ASSIGNMENT COVERSHEET

PRAGUECITY UNIVERSITY

Student Name: Daniel Shutov	
Class: Programming fundamentals (FT/BL)	
Assignment: Analyze data, design and implement visualization	
Lecturer: Mohamed Bettaz	Semester: 2201
Due Date: 19 January 2023 - 23:59	Actual Submission Date: Submission date

Evidence Produced (List separate items)	**Location (Choose one)**	
	X	Uploaded to the Learning Center (Moodle)
		Submitted to reception
Note: Email submissions to the lecturer are not valid.		

Student Declaration:	
I declare that the work contained in this assignment was researched and prepared by me, except where acknowledgement of sources is made. I understand that the college can and will test any work submitted by me for plagiarism. **Note:** The attachment of this statement on any electronically submitted assignments will be deemed to have the same authority as a signed statement	
Date: March 3, 2023	Student Signature: Daniel Shutov

A separate feedback sheet will be returned to you after your work has been graded.
Refer to your Student Manual for the Appeals Procedure if you have concerns about the grading decision.

Student Comment (Optional)
Was the task clear? If not, how could it be improved?
Was there sufficient time to complete the task? If not, how much time should be allowed?
Did you need additional assistance with the assignment?
Was the lecturer able to help you?
Were there sufficient resources available?
How could the assignment be improved?

Analyze data, design and implement visualization

Daniel Shutov

March 3, 2023

Contents

List of Figures

Listings

1 List of acronyms

Python: Is an object-oriented, high-level, interpretable programming language that has dynamic semantics.

Jupyter notebook: Enables the creation and sharing of documents containing live code, mathematics, graphics, and narrative text. Data cleansing and transformation, numerical simulation, statistical modeling, and data visualization are among its applications.

Dash: Dash is an open-source data visualization interface development framework. Since its 2017 debut as a Python library. Dash enables data scientists to construct analytical online applications without significant web development expertise.

Plotly: Plotly is the name of a module that provides functions that can generate full figures simultaneously. Plotly is an integral component of the Plotly library and the preferred starting point for the creation of the majority of figures.

Pandas: As pandas.pydata.org (2022) stated, "Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python."

NumPy: As w3schools.com (2022) stated, "NumPy is a Python library used for working with arrays. It also has functions for working in the domain of linear algebra, Fourier transform, and matrices. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists."

Library: A programming library is a collection of pre-written code that may be used to optimize work by programmers. Typically, this library of reusable code targets specific common problems.

Function: As futurelearn.com (2019) said, "A function is simply a “chunk” of code that you can use over and over again, rather than writing it out multiple times. Functions enable programmers to break down or decompose a problem into smaller chunks, each of which performs a particular task."

Data purification: Data purification is the procedure of correcting or deleting incorrect, corrupted, improperly formatted, duplicate, or insufficient data from a dataset. When several data sources are combined, there are numerous chances for data duplication and mislabeling.

2 Introduction

This technical report will discuss the problem that was assigned in the ICA "Implementing and deploying a python web application using Dash library and data cleaning technics". The report must include explanation of: How the writer purified the data and why, proper application design with justification, specification and analysis. The assignment was completed only by using the prior knowledge and official documentation.

3 Problem analysis

During the COVID-19 epidemic, the entire world sought accurate data represented by simple but effective graphs that could display numbers as pictures. As stated in the assignment brief, "the student is a freelance data scientist that assists various organizations with their data visualization." it is clear that a reputable data source must be located so that the data may be represented with great precision and clarity. According to the author of this technical report, the solution is pretty straightforward; it uses the "Johns Hopkins Center for Science and Engineering" dirty database as a reputable source, clean it using different librarys like: NumPy, Pandas etc. and visualizes the data with the Dash library on a short web page. This page will be publically accessible via the Heroku platform, and it will be intended for anyone who wish to view the global situation in its entirety.

Figure 1: Covid-19 logo

4 From zero to hero

Before the issue may be resolved, it is important to take a number of measures. Installing the required tools for data cleansing, modification, rapid processing, and visualization is the first step. Second, data retrieval from the source. Third, cleansing the data, for instance by removing irrelevant or empty information .CSV cells and None data, with column names modified and groups joined as required. Fourth, while accessing the relevant fields or columns and altering the clean data for display, it is more important than ever to conduct tiny tests for each function individually. Fifth, testing the program and how each function interacts, ideally without any issues. The sixth step is platform deployment. These six actions are the most crucial for the start.

