

Data Preparation and Setup

1. Database System -

MySQL Workbench, version: 6.3.8 (a newer version may work too)

Link: <https://www.mysql.com/downloads/>

2. Data -

us_counties_infections.csv (sample size), us_state_vaccinations.csv, and population-counties.csv data have been provided in our zip file for convenience.

A. Cases:

<https://github.com/nytimes/covid-19-data/blob/master/us-counties.csv>

Export to csv, save in the app folder as us_counties_infections.csv and then run the script cumulSetup.py by typing “python cumulSetup.py” in a command prompt/terminal.

*For our submitted sample dataset, we only provided county cases data starting from 2/1/2021. However, for our demo we used the original/whole data set that started on 3/23/2020. This is to limit the file size for upload purposes. You can download the full data from the link mentioned above.

B. Vaccinations:

Vaccinations data was obtained from

https://github.com/owid/covid-19-data/blob/master/public/data/vaccinations/us_state_vaccinations.csv (by CDC).

Export only date, location, total_vaccinations, total_distribution, people_fully_vaccinated, and daily_vaccinations to csv, and save in the app folder as us_state_vaccinations.csv. After saving, the date format will need to be updated via Excel tools to the following format “YYYY-MM-DD”. There is a section of code at the end of our schema.sql file which will import this data to the database, given that you save this csv file in the location: “/var/lib/mysql-files/us_state_vaccinations.csv”

Note: since the vaccinations data file was 321 kb, we included it in our zip file submission for convenience.

C. Population data:

https://github.com/balsama/us_counties_data/blob/main/data/counties.csv

The entire population data is necessary for our app to work. It should be named population-counties.csv. There is a section of code at the end of our schema.sql file which will import this data to the database, given that you save this csv file in the location:

“/var/lib/mysql-files/population-counties.csv”

Changes we made to NY population -

To update New York's county populations since they were recorded inaccurately,

<https://www.census.gov/data/datasets/time-series/demo/popest/2010s-counties-total.html> was used to get the populations “POPESTIMATE2019” for each New York counties. Then directly updated the population-counties.csv file with the correct population.

Note: since the population data is 136.9 kb, we included it in our submission upload for convenience.

2. To load this data into the database system:

We used SQL to design our database for this project. Using MySQL, you can run schema.sql which contains all of the CREATE TABLE statements as well as the loading statements to pull data from the csv. After running schema.sql is when you would run the cumulSetup.py file mentioned before to load the infection data. In cumulSetup.py you will need to enter your MySQL db password near the top of the file. This script calculates daily counts given the cumulative data. After this, you would run both procedures.sql and county_pop_proc.sql to create the additional procedures necessary for the app to run.

Application and code

1. Programming language: Python, version: 3.6

Also used HTML, CSS, JS. (No installation needed)

2. This application runs Flask with Python 3 (version 3.6 or greater). To install the requirements, assuming you have Python 3 and pip installed, type “pip install -r requirements.txt”. This will install flask and flask-mysql. (for reference you can also install Flask here: <https://flask.palletsprojects.com/en/1.1.x/>)

3. GUI: To run the app, first export the FLASK_APP environment variable, then run the flask run command:

```
"export FLASK_APP=app.py
```

```
flask run"
```

Before being able to run the commands, you will need to enter your db password near the top of the app.py file. There is a comment denoting where to place your password.

Code Documentation and References

1. Existing code references and changes:

For the D3 aspect of our app, the maps, we used existing templates. The template we used for the county view can be found here: <https://gist.github.com/msbarry/ab59a22abacc916cd7e8b37302754cf3> (Github gist). We changed the color scheme to better reflect the theme of our data, and we changed the number scale and legends since our domain of numbers was different. For example, in one case our numbers were much larger, and in another case, we had to change the scale to be in percentage format. We also changed some variables to better reflect our data. Additionally, we changed the file to pull data from a csv file name which was passed in, instead of pulling from a single statically named csv file. We also changed the titles to be relevant to our data.

In addition to the map, we added to this html file the user input field of date. We also set it up so this field input was sent as a POST request when the user hit the submit button. This part was not included in the template D3 code. We also added the back button which would take users back to the home page.

For the state view we used this template: <https://bl.ocks.org/dnprock/5215cc464cfb9affd283>. We changed the value format displays. Originally it displayed just a number, but we made it so that the vaccination/population ratio map would show a percentage value, and the vaccination distribution map would show in "K"(thousands), "M"(millions), and "B"(billions). As mentioned with the county view map, we added the user input date field, submit, and back buttons as well.

2. Files written by us:

Schema.sql

Procedures.sql

County_pop_proc.sql

App.py

cumul_Setup.py

Requirements.txt

Index.html

README.md

view_cases_county.html*

view_county_ratio.html*

view_daily_cases.html*

view_deaths_county.html*

view_vac_dis.html*

view_vac_ratio.html*

*These include the d3 templates mentioned above. So these * files are partially written by us, and partially the existing templates.

3. Images of our app working:





