

Visualizing the New York Times' Democratic Primary Coverage

1 Basic info

- Roger Yu-Hsiang Lo
- Jeff Miiller
- Mike Powar

2 Overview

If humans are, as Aristotle put it, political animals, then open and fair news media plays a key role in a healthy democracy. In the media landscape, the New York Times (NYT) is often regarded as a cornerstone institution, and this project aims to visually assess its coverage of the primary race of the upcoming US presidential election for the Democratic Party. One common and intuitive way to evaluate a news article is to quantify its affective states — its sentiment — that span a continuum from negative through neutral to positive. The data visualization in the current project takes this as the starting point and presents a dataset that involves a sentiment analysis of the coverage of the candidates (i.e., we restrict the scope of the discussion to Joe Biden, Bernie Sanders, Elizabeth Warren, Michael Bloomberg, and Pete Buttigieg) trying to become the Democratic Party's nominee for the election. Functionally, the visualization will allow the user to get a quick glimpse of how the NYT coverage presented the candidates to its readers in terms of its sentiment and will also show how the coverage changed over time throughout the primary race, aided by other data that show key events and polling results.

3 Data

All the data used in the current project can be found in the `data` folder of the project repository on GitHub. In what follows, we first provide a description for our datasets and then elaborate on the preprocessing pipelines taken to obtain these datasets.

3.1 Data description

In the following sections, we describe the individual datasets used for the three views. We first provide a snippet of the data; then more explanations are given for the columns whose names are not self-explanatory before we move on to summarize data type and cordiality for each column.

3.1.1 View #1 data

```
"Date", "Candidates", "NewsDesk", "Category", "Headline", "SentScore(headline)",  
"2020-02-01", "Bloomberg", "politics", "politics", "Bloomberg Proposes...", 0.56,  
"2020-02-04", "Buttigieg", "politics", "politics", "Buttigieg Seizes on...", 0.59,  
"2020-02-05", "Buttigieg", "nytnow", "other", "Pete Buttigieg, Coronavirus...", 0.0,  
"2020-02-05", "Biden", "politics", "politics", "What Went Wrong for Joe Biden...", -0.48,  
"2020-02-06", "Bloomberg", "politics", "politics", "Bloomberg Pursues...", 0.19,  
  
"SentScore(snippet)", "SentScore(lead)", "SentScore(avg)", "Count"  
0.28, 0.02, 0.29, 996  
0.53, 0.58, 0.56, 129
```

0.0,0.0,0.0,1626
 -0.15,-0.87,-0.5,2102
 0.75,0.0,0.31,1580

This dataset contains information about a subset of NYT articles published between 5 December, 2018 and 6 April, 2020. The articles included all have one common feature — their headline has to mention one and only one of the five interested candidates. The published date, the candidate mentioned, the news desk, and the headline of each articles are stored under `Date`, `Candidates`, `NewsDesk`, and `Headline` respectively. In the `Category` column, each article is further classified into one of the four broader categories — ‘business’, ‘politics’, ‘opinion’, and ‘other’ — as detailed in Section 3.2.1. The columns `SentScore(headline)`, `SentScore(snippet)`, `SentScore(lead)`, and `SentScore(avg)` contain the sentiment scores estimated based on the headline, snippet, lead paragraph, and their average respectively. Finally, the `Count` column contains the number of words in each article. Note that, even though there are four sentiment scores, all the visualizations in the current project are based on `SentScore(avg)`.

Data abstraction of this tabular dataset with 1127 rows is summarized in the following table:

Column	Type	Range/cardinality
<code>Date</code>	quantitative	2018-12-05 to 2020-04-06
<code>Candidates</code>	categorical	5 levels: Biden, Bloomberg, Buttigieg, Sanders, Warren
<code>NewsDesk</code>	categorical	32 levels, including politics, science, business, etc.
<code>Category</code>	categorical	4 levels: business, politics, opinion, other
<code>Headline</code>	categorical	1127 levels
<code>SentScore(headline)</code>	quantitative	theoretical: $[-1, 1]$; actual: $[-0.86, 0.86]$
<code>SentScore(snippet)</code>	quantitative	theoretical: $[-1, 1]$; actual: $[-0.90, 0.94]$
<code>SentScore(lead)</code>	quantitative	theoretical: $[-1, 1]$; actual: $[-0.96, 0.96]$
<code>SentScore(avg)</code>	quantitative	theoretical: $[-1, 1]$; actual: $[-0.78, 0.83]$
<code>Count</code>	quantitative	$[15, 8179]$

