TwoFortyNine :: Kazi Jamal, Eric Lau, and Raymond Lee
SoftDev1 pd9
P04 — Let the Data Speak
2020-05-11

coviD3.js by TwoFortyNine

Roster and Roles

- **PM Kazi Jamal:**
  - Project management
  - Set milestones and track progress of team
  - Update design doc and make sure devlog is being maintained
  - Work on frontend and styling
  - Handle minor backend tasks as necessary
- **Eric "Morty" Lau:**
  - Primarily work on transportation section
  - Preprocessing raw data for MTA
  - Create Flask routes to retrieve data
  - Use D3 to create data visualizations for the transportation section
- **Raymond "ray. lee." Lee:**
  - Primarily work on sentiment analysis section
  - Preprocess raw data for public media
  - Create Flask routes to retrieve data
  - Use D3 to create data visualizations for the public media section

Description

coviD3.js is a website run by TwoFortyNine. We specialize in telling stories through beautiful and engaging data visualizations. With coviD3.js, we plan on analyzing the ways the coronavirus pandemic affects society outside of the hospital. For our first week, we plan on publishing two articles on changes in media sentiment and transportation.

Features

Yellow - Additional

- **Dashboard: "/"**
  - Display coronavirus cases data
    - Data: https://www.kaggle.com/sudalairajkumar/covid19-in-usa
  - Display links to other features
- **Sentiment Analysis: "/sentiment"**
  - Display patterns and analysis of public opinion during COVID-19 using sentiment analysis and natural processing Python libraries on the datasets
  - **Public Media: "/sentiment/media"**
    - Data: https://www.kaggle.com/jannalipenkova/covid19-public-media-dataset
    - Word cloud
    - Positivity/negativity
      - Analyze the relationship between number of COVID cases across time and effects on media
      - Analyze the differences in sentiment among news outlets

    - Analyze the frequency of named entities (ex: China, U.S., COVID-19)
      - Plot the frequencies of each day onto a barplot, and swap the bars as the animation transitions from start date to end date
      - http://bl.ocks.org/bycoffe/21061661b1450a4db92a
  - **Trump Tweets: “/sentiment/trumptweets”**
    - Data: https://www.kaggle.com/austinreese/trump-tweets
    - Word cloud
    - Positivity/negativity
      - Analyze the relationship between number of COVID cases across time and effects on media
- **Transportation: “/transportation”**
  - Display transportation changes in New York City during quarantine
  - **MTA / CitiBike: “/transportation/nycpublic”**
    - MTA Ridership Data: http://web.mta.info/developers/turnstile.html
      - We will only process the data once rather than making requests multiple times. We are creating Python utility files to do this processing for us. However, if we finish all of our MVP features, we will attempt to create a GitHub Actions script that will run our processing file every weekend and update our CSV.
    - CitiBike Data: https://www.citibikenyc.com/system-data
    - Medium Income Data: currently looking for a data source
    - Compare MTA / CitiBike usage from previous years to 2020
      - Analyze usage changes by borough
    - Mask usage changes on top of average income of communities
    - What stations are people more likely to get on or get off at
    - Have buttons to transition data between last 30 days, last month, or last year
    - Data on different durations will have different scales (days vs weeks)
    - Analyze age and gender of CitiBike users
  - **Mobility: “/transportation/mobility”**
    - Data: https://www.apple.com/covid19/mobility/
    - Compare changes in mobility between different countries / regions
- **Numbers Section: “/numbers”**
  - Display general statistics and numbers for various topics
    - Netflix
    - TikTok
    - Video Games (Nintendo Switch and Animal Crossing: New Horizons)
    - Unemployment

