge list 套件
install.packages("igraph")

# 安裝 dplyr 套件 (次數)
```

```r
install.packages("dplyr")

# 安裝 Rtsne 套件
install.packages("Rtsne")

# 自定義檔名函數
file_name_function = function(i){
  if (i < 10)
    file_name <- paste0(" 紅樓夢 _ 第", 0, 0, i, " 回")
  else if (i < 100)
    file_name <- paste0(" 紅樓夢 _ 第", 0, i, " 回")
  else
    file_name <- paste0(" 紅樓夢 _ 第", i, " 回")
  return(file_name)
}

# 字數統計函數
stat_words <- function(file_name){
  returnValue(sum(nchar(gsub("[，。!?:「」；《》『』——　【】1234567890]",
  ↪  "", gsub("\\[", " ", gsub("\\]", " ", gsub("-", " ", gsub("\\
  ↪  “\\” \\ ‵\\″ ", " ", file_name)))))))
}

# 統計出現幾次特定文字函數
stat_name <- function(name_matrix, num.hei, name){ # name_matrix <-
↪  欲統計的字詞, num.hei <- 總回數, name <- 儲存的矩陣

  # 名稱的替代 (方便處理)
  name_matrix_change <- paste("person_", 1:length(name_matrix), "~",
  ↪  sep = "")
  name_matrix_change2 <- paste("person_", 1:length(name_matrix), sep
  ↪  = "")

  # 創建矩陣存放各章節人物出現次數
```

```r
  name <- matrix(0, nrow = length(name_matrix), ncol = num.hei,
  ↪ dimnames = list(name_matrix, c(1:num.hei)))

  for (chapter in 1:num.hei){
    for (i in 1:length(name_matrix)){
      temp_store_name <- gsub(name_matrix[i], name_matrix_change[i],
      ↪ get(file_name_function(chapter)), fixed = TRUE) # 替換人物
      ↪ 名稱並存入 temp_store_name
      temp_store_name1 <- strsplit(temp_store_name, "~") # 切割字串並
      ↪ 以"~" 分割
      name[i, chapter] <- length(grep(name_matrix_change2[i],
      ↪ temp_store_name1[[1]])) # 計算字串被切割幾次並儲存到 name
    }
  }
  return(name)
}

# 文字雲 自訂函數
library(tm)
library(wordcloud)
cloud <- function(input_array, cloud_title) { # imput_array <- 資料,
↪ cloud_title <- 文字雲標題
  wordcloud(names(input_array), input_array, scale = c(4,1),
  ↪ random.order = FALSE, rot.per = 0.1, min.freq = 1, colors =
  ↪ rainbow (40)) # 文字雲描述出現次數
  title(cloud_title)
}
# words 參數指定需要繪製詞云的詞；freq 參數指定每個詞的頻率；scale 參
↪ 數控制字體大小；min.freq 參數指定至少需要出現的次數；random.order
↪ 參數控制是否隨機排列詞；rot.per 參數控制字體旋轉的比例；colors 參
↪ 數指定顏色；... 爲其他附加參數

# MDS 自訂函數
mds <- function(data_matrix, mds_title){
```

```r
  # Load required packages
  library(magrittr)
  library(dplyr)
  library(ggpubr)
  # Cmpute MDS
  mds_oper <- data_matrix %>%
  dist() %>%
  cmdscale() %>%
  as_tibble()
  colnames(mds_oper) <- c("Dim.1", "Dim.2")
  # Plot MDS
  ggscatter(mds_oper, x = "Dim.1", y = "Dim.2",
          label = rownames(data_matrix),
          size = 1,
          repel = TRUE,
          main = mds_title)
}

#jaccard 自訂函數
#Jaccard 係數是用來衡量兩個集合的相似度。它的計算方法是將兩個集合的交集除以它們的聯集，結果介於 0 和 1 之間，數值越大，表示集合之間的相似度越高。
library(vegan)
jaccard <- function(data_matrix, jaccard_title){
  temp <- vegdist(data_matrix, method = "jaccard")
  temp <- as.matrix(temp)
^^Iprint(temp)
  mds(temp, jaccard_title)
}

jaccard_1 <- function(data_matrix){
  temp <- vegdist(data_matrix, method = "jaccard")
  temp <- as.matrix(temp)
  return(temp)
```