3.1.2 View #2 data

Dataset 1:

`Debate,Date,Time(ET),Viewers,`
 1A,"June 26 2019",9-11 p.m., "~24.3 million (15.3m live TV; 9m streaming)",
 1B,"June 27 2019",9-11 p.m., "~27.1 million (18.1m live TV; 9m streaming)",
 2A,"July 30 2019",8-10:30 p.m., "~11.5 million (8.7m live TV; 2.8m streaming)",

`Location,Sponsor(s),Moderator(s)`
 "Arsht Center,Miami, Florida","NBC News, MSNBC, Telemundo","Jose Diaz-Balart, ..."
 "Arsht Center,Miami, Florida","NBC News MSNBC, Telemundo","Jose Diaz-Balart, ..."
 "Fox Theatre, Detroit, Michigan",CNN,"Dana Bash, Don Lemon, Jake Tapper"

This dataset contains information pertaining to all Democratic Presidential Primary from 26 June, 2019 to 25 February, 2020. `Debate` refers to the order of the debates, with letters used to indicate if the debate candidates were divided over multiple nights. `Date` is self-explanatory. `Time(ET)` is the EST time range the debate occurred over. `Viewers` refers to the number of viewers of the debate. `Location` refers to the location the debate was hosted. `Sponsors` refers to the media organizations that sponsored and organized that specific debate. `Moderators` refers to the media personalities and journalists who moderated the debate.

Column	Type	Range/cardinality
Debate	ordinal	12 levels
Date	quantitative	2019-06-26 to 2020-02-25
Time(ET)	categorical	5 levels
Viewers	quantitative	theoretical: [0, the population in USA]; actual: [0, 33.16 million]
Location	categorical	10 levels
Sponsor(s)	categorical	10 levels
Moderator(s)	categorical	10 levels

Dataset 2:

Date,Description,Url

"December 31, 2018","Senator Elizabeth Warren of Massachusetts forms...","https://..."
 "January 23, 2019","List of mayors of South Bend, Indiana of South Bend...","https://..."

This dataset contains information pertaining to crucial events in the Democratic Presidential Primary from 31 December, 2018 to 10 March, 2020. Date is self-explanatory. Description is a short description of the key event. Url is a URL to an article in a major news organization that reported on the event.

Column	Type	Range/cardinality
Date	quantitative	2018-12-31 to 2020-03-10
Description	categorical	~35 levels
URL	categorical	~35 levels

Dataset 3:

Date,Description

"February 9, 2019","Elizabeth Warren formally declares her candidacy for the Democratic..."
 "February 19, 2019","Bernie Sanders formally declares his candidacy for the Democratic..."
 "March 5, 2019","Mike Bloomberg declares that he will not run for the Democratic..."

This dataset contains manually curated commentary on the narrative of the Democratic Presidential Primary. Date is self-explanatory. Description contains our self-written description of a key event in the democratic primary.

Column	Type	Range/cardinality
Date	quantitative	2019-02-09 to 2020-03-10
Description	categorical	~20 levels

3.1.3 View #3 data

"Candidates", "Date", "SentScore(headline)", "SentScore(snippet)", "SentScore(lead)",
 "Biden", 2020-02-03, 0.0428571428571429, -0.13, 0.0485714285714286,
 "Bloomberg", 2020-02-03, 0.3075, -0.0775, 0.2025,
 "Buttigieg", 2020-02-03, 0.1733333333333333, 0.0883333333333333, 0.166666666666667,
 "Sanders", 2020-02-03, 0.57, 0.145, 0.125,

