**PRACTICE #02**

**Matplotlib**

# Thông tin cá nhân:

**MSSV**: 19127614

**Họ** **Tên**: Nguyễn Anh Tuấn

# Trực quan hóa dữ liệu Covid19 ở Việt Nam

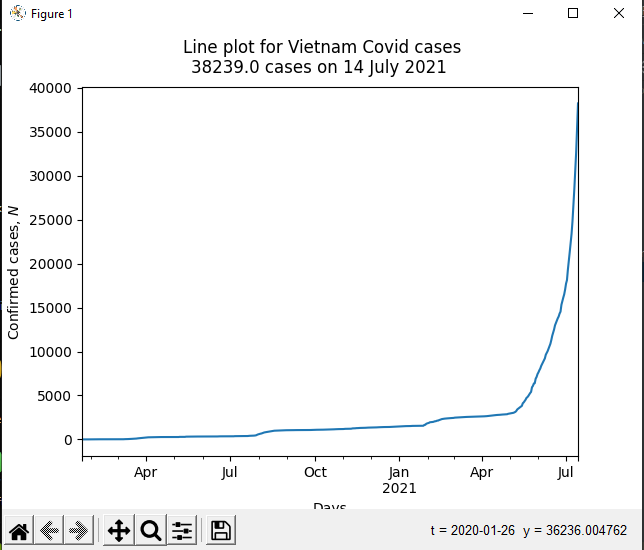
Phần này sử dụng 2 thư viện chính là:

Pandas: dùng để truy xuất, hiệu chỉnh và thống kê dữ liệu từ file CSV

Matplotlib: Dùng để vẽ biểu đồ, trực quan hóa dữ liệu

Data sẽ được lấy từ file: covid-19-cases.csv

## Line Plot (biểu đồ đường)



**Nhận xét:**

Do dữ liệu covid qua hàng ngày là liên tục, nên việc sử dụng biểu đồ đường theo em là một lựa chọn rất tốt. Do dữ liệu nhiều nên đã chia hàng X theo thành tháng.

Qua biểu đồ, có thể thấy được được số lượng các ca covid ở Việt Nam tăng theo từng ngày – tháng, và đỉnh điểm là từ tháng 4 đến tháng 7 tăng cao kỷ lục.

Ưu điểm:

* Hữu ích để thể hiện dữ liệu liên tục, chẳng hạn như thay đổi theo thời gian
* Có thể dùng để so sánh 2 hay nhiều thực thể
* Sử dụng biểu đồ đường để so sánh một số chuỗi dữ liệu hoặc cho các giá trị vô hình riêng lẻ như tỷ lệ.

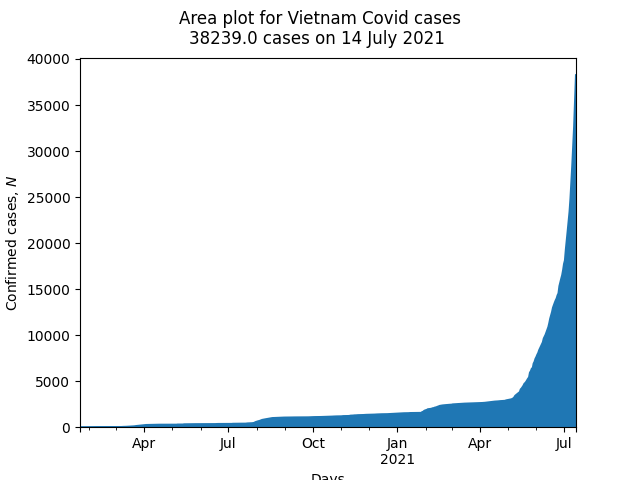
Khuyết điểm:

* Nếu dữ liệu quá nhiều, rất khó để so sánh chính xác các giá trị tại thời điểm nhất định.
* Nếu có các giá trị giống nhau sẽ khiến biểu đồ trở nên khó khăn hơn trong việc so sánh.

