ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

TRỰC QUAN HÓA DỮ LIỆU

GVLT: BÙI TIẾN LÊN
GVTH: LÊ NGỌC THÀNH

BÁO CÁO LAB 01

HOÀNG PHƯỚC NGUYÊN - 20127258
MAI QUÝ TRUNG - 20127370
NGUYỄN ĐỨC BẢO - 20127005
TRẦN NHẬT TRƯỜNG - 20127376
NGUYỄN NGỌC BẢO TRÂM - 20127084

Ngày 12 tháng 3 năm 2023

Mục lục

BÁO CÁO LAB 01

1. Báo cáo tiến độ:

1.1 Thông tin thành viên và mức độ hoàn thành:

MSSV	Họ và tên	Mức độ hoàn thành
20127258	Hoàng Phước Nguyên	100%
20127370	Mai Quý Trung	100%
20127005	Nguyễn Đức Bảo	100%
20127376	Trần Nhật Trường	100%
20127084	Nguyễn Ngọc Bảo Trâm	100%

Công việc:

- 20127258: Tiền xử lý dữ liệu, toàn bộ 2.3.1, tổng hợp báo cáo.
- 20127370: Thu thập dữ liệu, 2.3.3 - biểu đồ 2, 3, 2.3.5 - biểu đồ 3, toàn bộ 2.3.6.
- 20127376: Thu thập dữ liệu, toàn bộ 2.3.2, kiểm tra thuật toán/mối quan hệ.
- 20127005: Phân tích dữ liệu, 2.3.3 - biểu đồ 1, 2.3.4.
- 20127085: Tiền xử lý dữ liệu 2.3.5 - biểu đồ 1, 2.

1.2 Mức độ hoàn thành tổng thể:

Yêu cầu	Mức độ hoàn thành
Thu thập số liệu	100%
Tiền xử lý dữ liệu	100%
Trực quan biểu đồ	100%
Xét quan hệ các trường dữ liệu	100%
Tổng:	100%

2. Báo cáo thuật toán:

2.1 Pha thu thập dữ liệu:

Bước 1: Gọi những thư viện cần thiết để thực hiện việc thu thập dữ liệu từ trang Worldometer, bao gồm:

- `request`: Gọi api đến trang web.
- `BeautifulSoup`: Hỗ trợ crawl dữ liệu từ các thẻ trong file html của trang web.
- `pandas`: Định hình bảng dữ liệu và lưu vào file csv.

```python
import pandas as pd
import requests
from bs4 import BeautifulSoup
```

Hình 1: Import thư viện cần thiết

Bước 2: Gọi API và lưu dữ liệu thô vào BeautifulSoup:

- Thực hiện gọi API bằng đường link tới trang thống kê dữ liệu và lưu vào BeautifulSoup với định dạng `html`.

```python
url = "https://www.worldometers.info/coronavirus/"

corona_content = requests.get(url, "html.parser")
soup = BeautifulSoup(corona_content.content)
```

Hình 2: Gọi API và lưu vào BeautifulSoup

- Tìm thẻ `table` với id là `main_table_countries_today` để lấy toàn bộ thẻ html trong bảng dữ liệu.

```python
table = soup.find("table", id="main_table_countries_today")
```

Hình 3: Tìm thẻ `table` với id `main_table_countries_today`

Bước 3: Trích xuất các hàng, cột và lưu dữ liệu thống kê:

- Trong quá trình lấy dữ liệu có một số cột bị lỗi (sai về định dạng chữ) làm cho không đúng theo format dữ liệu mong muốn khi đưa vào file csv. Để khắc phục, nhóm đã tạo 1 dictionary với key là thuộc tính, value là mảng chứa dữ liệu.
- Lấy dữ liệu các cột trong bảng dữ liệu bằng cách tìm các thẻ <tr> trong <thead>.
- Tiếp tục thực hiện lấy nội dung từ thẻ <tr> trong thẻ <tbody>.

```python
data = {
    "Country": [],
    "Total Cases": [],
    "New Cases": [],
    "Total Deaths": [],
    "New Deaths": [],
    "Total Recovered": [],
    "New Recovered": [],
    "Active Cases": [],
    "Serious, Critical": [],
    "Tot Cases/1M pop": [],
    "Deaths/1M pop": [],
    "Total tests": [],
    "Tests/1M pop": [],
    "Population": [],
    "Continent": [],
    "1 Case every X ppl": [],
    "1 Death every X ppl": [],
    "1 Test every X ppl": [],
    "New Cases/1M pop": [],
    "New Deaths/1M pop": [],
    "Active Cases/1M pop": [],
}

header_row = table.find("thead").find("tr")
data_rows = table.find("tbody").find_all("tr")
```

