Chapter 5

Akanksha Garlapati

2024-09-27

1. **Find the ID of the novel “Pride and Prejudice”** using str\_detect:

* library(gutenbergr)  
  library(dplyr)
* ##   
  ## Attaching package: 'dplyr'
* ## The following objects are masked from 'package:stats':  
  ##   
  ## filter, lag
* ## The following objects are masked from 'package:base':  
  ##   
  ## intersect, setdiff, setequal, union
* library(stringr)  
  pride\_id <- gutenberg\_metadata %>%  
   filter(str\_detect(title, "Pride and Prejudice"))  
  pride\_id
* ## # A tibble: 7 × 8  
  ## gutenberg\_id title author gutenberg\_author\_id language gutenberg\_bookshelf  
  ## <int> <chr> <chr> <int> <chr> <chr>   
  ## 1 1342 Pride an… Auste… 68 en "Best Books Ever L…  
  ## 2 20686 Pride an… Auste… 68 en "Best Books Ever L…  
  ## 3 20687 Pride an… Auste… 68 en "Harvard Classics/…  
  ## 4 26301 Pride an… Auste… 68 en "Best Books Ever L…  
  ## 5 37431 Pride an… Auste… 68 en ""   
  ## 6 37431 Pride an… MacKa… 38839 en ""   
  ## 7 42671 Pride an… Auste… 68 en "Best Books Ever L…  
  ## # ℹ 2 more variables: rights <chr>, has\_text <lgl>

1. **Use gutenberg\_works()** to filter and find the correct ID for “Pride and Prejudice”:

* pride\_id\_filtered <- gutenberg\_works() %>%  
   filter(str\_detect(title, "Pride and Prejudice"))  
  pride\_id\_filtered
* ## # A tibble: 3 × 8  
  ## gutenberg\_id title author gutenberg\_author\_id language gutenberg\_bookshelf  
  ## <int> <chr> <chr> <int> <chr> <chr>   
  ## 1 1342 Pride an… Auste… 68 en "Best Books Ever L…  
  ## 2 37431 Pride an… Auste… 68 en ""   
  ## 3 37431 Pride an… MacKa… 38839 en ""   
  ## # ℹ 2 more variables: rights <chr>, has\_text <lgl>

1. **Download the text of “Pride and Prejudice”** using the gutenberg\_download function:

* book <- gutenberg\_download(pride\_id\_filtered$gutenberg\_id)
* ## Determining mirror for Project Gutenberg from https://www.gutenberg.org/robot/harvest
* ## Using mirror http://aleph.gutenberg.org
* ## Warning: ! Could not download a book at http://aleph.gutenberg.org/1/3/4/1342/1342.zip.  
  ## ℹ The book may have been archived.  
  ## ℹ Alternatively, You may need to select a different mirror.  
  ## → See https://www.gutenberg.org/MIRRORS.ALL for options.

1. **Create a tidy table of words** from the book text using the tidytext package:

* library(tidytext)  
  words <- book %>%  
   unnest\_tokens(word, text)

1. **Add a column for the word number** (useful for later plotting):

* words <- words %>%  
   mutate(word\_number = row\_number())

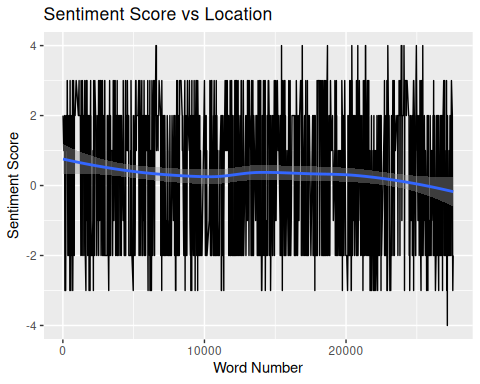
1. **Remove stop words and numbers** from the words object using anti\_join:

* data("stop\_words")  
  words\_clean <- words %>%  
   anti\_join(stop\_words, by = "word") %>%  
   filter(!str\_detect(word, "^\\d+$"))

1. **Assign sentiment values** to each word using the AFINN lexicon:

* afinn <- get\_sentiments("afinn")  
  words\_sentiment <- words\_clean %>%  
   inner\_join(afinn, by = "word")

1. **Plot sentiment score versus word location** and add a smoother:

* library(ggplot2)  
  ggplot(words\_sentiment, aes(x = word\_number, y = value)) +  
   geom\_line() +  
   geom\_smooth(method = "loess") +  
   labs(x = "Word Number", y = "Sentiment Score", title = "Sentiment Score vs Location")
* ## `geom\_smooth()` using formula = 'y ~ x'
* 

1. **Convert word locations to pages** (assuming 300 words per page) and plot average sentiment by page:

* words\_sentiment <- words\_sentiment %>%  
   mutate(page = word\_number %/% 300)  
    
  page\_sentiment <- words\_sentiment %>%  
   group\_by(page) %>%  
   summarize(avg\_sentiment = mean(value, na.rm = TRUE))  
    
  ggplot(page\_sentiment, aes(x = page, y = avg\_sentiment)) +  
   geom\_line() +  
   geom\_smooth(method = "loess") +  
   labs(x = "Page", y = "Average Sentiment", title = "Average Sentiment by Page")
* ## `geom\_smooth()` using formula = 'y ~ x'
* 