**Source Code:**

def make\_line\_plot(country):  
 #sort data from country  
 c\_df = df\_countries.loc[country, df\_countries.columns[3:]]  
 # convert index to a proper datetime object  
 c\_df.index = pd.to\_datetime(c\_df.index)  
 #make Line plot  
 line\_plt = c\_df.plot(kind="line")  
 #Set name for x and y label  
 plt.xlabel('Days')  
 plt.ylabel('Confirmed cases, $N$')  
 # add a title reporting the latest number of cases available  
 title = 'Line plot for {} Covid cases' \  
 '\n{} cases on {} '.format(country, c\_df[-1],  
 c\_df.index[-1].strftime('%d %B %Y'))  
 plt.suptitle(title)  
 plt.show()

## Area Plot (biểu đồ vùng)



**Nhận xét:**

Sử dụng biểu đồ đường sẽ hửu ích hơn nếu có 2 hoặc nhiều hơn các thực thể cần so sánh, ở đây ta chỉ có mỗi 1 thực thể là Việt Nam nên nó trông không khác gì lắm biểu đồ đường.

Qua biểu đồ, có thể thấy được được số lượng các ca covid ở Việt Nam tăng theo từng ngày – tháng, và đỉnh điểm là từ tháng 4 đến tháng 7 tăng cao kỷ lục.

Ưu điểm:

* Hữu ích để so sánh hai hoặc nhiều đại lượng, thực thể.
* Dữ liệu đơn giản được trực quan và dễ hiểu
* Hoạt động tốt với dữ liệu liên tiếp theo thời gian
* Dùng biểu đồ vùng cho nhiều chuỗi dữ liệu có mối quan hệ một phần với toàn bộ, cho chuỗi dữ liệu riêng lẻ đại diện cho một tập hợp các dữ liệu có thể đếm được hoặc chuỗi giá trị tích lũy.

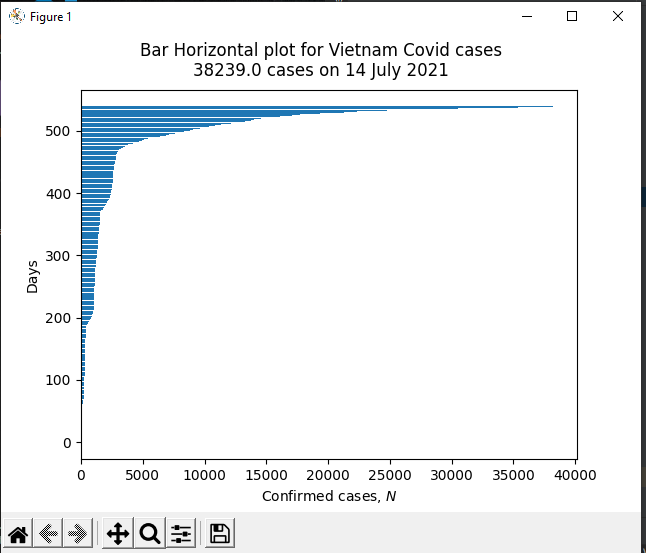
Khuyết điểm:

* Do các dữ liệu rất gần kề nhau theo mặt thời gian, nhưng biểu đồ lại có giới hạn, nên rất khó để so sánh chính xác các dữ liệu tại một thời điểm thích hợp
* Khi dữ liệu quá lớn, hoặc có quá nhiều thực thể, các dữ liệu sẽ bị chồng chéo, gây khó cho việc so sánh

**Source Code:**

def make\_area\_plot(country):  
 # sort data from country  
 c\_df = df\_countries.loc[country, df\_countries.columns[3:]]  
 # convert index to a proper datetime object  
 c\_df.index = pd.to\_datetime(c\_df.index)  
 #make Area plot  
 area\_plt = c\_df.plot(kind="area")  
 # Set name for x and y label  
 plt.xlabel('Days')  
 plt.ylabel('Confirmed cases, $N$')  
 # add a title reporting the latest number of cases available  
 title = 'Area plot for {} Covid cases' \  
 '\n{} cases on {} '.format(country, c\_df[-1],  
 c\_df.index[-1].strftime('%d %B %Y'))  
 plt.suptitle(title)  
 plt.show()

## Bar Horizontal Plot (biểu đồ cột ngang)



**Nhận xét:**

Sử dụng biểu đồ cột ngang có một bất lợi trong việc trực quan hóa các dữ liệu liên tiếp, đó là việc các dữ liệu nhiều và liên tiếp về mặt thời gian sẽ khiến cho biểu đồ bị chồng chéo các cột lên nhau, rất khó coi. Tuy nhiên nếu để các cột giá trị thời gian quá ít (ví dụ như theo tháng ở 2 biểu đồ trên) sẽ khiến dữ liệu các cột bị thưa, không chính xác. Thế nên ở biểu đồ này sẽ chia số ca covid theo từng ngày từ ngày 0 đến ngày thứ 500 chứ không theo tháng như ở trên.

Qua biểu đồ, có thể thấy được được số lượng các ca covid ở Việt Nam tăng theo từng ngày, bắt đầu tăng mạnh từ khoảng ngày 400 và khoảng tăng đỉnh điểm vào sau ngày thứ 500

Ưu điểm:

* Sẽ có khoảng không để đánh nhãn dữ liệu có tên dài, không khiến các nhãn va chạm, chồng chéo nhau.
* Giúp ích trong việc nghiên cứu các mẫu trong một thời gian dài.
* Dễ hiển thị nhiều danh mục.
* Hiển thị dữ liệu âm một cách trực quan hơn 2 biểu đồ trên.
* Rất dễ nhìn, hiểu.
* Hoạt động rất tốt với các dữ liệu rời rạc

Khuyết điểm:

* Khó thể hiện được các mối quan hệ của dữ liệu
* Chỉ hoạt động tốt với các dữ liệu rời rạc.
* Do dữ liệu là rời rạc nên khó để đánh giá tác động của các sự kiện bất ngờ lên đối tượng dữ liệu.

**Source Code:**

def make\_barh\_plot(country):  
 # sort data from country  
 c\_df = df\_countries.loc[country, df\_countries.columns[3:]]  
 # convert index to a proper datetime object  
 c\_df.index = pd.to\_datetime(c\_df.index)  
 # get the number of column  
 n = len(c\_df)  
 # setting bar with corresponding values  
 plt.barh(range(n), c\_df.values)  
 # setting x and y label  
 plt.ylabel('Days')  
 plt.xlabel('Confirmed cases, $N$')  
 # add a title reporting the latest number of cases available  
 title = 'Bar Horizontal plot for {} Covid cases' \  
 '\n{} cases on {}'.format(country, c\_df[-1],  
 c\_df.index[-1].strftime('%d %B %Y'))  
 plt.suptitle(title)  
 plt.show()

# Phần Bonus

## Tập dữ liệu và các thư viện

* **Các thư viện**:

Phần này sử dụng 2 thư viện chính là:

Pandas: dùng để truy xuất, hiệu chỉnh và thống kê dữ liệu từ file CSV

Matplotlib: Dùng để vẽ biểu đồ, trực quan hóa dữ liệu

Seaborn: Dùng để điều chỉnh biểu đồ cho đẹp hơn

* **Tập dữ liệu:**

**Tên** : Countries Population from 1995 to 2020

**Link**: <https://www.kaggle.com/datasets/themlphdstudent/countries-population-from-1955-to-2020>

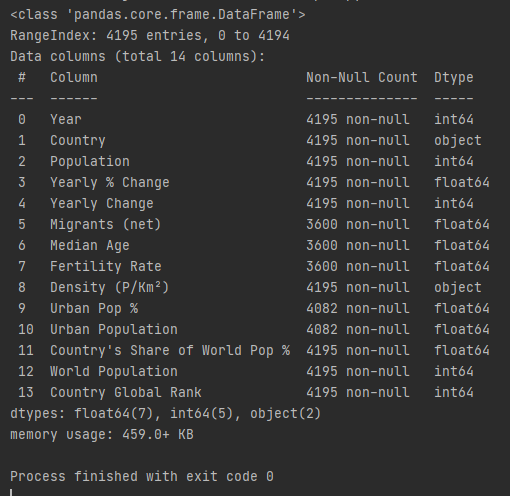
**Miêu Tả**: Là tập dữ liệu về dân số của 235 quốc gia trên thế giới từ năm 1995 đến 2020 với tổng cộng 4195 mẫu và 14 loại dữ liệu.