Hình 4: Dictionary, lấy dữ liệu từ thẻ <thread và <tbody>

- Sau khi có được nội dung, lưu từng giá trị của các cột vào dictionary đã tạo.

```python
for row in data_rows[8:]:
    columns = row.find_all("td")[1:]
    for i, k in zip(columns, data):
        data[k].append(i.text.strip())
```

Hình 5: Lưu từng giá trị các cột vào dictionary

Bước 5: Convert dữ liệu thành dạng DataFrame rồi lưu vào file csv.

- Convert dữ liệu sang DataFrame bằng các hàm phụ trợ của thư viện `pandas` và cuối cùng lưu vào file csv.

```python
# Create a Pandas DataFrame from the dictionary
df = pd.DataFrame(data)

# Save the data to a CSV file
df.to_csv(f"{datetime}_corona_data.csv", index=False)
```

Python

Hình 6: Chuyển đổi và lưu dữ liệu

2.2 Pha tiền xử lý dữ liệu:

Tiền xử lý ở bộ dữ liệu đã thu thập được nhằm giúp cho dữ liệu thô ban dầu trở nên dễ tính toán hơn, tránh sai sót trong quá trình phân tích, cụ thể nhóm em đã thực hiện các biến đổi như sau:

- Điền vào ô khuyết ở cột `Continent` bằng giá trị **"Unknown"**:

```python
covid_df['Continent'] = covid_df['Continent'].fillna('Unknown')

covid_df.head(10)
```

Python

Hình 7: Điền giá trị Unknown vào cột Continent

- Xóa hết các dấu "+" có trong trường dữ liệu `New Cases` và `New Recovered`:

```python
covid_df['New Cases'] = covid_df['New Cases'].apply(lambda x: x.replace('+', '')
                                                    if isinstance(x, str)
                                                    else x)
covid_df['New Recovered'] = covid_df['New Recovered'].apply(lambda x: x.replace('+', '')
                                                            if isinstance(x, str)
                                                            else x)

covid_df.head(10)
```

Hình 8: Xóa dấu + ra khỏi bộ dữ liệu

- Chuyển kiểu dữ liệu các trường có kiểu là object thành kiểu numeric, đồng thời thực hiện bỏ dấu "**,**" ở mỗi số.
  - Các trường dữ liệu ngoại trừ `Country`, `Continent` là các kiểu dữ liệu định danh là không chuyển đổi qua numeric.
- Sau khi chuyển xong, thực hiện điền khuyết tất cả các giá trị số thành giá trị 0.

```python
numeric_cols = ['Total Cases', 'New Cases', 'Total Deaths', 'Total Recovered',
                'New Recovered', 'Active Cases', 'Serious, Critical', 'Tot Cases/1M pop',
                'Deaths/1M pop', 'Total tests', 'Tests/1M pop', 'Population',
                '1 Case every X ppl', '1 Death every X ppl', 'Active Cases/1M pop']

covid_df[numeric_cols] = covid_df[numeric_cols].apply(lambda x: x.str.replace(',', '')).astype(float)
covid_df = covid_df.fillna(0)
```

Hình 9: Chuyển đổi kiểu dữ liệu và điền 0

- Drop 2 quốc gia Diamond Princess và MS Zaandam: 2 quốc gia Diamond Princess và MS Zaandam thiếu rất nhiều dữ liệu, không giúp được gì cho quá trình quan sát các mối quan hệ nên nhóm quyết định drop 2 dòng này.

```
1  to_drop = covid_df[covid_df["Country"].isin(["Diamond Princess", "MS Zaandam"])].index
2  covid_df = covid_df.drop(to_drop)

✓ 0.1s                                                                    Python
```

Hình 10: Xóa 2 quốc gia Diamond Princess và MS Zaandam

- Cuối cùng lưu lại vào file csv để tái sử dụng cho giai đoạn sau.