"Warren", 2020-02-10, 0, 0.356666666666667, -0.233333333333333,

"SentScore(avg)", "n"
 -0.0114285714285714, 7
 0.1425, 4
 0.141666666666667, 6
 0.28, 2
 0.04, 3

This dataset is derived from the View #1 dataset and contains weekly average sentiment scores for each candidate spanning between December 2018 and April 2020. Each week is represented by a date in the Date column that corresponds to the Monday of the week. SentScore(headline), SentScore(snippet), SentScore(lead), and SentScore(avg) columns here are mean sentiment scores based on the corresponding columns in the View #1 dataset. The final column n is the number of articles in that particular week.

Data abstraction of this tabular dataset with 251 rows is summarized in the following table:

Column	Type	Range/cardinality
Candidates	categorical	5 levels: Biden, Bloomberg, Buttigieg, Sanders, Warren
Date	quantitative	2018-12-03 to 2020-04-06
SentScore(headline)	quantitative	theoretical: [-1, 1]; actual: [-0.73, 0.80]
SentScore(snippet)	quantitative	theoretical: [-1, 1]; actual: [-0.77, 0.88]
SentScore(lead)	quantitative	theoretical: [-1, 1]; actual: [-0.96, 0.95]
SentScore(avg)	quantitative	theoretical: [-1, 1]; actual: [-0.68, 0.74]
n	quantitative	[1, 29]

3.1.4 View #4 data

"State", "Percentage", "Date", "Candidates"
 "National", 26.41957, 2020-02-03, "Biden"
 "National", 8.775494, 2020-02-03, "Bloomberg"
 "National", 7.012091, 2020-02-03, "Buttigieg"
 "National", 22.18171, 2020-02-04, "Sanders"
 "National", 14.63216, 2020-02-03, "Warren"

This dataset contains information about national polling averages for each candidate for each day, starting from 1 March, 2019 to 8 April, 2020. The values under the State column are always National. The Percentage column represents polling percentages that go to each candidate.

Data abstraction of this tabular dataset with 1918 rows is summarized in the following table:

Column	Type	Range/cardinality
State	categorical	1 level: National
Percentage	quantitative	theoretical: [0%, 100%]; actual: [0.2%, 57.1%]
Date	quantitative	2019-03-01 to 2020-04-08
Candidates	categorical	5 levels: Biden, Bloomberg, Buttigieg, Sanders, Warren

3.2 Data acquisition and preprocessing

3.2.1 View #1 data

The data that View #1 is based on were scraped from the NYT, using the application programming interface (API) service of the NYT (<https://developer.nytimes.com/apis>). The Python script that did the scraping and handled data preprocessing is included in `script` folder. In what follows, we provide a detailed data preprocessing pipeline.

- The original scraped data contain information about NYT articles published between 5 December, 2018 and 6 April, 2020 in the JavaScript Object Notation (JSON) format, with a snippet of the raw data shown below:

```
{
  "_id": "nyt://article/65afb39e-cb64-5243-a4d6-f6367f4efb43",
  "abstract": "President Trump invites dirty tricks in a filthy way.",
  "blog": [...],
  "byline": {...},
  "document_type": "article",
  "headline": {
    "content_kicker": null,
    "kicker": "Op-Ed Columnist",
    "main": "A Down and Dirty White House",
    "name": null,
    "print_headline": "A Down and Dirty White House",
    "seo": null,
    "sub": null
  },
  "keywords": [...],
  "lead_paragraph": "WASHINGTON \u2014 It is very disorienting when those...",
  "multimedia": [...],
  "news_desk": "OpEd",
  "print_page": "11",
  "pub_date": "2019-06-15T19:18:14+0000",
  "section_name": "Opinion",
  "snippet": "President Trump invites dirty tricks in a filthy way.",
  "source": "The New York Times",
  "subsection_name": "Sunday Review",
  "type_of_material": "Op-Ed",
  "uri": "nyt://article/65afb39e-cb64-5243-a4d6-f6367f4efb43",
  "web_url": "https://www.nytimes.com/2019/06/15/opinion/sunday/dowd-trump-.html",
  "word_count": 885
}
```

- The values from the following fields — `main`, `headline`, `lead_paragraph`, `news_desk`, `pub_date`, `snippet`, `word_count` — were extracted from each article. However, only the articles whose main headline mentioned one of the five candidates (i.e., Joe Biden, Michael Bloomberg, Pete Buttigieg, Bernie Sanders, and Elizabeth Warren) were included in the final dataset. Note also that in the current project we focus only on articles of which the headline only contained one and only one candidate.

