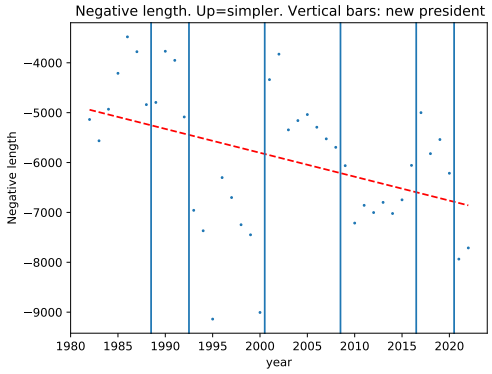
CS 1110 Spring 2022, Assignment 3 solutions*

The solution files are here: http://www.cs.cornell.edu/courses/cs1110/2022sp/assignments/assignment3/a3_solns.zip. Take a look![1]

Included in the solution files is numerical_results.txt, which is a printout of the values our code produced on the real State of the Union (SOTU) data. Below, we show the graphical plots of these results.

- Length: in the "A3 optional" handout, you already saw these results for the negative[2] lengths of each SOTU.



The red dotted line is a linear fit to the data, So if we use negative length as a measure, it seems like recent SOTUs are getting *less* simple, in contrast to what was asserted by the motivating quote for the main A3.

Vertical lines demarcate changes in who was the U.S. President. Observe that different presidents[3] seem to have distinct length preferences, for the most part.[4] It even seems that for later presidents, there's been an oscillation in length preferences.

- Type-token ratio:

*Authors: Lillian Lee

[1] You will see `# BEGIN REMOVE` and `#END REMOVE` comments in a3.py. These come about as follows. The way we create assignment code is that we first code up a working code base. Then, we use `# BEGIN REMOVE` and `#END REMOVE` comments to demarcate portions for students to create themselves. Then, we run code that creates skeleton files automatically by removing the demarcated sections.

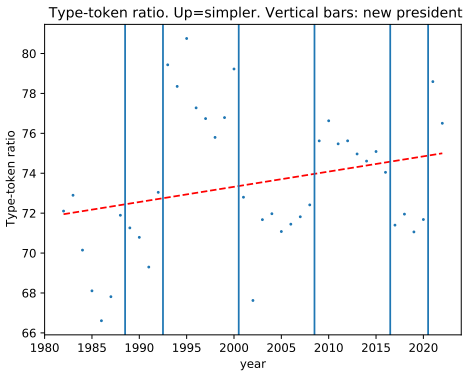
[2] Negating the lengths means that larger values correspond to greater simplicity, for consistency with the other functions considered in A3.

[3] Or at least their speechwriters.

[4] You may find it interesting to note that our results are quite close to, but not quite the same as, the length statistics produced by the American Presidency project. They describe their (more-careful-than-ours) processing of tokens as follows:

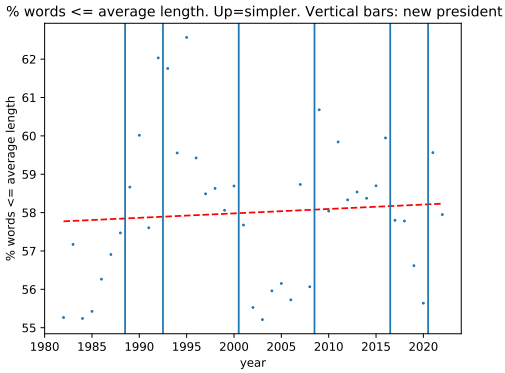
> Length in words is computed by using the word count feature in Microsoft Word. The APP first uses the transcript provided by the White House, and then provides a modified word count about a week later after the official transcript is provided by the National Archives in the "Daily Compilation of Presidential Documents". The following elements are removed from the transcript prior to generating the word count: 1) audience reaction notations such as "applause", etc.; 2) Prompts indicating a change in the speaker; 3) All audience comments included in the transcript; 4) All "em dashes" between clauses in sentences; 5) All words in brackets that reflect a White House correction to the word actually spoken; and 6) all HTML or other coding to produce paragraph breaks, among others. The word count is then generated using a single blob of words separated only by spaces and normal punctuation.

For example, our program does not strip out audience interjections, some of which have been considered newsworthy.



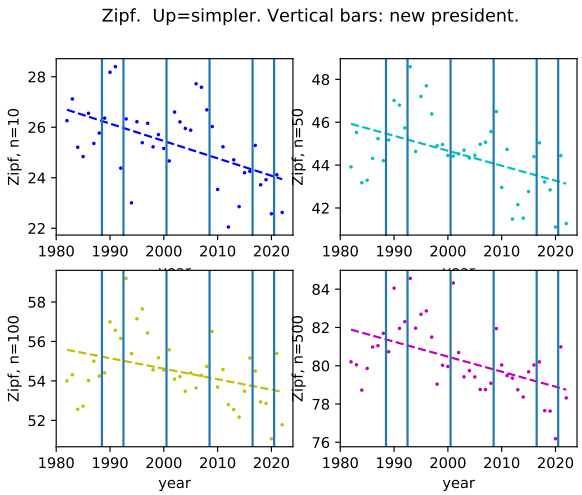
So in contrast to the negative-length results, for the type-token-ratio measure, SOTUS appear to be getting more simple over time.

- Here is percent_avg_or_shorter().



Once again, a change in measure seems to yield a different story: The trend line looks fairly flat, suggesting that there hasn't really been change in complexity recently.

- For function zipf(), we have the issue of needing to pick a value for parameter `n`. Our code prints/plots the results for the somewhat arbitrary choices of `n` in `[10, 50, 100, 500]`:



What do *you* conclude?