**UNIVERSITY OF SCIENCE**

**FACULTY OF INFORMATION TECHNOLOGY**

**A blue and white logo

Description automatically generated**

**MULTIVARIATE**

**STATISTICAL ANALYSIS**

**Report Practice 02  
Matplotlib**

**Lecturers**  Lý Quốc Ngọc

Nguyễn Mạnh Hùng

Phạm Thanh Tùng

**Student** Võ Nguyễn Hoàng Kim

21127090

**CONTENT**

[A. SELF-ASSESSMENT FORM 3](#_Toc161681166)

[B. IMPLEMENTATION 3](#_Toc161681167)

[I. Dataset 3](#_Toc161681168)

[II. Visualization data 4](#_Toc161681169)

[**1.** **Line plot** 4](#_Toc161681170)

[**2.** **Bar plot** 5](#_Toc161681171)

[**3.** **Pie plot** 6](#_Toc161681172)

[C. BONUS 8](#_Toc161681173)

[I. DATASET 8](#_Toc161681174)

[II. VISUALIZATION AND ANALYSIS 8](#_Toc161681175)

[**1.** **Sample 1** 8](#_Toc161681176)

[**2.** **Sample 2** 8](#_Toc161681177)

[III. OTHER VISUALIZATION LIBRARIES 9](#_Toc161681178)

[D. REFFERENCES 9](#_Toc161681179)

# **SELF-ASSESSMENT FORM**

|  |  |  |
| --- | --- | --- |
| **Features** | **Note** | **Level of completion** |
| Read data | From the given Covid-19 cases CSV file | 100% |
| Visualization data | With 3 different graph using Matplotlip (line, bar and pie) | 100% |
| Comments, analytic | Do on corresponding graphs | 100% |
|  | | |
| **Bonus** | **Note** | **Level of completion** |
| Read other data CSV files | From the given Covid-19 deaths CSV file | 100% |
| Visualization data Comments, analytic | With 2 different graph using Matplotlip (line, bar and pie) Do on corresponding graphs | 100% |
| Comments, analytic | With 2 different libraries using Pandas plot and Seaborn | 100% |

# **IMPLEMENTATION**

Note: In this program, we use 2 main libraries:

* **Matplotlib.pyplot** (call as **plt**) for visualizing data in graphs [1].
* **Pandas** (call as **pd**) for data processing and refinement [2].

## **Dataset**

1. **Read data**

* Dataset: The given Covid-19 cases CSV file.
* Read data: To read data from the file, we use the **pd.read\_csv()** function with the parameter being the path to the data file. Store the data that has just been read into a variable (call as *df*) in the program (this variable will be used for processing, editing, etc., the data for visualization later).
* The value of *df* after we reading the data file as below (The following figure depicts just a portion of the data table stored in the variable *df*):

A screenshot of a computer screen

Description automatically generated

1. **Data group**

* To save time with implementing some visualizations below, we first group the data by country/region. This is done using the pd.groupby() function: the data in original dataframe *df* is grouped by the countries in the "Country/Region" column.
* Then, it sum over different states/province of each country. The final result is stored in *df\_countries* variable (a part of its value as below):

A screenshot of a computer

Description automatically generated

## **Visualization data**

*Note:* In this program, we use several helper functions from the Matplotlib.pyplot library to set parameters and characteristics for the graph, such as:

* **plt.xlabel()**: Set the label for the X-axis.
* **plt.ylabel()**: Set the label for the Y-axis.
* **plt.title()**: Set the title of the chart.
* **plt.show()**: Display the graph.

### **Line plot**

1. **Purpose**

* Observing the data, we realize that it is aimed at recording the number of Covid-19 infections by each day in each country. The recording period spans from January 22, 2020, to July 14, 2021 (a total of 540 days).
* With the provided data, we can construct a Line graph to observe the trend of the pandemic situation over the statistical period for a particular country.

1. **Implement**

This implementation utilizes the data frame previously processed, which is grouped by country/region.

To make this plot, we initialize a function that is called **visualizeCasesInCountry()** with one parameter is *country* to specify the country (or region) that we want to plot its data.

* Firstly, we extract the data (columns recording the number of infections per day from 01/22/2020 to 07/14/2021) of the corresponding country, resulting in a new data frame named *country\_df*.
* By calculating the length of *country\_df*, we can determine the number of columns it has, which is equivalent to the number of days during the statistical period (Represent for X axis)
* Additionally, country\_df.values provides the corresponding number of infections for each day (representing the Y-axis).
* With the data for the X-axis and Y-axis already available, we use the **plt.plot()** function to create a line graph. Additionally, we can use other functions in the *Note* section to set various characteristics of the graph.

1. **Result**

A graph with a line

Description automatically generated

1. **Analysis**

* As we know, China is where the outbreak of Covid-19 was first detected, and the first cases were recorded.
* Based on the chart, this country recorded a very high number of infections and experienced continuous and robust growth during the initial period of the statistical observation, reaching a milestone of 80,000 cases within the first 100 days.
* However, during the remaining period, although the number of infections in China continued to increase, there was a slowdown. Evidence of this is that within the around subsequent 400 days, the number of infections they recorded was only higher by 20,000 compared to the initial period.
* With this chart, we can track the rate of development and spread of the disease daily.

### **Bar plot**

1. **Purpose**

* Observing the data, we realize that it is aimed at recording the number of Covid-19 infections by each day in each country.
* As a result, we can statistic and visualize the top 5 countries with the highest number of infections on a specific day by using a bar graph.

1. **Implement**

This implementation utilizes the data frame previously processed, which is grouped by country/ region.

To make this plot, we initialize a function that is called **visualizeTopCountryInDay()** with one parameter is *date* to specify the date that we want to plot its data.

* First, we sort the *df\_countries* data frame in descending order based on the values of the *date* column, using the **sort\_values()** function.
* From the sorted result, we extract the first 5 countries in the table (corresponding to the top 5 countries with the highest number of Covid infections) using the **head()** function. Record the data of these 5 countries in the variable *top5\_df*.
* What we want from the graph is to represent the top countries with the highest number of infections on a specific day. Therefore, the X-axis will represent each country, and the Y-axis will represent the number of cases recorded for that country:
  + For X axis, we use *top5\_df.index* to represent countries
  + For Y axis, we use *top5\_df[date]* to represent the corresponding number of Covid-19 cases of each country.
  + In this plot, we use a **plt.text()** function to make clearly the number of Covid-19 cases of each country.
* Using **plt.bar()** function to plot a bar graph that perform top countries or regions has the highest Covid-19 cases in the *date*.

1. **Result**

A graph of covid-19 cases

Description automatically generated

1. **Analysis**

* At the time of selecting to draw this bar graph, the order of countries by the number of Covid-19 infections is as follows: US, India, Brazil, France, and Russia, ranked in descending order.
* The number of Covid-19 infections recorded in the US is 33,947,230 cases, significantly surpassing the other countries, nearly 5.8 times the number of cases recorded in Russia.
* Thanks to the above graph, we can clearly grasp the alarming situation of the US at that time.

### **Pie plot**

1. **Purpose**

* With the initial data, the statistics include information on the number of infections in the states/provinces of each country.
* Therefore, the pie chart is utilized for the purpose of analyzing and statistic the situation of the pandemic occurring within the states or provinces of a country.

1. **Implement**

This implementation utilizes the original data frame is *df*.

To make this plot, we initialize a function that is called **visualizeCaseInStatesOfCountry()** with two parameters are country and threshold to specify the country that we want to plot its data and a threshold is a filtering parameter for the number of infections used to evaluate the "contribution" of the states/provinces in the graph.

*Note*: For the purpose stated above, this chart is only drawn when a country has recorded data for its states/provinces in the dataset. Therefore, if an inappropriate country is intentionally passed as an argument, the function will return an "Invalid" message, and no graph will be drawn.

* Firsly, we extract the data of the specified country from the original dataframe *df*, resulting in a new dataframe *df\_subset*.
* Check the validity of the function: verify if the input country has data of states/provinces recorded in the dataset (by checking the length of *df\_subset*).
* Reset the index for *df\_subset* using the **set\_index()** function, with the new index value set as "Province/State".
  + Setting the index for *df\_subset* to ensure that the names of states/provinces are retained for further steps.
* Calculate the total number of Covid-19 infections for each province/state. Use*sum\_rows* variable to store the corresponding results for each province/state.
  + In this step, use the **iloc[]** function to access the necessary positions and calculate the sum using the **sum()** function with *axis = 1* to sum the number of cases during the statistical period (sum by horizontal direction).
* Use a threshold to filter the values:
  + *greater\_cases*: cases where the province/state has a number of Covid-19 infections greater than or equal to the *threshold*.
  + *small\_cases*: cases where the province/state has a number of Covid-19 infections less than the *threshold*.
  + For elements belonging to *small\_cases*, combine their data into a new component named "Other" (the label used for a component in the graph), and store it as a new data frame *sum\_small\_cases* (using the **pd.Series()** function).
* Combine the components in *greater\_cases* and *sum\_small\_cases* using the **pd.concat()** function to obtain a new dataframe *df\_final*.
* With data in *df\_final*, use the **plt.pie()** function to represent the percentage of infections in each province/state of a country.
  + In this pie plot, we use the **plt.legend()** function to add a legend about the labels in the graph.

