# DATA 300 3 Homework 5 Solution

Aadarsha Gopala Reddy

November 17, 2022

## Contents



## 0. Loading the libraries

```
# load the libraries
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidytext)
```

```
## Warning: package 'tidytext' was built under R version 4.2.2
```

```
library(tidyr)
library(janeaustenr)
```

```
## Warning: package 'janeaustenr' was built under R version 4.2.2
```

---

# 1. Setup

Import the datasets into R and load any packages that you will need for your analysis.

`read_csv()` reads a csv file into the program

`dim()` returns the dimensions of a data frame

```
# load the datasets
askreddit <- read_csv("askreddit.csv")
```

```
## Rows: 1000 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): title
## dbl (3): ups, downs, num_comments
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
funny <- read_csv("funny.csv")
```

```
## Rows: 1000 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): title
## dbl (3): ups, downs, num_comments
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
movies <- read_csv("movies.csv")
```

```
## Rows: 999 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): title
## dbl (3): ups, downs, num_comments
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
worldnews <- read_csv("worldnews.csv")
```

```
## Rows: 1000 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): title
## dbl (3): ups, downs, num_comments
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# dimensions of each datasets
dim(askreddit)
```

```
## [1] 1000    4
```

```
dim(funny)
```

```
## [1] 1000    4
```

```
dim(movies)
```

```
## [1] 999   4
```

```
dim(worldnews)
```

```
## [1] 1000    4
```

---

## 2. Tidy the data

Combine the four separate datasets into a single tibble or data.frame in R. This new dataset should have the four variables from the original data, plus an additional variable that records the subreddit where each post was made. (This additional variable should take the values "Movies," "World News," "Ask Reddit," or "Funny.")

`mutate()` adds a new column to a data frame

`bind_rows()` combines two data frames into one

```
# combine the datasets
reddit_data <- bind_rows(askreddit, funny, movies, worldnews) %>%
  mutate(subreddit = c(
    rep("Ask Reddit", 1000),
    rep("Funny", 1000),
    rep("Movies", 999),
    rep("World News", 1000)
  ))
```

---

## 3. Tokenize

Create a new tibble or data.frame in R in which each row is a word. This new dataset should retain information about which subreddit each word was posted in. (At this stage, you do not need to de-duplicate the words. The same word can appear multiple times in this dataset.)

`unnest_tokens()` tokenizes the text

```
# tokenize the data
reddit_data_tokenized <- reddit_data %>%
  unnest_tokens(word, title)
```

---

## 4. Remove Stop Words

Remove filler words (like "the," "I," and "a") from the dataset that you created in the previous question. You can find a list of stop words by loading the tidytext package and calling the stop_words object.

`anti_join()` removes rows from one data frame that match rows in another data frame

```
# remove stop words
reddit_data_tokenized <- reddit_data_tokenized %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

---

## 5. Aggregate by Words

Create a new tibble or data.frame in R that records unique words within each topic. This object should have, at a minimum, three columns: the subreddit, the word within that subreddit, and the number of times that the word appeared. Words should be unique within subreddits – there should only be one entry for the word "day" within the Ask Reddit subreddit – but the same word may appear in multiple subreddits.

`count()` counts the number of times a value appears in a column

