

Learning Lists and Dictionaries by Building Web Dashboards with Live Data

Nifty Assignment

Eric D. Manley

Department of Mathematics and Computer Science

Drake University

Des Moines, IA 50311

`eric.manley@drake.edu`

This Nifty Assignment tasks students with creating an interactive, web-based data visualization dashboard connected to a live data source. The assignment is targeted for late-CS1/early-CS2 after students have experience with loops and know how to access individual items in lists and dictionaries. Students are walked through the use of Python libraries for making HTTP data request, creating visualizations, and integrating them into a web framework with user interface components. The challenge of the assignment comes in working with the received data, containing nested lists and dictionaries. Students learn to explore the data, selecting needed fields as well as iterating through and filtering. Overall, there are several benefits of utilizing these tools:

- students get experience working with real-world, non-trivial examples of nested lists and dictionaries
- the application students develop is visually appealing, fun to use, and can be deployed and shared
- students get a taste of how to build professional-quality web applications
- the data visualization aspect provides early exposure to data science work for students who might be interested in that area
- the assignment serves as the springboard for many creative projects through substitution of different data sources and visualization types

An example of an end-result of the assignment be seen in Figure 1. Assignment materials can be found at

<https://analytics.drake.edu/~manley/nifty/NiftyCOVIDDashboard.html>

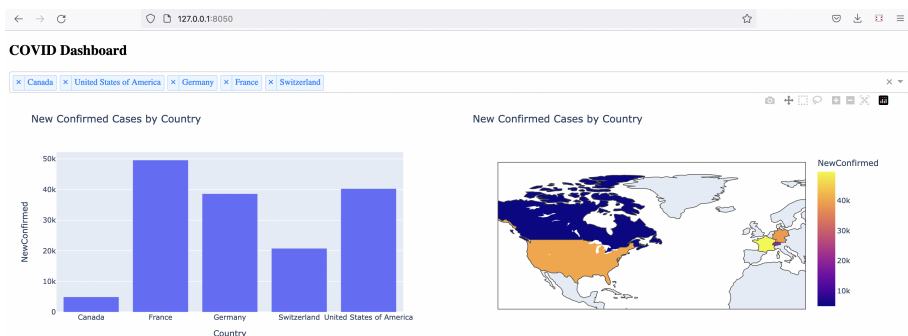


Figure 1: an example of dashboard resulting from the assignment

The assignment is presented in three parts, covering the three components of the application, and instructors could choose to use one, two, or all three parts.

Part 1: Web APIs: Part 1 introduces the Python `requests` library for making HTTP requests to Web APIs. We utilize a free COVID-19 API [3], which is easy-to-use and does not require any authentication. There are a number of different endpoints that developers can use, and the JSON data is returned as a Python object containing nested lists and dictionaries.

Part 2: Plotly Visualizations: Plotly [2] is an open-source Python library for creating interactive data visualizations. It includes all major types of charts (line, bar, scatter, etc.) with many features like mouse hover effects. Many kinds of charts recognize data in the list-of-dictionary-records format (making it perfect for the data from Part 1) and can be generated with one line of code. To maximize the usefulness of these visualizations, though, students will first get to do some processing of their data.

Part 3: Dash Application: Dash [1] is an open-source framework enabling rapid development of web applications. Dash applications sit on top of a Flask server and generates React front ends. Developers need not know any HTML, JavaScript, Flask, or React. Getting started is just as easy as other Python GUI frameworks like `tkinter`. Dash was developed by the Plotly company and by design works seamlessly with Plotly visualizations, so students can easily build a web UI around their work from Part 2. The assignment materials include an optional lab that can get students up-to-speed with Dash before using it in the assignment.

References

- [1] Dash python user guide. <https://dash.plotly.com/>.
- [2] Plotly open source graphing library for python. <https://plotly.com/python/>.
- [3] Kyle Redelinguys. Covid 19 api. <https://covid19api.com/>.