5 Libraries

To install the needed libraries, one must use the **CMD** (windows) to write the following PIP commands using shell scripting. After the installation, one could import the libraries to any .py file where they could be used fully.

```
pip install numpy
pip install pandas
pip install plotly
pip install dash
pip install requests
pip install gunicorn
```

6 Data purification

Following the assignment of databases to variables. One would desire to alter the column names with the **.rename()** method to more pleasant and intelligible ones or changing the data from string to integer using **.astype()**, such as "CoUnTrY/StaTE" to "country" etc., for future convenience. Next, one must utilize the **.drop()** and **.dropna()** methods for clearing, or to be more precise - eliminating the unneeded columns and None values located within the columns, such as:

```
import numpy as np
import pandas as pd

death_df = pd.read_csv("csse_covid_19_data.csv")
confirmed_df = pd.read_csv("time_series_covid19_confirmed_global.csv")
recovered_df = pd.read_csv("time_series_covid19_recovered_global.csv")
country_df = pd.read_csv("cases_country.csv")

confirmed_df = confirmed_df.rename(columns={'pro/sta': 'state', 'cou/reg': '
    country'})
confirmed_df = df.astype({"population": int})
country_df=country_df.drop(columns=['recovered', 'active', 'uid'])
confirmed_df=confirmed_df.dropna(subset=['long','lat'])
```

Listing 1: Python example

Then, one must delete any duplicates using the **.drop_duplicates()** method, whether they exist or not. Safer to be cautious than sorry. After removing all duplicates, it is recommended to sort and correctly index the clean database using **.sort_values()** and **.set_index()** methods. It is strongly suggested, as using a sorted database would yield more accurate and precise results for the visualization of the numbers and names. For instance, the author of this technical paper wished to illustrate the global scope of the pandemic; therefore, a database arranged from "the worst to the best cases" will be much more effective. In specific cases, one would use **.groupby()** method to group rows in the **.csv** database and the **.last()** method to check just the latest data only.

```
dfall.set_index(["location", "date"])
dfplot = (dfv.sort_values(["iso_code", "date"])
    .groupby("iso_code", as_index=False)
    .last()
    .sort_values("people_fully_vaccinated_per_hundred", ascending=False))
```

Listing 2: Python example

6.1 Justification of the purification

After these stages, the database will be at a "good starting point" prior to manipulation and visualization processing. A "good starting point" means that all the following steps are done:

- Data retrieval and storage in a variable for convenient use.
- Eliminating unnecessary columns and clearing empty or None cells for clean and accurate results.
- Changing the names and data variables so that they are more accessible and "user-friendly".
- Joining fields/columns from many databases into a single database for increased efficiency and usability.
- Sorting the database according to the specified value for easy data manipulation.
- Reindexing database tables because the sorting altered the database's order, necessitating a reindexing.

7 Visualization

As was mentioned in section number four, visualization is the next step, thus one should consider a couple of things:

- What graph one should use.
- How one would show the numbers in a more accessible and "user-friendly" way.
- Which fields one should display. (what's important and what not)
- What kind of representation each country one should have, and why.

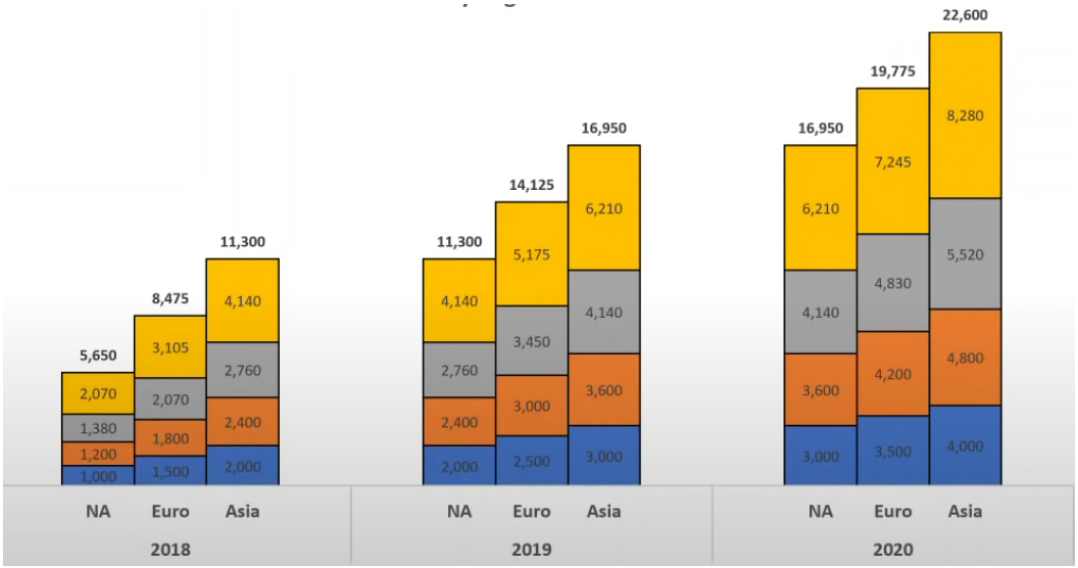
7.1 Graph types and their uses

One should choose carefully the type of the graph, because each graph has its own purpose.

7.1.1 Stacked bar chart

One can use stacked bar charts to visualize the composition of a total by category, or to display data split down to demonstrate the components of a whole. Typically, the various sections of the bar will be colored differently to represent their unique contributions to the bar's height.

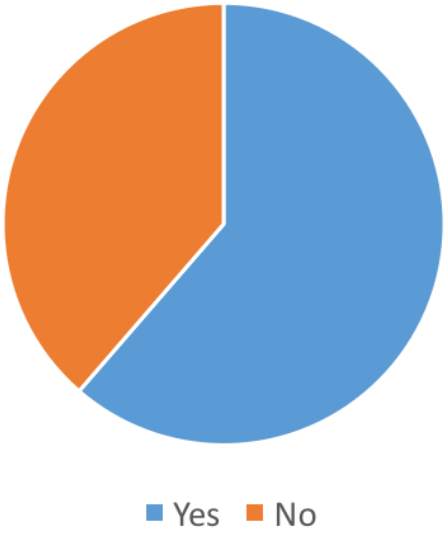
Figure 2: Stacked chart



7.1.2 Multiple KPI Pie chart

Multiple KPI Pie charts have the same capabilities as traditional pie graphs, with the added ability to select certain, frequently unrelated KPIs for their structure.

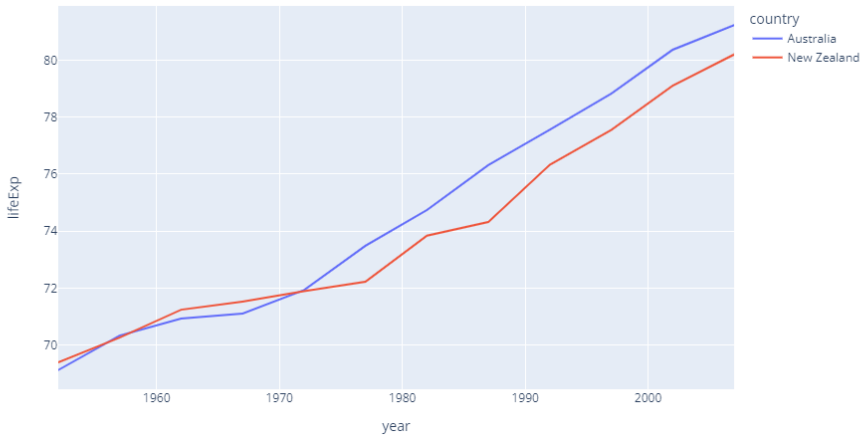
Figure 3: Pie chart



7.1.3 Line chart

A line graph displays the values or measurements of continuous data through time. They are an excellent option for tracking metrics over time, such as stock or share prices or in this technical paper - COVID-19 pandemic. Almost always, line graphs include time or dates on the x-axis and comparison plots on the chart.

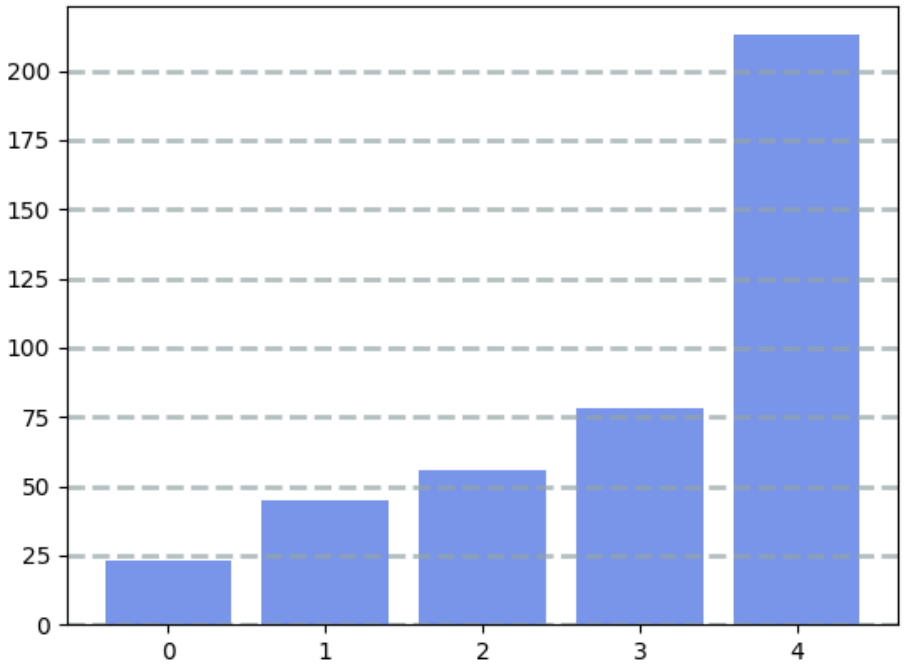
Figure 4: Line chart



7.1.4 Bar chart

A bar chart is one of the most prevalent chart types. They are useful for summarizing data that has been subset by categories. Typically, these categories would be represented by a sequence of horizontal bars whose totals correspond to the category values.

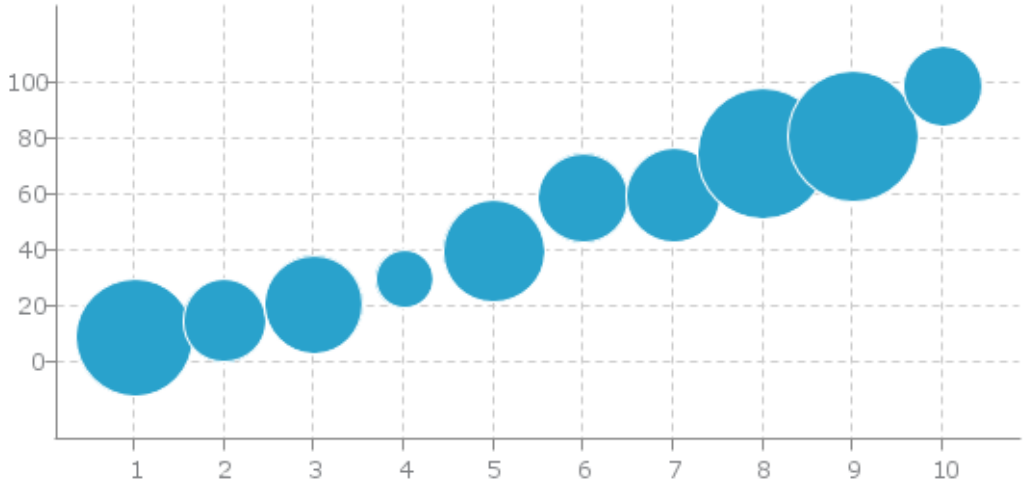
Figure 5: Bar chart



7.1.5 Bubble chart

Bubble charts, are utilized when data requires a third dimension to offer viewers with more comprehensive information. The purpose of a bubble plot is to compare three variables.

Figure 6: Bubble chart



7.2 Graph justification

The author of this technical article chose the following charts because: First, in the top ten representation, the **column chart** is the most convenient, and it is easy to distinguish between the pandemic-affected countries, for example: mortality rate, confirmed cases etc. Second, as stated previously, **bubble charts** can display information in three dimensions; therefore, the "60 worst hit countries (death)" were shown using a bubble chart since it is simpler to determine from the graph which circle is larger and higher and which country was affected. Thirdly, the **line chart** was chosen based on the above statement, "A line graph illustrates the values or measurements of continuous data over time." Therefore, it will be simpler to visualize the confirmed cases, fatalities, and recovered persons around the world. The final chart is a combination of a **stacked chart** and a **line chart**. The stacked chart was the best option for displaying which vaccine each county utilized and in what quantity, and the line chart further clarifies what percentage of the population is fully vaccinated. It is vital to note that the graphs are directly linked to internet databases, thus the information is always current. (assuming the source will continue to exist)