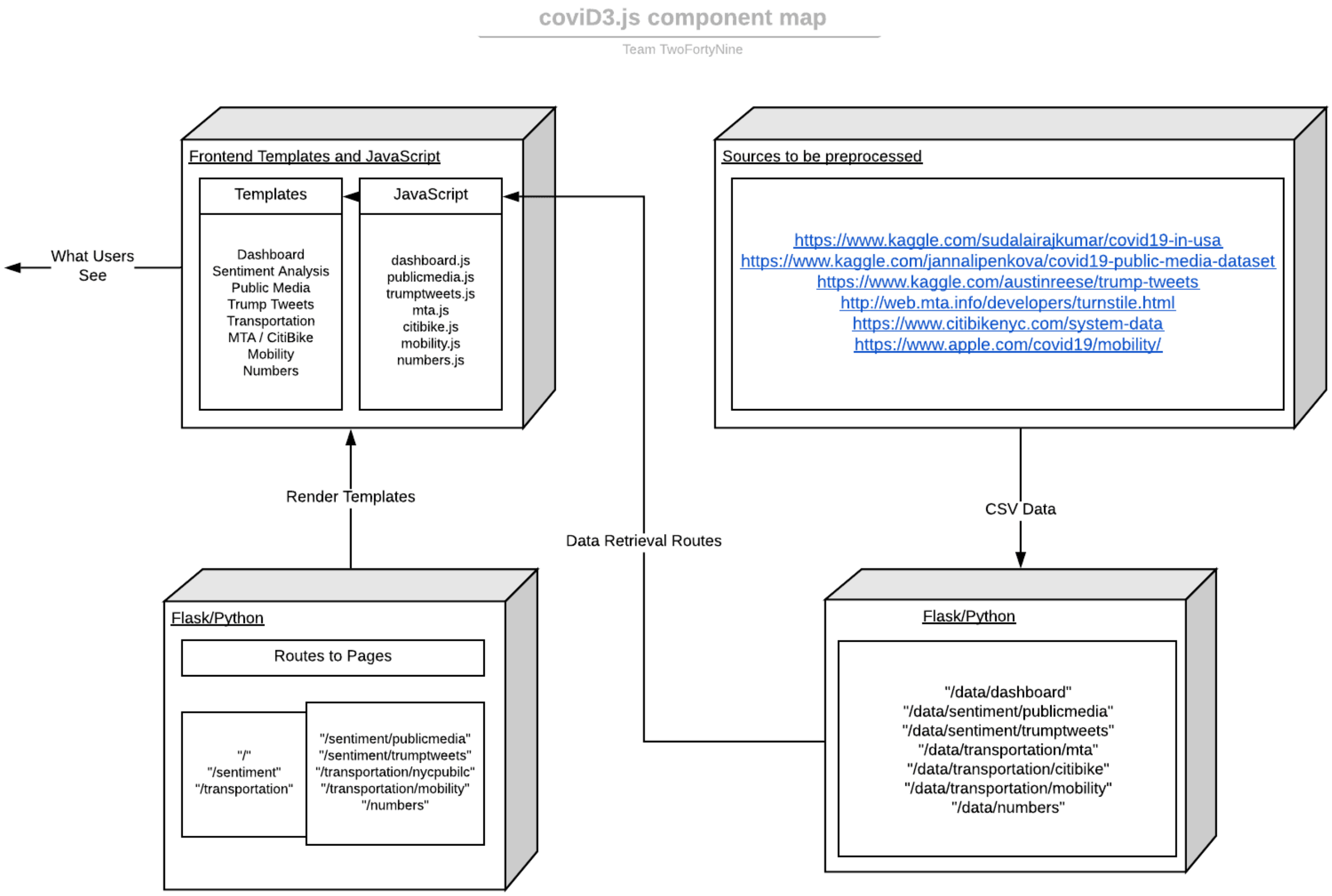
Visualizations

- Dashboard
  - Number of US cases and number of US deaths
  - Line chart of US cases and deaths
  - Choropleth of US states and cases
- Public Media
  - Line chart describing number of articles written over time on a daily, weekly, and monthly basis
    - Second line showing US cases

    - Bar chart for subjectivity/objectivity among news outlets
    - Word cloud showing most used words in the articles
- Trump Tweets
    - Line chart or scatter plot for sentiment polarity (positivity/negativity)
        - y-axis -1 to 1 and x-axis January 1, 2020 to latest tweet date
        - Each tweet is a data point
    - Bar chart for sentiment polarity (positivity/negativity)
- MTA
    - Line chart describing MTA ridership on a daily and weekly basis
        - Second line showing NYC cases
    - Separate line chart describing MTA ridership only in 2020
    - Choropleth that breaks down cases by zip codes
        - Circles showing subway stops
    - Choropleth that breaks down cases by borough
    - Multi-line chart that shows ridership by borough
    - Maybe choropleth that breaks down cases by medium income

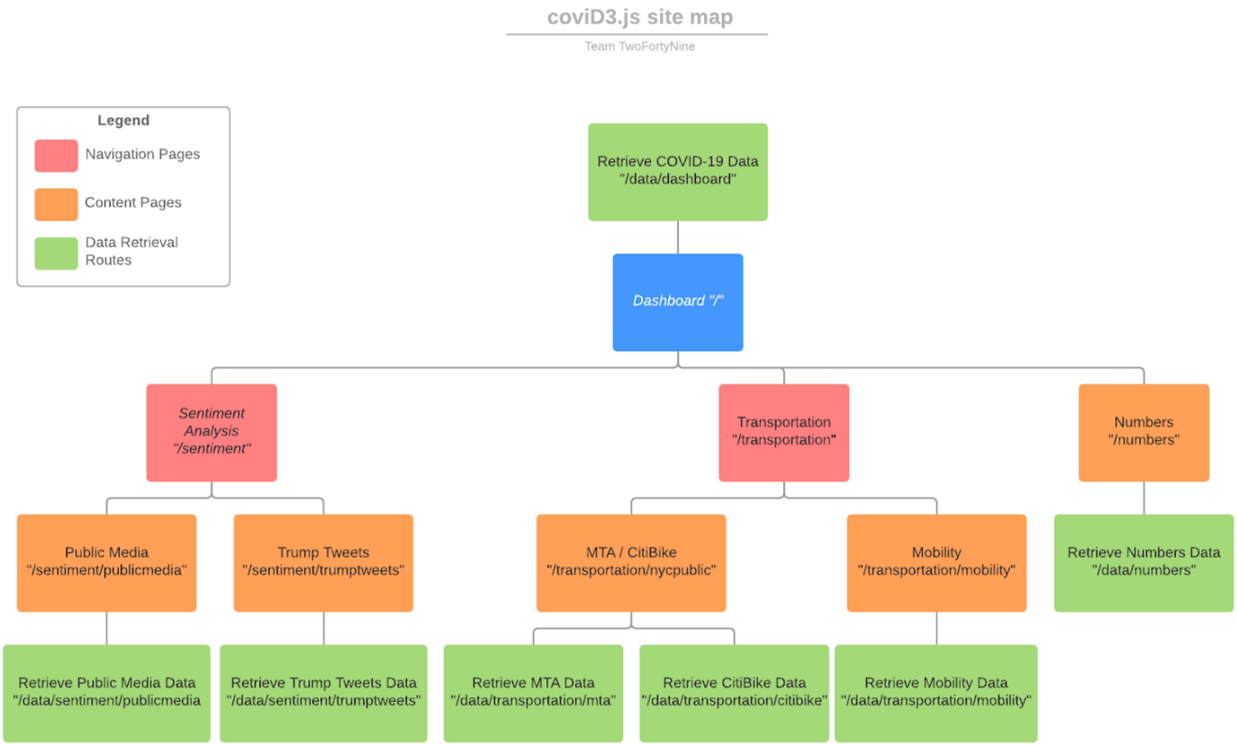
Component Map

https://www.lucidchart.com/documents/edit/570c4b9e-74be-4e99-9fd8-99b96d134f23/0_0?beaconFlowId=972DE51459513445


coviD3.js component map
Team TwoFortyNine
What Users See
Frontend Templates and JavaScript
Templates
Dashboard
Sentiment Analysis
Public Media
Trump Tweets
Transportation
MTA / CitiBike
Mobility
Numbers
JavaScript
dashboard.js
publicmedia.js
trumptweets.js
mta.js
citibike.js
mobility.js
numbers.js
Sources to be preprocessed
https://www.kaggle.com/sudalairajkumar/covid19-in-usa
https://www.kaggle.com/jannalipenkova/covid19-public-media-dataset
https://www.kaggle.com/austinreese/trump-tweets
http://web.mta.info/developers/turnstile.html
https://www.citibikenyc.com/system-data
https://www.apple.com/covid19/mobility/
Render Templates
Data Retrieval Routes
CSV Data
Flask/Python
Routes to Pages
"/"
"/sentiment"
"/transportation"
"/sentiment/publicmedia"
"/sentiment/trumptweets"
"/transportation/nycpubilc"
"/transportation/mobility"
"/numbers"
Flask/Python
"/data/dashboard"
"/data/sentiment/publicmedia"
"/data/sentiment/trumptweets"
"/data/transportation/mta"
"/data/transportation/citibike"
"/data/transportation/mobility"
"/data/numbers"

Site Map

https://www.lucidchart.com/documents/edit/1259f871-b0cf-4406-bf9d-39c5888ae3b0/0_0?beaconFlowId=C3D030618B147B4E


coviD3.js site map
Team TwoFortyNine
Legend
🟥 Navigation Pages
🟧 Content Pages
🟩 Data Retrieval Routes
Retrieve COVID-19 Data "/data/dashboard"
Dashboard "/"
Sentiment Analysis "/sentiment"
Transportation "/transportation"
Numbers "/numbers"
Public Media "/sentiment/publicmedia"
Trump Tweets "/sentiment/trumptweets"
MTA / CitiBike "/transportation/nycpublic"
Mobility "/transportation/mobility"
Retrieve Numbers Data "/data/numbers"
Retrieve Public Media Data "/data/sentiment/publicmedia
Retrieve Trump Tweets Data "/data/sentiment/trumptweets"
Retrieve MTA Data "/data/transportation/mta"
Retrieve CitiBike Data "/data/transportation/citibike"
Retrieve Mobility Data "/data/transportation/mobility"

Data Transfer

- Dashboard
  - Collect data from various sources and create a CSV containing that data
  - Create Flask route to pass that CSV into JavaScript
  - JavaScript will parse the CSV using d3.csv
- Sentiment Analysis
  - Public Media
    - Tokenize each sentence of the article content for each row in the csv.
    - Preprocess the tokens
      - Remove stop words
      - Remove stem endings (lemmatization)
      - Record the frequency of words (unigrams, bigrams)
    - Store processed data into a CSV file(s)
    - Create Flask route to pass that CSV into JavaScript
    - JavaScript will parse the CSV using d3.csv
  - Trump Tweets
- Transportation
  - Turnstile data
    - Preprocess raw turnstile data
      - Send HTTP requests to that link and retrieve data as text
      - Retrieve data from January 2019 to today
    - Store processed data into a CSV file
    - Create Flask route to pass that CSV into JavaScript
    - JavaScript will parse the CSV using d3.csv
  - Map data
    - Store map data on subway stops and New York City in json files
    - Read files from JavaScript with d3.json
- Numbers Section
  - Collect data from various sources and create a CSV containing that data
  - Create Flask route to pass that CSV into JavaScript
  - JavaScript will parse the CSV using d3.csv

Frontend Framework

We will be using Bootstrap as our frontend framework because we have all used it extensively and are comfortable with it.

Extra Packages

- spaCy for most of the natural language processing
  - Use spaCy to remove stop words and extract named entities out of text
- TextBlob for sentiment analysis
  - Use TextBlob to identify the positivity and negativity of each