```r
}

# 統計最多的前 number 名函數
top_stat <- function(input_matrix, output_character, number){
  output_character <- sort(apply(input_matrix, 1, sum), decreasing =
  ↪  TRUE)
  output_character <- output_character[1:number]
  return(output_character)
}

# 將所有章節輸入 R
for (i in 1:120){
  file_name <- file_name_function(i)
  assign(file_name, readLines(paste0(file_name, ".txt")))
  temp <- get(file_name)
  for (j in 2:length(get(file_name))){
  temp[1] <- paste0(temp[1], temp[j])
  }
  assign(file_name, temp[1])
}

# 各章節字數統計
words_per_chapter <- matrix(0, ncol = 120, dimnames = list("words",
↪  c(1:120)))
for (i in 1:120){
  words_per_chapter[1,i] <- stat_words(get(file_name_function(i)))
}
print(words_per_chapter) # 測試字數輸出

# 定義欲探勘之章回
start_file <- 1 # 起始章節
end_file <- 120 # 結束章節

total_words <- 0
```

```r
cat(paste(paste(" 起始章節：", file_name_function(start_file)),
  paste(" 結束章節：", file_name_function(end_file)), sep = '\n'))

# 計算字數
if (start_file > end_file){
  print(paste(" 錯誤！起始章節不可大於結束章節！", start_file, ">"
    ,end_file))
  }else{
  for (i in start_file:end_file){
    total_words <- total_words + words_per_chapter[1,i]
  }
}
print(paste(" 共計", total_words, " 字（去除空白與標點符號）"))

# 定義人物名稱

# 主要人物名稱
name_main <- c(" 賈寶玉", " 林黛玉", " 薛寶釵", " 賈元春", " 賈探春",
  " 史湘雲", " 妙玉", " 賈迎春", " 賈惜春", " 王熙鳳", " 賈巧姐", "
  李紈", " 秦可卿")

# 別名
nickname_ 賈寶玉 <- c(" 寶玉", " 此石", " 寶二爺", " 怡紅公子", " 絳洞
  花王")
nickname_ 林黛玉 <- c(" 黛玉", " 顰顰", " 顰兒", " 林姑娘", " 林丫頭",
  " 瀟湘妃子")
nickname_ 薛寶釵 <- c(" 寶釵", " 蘅蕪君", " 寶姐姐", " 寶丫頭")
nickname_ 賈元春 <- c(" 元春", " 賈妃", " 元妃", " 貴妃", " 大姑娘")
nickname_ 賈探春 <- c(" 探春", " 蕉下客", " 三姑娘")
nickname_ 史湘雲 <- c(" 湘雲", " 枕霞舊友")
nickname_ 妙玉 <- c(" 妙玉", " 檻外人")
nickname_ 賈迎春 <- c(" 迎春", " 二木頭", " 二姑娘")
nickname_ 賈惜春 <- c(" 惜春", " 藕榭", " 四姑娘")
nickname_ 王熙鳳 <- c(" 熙鳳", " 鳳辣子", " 璉二奶奶")
```

```r
nickname_賈巧姐 <- c(" 巧姐", " 妞妞", " 大姐兒")
nickname_李紈 <- c(" 李紈", " 宮裁", " 稻香老農")
nickname_秦可卿 <- c(" 可卿", " 蓉大奶奶", " 兼美"," 秦氏")

# 其餘人物名稱
other_name <- c(" 賈母", " 劉姥姥", " 香菱", " 平兒", " 晴雯", " 襲人", " 紫鵑", " 鴛鴦", " 薛寶琴")

# 定義要搜尋的人物名稱
name_matrix <- " 小熊維尼" # 給定預設值 (後面不會用到)
for (i in 1:length(name_main)){
  person_names <- paste0("nickname_", name_main[i]) # 人物別名指定
  name_matrix <- c(name_matrix, get(person_names)) # 合併至 name_matrix
}
name_matrix <- c(name_matrix[-1], other_name) # 刪除" 小熊維尼" 並合併其餘人物
name_matrix

