Polls have been an important tool for gauging public opinion and even predicting or forecasting elections. However as the internet and methods of analysing big data have proliferated, individual polls drive stories less and poll aggregators become more important. Now, when you google 2024 polls, the top results you get will all be poll aggregators. Many people in the media have questioned the accuracy of The Polls (not individual polls but polling as a method) with headlines like “Polls can't predict who will be our next president”.1 It is important for the media to remain skeptical at all times. However, It is this media discussion that motivates my research question, “How accurate are the polls and are they any more accurate than the sentiment of articles from the NYTimes?”

In terms of how I would test this, I would run a fairly simple regression that looks like:

Whichever result gave a stronger prediction would be a better guide to predict the election outcome. By no means am I arguing that polling averages or sentiment cause vote share, but that they could be a useful indicator if someone were to attempt to predict the election based on this information and some control information.

The coding had some very easy parts and some very difficult parts. There was the full range of experience. I decided to structure my code the way that I did, 4 r script files plus a shiny app, because there were many tasks that needed to be completed with intermediate steps that needed to be saved. All NYTimes API code and webscraping is in one R script file. All sentiment analysis is in another. All data cleaning for election results and poll results is in one

file. All data analysis is done in analysis.r. This allowed me to work on the project while hitting roadblocks in different areas. All code in poll\_result\_wrangling.r, analysis.r, and app.r was fairly straightforward. The data was easy to find and the data was in forms that were fairly simple to work with (for analysis.r and app.r that was due to careful data management and organization.)

However, the difficult part was the sentiment analysis. The NYTimes API has some unique syntax (might be HTML but I had to learn it) and with the limits of the API (5 per minute and 500 per day) I had to make absolutely sure that I was making a good attempt. No throw away tests to just see what would happen. Then, when I got a working prototype, it was quite time consuming to get the data and then sample the URLs to see if there were any obvious issues. Then, when you attempted to webscrape there would be an issue with article 2376 and you’d need to see why that URL didn’t work. Usually I would have to tweak the search terms to be more restrictive. I ended up finding a pretty slick search query that found (mostly) everything that I wanted. Once I got a usable dataset of text, the sentiment analysis was tricky to do because of the complex nature of the writing. I think I found the beginnings of a solution. I think by running another layer of parent, children relationships, I might be able to find more sentiment words. There were very few in the child/parent layer but many more on the grandchild/grandparent layer. Once these were extracted, it was fairly simple to attach sentiments to the words using bing and afinn.2 3

Most of the limitations of the analysis come from the sentiment portion. The polling data is quite complete. One problem with the sentiment analysis is that many of the articles are not complete. Since the NYTimes requires a subscription, you can’t access the full articles without logging in and I have yet to figure out how exactly to do that in R. Additionally, due to the limits of the NYTimes API, I only gathered 100 partial articles per candidate. This means that I did not get a full sample of the entire campaign period. The latest articles that I saw were in Mid-October, which is quite a few months before Late July. Future analysis will go deeper in the API and look deeper at the relationships between word tokens and look for better ways to find sentiment words attached to each candidate. I would have taken a more blunt approach for sentiment. However, articles often reference both campaigns in the article so getting a sentiment

for the entire article for one candidate seems wrong for what I am trying to do.