Tom Recht
5.26.20

ETL Project Report

For my project, I decided to compile data on COVID-19 cases and air quality in the US. I wanted to put together a database with county-level data on different parameters of air quality and on reported cases and deaths from COVID-19, to facilitate investigating the possible effect of air quality on disease incidence.

Extraction

For COVID-19 data, I used the New York Times county-level database, available at https://github.com/nytimes/covid-19-data/. This contains a file "us-counties.csv" which gives reported case and mortality data at the county level, ordered by date. I downloaded this file on 5/20, so my data only goes up to that date.

For air quality data, I used files from the EPA's AirNow website at https://docs.airnowapi.org/files (signup required for access). This contains folders with various types of data files (e.g. hourly, daily totals etc.), one folder per date. I ascertained which specific data file I needed ("daily_data_v2.dat", which gives overall data for a given date) and wrote a Python script to download all such files for dates from 2/1 to 5/20. Although these are ".dat" files, they are in CSV format. Each file contains one day's readings for various air quality parameters, organized by reporting site (with site name and ID but no state/county names). The air quality parameters reported are: ozone 1-hour, ozone 8-hour, CO2 8-hour, particulate matter 2.5μm 24-hour, particulate matter 10μm 24-hour, and sulfur dioxide 24-hour. Most sites do not report all six parameters for every date.

I also wanted to get county population data so that cases could be calculated in proportion to population. I found an Excel table with this data on the Census Bureau site, https://www.census.gov/data/datasets/time-series/demo/popest/2010s-counties-total.html, giving data from 2010-2019 for each US county.

The EPA files are organized by site name and ID, which does not usually correspond to a county name, so I needed a way to match up the EPA data with the NYT data. The NYT data identifies counties by 5-digit FIPS code (a unique federal identifier), but this did not seem to appear in the EPA data. I eventually found that the EPA's 12-digit site identifiers contained the FIPS as a substring. But the Census Bureau data did not include FIPS at all, so I found an online table giving FIPS codes by county and state (https://www.census.gov/data/datasets/time-series/demo/popest/2010s-counties-total.html) and downloaded this into a pandas dataframe.

Transformation

Most of the transformation work was on the EPA files. First, these include some non-US data which I had no use for. I looked into the 12-digit site ID codes and found that the first three digits were the ISO country code, so I deleted all data that did not begin with '840' (the US country code). I then extracted the substring of the site ID that corresponds to the FIPS and added that in a new column. I also dropped several columns that seemed irrelevant (e.g. latitude and longitude of the reporting sites).

At first I created dataframes for each of the EPA files, i.e. one dataframe per date, but I then decided to join them into a single master dataframe. Since there are six parameters of air quality reported, I also divided this master dataframe into six separate dataframes, one for each parameter.

For the county population data, I dropped all but the 2019 values in Excel.

The NYT and EPA tables use different date formats, so I changed the format of the former in Python. Some of my sources used two-letter abbreviations for the state names and others used the full name, so I changed the latter to the abbreviations. Since county names are not unique across states, I added a 'full county name' column to all tables with county and state columns (e.g. 'Oakland, CA' in addition to 'Oakland' and 'CA'). I also made some other formatting and data type changes, for example changing population values from strings with thousands commas to integers.

Loading

I created the following SQL tables:

- one table for the NYT COVID-19 data, with date, county, FIPS, cases and deaths reported
- six tables for the air quality data, corresponding to the six reported parameters; these contain date, site name, value reported, air quality index (for some of the parameters, since this is not calculated for all parameter types), and FIPS
- one table giving FIPS code for each US county
- one table giving 2019 population for each US county

I populated these tables from CSV files created from my pandas dataframes. I kept the CSV files for possible future use in exploring correlations etc.

I considered using MongoDB, but ended up going with SQL. A reason to use MongoDB would have been that the air quality reports from each site don't all report the same parameters, so the data structure is different from one record to the next. But since there are only six total parameters, I decided the best way to deal with this was simply to create six parameter-specific SQL tables.

With this dataset, it should be possible to (a) calculate COVID-19 incidence over time per county in proportion to population, and (b) correlate that with air quality measurements for the six reported parameters and/or with air quality index values. One wrinkle would be that if air quality does have an effect on COVID-19 incidence, there would presumably be a few days' lag between a low air-quality event and a spike in reported COVID-19 cases, so you would want to correlate air quality measurements at date d with COVID-19 reports at date $d+n$, where n is some number of days that it might reasonably take for the effect to become apparent.