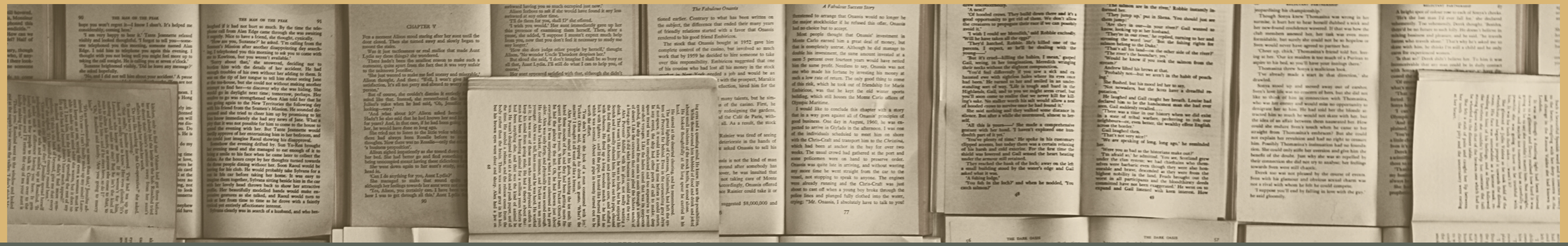
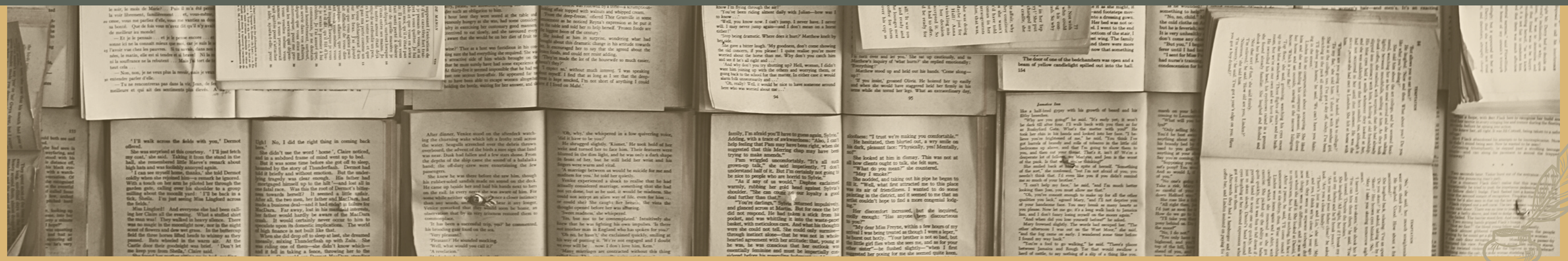


# 十九與二十世紀各國經典文學作品高頻字與主題標記分析

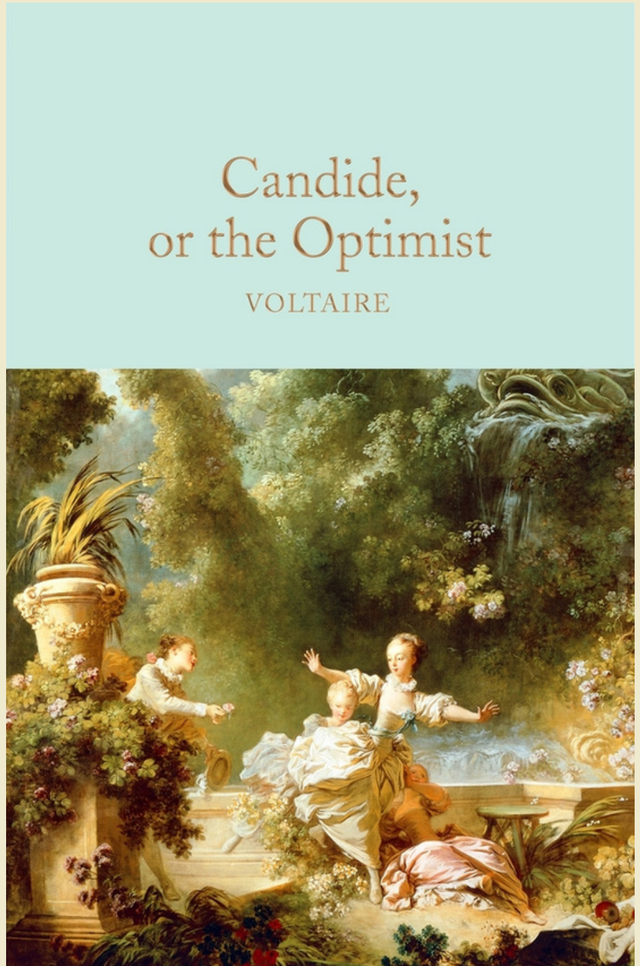
組名：Sherry Don't Go



02

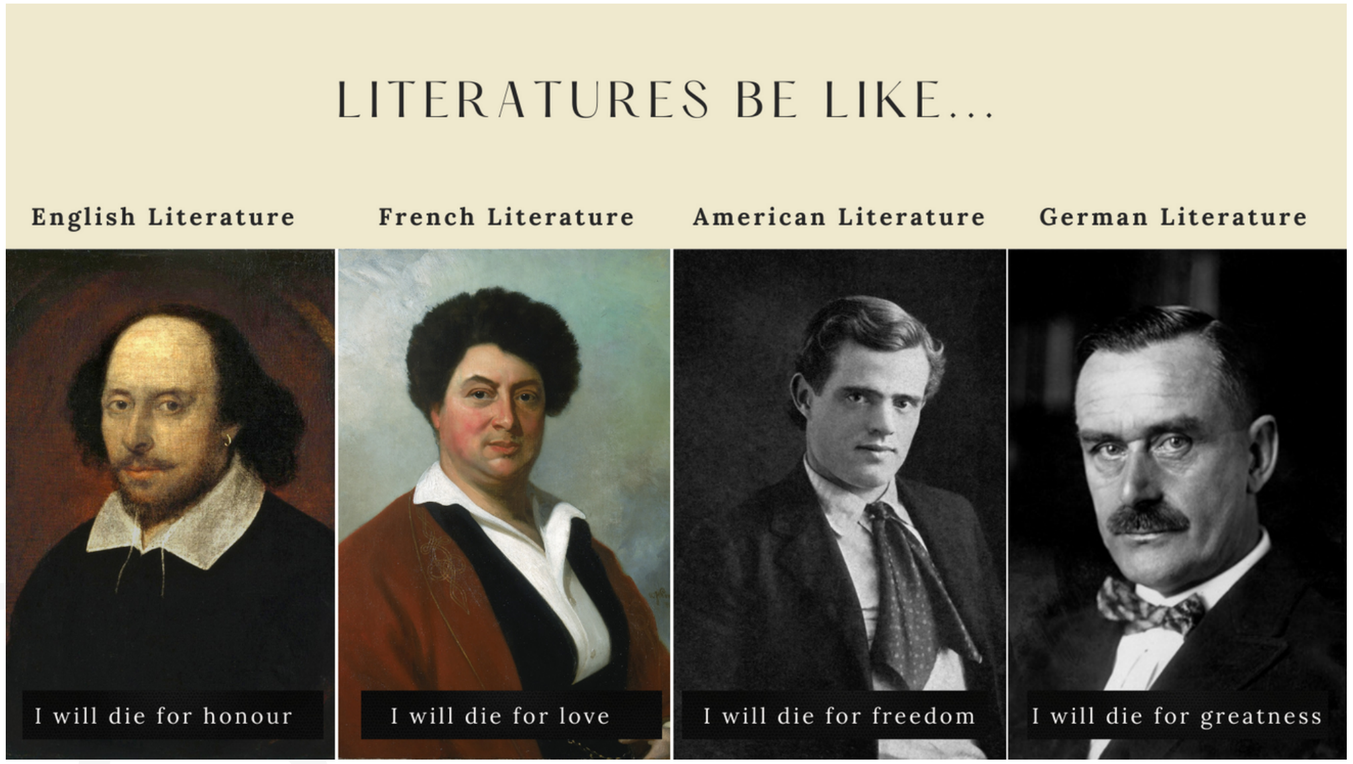
# *Content*





# Motivation & Goal

Looking back what we've learned in all those literature courses....




MOTIVATION

1. 想要回顧4年在外文系讀過的文學作品
2. 想知道各國文學的主題是否符合某些的刻板印象

GOAL

1. 找出不同國家的文學作品中常見的用字與主題
2.分析專案結果，檢視是否符合我們的預期

# *Book List*

## ENGLISH LITERATURE

- A Tale of Two Cities
- Dubliners
- Gulliver's Travels into Several Remote Nations of the World
- Monday or Tuesday
- Pride and Prejudice

## AMERICAN LITERATURE

- Adventures of Huckleberry Finn
- Little Women
- Moby Dick
- The Great Gatsby
- The Scarlet Letter

## GERMAN LITERATURE

- Metamorphosis
- Royal Highness
- Siddhartha
- The Sorrows of Young Werther
- The Trial

## FRENCH LITERATURE

- Candide
- Madame Bovary
- Notre-Dame de Paris
- The Flowers of Evil
- The Phantom of the Opera

05

# METHOD & PROCESS

## STEP 1 - DATA ASSEMBLE

Gutenbergr

Web scrape: httr, rvest

## STEP 2 - DATA PROCESSING

tidyverse, stopwords,  lemmatization

## STEP 3 - TF-IDF

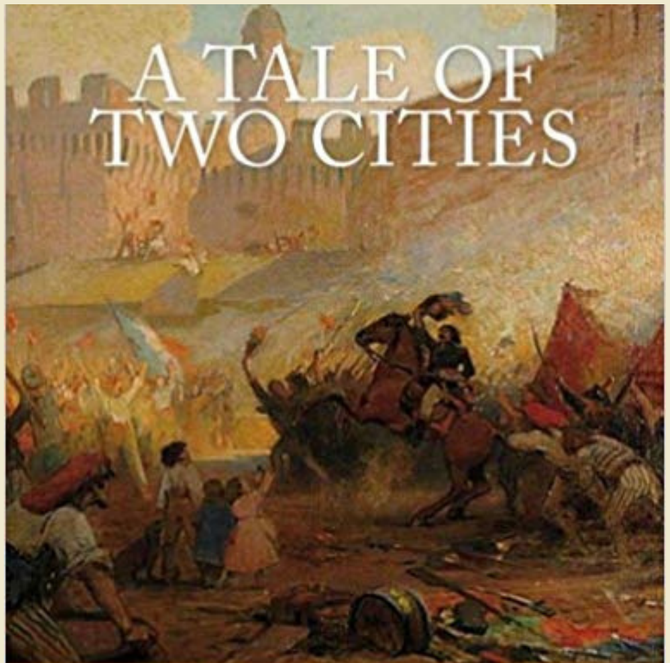
tidytext

## STEP 4 - TOPIC MODELING

topicmodels

## STEP 5 - VISUALISATION

ggplot2, shiny




# Step 1: Data Assemble

```
# some books are not in the gutenbergr package :(((
if (title == "character(0)"){
    # using web scraping to download from site
    title_url<-paste0("https://www.gutenberg.org/ebooks/",i)
    url<-paste0("https://www.gutenberg.org/files/",i,"/",i,"-h/",i,"-h.htm")
    # finding title
    title_new<-GET(title_url) %>%
        content() %>%
        html_nodes("#content > h1") %>%
        html_text() %>%
        str_to_title() %>%
        str_remove_all("By .+") %>%
        str_trim() %>%
        str_replace_all(" ", "_")
    # assigning text object according to its title
    assign(title_new, GET(url) %>%
               content() %>%
               html_nodes("p") %>%
               html_text() %>%
               str_trim() %>%
               as.data.frame() %>%
               setNames("text") %>%
               mutate(book=title_new) %>%
               select(book, text), envir = .GlobalEnv)
    # adding text object to a book list
    shelf<-append(shelf, title_new)
```

```
# assigning text object according to its title
assign(title, gutenberg_works()%>%
           filter(gutenberg_id == i)%>%
           select(gutenberg_id)%>%
           as.vector()%>%
           gutenberg_download(strip = TRUE)%>%
           mutate(book=title) %>%
           select(book, text), envir = .GlobalEnv)
```

▲ 使用Gutenbergr 套件下載文章

◀ 沒有收錄在套件中的文章則直接至網站爬文本

# Step 2: Data Processing

```
# remove stop words with anti_join
other_stopwords<-c("de", "von", "er", "mo", "hl", "ly", "ter", "zu", "ye", "da", "la", "aide")%>%
    as.vector()%>%
    as.data.frame()
colnames(other_stopwords)<-"word"
stop_words_new<-stop_words %>%
    select(word) %>%
    rbind(other_stopwords)
```

▲ 建立新的stop_words data frame

▼ 進行文本前處理

```
text %>%
    unnest_tokens(word, text, to_lower=F) %>% # unnest tokens (to_lower=F so for easier name removal)
    mutate(word = lemmatize_words(word)) %>%
    anti_join(stop_words_new, by=c("word"="word")) %>%
    filter(!grepl(pattern = ".*[A-Z].*", x=word)) %>% # remove capitalized name entities
    filter(!grepl(pattern = ".*[0-9_''].*", x=word)), envir = .GlobalEnv) # remove numbers
```

# Step 3: TF-IDF

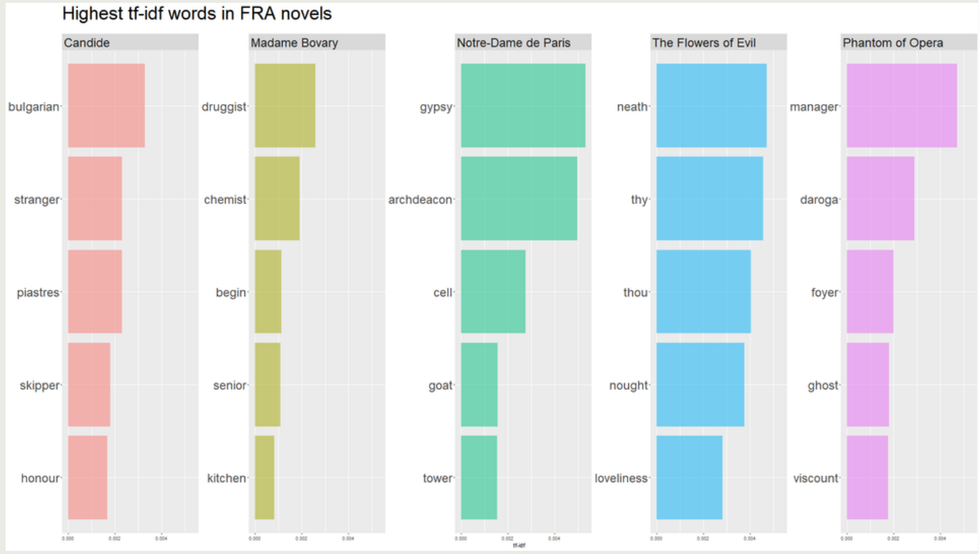
08

```
assign(paste0("tf_idf_", country),
       get(paste0("cleaned_books_", country)) %>%
           count(book, word, sort = TRUE) %>%
           bind_tf_idf(word, book, n) %>%
           arrange(-tf_idf) %>%
           group_by(book) %>%
           top_n(topwords) %>%
           ungroup %>%
           mutate(word = reorder_within(word, tf_idf, book)), envir = .GlobalEnv)
```

▼ 圖像化TF-IDF

```
plot<-get(paste0("tf_idf_", country)) %>%
    ggplot(aes(word, tf_idf, fill = book)) +
    geom_col(alpha = 0.5, show.legend = FALSE) +
    facet_wrap(~ book, scales = "free_y", ncol = 5,
               labeller = labeller(book = book_acronym)) +
    scale_x_reordered() +
    coord_flip() +
    theme(plot.title = element_text(size=36),
          strip.text.x = element_text(size=24, hjust = 0.0),
          axis.text.y.left = element_text(size=24, hjust = 1.0)) +
    labs(x = NULL, y = "tf-idf",
         title = paste("Highest tf-idf words in", country, "novels"),
         subtitle = "")
```