**Các dữ liệu quan trọng**:

* Year: Năm
* Country: Tên quốc gia)
* Population: Dân số
* WorldPolulation: Tổng dân số thế giới
* Median Age: Độ tuổi trung bình

…

Tổng quan về tập dữ liệu được xuất bằng hàm data.info() của thư viện Pandas



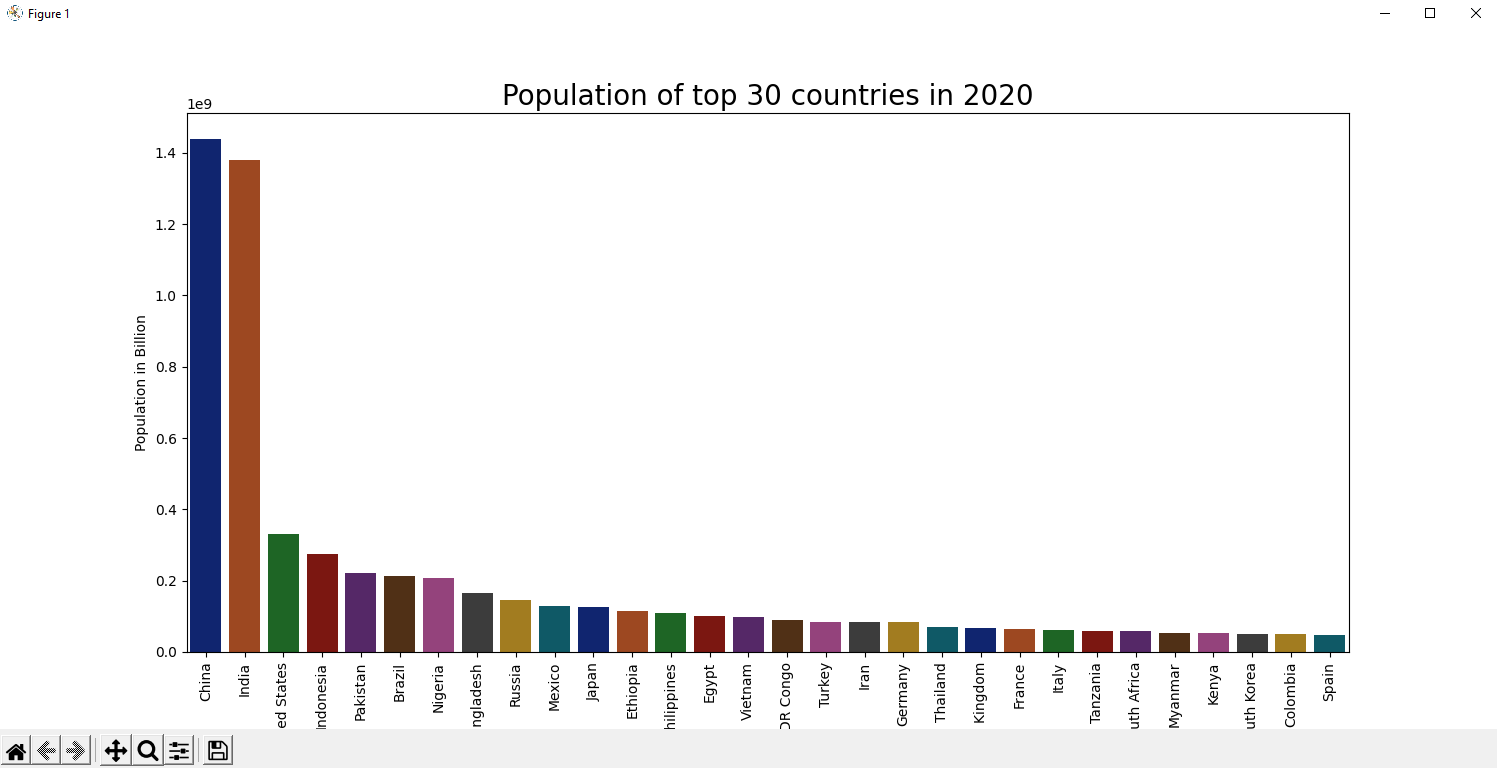
## Trực quan hóa và nhận xét

* **Top 30 quốc gia có dân số đông nhất:**

Source Code

def top\_30\_bar\_plot(year):  
 # year = year you want to show  
 # Get data top 30 population data of year  
 current\_population = population[population['Year'] == year][:30]  
 # Fig size  
 plt.rcParams['figure.figsize'] = (15, 7)  
 ax = sns.barplot(x=current\_population['Country'][:30], y=current\_population['Population'][:30], palette='dark')  
