Parts I & II

Elizabeth Miller

Part I: Web Scraping

Go to the website https://www.scrapethissite.com/pages/simple/ and scrape the data to create a table with four variables: Country, Capital, Population, and Area. The table will have a total of 250 observations.

```
url <- "https://www.scrapethissite.com/pages/simple/"
html <- read_html(url)

country <- html |>
  html_elements("h3.country-name") |>
  html_text2()

capital <- html |>
  html_elements("span.country-capital") |>
  html_text2()

population <- html %>%
  html_elements("span.country-population") %>%
  html_text2()

area <- html |>
  html_elements("span.country-area") |>
  html_text2()

# Build tibble
country_table <- tibble(country, capital, population, area)
country_table
```

```
# A tibble: 250 x 4
   country              capital          population area
```

```
   <chr>                <chr>            <chr>      <chr>
 1 Andorra              Andorra la Vella 84000      468.0
 2 United Arab Emirates Abu Dhabi        4975593    82880.0
 3 Afghanistan          Kabul            29121286   647500.0
 4 Antigua and Barbuda  St. John's       86754      443.0
 5 Anguilla             The Valley       13254      102.0
 6 Albania              Tirana           2986952    28748.0
 7 Armenia              Yerevan          2968000    29800.0
 8 Angola               Luanda           13068161   1246700.0
 9 Antarctica           None             0          1.4E7
10 Argentina            Buenos Aires     41343201   2766890.0
# i 240 more rows
```

Part II: Text Analysis

Use the artofwar dataset and conduct a text analysis.

Tokenize the data, compute word counts, remove stop words, and create bar plot showing dataset's top ten used words, flipping the axes.

```
# Tokenize the data
text_data <- read_csv("artofwar.csv")
```

```
Rows: 396 Columns: 1
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): x

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.
```

```
text_data <- text_data %>%
  rename("origcol" = x) # the column name was giving me trouble, so i renamed it

tokens <- text_data %>%
  unnest_tokens(word, origcol)

# Compute word counts
tokens %>%
  count(word)
```

```
# A tibble: 2,222 x 2
   word        n
   <chr>   <int>
 1 1          24
 2 10         14
 3 100,000     1
 4 11         14
 5 12         14
 6 13         13
 7 13,14       1
 8 14         12
 9 15         12
10 16         12
# i 2,212 more rows
```

```
# Remove stop words
tokens %>%
  anti_join(stop_words)
```

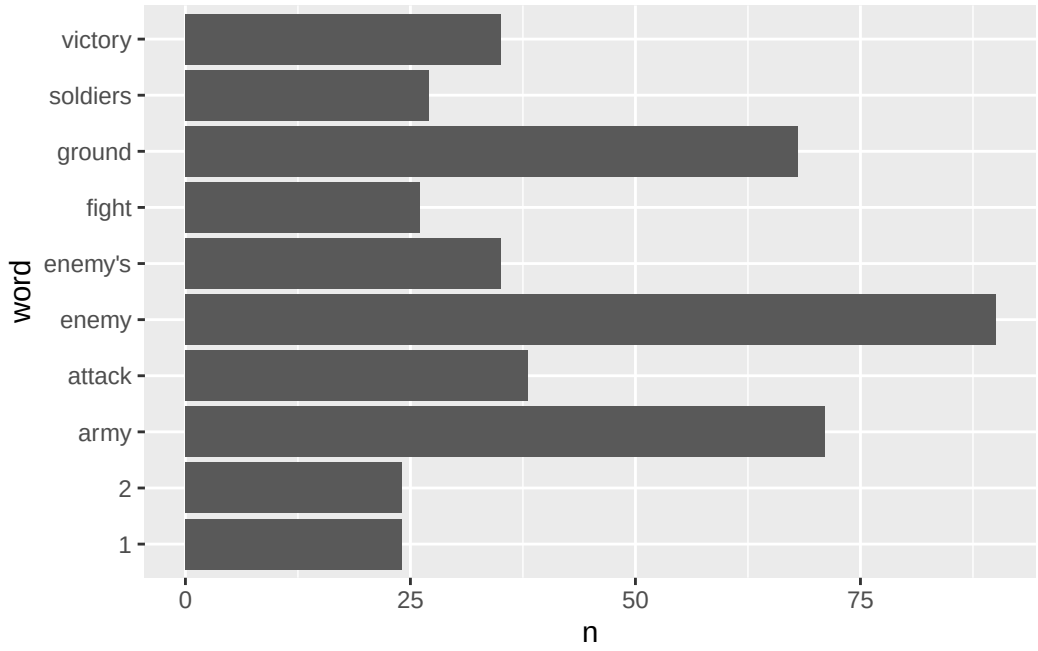
```
Joining with `by = join_by(word)`
```

```
# A tibble: 4,367 x 1
   word
   <chr>
 1 chapter
 2 1
 3 laying
 4 plans
 5 1
 6 sun
 7 tzu
 8 art
 9 war
10 vital
# i 4,357 more rows
```

```
# create bar plot showing dataset's top 10 used words
tokens %>%
  anti_join(stop_words) %>%
  count(word) %>%
  arrange(desc(n)) %>%
```

```
  slice(1:10) %>%
  ggplot(aes(x = word, y = n)) +
  geom_col() +
  # Flip the plot coordinates
  coord_flip()
```

```
Joining with `by = join_by(word)`
```



Let's make the visualization better by creating custom stop words to remove all numbers. Compute word counts again and create a bar plot that shows top 20 used words. Create a bar plot showing the dataset's top 20 used words, remembering to flip the axes for better visualization

```
# custom stop words
stop_num <- (1:20)
stop_num_words <- c("one", "two", "three", "four", "five",
                    "six", "seven", "eight", "nine", "ten",
                    "eleven", "twelve", "thirteen", "fourteen",
                    "fifteen" , "sixteen", "seventeen", "eighteen",
                    "nineteen", "twenty")

stop_num <- as.character(stop_num)
```

```r
stop_numbers_list <- c(stop_num, stop_num_words)

stop_numbers <- tibble(word = stop_numbers_list)

# compute word counts again
tokens %>%
  anti_join(stop_words) %>%
  anti_join(stop_numbers) %>%
  count(word) %>%
  arrange(desc(n))
```

```
Joining with `by = join_by(word)`
Joining with `by = join_by(word)`
```

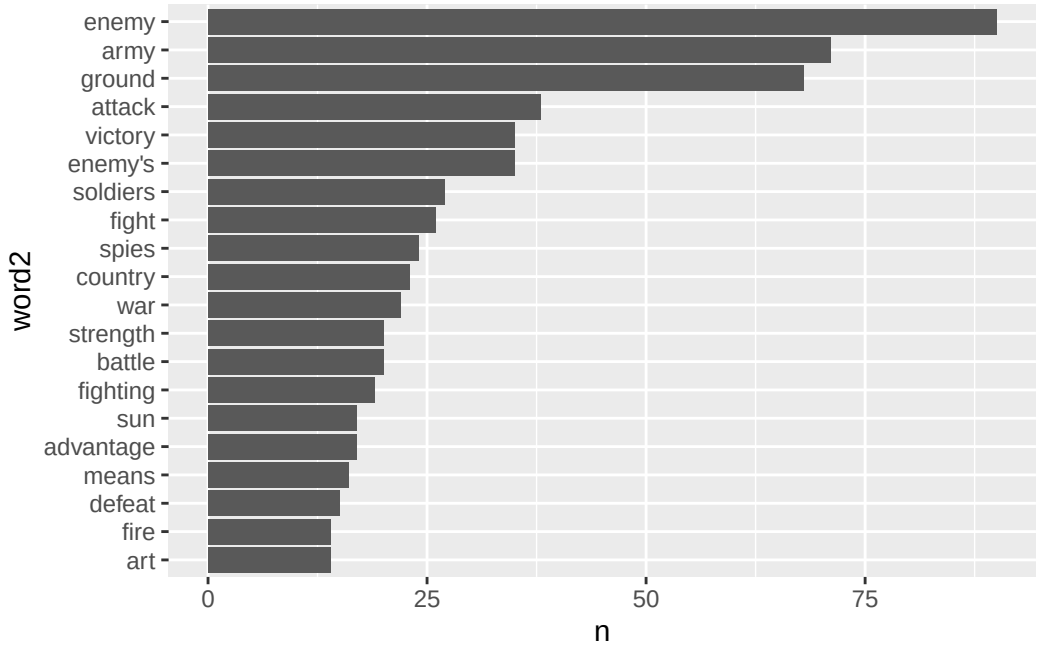
```
# A tibble: 1,849 x 2
   word         n
   <chr>    <int>
 1 enemy       90
 2 army        71
 3 ground      68
 4 attack      38
 5 enemy's     35
 6 victory     35
 7 soldiers    27
 8 fight       26
 9 spies       24
10 country     23
# i 1,839 more rows
```

```r
# make bar plot

tokens %>%
  anti_join(stop_words) %>%
  anti_join(stop_numbers) %>%
  count(word) %>%
  arrange(desc(n)) %>%
  mutate(word2 = fct_reorder(word, n)) %>%
  slice(1:20) %>%
  ggplot(aes(x = word2, y = n)) +
  geom_col() +
```

```
  # Flip the plot coordinates
  coord_flip()
```

```
Joining with `by = join_by(word)`
Joining with `by = join_by(word)`
```