▲ 建立詞頻表



# Step 4: Topic Modeling

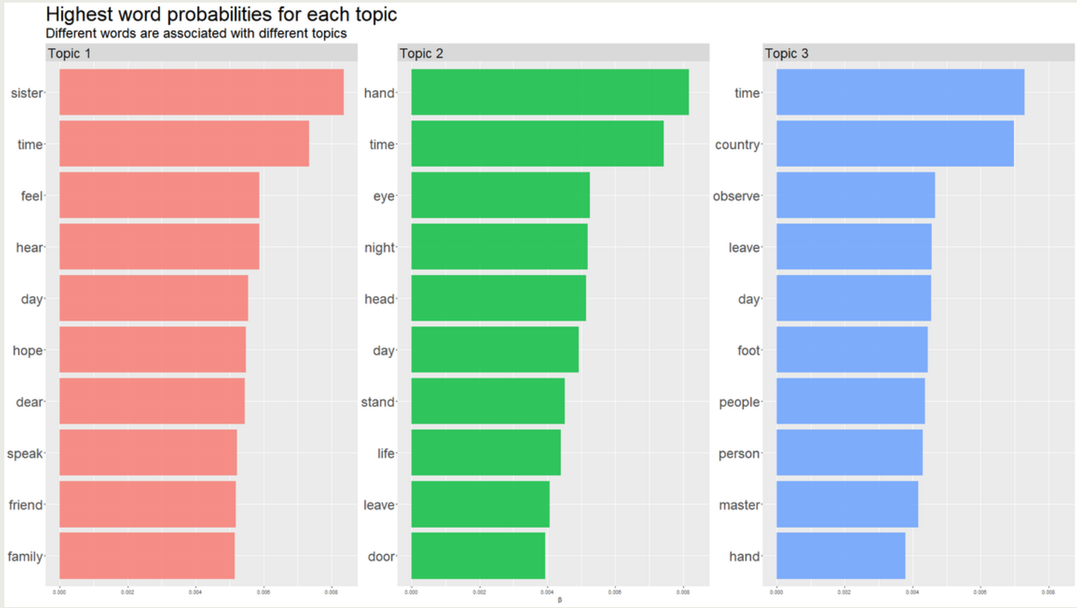
```
topic_modeling<-function(x){
    XXX<-deparse(substitute(x))%>%strsplit("_")%>%unlist()
    dfm <- get(paste0("cleaned_books_", XXX[3])) %>%
        count(book, word, sort = TRUE) %>%
        mutate(book=as.character(book))%>%
        cast_dfm(book, word, n)

    sparse <- get(paste0("cleaned_books_", XXX[3])) %>%
        count(book, word, sort = TRUE) %>%
        mutate(book=as.character(book))%>%
        cast_sparse(book, word, n)

    assign(paste0("topic_model_", XXX[3]), stm(dfm, K = 3, verbose = FALSE, init.type = "Spectral"))

    td_beta <- tidy(get(paste0("topic_model_", XXX[3])))

    td_gamma <- tidy(get(paste0("topic_model_", XXX[3])), matrix = "gamma",
                     document_names = rownames(dfm))
```

▲ 建立topic model

# ≡ step 5 - Visualisation

▼圖像化topic model

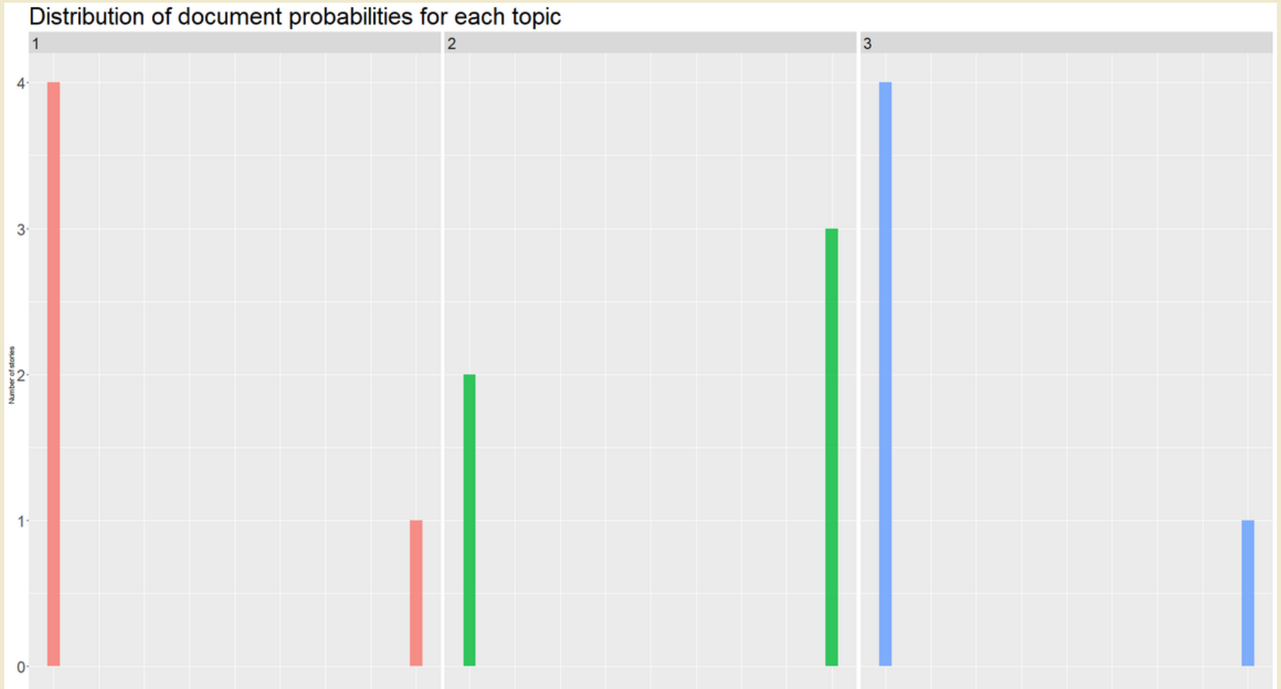
```
plot1<-td_beta %>%
    group_by(topic) %>%
    top_n(10, beta) %>%
    ungroup() %>%
    mutate(topic = paste0("Topic ", topic),
           term = reorder_within(term, beta, topic)) %>%
    ggplot(aes(term, beta, fill = as.factor(topic))) +
    geom_col(alpha = 0.8, show.legend = FALSE) +
    facet_wrap(~ topic, scales = "free_y") +
    coord_flip() +
    scale_x_reordered() +
    theme(plot.title = element_text(size=36),
          plot.subtitle = element_text(size=24),
          strip.text.x = element_text(size=24, hjust = 0.0),
          axis.text.y.left = element_text(size=24, hjust = 1.0)) +
    labs(x = NULL, y = expression(beta),
         title = "Highest word probabilities for each topic",
         subtitle = "Different words are associated with different topics")
```



# step 5 - Visualisation

建立主題分布圖

```
plot2<-ggplot(td_gamma, aes(gamma, fill = as.factor(topic))) +
    geom_histogram(alpha = 0.8, show.legend = FALSE) +
    facet_wrap(~ topic, ncol = 3) +
    theme(plot.title = element_text(size=36),
          plot.subtitle = element_text(size=24),
          strip.text.x = element_text(size=24, hjust = 0.0),
          axis.text.y.left = element_text(size=24, hjust = 1.0)) +
    labs(title = "Distribution of document probabilities for each topic",
         y = "Number of stories", x = expression(gamma))
```



# step 5 - Visualisation

```r
library(shiny)
library(shinythemes)
library(shinyWidgets)

# Define UI (web interface)
ui <- fluidPage(theme = shinytheme("sandstone"),
                navbarPage("",
                           tabPanel{⟷},
                           tabPanel{⟷},
                           tabPanel{⟷})

# Define server logic (how R process user's inputs)
server <- function(input, output) {
    output$dfll <- renderImage({⟷})
    output$bookOut <- renderText(input$bookname)
    output$countryOut <- renderText(input$country)
    output$country_tfidf <- renderText({paste0(input$country1,"  TF-IDF")})
    output$country_topic <- renderText({paste0(input$country2,"  Topic Modeling")})
    output$tfidf_rank <- renderImage({⟷})
    output$TOPIC <- renderImage({⟷})
    output$TOPIC2 <- renderImage({⟷})
}

# Run the application
shinyApp(ui = ui, server = server)
```