8 Dash

Before the testing stage, it is important to initialize a dash instance in the code .py file and to start the server that will run the whole main file. Ideally, it should be written at the beginning of the .py file.

```
app = dash.Dash(__name__)
server = app.server
```

Listing 3: Python example

9 Testing

The testing stages of any web application by Timotic (2018) should be:

```
Step 1: Functional Testing.
Step 2: Usability Testing.
Step 3: Interface Testing.
Step 4: Compatibility Testing.
Step 5: Performance Testing.
Step 6: Security Testing.
```

The author of this technical article elected to follow these steps precisely. First, it was determined if the application was retrieving the necessary data from GitHub, if it was manipulable in the selected IDE, and if the logic was functioning as expected. The second phase was to determine whether the website is accessible and interactive, allowing users to select/deselect nations, etc. Jupyter notebook made it simpler to see each step as one progressed through the code. After the deployment of the third stage, it was straightforward to determine whether the website is live, "user-friendly," and continues to function as intended. The fifth phase has been accomplished as a result of the new python update (3.11), which is 10 to 60 percent faster than its predecessor (3.10) as was stated by Salgado (2022). The author has no control over the fourth and sixth steps because the web application was pushed to Heroku. Heroku is responsible for ensuring that the servers are secure and operational.

10 Deployment

Unfortunately, Heroku only supports Python 3.7.9 or lower, resulting in a significant decrease in code speed. As a result, the author of this technical article decided to make the code as efficient as as, without any callback functions (it is well known that callback function can make your efficiency drop in large scales) or sophisticated visuals, so that the code could run faster while still displaying all the required data for the ICA. During the deployment phase, while producing the "requirements.txt" file for Heroku, a version difference was observed between the "Procfile.txt" file for hosting information. Actual data was retrieved from the master repository on GitHub. After upgrading Python from version 3.11 to version 3.7.9, the web application was successfully deployed.

Figure 7: Deploying to Heroku

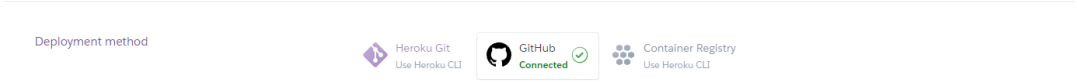


Figure 8: Deploying branch

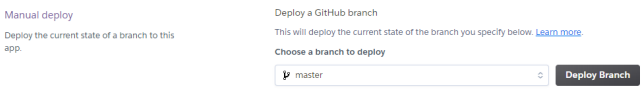


Figure 9: Heroku log

```
-----> Building on the Heroku-20 stack
-----> Python app detected
-----> No change in requirements detected, installing from cache
-----> Installing pip 20.1.1, setuptools 47.1.1 and wheel 0.34.2
-----> Installing SQLite3
-----> Installing requirements with pip
-----> Discovering process types
       Procfile declares types -> web
-----> Compressing...
       Done: 144.7M
-----> Launching...
       Released v5
```

11 Solution for problem

The visual representation can be seen the Dash application will be presented during the session of Friday 20/January/2023 or on the Heroku website or to run the .py file and to start it locally on "http://127.0.0.1:8050" and the code in the Jupyter notebook or in the .py file that came with this technical paper.

12 Suggested improvements

There are some example improvements that can be done:

- Creating a function that accepts a "dirty" database and returns a "clean" one, thereby increasing the efficiency of the cleaning process. Thus, it may be possible to pass only a list of databases and obtain clean ones relatively quickly.
- Utilizing more accurate and current databases.
- Different platform for deployment that will allow more functionality for the programmer.

13 List of figures

References

futurelearn.com ((2019)). “What is a function?” en. In: URL: https://www.futurelearn.com/info/courses/programming-102-think-like-a-computer-scientist/0/steps/53095.

pandas.pydata.org (May (2022)). “Package overview”. en. In: URL: %22https://pandas.pydata.org/docs/getting_started/overview.html.

Salgado, Pablo Galindo (June (2022)). “What’s New In Python 3.11”. en. In: URL: https://docs.python.org/3/whatsnew/3.11.html.

Timotic, Milos ((2018)). “Testing stages”. en. In: URL: tms-outsource.com.

w3schools.com (June (2022)). “NumPy Introduction”. en. In: URL: https://www.w3schools.com/python/numpy/numpy_intro.asp.