Let's perform sentiment analysis using 'nrc' sentiment dictionary. Append dictionary to the subset created in (e). Create a bar plot of the word counts colored by sentiment. Show only the top 10 words for each sentiment using facet wrap.

```
# Append the dictionary to the subset created in part (e)
tokens_no_num <- tokens %>%
  anti_join(stop_words) %>%
  anti_join(stop_numbers)
```

```
Joining with `by = join_by(word)`
Joining with `by = join_by(word)`
```

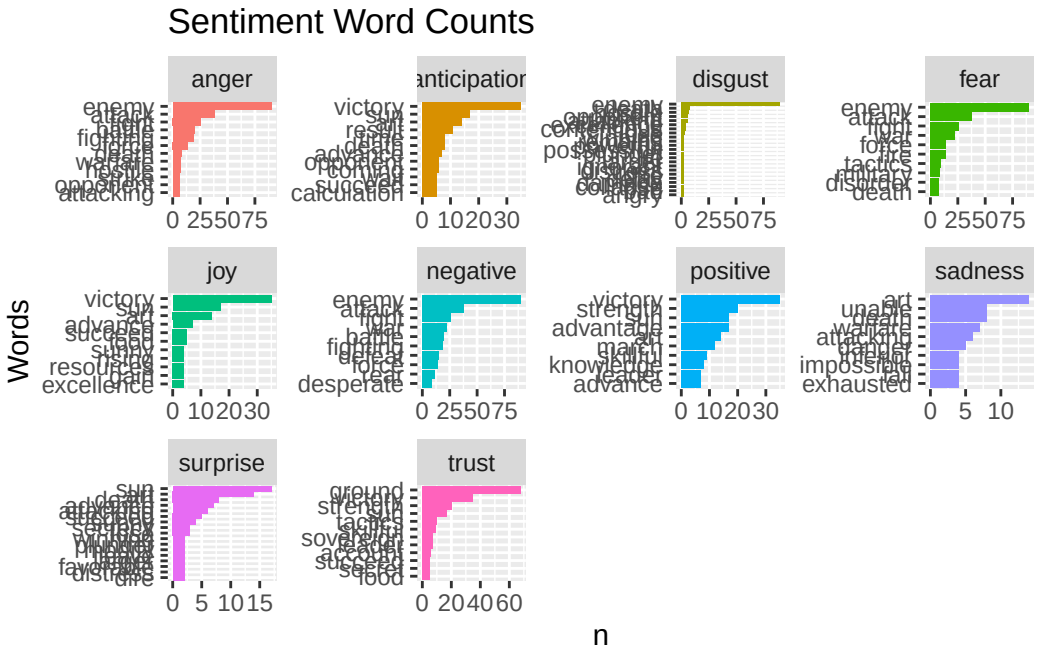
```
war_sentiment<- tokens_no_num %>%
  inner_join(get_sentiments("nrc"),
             relationship = "many-to-many")
```

```
Joining with `by = join_by(word)`
```

```
# Create a bar plot of the word counts colored by sentiment. Show top 10 words
war_sentiment <- tokens_no_num %>%
  inner_join(get_sentiments("nrc"),
             relationship = "many-to-many") %>%
  count(word, sentiment) %>%
  group_by(sentiment) %>%
  slice_max(n, n =10) %>%
  ungroup() %>%
  mutate(word2 = fct_reorder(word, n))
```

```
Joining with `by = join_by(word)`
```

```
# bar plot
ggplot(war_sentiment, aes(x = word2, y = n,
                          fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  # Create a separate facet for each sentiment with free axes
  facet_wrap(~ sentiment, scales = "free") +
  coord_flip() +
  # Title the plot "Sentiment Word Counts" with "Words" for the x-axis
  labs(
    title = "Sentiment Word Counts",
    x = "Words"
  )
```



What would you say about the sentiments displayed in this book?

Sentiments are largely negative or antagonistic, which makes sense because these are excerpts from The Art of War. Positive sentiments that are included in the text are usually still related to war, such as "victory" or "advance."