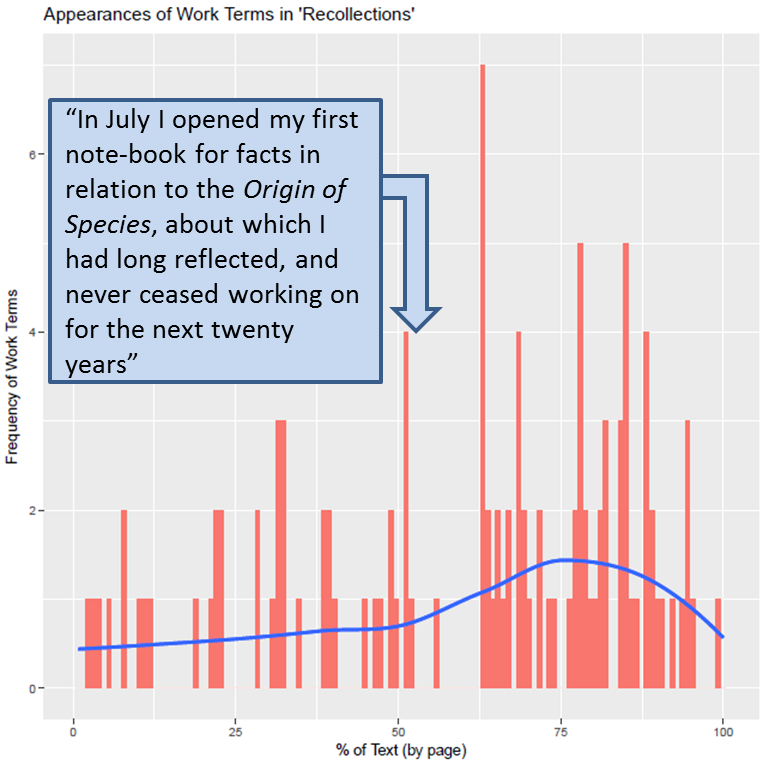
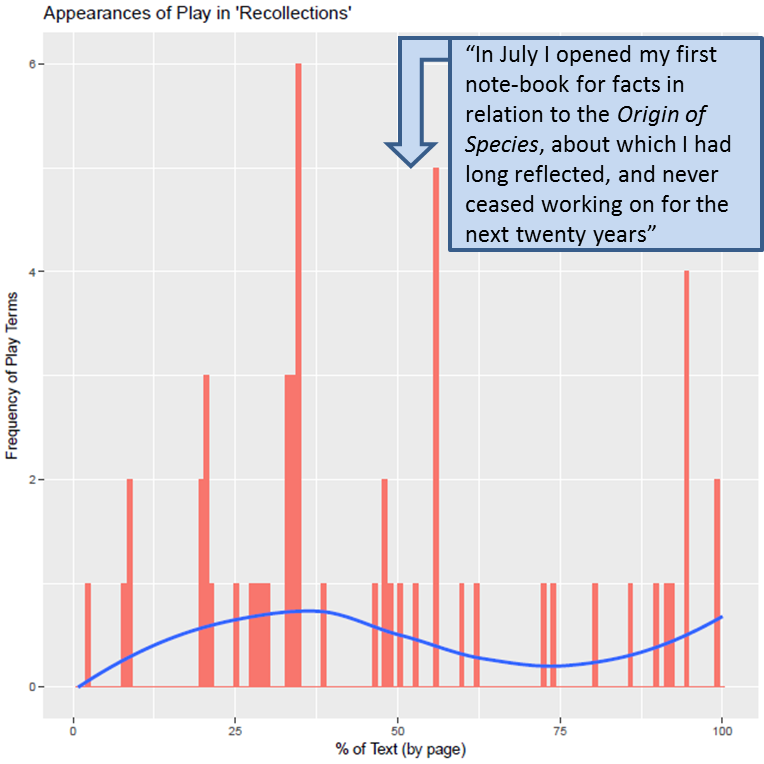
Inspired by Kurt Vonnegut’s work on the shapes of stories (<https://www.youtube.com/watch?v=oP3c1h8v2ZQ>), I’ve spent the last few years experimenting with a type of data visualization that I refer to as “Word Flagging,” wherein I choose a set of words and attempt to determine how frequently they occur within a certain unit of the text (the sentence, the page, the chapter, etc.). In short, my Word Flagging scripts make use of Lincoln Mullen’s “tokenizers” package to break a txt files into a string of words, assigns each word a number to identify it, and then searches that string for matches to certain key words.

The Word Flagging scripts I’ve developed are somewhat similar to the Bubblelines tool available in Voyant Tools (<http://docs.voyant-tools.org/tools/bubblelines/>), which visualizes the frequency with which certain terms appear within a corpus. Where the Bubblelines tool uses bubbles, I generally prefer to visualize in the form of a bar chart, which I feel gives a better impression of how the appearance of words within a text can reveal aspects of their narrative.

I began this work, for instance, by looking for the appearance of words associated with work and play in Charles Darwin’s autobiography. Visualizing these terms suggests demonstrates that Darwin presented his life’s story as a development from a sporting gentleman into a serious scientific worker.



Recently I have been updating my R scripts to allow Word Flag data to be collected alongside information about the context in which the word appeared (what is commonly known as the Key Word in Context, or KWIC). I realized that Lincoln Mullen’s “tokenizers” package would make it easy enough to preserve information about the sentence; however, it would make my scripts run significantly slower if I attempted to collect information about the sentences AND to preserve information about where EXACTLY a specific word appeared in the text. I could collect information about which sentence a key word appeared (for instance, sentence #5 out of 392), or I could collect information about which number in the character string matched my key word (for instance, word #213 out of 133,432). Collecting both, however, led to a run time that was at least 5 times longer.

As a result, I thought it was important to experiment to see whether assigning matched terms a sentence identification number rather than a word identification number would lead to notably different visualizations.

The scripts for this experiment can be found here: ~~~. I returned to Darwin’s autobiography as my test case, as it was a visualization I was quite familiar with.