That is, articles whose headline contained the names of two or more candidates were excluded from the dataset. We did this to simplify our analysis, as we do not have a good way to associate a single sentiment score calculated from an article with multiple candidates. We also excluded articles that had a `word_count` of 0. In total, these exclusion criteria resulted in 1,127 unique data entries in the final dataset.

- We then calculated the sentiment score for each article using the polarity sentiment analyzer from the Python library `vaderSentiment`. For each article, we calculated four **compound polarity scores**, on the basis of `main_headline`, `snippet`, `lead_paragraph`, and the average of the previous three scores respectively. That is, each article has four sentiment scores: `SentScore(headline)`, `SentScore(snippet)`, `SentScore(lead)`, and `SentScore(avg)`.¹ For simplicity, the visualization is built only on `SentScore(avg)`.
- For each article, we also classified it into one of the four categories — `business`, `politics`, `opinion`, and `other`, based on the `news_desk` field of the article. The category ‘business’ has articles from the news desks of business, Business Day, and Sunday Business; `politics` has those from Politics, National, U.S., and Washington; `opinion` contains those from Editorial, Opinion, Op-Ed, and Upshot; finally, `other` consists of the articles that do not fall into one of the aforementioned categories.

3.2.2 View #2 data

Dataset 1: Dataset 1 comes from the 2020 Democratic Party Presidential Debate schedule on [Wikipedia](#). This data has been scraped and formatted into the CSV format.

The data processing pipeline required us to remove some artifacts from the scraping, which is done in `js/timeline/data-cleaning.js`. Furthermore, we were required to parse their date format into JavaScript Date objects.

Dataset 2: Dataset 2 comes from an overall timeline of the 2020 Democratic Presidential Primary on [Wikipedia](#). The data was filtered to only include events pertaining to Buttigieg, Bloomberg, Biden, Warren, and Sanders. Key fields were then removed from the remaining data and placed into the CSV format.

Dataset 3: Dataset 3 was manually built, consisting of our attempt to highlight some particularly crucial events in the Democratic Presidential Primary timeline.

3.2.3 View #3 data

These data were derived from the data used for View #1. Specifically, instead of using raw sentiment scores, we calculated weekly mean sentiment scores spanning between December 2018 and April 2020 for each candidate. The R script `calculated_average_sentscore.R` used to derive the numbers can be found in the `script` folder.

3.2.4 View #4 data

This dataset is reconstructed from FiveThirtyEight’s presidential primary polling averages data downloaded from their [GitHub repository](#). The code for preprocessing is included in an R script `filter_poll_data.R` in the `script` folder. The major preprocessing step is just a filtering function that retains only national averages and interested candidates.

¹Note that the data from December 2018 do not have a JSON `lead_paragraph` field, so the scores based on this field for this month were artificially set to `-100`, and the average scores only came from `SentScore(headline)` and `SentScore(snippet)`.