```
# aggregate by words
reddit_data_tokenized <- reddit_data_tokenized %>%
  count(subreddit, word, sort = TRUE)
```

---

## 6. Visualize Popular words by Subreddit

Using a method of your choosing, visualize the 10 most popular words within each subreddit. You might choose to do this using a bar chart or a word cloud, for example.

`group_by()` groups the data by a column

`top_n()` returns the top n rows of a data frame

`ungroup()` removes the grouping of a data frame

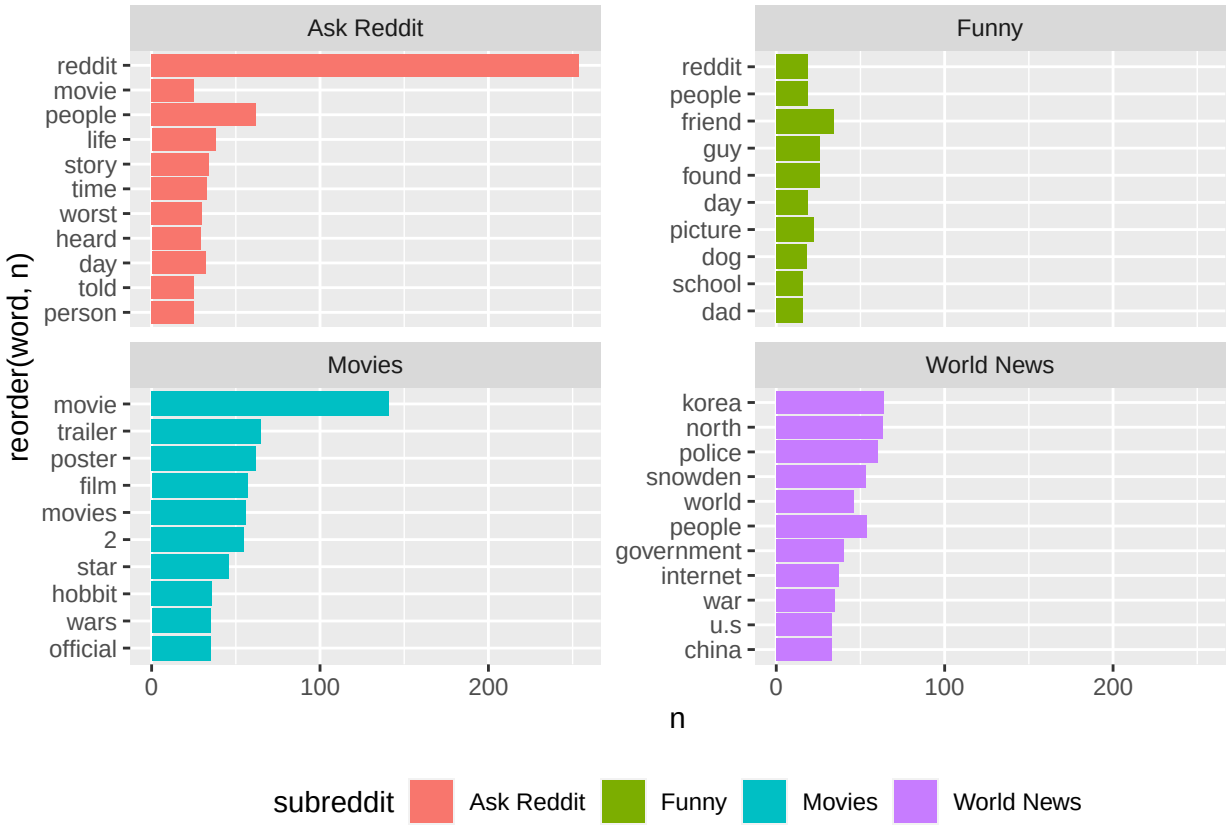
`ggplot()` creates a plot

`geom_col()` creates a bar chart

`facet_wrap()` creates a facetted plot

`coord_flip()` flips the x and y axis

```
# visualize the 10 most popular words within each subreddit
reddit_data_tokenized %>%
  group_by(subreddit) %>%
  top_n(10, n) %>%
  ungroup() %>%
  ggplot(aes(x = reorder(word, n), y = n, fill = subreddit)) +
  geom_col() +
  facet_wrap(~subreddit, scales = "free_y") +
  coord_flip() +
  theme(legend.position = "bottom")
```




---

## 7. Sentiment Analysis

Use the "bing" sentiment dictionary from the tidytext package to add a "sentiment" column to the words in your dataset. Words within your dataset should be categorized as positive, negative, or NA (if they do not appear in the bing sentiment dictionary).

`inner_join()` joins two data frames by matching rows

```
# sentiment analysis
reddit_data_tokenized <- reddit_data_tokenized %>%
  inner_join(get_sentiments("bing")) %>%
  mutate(sentiment = ifelse(sentiment == "positive", 1,
    ifelse(sentiment == "negative", -1, 0)
  ))
```

```
## Joining, by = "word"
```

---

## 8. Compare Sentiments

Within each subreddit, find:

- The number of positive words;
- The number of negative words;

- Positive words as a percentage of the total number of words;
- Negative words as a percentage of the total number of words.

Which subreddit has the highest percentage of positive words? Which subreddit has the highest percentage of negative words?

***The movies subreddit has the highest percentage of positive words.***

***The world news subreddit has the highest percentage of negative words.***

`summarize()` summarizes the data

```
# compare sentiments
reddit_data_tokenized %>%
  group_by(subreddit) %>%
  summarise(
    pos_words = sum(sentiment == 1),
    neg_words = sum(sentiment == -1),
    pos_words_perc =
      pos_words / (pos_words + neg_words),
    neg_words_perc =
      neg_words / (pos_words + neg_words)
  )
```

```
## # A tibble: 4 x 5
##   subreddit  pos_words neg_words pos_words_perc neg_words_perc
##   <chr>          <int>     <int>          <dbl>          <dbl>
## 1 Ask Reddit       162       321          0.335          0.665
## 2 Funny            115       163          0.414          0.586
## 3 Movies           134       180          0.427          0.573
## 4 World News       171       451          0.275          0.725
```

---