1. **Result**

A green circle with red and blue text

Description automatically generated

1. **Analysis**

* During the statistical period, in China, the number of Covid-19 infections in Hubei province accounted for approximately ¾ of the total number of infections recorded in China.
* Meanwhile, other provinces, such as Hong Kong and Guangdong, accounted for a very small proportion, less than 10% of the total number of infections in the country.
* The remaining portion comprises other provinces and states, totaling approximately 17%.
* With this chart, a country can be analyzed to understand the complex situation of the pandemic and its most affected provinces/states.
* If more detail is desired, one can simply decrease the *threshold* value.

# **BONUS**

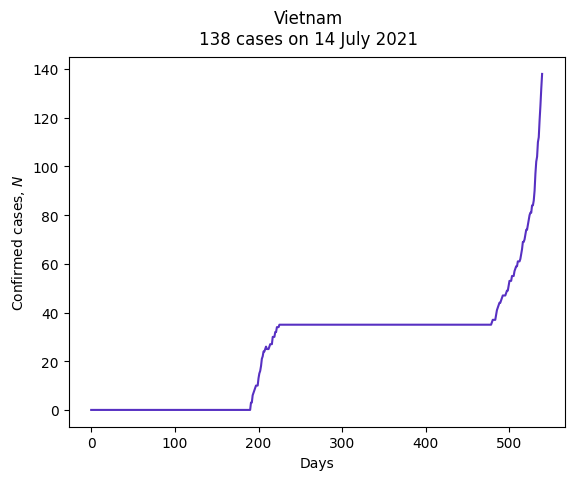
## **DATASET**

* In this section, we use data from the Covid-19 deaths CSV file. We use the pd.read\_csv() function to read the data from the file.

## **VISUALIZATION AND ANALYSIS**

### **Sample 1**

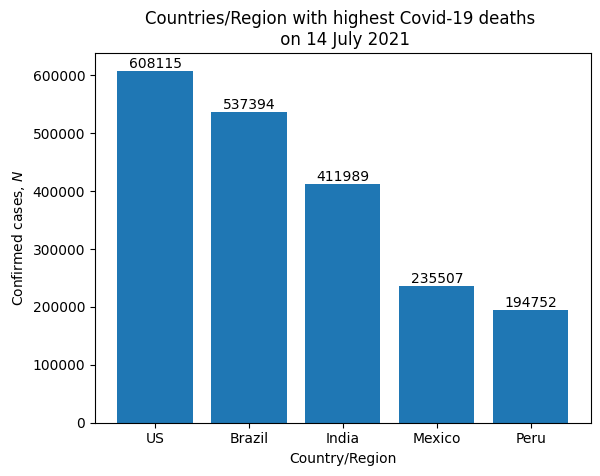
* Purpose
  + Observing the data, we realize that it is aimed at the number of people who died from Covid-19 recorded daily in each country. The recording period spans from January 22, 2020, to July 14, 2021 (a total of 540 days).
  + With the provided data, we can construct a Line graph to observe the trend of the pandemic situation over the statistical period for a particular country.
* Visualization



* Comments, analytic
  + The chart clearly depicts the number of deaths due to Covid-19 in Vietnam during the period from January 22, 2020, to July 14, 2021.
  + In the first 200 days of the statistics, Vietnam did not have any reported deaths.
  + However, towards the end of the statistical period, the number of deaths due to Covid-19 increased rapidly, reaching around 150 cases, with no signs of slowing down.

### **Sample 2**

* Purpose
  + Observing the data, we realize that it is aimed at recording the number of people who died from Covid-19 by each day in each country.
  + As a result, we can statistic and visualize the top 5 countries with the highest number of deaths on a specific day by using a bar graph
* Visualization



* Comments, analytic
  + At the time of selecting to draw this bar graph, the order of countries by the number of people died from Covid-19 is as follows: US, Brazil, India, Mexico, and Peru, ranked in descending order.
  + The number of deaths by Covid-19 recorded in the US is 608115 cases, significantly surpassing the other countries, nearly 3 times the recorded number of Peru.

## **OTHER VISUALIZATION LIBRARIES**

* In addition to Matplotlib, we can use other libraries such as Pandas or Seaborn to visualize data. Below are the graphs created by these libraries using the same dataset and interpretation.

|  |  |
| --- | --- |
|  |  |

* Even though they yield similar results, the syntax and methods to visualize data on the dataset of each library are different. Therefore, there is a need for being carefull in adjusting the data for visualization.

# **REFFERENCES**

|  |  |
| --- | --- |
| [1] | The Matplotlib development team., "matplotlib.pyplot," Matplotlib, 2012-2024. [Online]. Available: https://matplotlib.org/stable/api/\_as\_gen/matplotlib.pyplot.plot.html. |
| [2] | pandas, "User Guide," pandas, 2024. [Online]. Available: https://pandas.pydata.org/docs/user\_guide/index.html. |