# 定義回數 120 回
num.hei <- 120

# 統計出現次數函數
name <- 0 # 宣告有 name 的存在
name <- stat_name(name_matrix, num.hei, name)
name

name_all <- name # 將所有名字存入 name_all 方便以後分析

# 合併所有重複人物
total <- 1
temp_1 <- 0
name <- matrix(0, nrow = length(c(name_main,other_name)), ncol = num.hei, dimnames = list(c(name_main,other_name), c(1:num.hei)))
```

```r
for (i in 1:length(name_main)){
  temp <- name_all[total:(total + length(get(paste0("nickname_",
  ↪  name_main[i]))) - 1), ]
  name[i,] <- apply(temp,2,sum)
  total <- total + length(get(paste0("nickname_", name_main[i])))
}
name[(i+1):length(c(name_main,other_name)), ] <-
↪  name_all[(length(name_matrix) - length(other_name) +
↪  1):(length(name_matrix)), ]

# 一回內出現最多的次數
if (start_file > end_file){
  print(paste(" 錯誤！起始章節不可大於結束章節！", start_file, ">"
  ↪  ,end_file))
  }else{
    times_max <- apply(t(name[ ,start_file:end_file]),2,max)
    times_max
    times_max <- matrix(0, nrow = length(c(name_main,other_name)),
    ↪  ncol = 2, dimnames = list(c(name_main,other_name),
    ↪  c("chapter", "times")))
    for (i in 1:dim(name)[1]){
      temp <- 0
      for (chapter in 1:num.hei){
        if (name[i,chapter] > temp){
          temp <- name[i,chapter]
          times_max[i,1] <- chapter
          times_max[i,2] <- temp
        }
      }
    }
}

# 長條圖
```

```r
name_stat <- apply(name[ ,start_file:end_file], 1, sum) # 將 name 矩陣欲探勘章節所有回數合併
number <- 10 # 前 number 名
name_stat_10 <- sort(name_stat, decreasing = T)[1:number] # 降冪排列取前 number 名人物
print(name_stat_10)
barplot(log = "y", name_stat_10, names.arg = names(name_stat_10), cex.names = 8/(as.numeric(number)), xlab = " 人名", ylab = "log(次數)", main = paste0(" 第", start_file, " 回至第", end_file, " 回", " 前", number, " 名人物總出現 log 次數統計表"))

# 人物有出現在哪一回矩陣
person_exist <- 1 * (name & TRUE) # 將人物是否出現轉爲 {0,1} 元素
person_exist
person_exist.times <- apply(person_exist, 1, sum) # 出現次數
number <- 10 # 前 number 名
person_exist.times.10 <- sort(person_exist.times, decreasing = TRUE)[1:number] # 降冪排列取前 number 名人物
print(person_exist.times.10)
barplot(person_exist.times.10, names.arg = names(person_exist.times.10), cex.names = 7/(as.numeric(number)), xlab = " 人名", ylab = " 次數", main = paste0(" 第", start_file, " 回至第", end_file, " 回", " 前", number, " 名人物出現次數統計表")) # 出現在欲探勘章節中次數最多的前 number 名人物長條圖

# 相關矩陣
if (start_file > end_file){
  print(paste(" 錯誤！起始章節不可大於結束章節！", start_file, ">" ,end_file))
  }else{
    name_cor <- cor(t(person_exist[ ,start_file:end_file]))
    #name_cor
  }

```

```r
library(corrplot)
# Create a corrgram
# 相關矩陣 (不相關集中在左下)
corrplot(name_cor,
         method = 'color',
         order = 'FPC',
         type = 'lower',
         tl.col = "black",
         diag = FALSE,
         addCoef.col = 'black',
         tl.cex = .5,
         cl.cex = .5,
         number.cex = .35,
         main = paste0(" 第", start_file, " 回至第", end_file, " 回人物的相關矩陣")
)

# 人物文字雲
cloud(name_stat, paste(" 第", start_file, " 回至第", end_file, " 回人物出現總次數文字雲"))

#  每回中出現的人物數量
apply(person_exist,2,sum)

#  平均多少字出現一位人物名字
words_per_chapter / apply(name,2,sum)

# 人物分析 (參考至網路 <https://zhuanlan.zhihu.com/p/388637831>)

library(networkD3) # 畫網絡圖
library(readxl)    # 讀取 excel
library(tidyverse) # 分組統計
library(jiebaR)    # 分詞

```

