Outbreakpy: Function Development, Data Smoothing, Parameter Testing, and Dynamics Vignette

Hritika Chaturvedi
Mentor: Emory Hufbauer

Scripps Research Translational Institute

ANDERSENlab The Scripps Research Institute

Andersen Lab

Incorporates computation, genomics, and large-scale data analysis to assess the behavior of epi/pandemics, viral resistance, and early detection to ultimately inform outbreak responses.

What Is Outbreak.info?

Platform that summarizes available COVID-19 and SARS-CoV2 epidemiological and genomic data and published research. Researchers, health officials, and the public have access to downloadable data and visualizations on cases, deaths, and genomic variants.[3]

Outbreakpy Project

The larger Outbreakpy (name for our python module for outbreak data) team intends to make outbreak.info available to a broader audience in python, whether that be in allowing real-time surveillance of the pandemic, presenting visualizations that make patterns in data apparent, or creating a comprehensive user experience. In order to further this, this work will be available as open-source on outbreak.info.[2]

Personal Objectives

- To work with developers via GitHub (version control), communicate with project managers, and understand the structure and function of different scientific computing projects in the lab
- To study the growth dynamics of the SARS-CoV-2 pandemic around the world by modeling data from the outbreakpy tool developed previously, noting the limitations of models, and critically reading scientific papers

My Contribution

Involved defining and testing a function (cases_by_location) that returns a data frame specific to the location requested by the user and includes either data regarding the increase in SARS-CoV2 cases by date (confirmed_numIncrease) or weekly averaged data (confirmed_rolling) depending on the argument entered for the pull_smoothed parameter. This feature for smoothed rolling cases provides a user-friendly library for retrieving data in that the "noise" in a dataset is reduced(Fig.1).[4] The function raises the appropriate error message if either the location or pull_smoothed arguments is unacceptable (Fig. 2).

Fig. 1

```
def cases_by_location(location, server=server, auth=auth, pull_smoothed=0):
    if isinstance(location, str):
        location = location.replace(" ", "")
        location = list(location.split(","))
    if not isinstance(location, list) or len(location) == 0:
        raise ValueError('Please enter at least 1 valid location id')
    if pull_smoothed == 0:
        confirmed='confirmed_numIncrease'
    elif pull_smoothed == 1:
        confirmed='confirmed_rolling'
    elif pull_smoothed == 2:
        confirmed='confirmed_rolling, confirmed_numIncrease'
```

Fig. 2

```
except:
    for i in location:
        raise Exception('{} is not a valid location ID'.format(i))
```

Although a work in progress, I worked to develop a function that runs the test cases necessary in verifying that the cases_by_location function is responding as expected to both acceptable (pull_smoothed = 0, 1,or 2; valid location ID) and unacceptable arguments.

Fig. 3

```
def test_confirmed(prev_data, saveCases, params_dic): #inner function
    out=outbreak_data.cases_by_location(**params_dic)
    csv_out = out.to_csv(index=False)
    assert(params_dic['location'] in out.admin1.unique(), 'missing location in data')
    assert(prev_data == csv_out, 'old csv data does not match the current csv')
    if saveCases:
            return csv_out
```

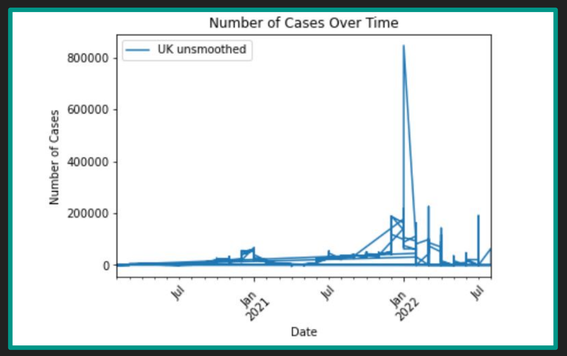
Part of a larger function, this inner portion runs cases_by_location for a case, saves the csv file of the data produced, ensures that the location ID is valid, and that the expected output for the case matches the saved output. If there is no discrepancy, the saved csv file is returned.

Tools

- Elasticsearch API: allows records to be searched without an index; data that's the most relevant to the user's query is returned[1]
- Python
  - Sys and requests: to access data from the API, parent directory, and files of interest
  - Pandas: to form visualizations and tables
  - Pytest: to test functions and ensure that they produce the expected output
- GitHub: to import data, merge changes, and run test cases
- Conda: package manager for scientific computing; gave it a list of dependencies for outbreakpy, including jupyter, and it installed them on my computer

Example Visualization

A plot of unsmoothed SARS-CoV2 cases in the UK produced using cases_by_location. Can be used to note trends and associate with supplemental data to hypothesize reason for progression and spike patterns. By using pull_smoothed=2, a comparison of smoothed/unsmoothed data can also be achieved.



References

1. Elasticsearch REST API, Overview and Tips. 2017 Oct 30. Logzio. [accessed 2022 Aug 8]. https://logz.io/blog/elasticsearch-api/#:~:text=This%20type%20of%20Elasticsearch%20AP
2. outbreak.info. GitHub. [accessed 2022 Aug 8]. https://github.com/outbreak-info.
3. outbreak.info. outbreakinfo. https://outbreak.info/.
4. 6.4.2. What are Moving Average or Smoothing Techniques? wwwitlnistgov. https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc42.htm.