4 Goals and Tasks

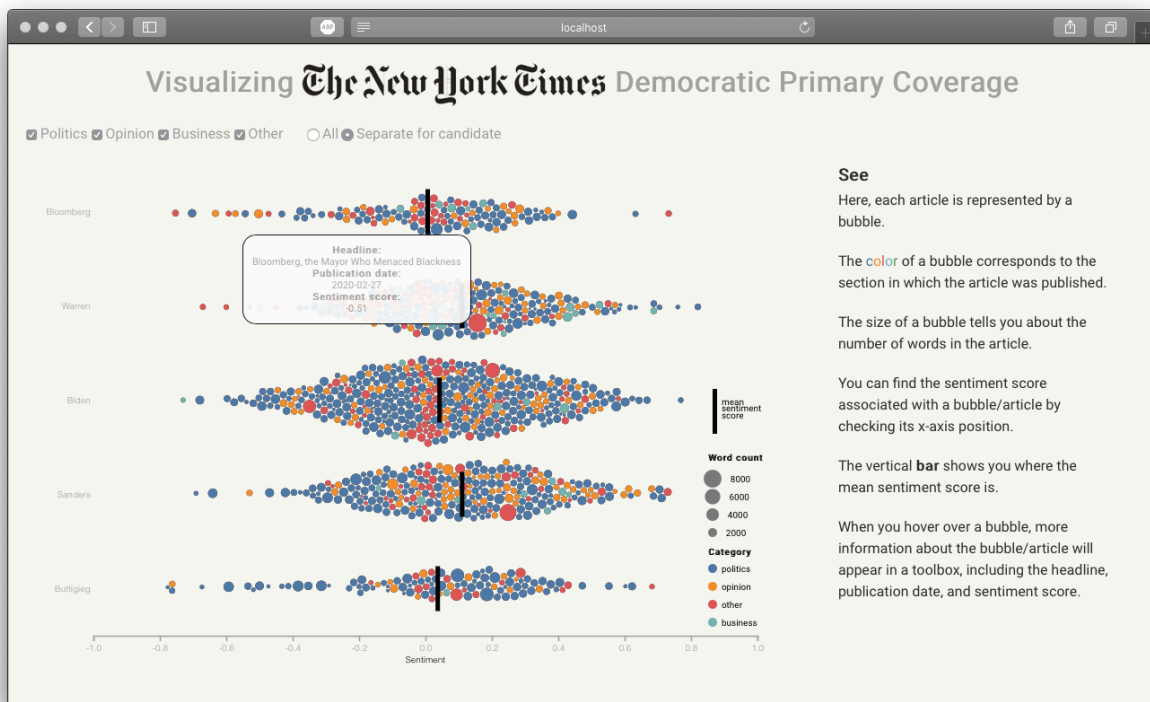
Ajay — an American student at the University of Wisconsin-Madison is interested Politics and is looking forward to participating in the process by voting in the Wisconsin Primary on April 7, 2020. Ajay is an avid and loyal reader of the NYT. After the first set of primaries, Ajay noticed a stark change in the way the media was talking about one of the candidates, Bernie Sanders, and the noticeable absence of coverage of another, Elizabeth Warren. Not wanting to jump to conclusions or swim in conspiracy theories, Ajay wants to get a sense of how fair the coverage of candidates is by the paper he reads and trusts the most.

The NYT boasts that it contains “all the news that’s fit to print”. With this set of visualizations, users like Ajay will be able to see how the NYT *shapes* all that is seen fit to print about the Democratic Party’s candidates for president. Ajay wants to [**explore**] the coverage of candidates from the early stages of the process all the way up to the Wisconsin primary to help him evaluate paper’s coverage and ultimately decide who to vote for in the Wisconsin State Primary. Ajay will [**compare**] the coverage of each candidate to see if they’ve been given a fair shot by the paper in terms of coverage frequency and whether the Times wrote about them in a positive or negative light. Ajay can [**identify and compare**] trends, [**identify or locate**] outliers, and [**browse**] features (moments in time where coverage changed) for a big picture ([**summarized**]) view of the primary process, *and* with the ability to filter by author and section of the paper for each candidate to drill down and see for himself if there are aspects *within* the paper that exhibit bias towards certain candidates compare that with how the paper *overall* treats those vying to be the democratic party’s presidential nominee.

5 Visualization

In this section, we describe the interface, visual encoding, and design rationales for each view. When different views are linked, we also provide the manner with which they are connected.

5.1 View #1

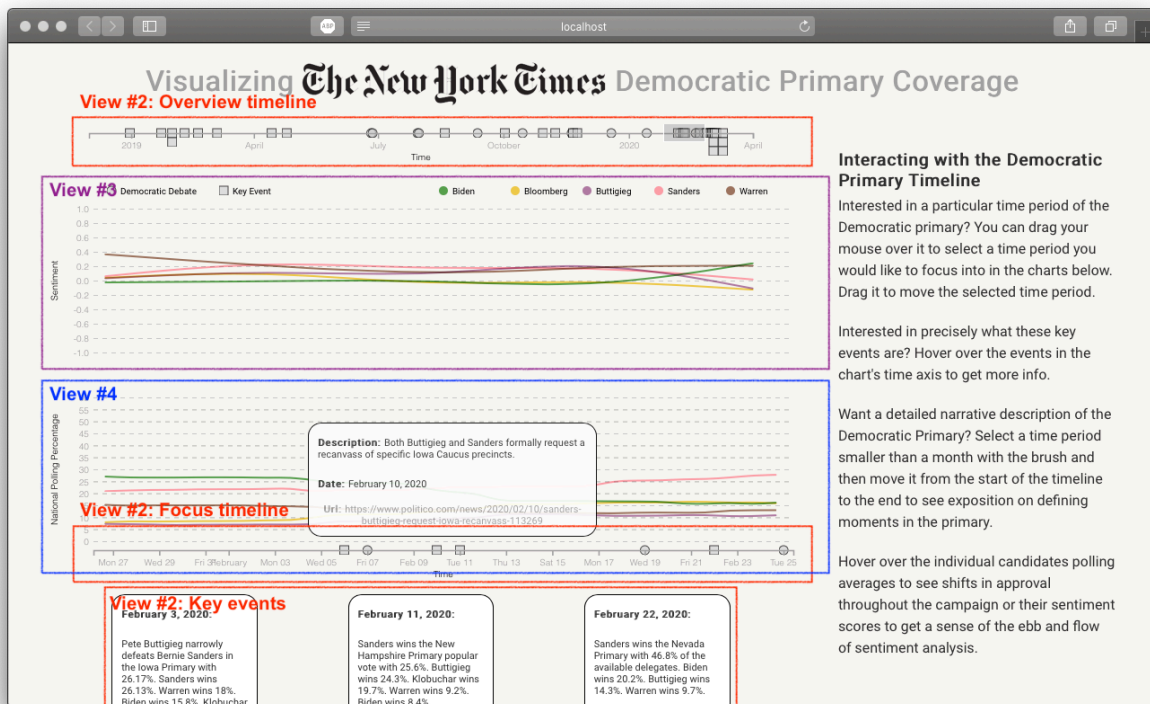


This view is independent from other views, and the interface is based on a bubble chart. The goal of this visualization is to help the user explore and compare the trends in terms of sentiment scores across candidates and over different article categories. There are two widgets the user can manipulate to interact with the bubble chart — one for controlling which article categories are visualized in the view, and the other for whether data points for different candidates should be spatially split. For categories, we opted for checkboxes as they allow the user to choose which combinations of categories to be shown. The other widget implemented is a pair of radio buttons that allow the user to choose whether to split the data on the basis of candidates. In addition, when the user hovers over a particular bubble, which stands for a single article, the headline, publication date, and sentiment score are displayed in a tooltip box.

We opted for a bubble chart mainly because it is visually engaging. Each entry in the dataset is marked with a point, with the horizontal position encoding the sentiment score ($\text{SentScore}(\text{avg})$), the vertical position encoding different candidates associated with entries in question (if the user chooses *Separate for candidates* radio button), the size of the marker representing word count (Count), and color hues standing for article categories (Category). As horizontal and vertical positions are dimensions that are visually prominent, we used them to encode two of the most important variables — sentiment scores and candidates. Word count of an article plays a less important role in the intended tasks, so it is encoded with a visually less sensitive dimension: size.

We also added bars (point marker), the horizontal positions of which reflect the mean of the sentiment scores of visible data entries. These bars are meant to help the user capture the trend in sentiment scores.