```r
# 人物名導入
name_df <- read_excel(" 人物表.xlsx")
# 多稱謂人物表導入
dupName_df <- read_excel(" 多稱謂人物.xlsx")
# 文檔導入
# texts = readLines("./雪中悍刀行.txt", encoding="gbk") # 上方已導入過，下方將進行串聯

# 將文章串聯 (自行設計)
temp <- get(file_name_function(start_file))
for (i in (start_file + 1):end_file){
  temp <- c(temp, get(file_name_function(i)))
}
texts <- temp

# 設置分詞器
engine1 = worker()
engine1$bylines = TRUE
# 分詞
seglist = segment(texts, engine1)
#head(seglist)

names = c()         # 姓名字典
relationships = list()  # 關係字典
lineNames = list()      # 每段內人物關係

for(i in 1:length(seglist)){
  line_i <- seglist[[i]]
  # 提取每個段落中的主要人物
  lineNames_i <- intersect(line_i,name_df$name)
  if(length(lineNames_i) >=2){
    # 如果該段落中包含至少兩個主要人物，則對不同主要人物形成詞對
    lineNames[[length(lineNames)+1]] <-lineNames_i
    for(i in 1:(length(lineNames_i)-1)) {
```

```
      for(j in (i+1):length(lineNames_i)) {
        if(i != j){
          # 提取共現關係
          relationships[[length(relationships)+1]] <-
           ↪  c(lineNames_i[i],lineNames_i[j])
        }
      }
    }
  }
}

head(relationships)

# 提取總人物
namelist <- unlist(lineNames)

# 提取共現關係
relationships_df <-
 ↪  data.frame(t(data.frame(relationships)),stringsAsFactors = F)
colnames(relationships_df) <- c("Sou",'Tar')
row.names(relationships_df) <- 1:nrow(relationships_df)

# 多稱謂人物合併
for (i in 1:ncol(dupName_df)) {
  name_i <- colnames(dupName_df)[i]
  # 每個主稱謂下的稱謂列表
  namelist_i <- unlist(dupName_df[name_i])
  # 將多稱謂人物轉爲主稱謂
  namelist[which(namelist %in% namelist_i)] <- name_i
  relationships_df$Sou[which(relationships_df$Sou %in% namelist_i)]
   ↪  <- name_i
  relationships_df$Tar[which(relationships_df$Tar %in% namelist_i)]
   ↪  <- name_i
}
```

```r

# 將詞對表中每行進行排序，保證每兩個人物間只有一種順序。
for(i in 1:nrow(relationships_df)){
  relationships_i <- unlist(relationships_df[i,])
  relationships_df[i,] <- relationships_i[order(relationships_i)]
}

head(relationships_df)

# 點數據
node_df <- data.frame(table(namelist))
# 設置索引
node_df <- node_df %>% mutate(Id = 0:(nrow(node_df)-1),name=namelist) %>%
  # 匹配分組-也是各個主要人物所在的勢力
  left_join(name_df)

# 設置邊節點對應列表
namline_source <- node_df %>% rename(source=Id,Sou=namelist) %>% select(Sou,source) # 起始點 ID 表
namline_target <- node_df %>% rename(target=Id,Tar=namelist) %>% select(Tar,target) # 終點 ID 表

# 邊數據統計詞頻
edge_df <- relationships_df %>% filter(Sou != Tar) %>% group_by(Sou,Tar) %>% summarise(Value=n()) %>% filter(Value > 5) ## 有錯誤
# 匹配邊節點 ID
edge_df <- edge_df %>% left_join(namline_source) %>% left_join(namline_target)
head(edge_df)

