Four Score and Seven Years Ago

Unit 3 - Lab 3

Directions: Follow along with the slides and answer the questions in **BOLDED** font in your journal.

Text as Data

- If you think about it, there's lots of information hidden in text.
- Writing is one of the principle ways people communicate with each other.
- Data scientists try to develop methods to let them tap into the info contained in text.
- They analyze tweets from Twitter to gauge whether people tweeting are feeling happy, sad, etc.
- In this lab, we'll look at a few simple methods to process and visualize text data.

Grabbing text from the web

- Similar to the previous lab, we can actually use *functions* in RStudio to grab text documents from off the web (No button required!)
- Take a look at the text found Here
- **What is this text? Why is it famous?**
- To grab this text and read it into R, we use the `readLines` function.
- Replace the *"Write text URL here"* with the actual URL of the text.

```
text <- readLines("Write text URL here")
```

Initializing text

- Text is very complicated for computers to understand.
- Computers don't necessarily understand sentence structures, punctuation, etc.
- Before we can analyze our text, we must first turn it into something that the computer can understand.
- That is, we must first *initialize* our text.
- This changes our text from a long list of symbols into a *corpus*.
- A *corpus* is just data science lingo for text that computers can easily deal with.
- To change our raw `text` into a *corpus*, run the following:

```
corpus <- InitializeText(text)
```

Text vs. Corpus

- Now that we've got our text turned into a corpus, we can see how the data are actually different by running the following two lines of code:

```
text
```

```
corpus
```

- **How are the outputs different for each line of code?**

Analyzing the Gettysburg Address

- There are many ways to analyze text data.
- We'll focus on two graphical methods.
- Specifically, *wordclouds* and *bargraphs*.
- To make a wordcloud of our corpus

```
MakeWordCloud(corpus)
```

- Wordclouds size the words based on how often they occur in the text.
- **Which word occurs most often in the Gettysburg address?**
- **Which word occurs the 3rd most often? Can you tell? Why or why not?**

Wordclouds vs. Bargraphs

- We'll use a special function to create bargraphs for our text data:

```
MakeWordBar(corpus)
```

- The height of the bars indicate how many times each word at the bottom occurred in the text.
- **What word occured the most often? How many more times did it occur than the next most frequent word?**
- **Which word occurs the 3rd most often?**
- **Between wordclouds and bargraphs, which plot do you think is easier to interpret and why?**

Drilling into our text

- The Gettysburg Address is one of the most important texts in US history . . .
- And all we've shown so far is that the word *the* occurs the most often.
- . . . Who cares!
- We're much more interested in finding the *important* words that President Lincoln used in his speech.
- Common words that are mostly uninteresting are called *stopwords* by data scientists.
- To remove these words, run the following code:

```
p_corpus <- ProcessText(corpus,
                        removestopwords=TRUE)
```

So what did we do?

```
p_corpus <- ProcessText(corpus,
                        removestopwords=TRUE)
```

- What we've done is created a new corpus called `p_corpus` (short for *processed corpus*).
- We took the words in our current corpus …
- … and then removed the stopwords (the ones that are mostly boring but occur really often, like 'the', 'and', 'but', etc.).
- **Create a wordcloud and word bargraph for your processed corpus.**
- **Write down the 5 words that occurred most often in the address.**

Tinkering with our plots

- The `MakeWordBar` and `MakeWordCloud` functions come with a few options we might be interested in.
- For example, by default, the functions only show words that occur at least 2 or more times. We can change this.
- We can also limit the number of words to show.
- **Run the following commands and describe how the plots have changed from those you made on the previous slide.**

```
MakeWordCloud(p_corpus, min.freq=1)
```

```
MakeWordBar(p_corpus, top=5)
```

On your own

- Use the following code to load a *mystery* text. (Don't *copy* & *paste* this code. Type it out by hand.)

```
text <-
  readLines("http://web.ohmage.org/mobilize/
          resources/ids/data/mystery.txt")
```

- **Using the code you learned in this lab, can you figure out what famous story the text is from?**
- **Write down the code for any plots you made which helped you figure it out.**