

# CAPP30122 – Final Completed Software

Team: python\_parser\_tongues

## **Team Members (CNetIDs):**

Ken Kliesner (kenkliesner)  
Gabriela Ayala (mariagabriela)  
Fabian Aranedo Baltierra (baltierra)  
Stefan Manuel (stefmanuel)

## **Project Overview:**

Project Modeling Covid-19 Vaccinations vs US Political Party Affiliation. The goal of our project is to analyze correlation between vaccination status (unvaccinated, fully vaccinated, or boosted) and 2020 US general election data. We reviewed both sets of data at the county-level and ran regressions on vaccination rates to identify all relevant control and explanatory variables (ie, possible controls: state density, income, race, etc.). There is an interactive data visualization component with a US State/County Map including components from each data source.

## **Overall Structure of Software:**

- 1) project -- main directory
- 2) install.sh -- creates virtual env and runs necessary packages
- 3) requirements.txt -- package requirements for virtual environment
- 4) data -- contains the project's data
  - i) data.csv -- final merged file with CDC data, election results and controls
  - ii) merge.py -- code that merges datasets
  - iii) election\_results.csv -- scraped data with 2020 Presidential Election results
  - iv) fips\_codes.csv -- list of FIPS codes used for merging purposes
  - v) controls.csv -- contains downloaded controls data at county level
  - vi) abbreviations.csv -- list of state names and abbreviations for merging purposes
  - vii) cdc\_api -- directory with code to extract data from CDC API and extracted data
  - viii) election\_crawler -- directory with crawler code
- 5) visual\_exp -- contains code for data visualization map
- 6) regression -- contains regression code of vaccination rates on election results
- 7) past\_deliverables -- includes deliverables 1 and 2
- 8) README.md -- description of this same structure

## **Coding Responsibilities by Team Member:**

**Ken Kliesner:** *Create Web Scraper for the US Election Data by County/State*

Found a 2020 US Election Data source to utilize as the main source of election results. Created a web scraper program to crawl through this website and access each different state's page and go down further to find the election results for each state's county. Also created and typed up the project deliverables/papers and helped organize project repository.

**Gabriela Ayala:** *Retrieve CDC Vaccination Data for API*

Retrieved vaccination data at county level from CDC API. Obtained vaccination data for each county's population, ages 18+ and 65+. Created code to merge vaccination data, election data, and controls into a single data.csv file. Also organized full project repository, creating files for \_\_init\_\_, \_\_main\_\_, README, etc. as necessary to link everything together.

# CAPP30122 – Final Completed Software

Team: `python_parser_tongues`

**Fabian Araneda Baltierra:** *Mapping Visualization*

Researched the best way to display the data using geographical information. Decided to use Plotly and Dash to showcase the relevant information in an interactive map of the US, colored by counties and offering different views of the information.

**Stefan Manuel:** *Collect Control Dataset for Regression Models and Run Regressions*

Tested different regression models, converted categorical variables into binary indicators, and built a weighed least squares regression, using population as a weight to prevent outliers. Regression model ultimately used to include predicted vaccine completeness results in the map visualization.

**How to Interact with Program and Functionality:**

To interact with the application, simply enter the following in the command line *after* accessing the virtual environment (`install.sh`):

**`./install.sh`**

**`python3 project/visualization/map_dem_rep.py`**

This will return a URL, which just needs to be copy and pasted into your preferred web browser, which will be an interactive map of the United States. You will be able to click on any state in the US and zoom in to see a view with its counties separated out. You will also be able to select different filtering options to separate out data on the map in different visualizations.

**What the Project Tried to Accomplish VS What It Actually Accomplished:**

With this program, we were able to accomplish multiple tasks within the software and were almost completely successful in achieving the goals we set from the beginning!

We were able to crawl through the election data mostly as expected. There were some slight issues due to the way the html code was programmed, but we were able to work through those with the crawler file. We obtained data on percentages that received a complete series (1 Johnson & Johnson shot or 2 Pfizer/Moderna shots) and corresponding booster for the county's population, ages 18+ and 65+. The code retrieves the most up-to-date data by epidemiological week. Excluded observations from counties were categorized as "unknown" for which no FIPS code was recorded. In building the regression model, we tested a variety of different models to see what fit, including OLS, Logistic, Quadratic, etc. We eventually settled on a Weighed Least Squares regression, using population as a weight to prevent outliers like limited supplies of vaccines or very sparse counties from affecting results.

One issue we had with the data visualization was that there was no way to add correlation between different regression variables we had prepared onto the map. Also, we were not able to color-code the map (ie, red for republican and blue for democrat), while also using a color gradient that gets darker for higher vaccination percentages, and lighter for lower percentages.