# 畫網絡圖
forceNetwork(Links = edge_df,# 線性質數據框
```

```
            Nodes = node_df,# 節點性質數據框
            Group = "group", # 節點分組 節點數據中對應的列名
            Source = "source",# 連線的源變量 邊數據中起始點 ID
            Target = "target",# 連線的目標變量 邊數據中終點 ID
            Value = "Value", # 邊的粗細值，邊數據中共現頻率列名
            NodeID = "name", # 節點名稱
            Nodesize = "Freq" , # 節點大小，節點數據框中節點頻率列名
            ### 美化部分
            fontSize = 30, # 節點文本標籤的數字字體大小（以像素爲單位）。
            linkColour="grey",# 連線顏色,black,red,blue,
            colourScale = JS("d3.scaleOrdinal(d3.schemeCategory10);"),
             #colourScale ,linkWidth,# 節點顏色,red，藍色 blue,cyan,yellow 等
            charge = -2000,# 數值表示節點排斥強度（負值）或吸引力（正值）
            opacity = 0.5, # 節點透明度
            #nodeColour="black",
            fontFamily = " 黑體",
            arrows=F, # 是否帶方向
            bounded=F, # 是否啓用限製圖像的邊框
            opacityNoHover=2, # 當鼠標懸停在其上時，節點標籤文本的不透明度比例的數值
            zoom = T, # 允許放縮，雙擊放大
            #clickAction = MyClickScript
            )

# 創建空的關係矩陣
relationship_matrix <- matrix(0, ncol = length(person_exist[,1]), nrow = length(person_exist[,1]))
colnames(relationship_matrix) <- c(name_main, other_name)
rownames(relationship_matrix) <- c(name_main, other_name)
edge_df <- c("", "")
```

```r

# 輸入資料
for(chapter in 1:num.hei){
  for (i in 1:length(person_exist[,1])) {
    for (j in 1:length(person_exist[,1])) {
      if(i != j && person_exist[i,chapter] != 0 &&
      ↪  person_exist[j,chapter] != 0){
        relationship_matrix[i,j] = 1
        relationship_matrix[j,i] = ""
      } else {
        relationship_matrix[i,j] = 0
        relationship_matrix[j,i] = ""
      }
    }
  }

# 輸出結果
print(chapter)
print(relationship_matrix)

# 創建邊數據
library(igraph)
g <- graph.adjacency(relationship_matrix)
temp <- get.edgelist(g)
edge_df <- rbind(edge_df, temp)
}
edge_df <- edge_df[-1,]

# 創建節點數據
freq <- data.frame(table(edge_df)) %>% arrange(match(edge_df,
↪  read_excel(" 人物表.xlsx")$name))
node_df <- data.frame(name = c(name_df$name), group =
↪  c(name_df$group), freq = freq[2])

```

```r
# 合併重複的邊

edge_df <- data.frame(from = c(edge_df[,1]), to = c(edge_df[,2]))
# 使用 count 函數創建包含重複次數的新數據框
edge_df_count <- count(edge_df, from, to)
# 將新數據框與原始數據框合併
edge_df_merged <- merge(edge_df, edge_df_count, by = c("from", "to"))
# 刪除重複的列
edge_df <- distinct(edge_df_merged)

# 關係次數大於 number 次
number <- 1
edge_df <- edge_df[round(log(edge_df$n*10)) > number,]

# 將邊數據轉換爲 networkD3 需要的格式
links <- data.frame(source = match(edge_df$from, node_df$name)  - 1,
                    target = match(edge_df$to, node_df$name)  - 1,
                    value = edge_df$n,
                    from = edge_df$from,
                    to = edge_df$to)

# x 軸標籤的方向
par(las=2)
# 人物出現頻率長條圖
node_df.order <- node_df[order(node_df$Freq, decreasing = TRUE),] #
↪  排序資料 (Freq)
barplot(node_df.order$Freq, names.arg = node_df.order$name, cex.names
↪  = 2/(as.numeric(length(node_df))), xlab = " 人物", ylab = " 出現
↪  頻率", main = " 人物出現頻率長條圖")

# x 軸標籤的方向
par(las=2)
# 人物關係 log 次數長條圖
```

```r
barplot(log(links$value), names.arg = paste0(links$from, "-", links$to), cex.names = 1/(as.numeric(length(links))), xlab = " 人物", ylab = " 出現頻率", main = " 人物關係 log 次數長條圖")