 #set x,y label  
 ax.set\_xlabel(xlabel='Countries', fontsize=10)  
 ax.set\_ylabel(ylabel='Population in Billion', fontsize=10)  
 #set tittle  
 title = 'Population of top 30 countries in ' + str(year)  
 ax.set\_title(label=title, fontsize=20)  
 plt.xticks(rotation=90)  
 plt.show()

**Biểu đồ**: Biểu đồ cột



**Nhận xét:**

Biểu đồ biểu hiện khách quan giá trị dân số của từng quốc gia theo năm. Do là dữ liệu rời rạc nên biểu đồ cột làm rất tốt việc trực quan hóa.

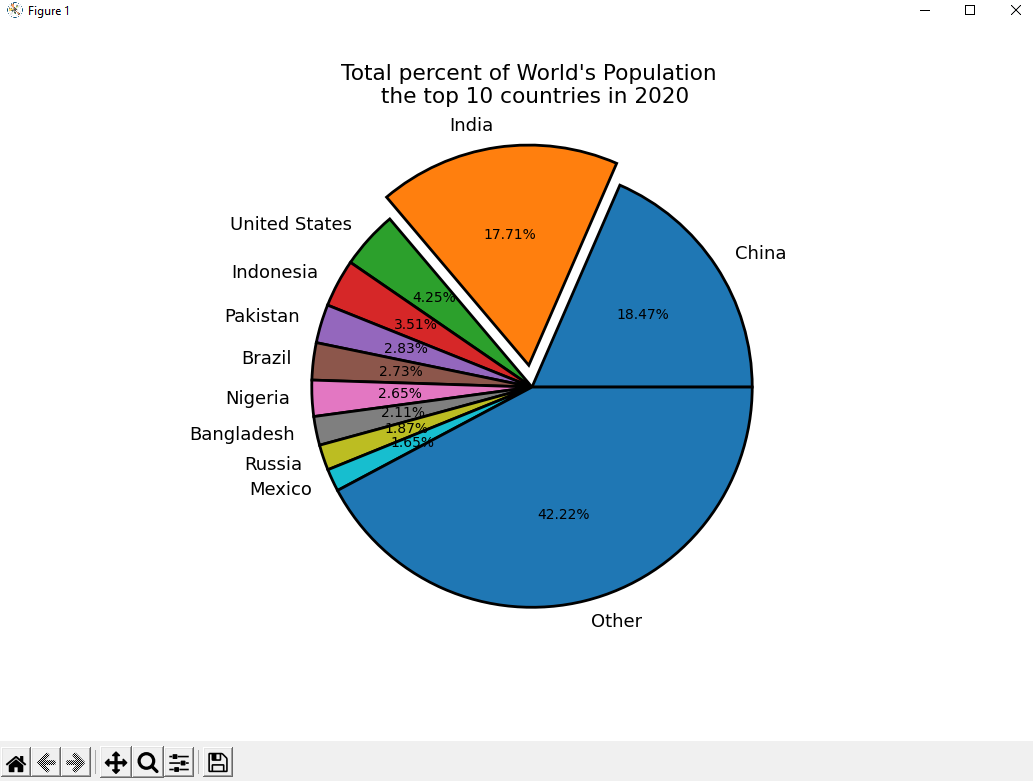
Qua biểu đồ, ta thấy được hai nước là Trung Quốc và Ấn độ có dân số lớn nhất và nhì, đồng thời hơn hẳn các nước từ top 3 trở xuống.