5.2 View #2



The timeline visualization consists of two separate parts:

- First, it contains an overview timeline, that sits above and separate from View #3 and View #4. The main purpose of the overview timeline is to provide a user with an overall picture of the timeline over which the Democratic Presidential Primary occurred. Furthermore, the overview timeline shows where key events in the primary occurred. The most crucial feature of the overview timeline is the time brush, which is linked with View #3 and View #4. Using their mouse, the user can select any give time range of any length, starting at any time. When the user does this, the time axis on View #3 and View #4 dynamically change to match the selected time period. Thus, this view allows the user to filter the data points they would like to see in View #3 and View #4.
- The second component is a focus timeline that is integrated into View #3 and View #4. The focus timeline serves as the time axis of View #3 and View #4 and, furthermore, uses the exact same encoding as the overview timeline. However, the focus timeline provides informative tooltips upon hovering over an event, which allows the user to dig deeper into the key events in the Democratic Presidential Primary. If they so choose, the user can navigate to more in-depth sources through a provided article URL for some of the key events. The main purpose of this view is to allow users to dig deeper into the key events of the primary and, through having a vertical hoverline along the time axis, they user can relate key events to changes in the candidates national polling percentage or the weekly average

sentiment analysis of articles of them by the NYT. For example, Sanders had a heart attack during the primary. This view allows users to (1) identify when the heart attack occurred and (2) answer the question: “Did this effect his national polling percentage or change how the New York Times covered him?”

The final, complementary piece of View #2 is a narrative component that further highlights key events in the primary that are crucial to understanding the primary. For example, who won which US states and, diving deeper, how much did they win by. And, in addition, when did candidates enter the race, exit the race, and who did they endorse upon exiting. Users will see this view appear on the bottom of the screen when they zoom into the timeline at an interval equal to or smaller than a month.

5.3 View #3

The multi-line visualization, as a single view, allows users to explore how sentiment scores from NYT articles changed over the course of the Democratic primary process. Users can hover over the sentiment scores of a particular candidate to highlight it and see how it compared with those of other candidates. Where the first view plots frequency of sentiment scores, in this view, the can contemplate trends in the scores themselves over time.

It is important to note that after plotting the original dataset, the one used in view #1, the graph was found to be too busy. The erratic movement of the lines when plotted made it difficult to read. Thus, as described in the data section earlier, a weekly average was derived to make the focus for the chart on aggregate trends as opposed to individual articles. In retrospect, a multi-line supported by a set of filtering widgets as in View #1, would have addressed this in a different way — one that would enable greater cross-functionality between the two views.

This multi-line chart gives a big picture view of how sentiment scores can shift over time. A multi-line with a manageable number of lines was chosen because it is accessible to a wide user base. Time across the x-axis and a score along the y-axis brings a familiar way of understanding data to a large audience. The familiarity and simplicity of the line chart make for a good base to add the context+focus brush of the time line.

5.4 View #4

The multi-line visualization of polling data for the five candidates completed an integrated, comparative presentation of a complex set of topics. Through a brush tool on the timeline, users are able to change their view over time and connect the occurrence of an event with the sentiment scores and polling data of one of the candidates by hovering or comparing all of them at the same time. Time is encoded on the horizontal is common for to all three views. What happened over time is a key element we wanted to have prominence, so that the user can examine the combination of attributes from three different datasets — events, sentiment scores, and polling numbers. The categorical elements — the candidates share color as a common channel, but their separation in two charts and two different sets of lines makes the distinction between the two more obvious.

The timeline with the brush interaction above and with the detailed tooltip shown on the bottom frames the full collection of views. As the user zooms in on a close enough time period, the significant events encoding event type appear and thus event, poll numbers, and sentiment scores come together. The task, of determining bias is a complex one, but choosing familiar anchoring points such as time on the horizontal, the ability to achieve the task is enhanced for a greater audience.

6 Reflection

6.1 Project development

Our original proposal was very feasible with regards to the technical limits of D3. If anything, our original proposal was not ambitious enough and failed to fully leverage the fully capabilities of D3. We had originally only planned for having a single, unintegrated timeline view, multi-line chart of average weekly article sentiment, and then a bubble chart of all the articles.