# x 軸標籤的方向
par(las=1)
# 人物關係次數分布長條圖
temp <- cut(links$value, breaks = seq(from = 0, to = 100, by = 10), right = FALSE, labels = seq(from = 0, to = 100, by = 10)[-1]) %>% table()
barplot(temp, xlab = " 間距", ylab = " 次數", main = " 人物關係次數分布長條圖")

# 創建網絡圖
forceNetwork(Links = links,# 線性質數據框
             Nodes = node_df,# 節點性質數據框
             Group = "group", # 節點分組 節點數據中對應的列名
             Source = "source",# 連線的源變量 邊數據中起始點 ID
             Target = "target",# 連線的目標變量 邊數據中終點 ID
             Value = "value", # 邊的粗細值，邊數據中共現頻率列名
             NodeID = "name", # 節點名稱
             Nodesize = "Freq" , # 節點大小，節點數據框中節點頻率列名
             ### 美化部分
             fontSize = 30, # 節點文本標籤的數字字體大小（以像素爲單位）。
             linkColour="grey",# 連線顏色,black,red,blue,
             colourScale = JS("d3.scaleOrdinal(d3.schemeCategory10);"),
              #colourScale ,linkWidth,# 節點顏色,red，藍色 blue,cyan,yellow 等
             charge = -2000,# 數值表示節點排斥強度（負值）或吸引力（正值）
             opacity = 0.5, # 節點透明度
             #nodeColour="black",
```

```
490             fontFamily = " 黑體",
491             arrows=F, # 是否帶方向
492             bounded=F, # 是否啓用限製圖像的邊框
493             opacityNoHover=2, # 當鼠標懸停在其上時，節點標籤文本的不
                ↪ 透明度比例的數值
494             zoom = T, # 允許放縮，雙擊放大
495             #clickAction = MyClickScript
496             )
497
498 # 創建空的關係矩陣
499 relationship_matrix <- matrix(0, ncol = length(person_exist[,1]),
    ↪ nrow = length(person_exist[,1]))
500 colnames(relationship_matrix) <- c(name_main, other_name)
501 rownames(relationship_matrix) <- c(name_main, other_name)
502 relationship_df_name <- c("", "")
503
504 # 輸入資料
505 for(chapter in 1:num.hei){
506   for (i in 1:length(person_exist[,1])) {
507     for (j in 1:length(person_exist[,1])) {
508       if(i != j && person_exist[i,chapter] != 0 &&
          ↪ person_exist[j,chapter] != 0){
509         relationship_matrix[i,j] = 1
510         relationship_matrix[j,i] = ""
511       } else {
512         relationship_matrix[i,j] = 0
513         relationship_matrix[j,i] = ""
514       }
515     }
516   }
517
518 # 輸出結果
519 print(chapter)
520 print(relationship_matrix)
```

```r

# edge name list
library(igraph)
g <- graph.adjacency(relationship_matrix)
temp <- get.edgelist(g)
relationship_df_name <- rbind(relationship_df_name, temp)
}
relationship_df_name <- relationship_df_name[-1,]
colnames(relationship_df_name) <- c("Sou",'Tar')
row.names(relationship_df_name) <- 1:nrow(relationship_df_name)

# 點數據
node_df_name <- data.frame(table(relationship_df_name))
# 設置索引
node_df_name <- node_df_name %>% mutate(Id =
↪  0:(nrow(node_df_name)-1),name=relationship_df_name) %>%
  # 匹配分組-也是各個主要人物所在的勢力
  left_join(name_df)

# 設置邊節點對應列表
namline_source <- node_df_name %>%
↪  rename(source=Id,Sou=relationship_df_name) %>% select(Sou,source)
↪  # 起始   ID 表
namline_target <- node_df_name %>%
↪  rename(target=Id,Tar=relationship_df_name) %>% select(Tar,target)
↪  #      ID 表

relationship_df_name_stat <- data.frame(from =
↪  c(relationship_df_name[,1]), to = c(relationship_df_name[,2]))
# 使用 count 函數創建包含重複次數的新數據框
relationship_df_name_count <- count(relationship_df_name_stat, from,
↪  to)
# 將新數據框與原始數據框合併
```