* **Thị phần dân số trên cả thế giới:**

**Source Code:**

def top\_share\_pie\_plot(year,top):  
 # year = year you want to show  
 # top = top n country you want to show  
 # get all country  
 unique\_countries = population['Country'].unique()  
 #change plot type  
 plt.style.use("seaborn-talk")  
 # set year  
 df\_last\_year = population[population['Year'] == year]  
 series\_last\_year = df\_last\_year.groupby('Country')['Population'].sum().sort\_values(ascending=False)  
 labels = []  
 values = []  
 country\_count = top # top country  
 other\_total = 0  
 #count the top number, all countries below top 10 will add to the "other"  
 for country in series\_last\_year.index:  
 if country\_count > 0:  
 labels.append(country)  
 values.append(series\_last\_year[country])  
 country\_count -= 1  
 else:  
 other\_total += series\_last\_year[country]  
 labels.append("Other")  
 values.append(other\_total)  
  
 #Edit pie plot  
 wedge\_dict = {  
 'edgecolor': 'black',  
 'linewidth': 2  
 }  
  
  
 explode = (0, 0.1, 0, 0, 0, 0, 0, 0, 0, 0, 0)  
 plt.title(f"Total percent of World's Population \n the top 10 countries in {year}")  
 plt.pie(values, labels=labels, explode=explode, autopct='%1.2f%%', wedgeprops=wedge\_dict)  
 plt.show()

**Biểu đồ:** Biểu đồ tròn (biểu đồ bánh)

****

**Nhận xét:**

Biểu đồ bánh làm rất tốt trong việc biểu thị dữ liệu dưới dạng có quan hệ với nhau, đặc biệt là các dữ liệu dưới dạng %.

Qua biểu đồ ta thấy được Trung Quốc và Ấn Độ vẫn là các nước có tỉ lệ dân số động nhất, đồng thời chỉ với top 10 nước đã chiếm hơn 50% tổng dân số thế giới.

## Các thư viện khác

**Thư viện**:

Bokeh

**Giới thiệu:**

Bokeh là một thư viện Python để tạo hình ảnh trực quan tương tác cho các trình duyệt web hiện đại. Nó giúp bạn xây dựng đồ họa đẹp mắt, từ các ô đơn giản đến các bảng điều khiển phức tạp với bộ dữ liệu trực tuyến. Với Bokeh, bạn có thể tạo hình ảnh trực quan hỗ trợ JavaScript mà không cần tự viết bất kỳ JavaScript nào. Đồng thời cũng là một thư viện tốt để trực quan hóa dữ liệu

**Source Code:**

from bokeh.io import show  
from bokeh.models import ColumnDataSource  
from bokeh.palettes import Spectral6  
from bokeh.plotting import figure  
from bokeh.transform import factor\_cmap  
  
#Data   
fruits = ['Apples', 'Pears', 'Nectarines', 'Plums', 'Grapes', 'Strawberries']  
counts = [5, 3, 4, 2, 4, 6]  
  
#Bokeh  
source = ColumnDataSource(data=dict(fruits=fruits, counts=counts))  
#create plot  
p = figure(x\_range=fruits, height=350, toolbar\_location=None, title="Fruit Counts")  
p.vbar(x='fruits', top='counts', width=0.9, source=source, legend\_field="fruits",  
 line\_color='white', fill\_color=factor\_cmap('fruits', palette=Spectral6, factors=fruits))  
  
#Edit Fig  
p.xgrid.grid\_line\_color = None  
p.y\_range.start = 0  
p.y\_range.end = 9  
p.legend.orientation = "horizontal"  
p.legend.location = "top\_center"  
  
show(p)

**Kết Quả:**