This lead us to greatly expand the technical goals of our project through adding more views with additional datasets, integrating and linking the timeline view with the two multiline views (i.e., national polling percentage multiline chart and the average weekly sentiment multi-line chart), adding a neat interactive time brush in the overview timeline that allowed users to select specific time period in the multiline charts and focus timeline, and adding a nice scrollytelling component that provided users with some information on our methodology and the views themselves.

As we continued to build this visualization, there were many further tantalizing technical goals we identified but ultimately decided we did not have enough time to complete. First and foremost, we considered adding a tooltip to our hoverline that, when moved, would display the value of each line at the point in time identified by the hoverline. Beyond that, we also considered have some means of identifying some set of articles that occurred after a key event (e.g., all articles a week after Buttigieg and Klobuchar dropped out) and looking at the average sentiment analysis of the these articles.

What is less expected, however, is that the patterns revealed in the visualizations are not as clear-cut as anticipated. In particular, the sentiment scores across candidates are largely very similar. While the lack of clear trends does not impact the design of our visualizations, it does undermine the kinds of claims we can draw from the data.

In what follows, we provide some causes for this lack of distinct patterns:

- As described in Section 3.2.1, we only considered articles whose headline mentioned one and only one candidates. We excluded articles that mentioned multiple candidates because there is no straightforward way to assign sentiment score for each candidate in such cases. Doing this can potentially hinder observable trends.
- We only calculated sentiment scores based on the headline, snippet, or lead paragraph of an article, as the entire text was not available with the NYT API. It is reasonable to expect that doing sentiment analysis on the entire text can potentially provide a more accurate picture.
- In some cases, the sentiment scores estimated using the three text chunks (i.e., headline, snippet, and lead paragraph) are not consistent — with one score being negative and another one positive. This is partially why we decided to use the average over the three scores, in an attempt to mitigate the effect of this inconsistency.

6.2 Future scope

We would like to implement more from the space where task abstraction and data abstraction mingle, including the ability to refine our visualizations further, to help facilitate deeper, and more rigorous analysis of whether algorithmic analysis like sentiment analysis could determine bias.

Some examples of this would have been to provide more tools to go back and forth between the micro and macro and to see how they feed into one another

Another experiment we would like to take is to bring the user directly into the data — to have them score the headlines and snippets of a few articles, and then have the results dynamically plotted alongside the machine-estimated sentiment scores.

7 Team Assessment

Name	Task	Responsibility
Roger	Data collection and preprocessing	Collected NYT data for View #1 using the NYT API and preprocessed the data to extract relevant fields and to calculate various sentiment scores.
	Scrollytelling View #1	Implemented the basic scrollytelling layout and functionality. Implemented all features associated with View #1: bubble chart, legend, widgets, force simulation.
	Aesthetics	Unified the styles across views.
	Milestones Submission	Basic set-up and wrote assigned parts. Submission of milestones.
Jeff	Data Processing View #2	Processed all data for the Timeline (View #2). Implemented all features associated with View #2: plotting all timeline data, hoverline, tooltip, legend, context+focus time brush, and linking brush to View #2, #3, and #4.
	Merged View #2 and #3	Combined View #2 and View #3 into one view that was linked with the brush from the Overview timeline.
	Narrative Text	I manually created the narrative text data for View #2. I also handled all the technical details with placing and removing the narrative text, which required working with CSS's flexbox extensively.
	Milestones	Wrote assigned sections.
Mike	Data collection	Collection and processing of timeline and polling data, including filtering and sorting data to find irrelevant and problematic pieces.
	View #3	Implemented Multi-line chart of sentiment analysis. Tried extensively, but ultimately struggled to completed a working version of the hoverline that would show the actual values for both sets of views, polling and sentiment. Moreover, extracted data for articles from sentiment analysis scores with the hopes of being able to incorporate within timeline (articles with lowest and highest scores for sentiment).
	View #4: Polling	Worked together with teammates to implement this view and integrate into larger view.
	Milestones	Wrote agreed upon sections of report.