**Overall Goal**

1. Use this as an introduction/motivation for analyzing text data
   1. Why do we care?
      1. We want people to be able to extract meaning from text data without having to learn how to code or understand the nitty gritty behind what is presented in the interface
      2. Gives people exposure to something they might not otherwise see - text analytics can be pretty high level but we want to give the layperson opportunities to use it as well
   2. Who would care?
      1. Undergraduate students enrolled in introductory courses such as Linguistics and English
      2. Undergraduate students just beginning to learn about data science
      3. Those interested in how what people are talking about can be used in marketing
2. Make easy to use interface broadly accessible
   1. Consequence: there is no novel/complicated text analysis going on here
   2. Everything displayed must be explained and easily understandable
   3. All plots should be easily readable and labeled well
   4. “Cover up” the behind the scenes clean up work that can be tedious and is unimportant to understanding the outputs
3. Make this interface fun/interesting to use
   1. Allow user to change inputs and guide themselves through the process interactively
   2. If they make a mistake or change their mind about a decision they’ve made (“Oh, I’d like to remove this word from the sentiment analysis since I saw it in the wordcloud and it doesn’t make sense for what I am trying to do”)
   3. Display outputs where user can actually learn the facts about what is going on in their text data

**Statistical Aspects**

1. Preprocessing the data
   1. Must be text data - .txt, .csv
   2. Remove “unimportant” lines - e.g. table of contents, copyright stuff
   3. Remove stop words - these are ‘filler’ words that are rather ordinary
      1. “The”, “these”, “and”, “or”
      2. The list I use has 1,149 words
2. What are the main ideas in the text?
   1. Frequency plot
   2. Word cloud with word frequencies - the larger it is on the plot, the more times it occured in the text
3. How do the words used in the document(s) convey emotion?
   1. Join different sentiment lexicons (basically word banks) to the data
   2. Calculate “word score”
   3. Plot words with “strongest” sentiment in a text - can be positive or negative
   4. Sentiment word cloud - color the words by ‘positive’ or ‘negative’ and size the words by their frequency
   5. Look at how words may be negated in sentiment by words preceding them
      1. I calculate word scores by single words - “happy” is positive - but when you consider two words together - “not happy” - should be categorized as a negative word in this case
4. How are the words connected in the text?
   1. What is the probability that two words will occur together? (within some number of lines - I have it coded as within two lines)
   2. What is the correlation between words?
   3. Difference between cooccurrence and correlation: Correlation explores *relationships* between words (how often they appear together versus not appear together) rather than cooccurrence, which only accounts for how many times they occur together (more reflective of how many times the individual words occur)

**Computer/human interface design - to make for new people**

1. Create an interface that is accessible to everyone
   1. I.e. you don’t need any major statistical understanding to be able to appreciate and understand what the interface is showing/doing
2. Are there particularly useful aspects of the Shiny that are useful outside of what we’re trying to do?
   1. Adjustable slider inputs - when you change certain aspects of how your data is processed, you can get vastly different results (a takeaway we should all learn in data wrangling/cleaning)
   2. Adjust text inputs - ability to filter data you don’t want and push the user to think about what their output truly means

**Shiny Technology**

1. Reactive map (show the process it goes through without saying “I did this then this then this”)
2. A large draw to this app will be the ability for the user to tinker with it themselves (more engaging than reading through a webpage)
   1. Shiny needs to be **reactive**
   2. User can upload their own data from their own machine
   3. User can change inputs and plots/tables/outputs will react accordingly
3. Process
   1. First tried to build up an interface that displays outputs for frequency plots, sentiment analysis, and network/correlation graphs for a single document
   2. Then expand to allow for multiple documents (a corpus)
   3. Do comparisons between documents