▲ 建立Shiny頁面

▶ Shiny首頁

# Shiny app

# Analysis

## TF-IDF

- 能夠概覽作品主題、手法
- 頑童歷險記：黑人方言
  ex. en(and), wuz(was)
- 流浪者之歌：修行、對真理的頓悟 ex.  oneness, ascetic

## TOPIC MODELING

- 即便同個國家的書，主題上幾乎沒有相關
- 把五本書分到三個主題下，會造成模型結果不佳
- 五個主題間區別不夠顯著，詞與詞關聯度低

## CONCLUSION

- 即便處在同時代和國家，不同作者表達的手法，很難如同前面的迷因圖，用一句話概括
- 以德國文學為例，有卡夫卡荒誕的寓言式故事，也有少年維特為愛殉道的理想主義 ≠ I will die for greatness

# Limitation

TOPIC MODEL有太多無效字

- 無法解釋topic幾乎都會包含time跟eye這兩個字

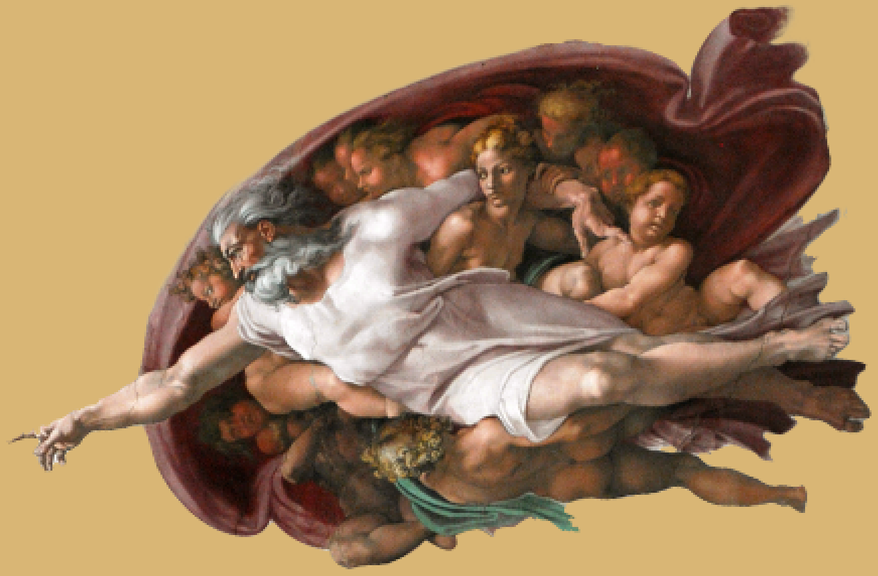
非英文文學的作者許多代表作在GUTENBERG上無英文翻譯

TOPIC MODEL效果不佳

- 主題間區別不顯著
- 主題內詞與詞關聯不高

文本長度不一，可能影響TOPIC MODEL結果

| 姓名 | 工作內容 | 投入程度 |
| --- | --- | --- |
| 陳聯輝 (code組) | 資料蒐集、資料處理、TF-IDF、Topic Modeling、結果視覺化 | 6 |
| | 簡報內容發想、圖片製作 | 4 |
| 周昕妤 (code組) | 資料蒐集、資料處理、Shiny介面製作、結果視覺化 | 6 |
| | 簡報內容發想、圖片製作 | 4 |
| 黃彙茹 (報告組) | 資料處理、Shiny介面製作 | 4 |
| | 簡報製作、結果分析<br>報告專案成果、分析與限制 | 6 |
| 吳錤 (報告組) | 資料處理、TF-IDF、Topic Modeling | 4 |
| | 簡報製作<br>報告專案動機、過程與方法 | 6 |



# Thank you!