Title page

1. The question

News agencies employ many tricks to sway viewers to their political position. Appeal to emotion, omission of context and even fake news are some of the methods news outlets have been known to use. As the world becomes more polarized we think it’s more important than ever to be aware how the news manipulates how readers feel. In this project we focused on how news outlets of different political outlets use sentimental language to influence readers. We were particularly interested in a some specific topics, namely the difference of reporting in Defense, sustainability, Guns and violence in America, Healthcare, Housing and jobs, and civil rights across different political positions.

1. How the data was obtained

We found a really useful website[] called All sides, which carefully considers how all perspectives current events are presented in news media of various political positions. The website includes positions from left wing media, right wing and central aligning media. Each Article from All sides provides a brief overview, its bias, and a URL to the original source website. This was where we scrapped our data from.

We imported several modules to accomplish this task:

import requests – To send http requests to the website and handle failures

from bs4 import BeautifulSoup – To parse html elements we are interested in from All Sides website

import csv – To export the dataset into csv format

import re – To use regular expressions

First we used a All Sides template URL, filled with our topic of choice, and we got a list of URLs of All sides article links. We will get a list for each topic we selected.

For each URL we recorded its Publish date, Title, All sides URL, Original source URL, News channel, All sides’ declared bias of the article and the body content of the article, all as a row in a dataframe. With an exception of the article body, this information was gotten by searching for specific html elements from the All sides page that we knew ahead of time.

To get the body content was a bit trickier. We used BeautifulSoup to select for a CSS element specific to each news channel and get the body content in that element. The list of selectors had to be created manually for each news article.

All the data could then be exported from a dataframe to a csv file.

1. Data analysis
2. Our conclusions
3. Visualizations
4. Limitations

As mentioned previous we did not find a way to automate getting the css selectors for the body text…