```r
relationship_df_name_merged <- merge(relationship_df_name_stat, relationship_df_name_count, by = c("from", "to"))
# 刪除重複的列
relationship_df_name_stat <- distinct(relationship_df_name_merged)

# 將邊數據轉換爲 networkD3 需要的格式
edge_df_name <- data.frame(source = match(relationship_df_name_stat$from, node_df_name$name) - 1,
                           target = match(relationship_df_name_stat$to, node_df_name$name) - 1,
                           value = relationship_df_name_stat$n)


# 邊數據統計詞頻
edge_df_name <- edge_df_name %>% filter(value > 25)

# 畫網絡圖
forceNetwork(Links = edge_df_name, # 線性質數據框
             Nodes = node_df_name, # 節點性質數據框
             Group = "group", # 節點分組 節點數據中對應的列名
             Source = "source", # 連線的源變量 邊數據中起始點 ID
             Target = "target", # 連線的目標變量 邊數據中終點 ID
             Value = "value", # 邊的粗細值，邊數據中共現頻率列名
             NodeID = "name", # 節點名稱
             Nodesize = "Freq" , # 節點大小，節點數據框中節點頻率列名
             ### 美化部分
            fontSize = 30, # 節點文本標籤的數字字體大小（以像素爲單位）。
             linkColour="grey",# 連線顏色,black,red,blue,
             colourScale = JS("d3.scaleOrdinal(d3.schemeCategory10);"),
              #colourScale ,linkWidth, # 節點顏色,red，藍色 blue,cyan,yellow 等
```

```
574             charge = -2000, # 數值表示節點排斥強度（負值）或吸引力（正
                ↪ 值）
575             opacity = 0.5, # 節點透明度
576             # nodeColour="black",
577             fontFamily = " 黑體",
578             arrows=F, # 是否帶方向
579             bounded=F, # 是否啓用限製圖像的邊框
580             opacityNoHover=2, # 當鼠標懸停在其上時，節點標籤文本的不
                ↪ 透明度比例的數值
581             zoom = T, # 允許放縮，雙擊放大
582             # clickAction = MyClickScript
583             )
584
585 # MDS
586 mds(name, "MDS 人物總出現次數")
587 mds(person_exist, "MDS 人物出現次數")
588
589 # 人物 jaccard 方法
590 jaccard(name, "MDS 人物總次數文字雲 (jaccard)")
591 jaccard(person_exist, "MDS 人物次數文字雲 (jaccard)")
592
593 # PCA
594 person_exist.jaccard <- jaccard_1(person_exist)
595 pca.result <- prcomp(person_exist.jaccard)
596 plot(
597   pca.result$x[,1],
598   pca.result$x[,2],
599   pch=19, xlab="PCA PC1",
600   ylab="PCA PC2",
601   main="Separation with PCA",
602   col=as.factor(gsub("_.*","",node_df$group))
603   )
```

```
text(pca.result$x[,1],
  pca.result$x[,2],gsub("_.*","",colnames(person_exist.jaccard)),pos=1,
  cex=0.7)

