**Capstone mini-project: Data Wrangling**

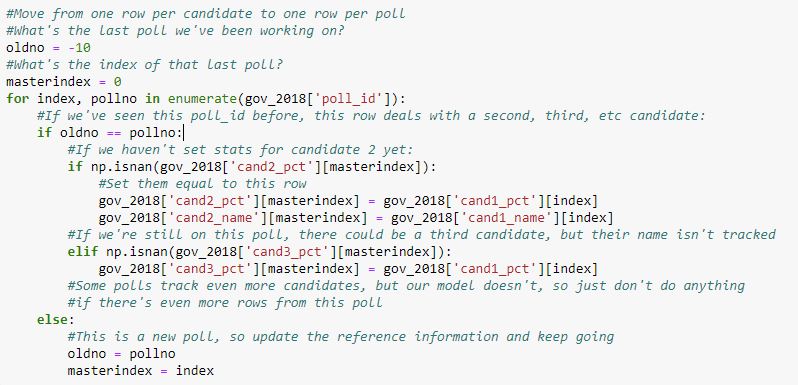
Raw data was collected from the GitHub files FiveThirtyEight provides at the following links:

1. <https://github.com/fivethirtyeight/data/tree/master/polls>
2. <https://github.com/fivethirtyeight/data/blob/master/pollster-ratings/2016/raw-polls.csv>

These files contain polls FiveThirtyEight has tracked from 2018 onward and polls from 2016 and before, respectively. FiveThirtyEight was selected because they have already aggregated polls from pollsters across America concerning senate, house, governors, and presidential elections. Additionally, they filter their data to remove polls they suspect to be falsified from sites such as CSP polling and Delphi Analytica, as discussed here: <https://fivethirtyeight.com/features/fake-polls-are-a-real-problem/>.

This aggregation and filtering makes FiveThirtyEight an excellent source of initial polling data. However, their 2018 data was stored with one candidate per poll per row, while the 2016 data stored all candidates in a given poll on one row. Furthermore, the 2018 data did not contain information on election results. As such, the data needed to be cleaned and another source of data for the 2018 elections was required.

To begin with, the following code was used to reformat the 2018 data from one row per candidate to one row per poll, matching the 2016 format:



Several columns of data were superfluous, either because they were unrelated to the data I was trying to collect, such as the source URL of the poll, because they were redundant, such as population and population\_full, or because they were not tracked in the 2016 data and were therefore only present in a small fraction of the total dataset. Additionally, FiveThirtyEight maintains a pollster rating system to track reliability and accuracy of individual pollsters, but given the goal of this project, determining which factors influence the accuracy and precision of polls, using this data would be cheating.

As the 2018 data did not track final election results, I obtained this data from the New York Times and Wikipedia and stored it in separate .csv files.

To convert the 2018 data from listing states by their full names to listing states by their abbreviations, in accordance with the 2016 format, I used a Python dictionary created by Mike Shultz. I then iterated though the data for governors, senate, and house races to incorporate the election results into the reformatted csv files and fill in the missing data.

The files were then ready to be merged with the 2016 dataframe.

The major issues that arose when tracking and wrangling the data were the intermittent presence of third-party candidates, named candidates, and single-party races. FiveThirtyEight stores polls and results as the percent of the vote received by ‘candidate 1’, followed by ‘candidate 2’. The convention is that, in partisan races, candidate 1 is the Democrat and candidate 2 is the Republican. However, in races such as the California 2018 senate election, both candidates were Democrats. The cand\_name columns attempt to clarify party affiliation, but these columns can contain names, such as ‘Kerry’, names followed by party affiliation, such as ‘Lamont (D)’ or only party affiliation, such as ‘Republican’, making an analysis of polling error by party much more difficult, as there was no concrete way to identify party affiliation from the data.

House polling tended to be sparser than Senate and Governor polling, especially pre-2018. One district, KS-1, received exactly one poll in 2014. This poll was off by 42.94 percentage points from the final results, skewing the data for that district tremendously. Aggregating the data by state both allows comparisons between house polling and all other polls and smooths out the data for each state. In the KS-1 example, averaging error across all house polls in Kansas gives an average error of 8.05 percentage points: notable, but not as outrageous as a 42.94 point average error in the KS-1 district.