library(Rtsne)
rtsne.result <- Rtsne(person_exist.jaccard, perplexity = 5)
plot(
  rtsne.result$Y,
  col=as.factor(gsub("_.*","",node_df$group)),
  pch=19,
  xlab="TSNE dim1",
  ylab="TSNE dim2",
  main="Separation with the Rtsne package"
  )
text(rtsne.result$Y[,1],rtsne.result$Y[,2],gsub("_.*","",colnames(
person_exist.jaccard)),pos=1, cex=0.5)
```

## 資料檔案

程式碼中需導入的資料檔案:

**《紅樓夢》各章回**（資料來源：維基文庫[1]）

紅樓夢 _ 第 001 回.txt

第一回　甄士隱夢幻識通靈　賈雨村風塵懷閨秀

【甲戌、庚、蒙批：此開卷第一回也。】

作者自：因曾歷過一番夢幻之後，故將眞事隱去，而借通靈之說，撰此《石頭記》一書也。故曰「甄士隱夢幻識通靈」。但書中所記何事，又因何而撰是書哉？自又：今風塵碌碌，一事無成，忽念及當日所有之女子，一一細推了去，覺其行止見識，皆出於我之上。何我堂堂之鬚眉，曾不若彼裙釵哉！實愧則有餘，悔又無益之大無可奈何之日也！當此時，則自欲將已往所賴，上賴天恩，下承祖德，錦衣紈　之時、飫甘饜美肥之日，背父母教育之恩，負師兄規訓之德，已至今日一事無成、半生潦倒之罪，編述一記，以告普天下人。我之罪固不能免，然閨閣中本自歷歷有人，萬不可因我之不肖，自護其短，則一併使其泯滅也。雖今日之茆椽蓬牖，瓦灶繩床，其風晨月夕，階柳庭花，亦未有傷於我之襟懷筆墨者。雖我未學，下筆無文，何爲不用假語村言，敷演出一段故事來，以悅人之耳目哉。故曰「風塵懷閨秀」，乃是第一回題綱正義也。開卷即曰「風塵懷閨秀」，則知作者本意原爲記述當日閨友閨情，並非怨世罵時之書矣。雖一時有涉於世態，然亦不得不敘者，但非其本旨耳，閱者切記之。

⋮

不知有何禍事，且聽下回分解。

表四 紅樓夢 _ 第 001 回.txt（節錄）

紅樓夢 _ 第 002 回.txt
紅樓夢 _ 第 003 回.txt
紅樓夢 _ 第 004 回.txt
⋮
紅樓夢 _ 第 120 回.txt

**人物表.xlsx**

| name | type | group |
|---|---|---|
| 賈寶玉 | nr | 賈家 |
| 林黛玉 | nr | 賈家 |
| 薛寶釵 | nr | 薛家 |
| 賈元春 | nr | 賈家 |
| 賈探春 | nr | 賈家 |
| 史湘雲 | nr | 史家 |
| 妙玉 | nr | 賈家 |
| 賈迎春 | nr | 賈家 |
| 賈惜春 | nr | 賈家 |
| 王熙鳳 | nr | 王家 |
| 賈巧姐 | nr | 賈家 |
| 李紈 | nr | 賈家 |
| 秦可卿 | nr | 賈家 |
| 賈母 | nr | 賈家 |
| 劉姥姥 | nr | 王家 |
| 香菱 | nr | 薛家 |
| 平兒 | nr | 王家 |
| 晴雯 | nr | 賈家 |
| 襲人 | nr | 賈家 |
| 紫鵑 | nr | 賈家 |
| 鴛鴦 | nr | 賈家 |
| 薛寶琴 | nr | 薛家 |

表五 人物表.xlsx

**多稱謂人物.xlsx**

| 賈寶玉 | 林黛玉 | 薛寶釵 | 賈元春 | 賈探春 | 史湘雲 | 妙玉 | 賈迎春 | 賈惜春 | 王熙鳳 | 賈巧姐 | 李紈 | 秦可卿 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 賈寶玉 | 林黛玉 | 薛寶釵 | 賈元春 | 賈探春 | 史湘雲 | 妙玉 | 賈迎春 | 賈惜春 | 王熙鳳 | 賈巧姐 | 李紈 | 秦可卿 |
| 寶玉 | 黛玉 | 寶釵 | 元春 | 探春 | 湘雲 | 妙玉 | 迎春 | 惜春 | 熙鳳 | 巧姐 | 李紈 | 可卿 |
| 此石 | 顰顰 | 蘅蕪君 | 賈妃 | 蕉下客 | 枕霞舊友 | 檻外人 | 二木頭 | 藕榭 | 鳳辣子 | 妞妞 | 宮裁 | 蓉大奶奶 |
| 寶二爺 | 顰兒 | 寶姐姐 | 元妃 | 三姑娘 | | | 二姑娘 | 四姑娘 | 璉二奶奶 | 大姐兒 | 稻香老農 | 兼美 |
| 怡紅公子 | 林姑娘 | 寶丫頭 | 貴妃 | | | | | | | | | 秦氏 |
| 絳洞花王 | 林丫頭 | | 大姑娘 | | | | | | | | | |
| | 瀟湘妃子 | | | | | | | | | | | |

表六　多稱謂人物.xlsx