2.3 Chi tiết các thuật toán thể hiện mối quan hệ dữ liệu:

Bằng dữ liệu được lấy trong ngày 06/03/2023, các thuật toán dùng để thể hiện mối quan hệ dữ liệu được thực hiện xoay quanh thời điểm đó.

2.3.1 Mối quan hệ xoay quanh trường dữ liệu New Cases (số ca nhiễm mới).

- Ngoài các thư viện sử dụng để xử lý dữ liệu, ở trường dữ liệu này, nhóm em dùng thư viện sau đây để trực quan thành biểu đồ:
  - `plotly`: Thư viện giúp tạo đồ họa tương tác.
- Các mối quan hệ được biểu diễn như sau:

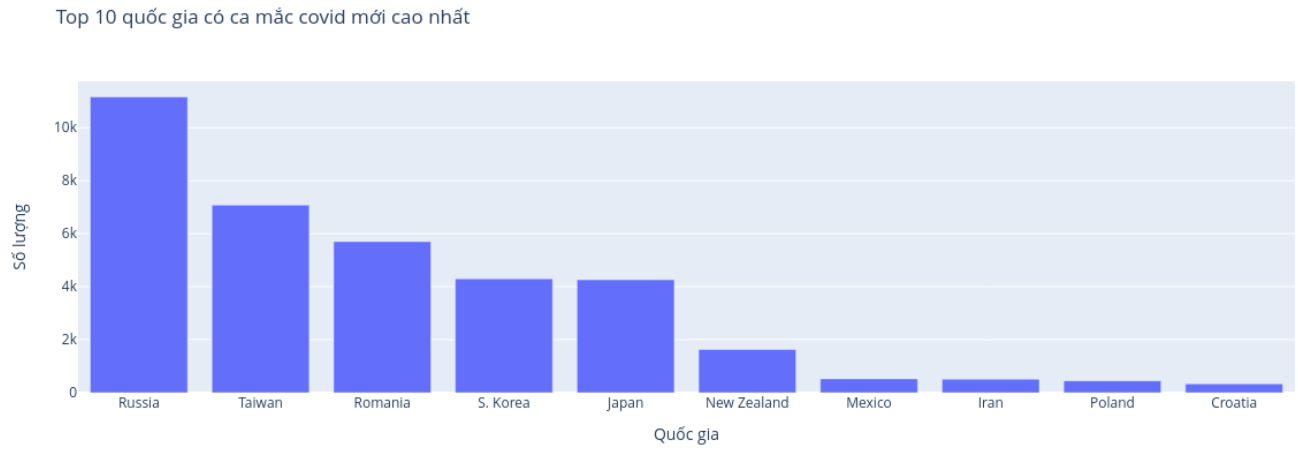
1. **Các quốc gia có số ca nhiễm covid mới cao nhất là các quốc gia nào?**
   - ***Trường dữ liệu được trực quan***: `New Cases`.
   - ***Biểu đồ sử dụng***: Bar chart.
   - ***Tính phù hợp của biểu đồ***: Bar chart thích hợp cho việc so sánh giữa các biến với nhau. Tính so sánh của nó được thể hiện qua chiều cao của các cột, cụ thể là số ca nhiễm mới giữa các quốc gia.
   - ***Mục đích của câu hỏi***: Để kiểm tra mức độ kiểm soát dịch của các quốc gia trên thế giới sau đại dịch rồi từ đó xem xét các mối quan hệ xung quanh nó.
   - ***Giải thích cách làm/thuật toán:***
     - Đầu tiên sử dụng thư viện `pandas` để lọc ra 2 trường dữ liệu `Country`, `New Cases` . Sau khi lọc thì tiến hành sắp xếp theo thứ tự giảm dần của số ca nhiễm mới, lấy ra 10 phần tử tương ứng với top 10 quốc gia có số ca nhiễm cao nhất.
     - Tiếp theo, sử dụng thư viện `plotly` để tạo ra biểu đồ cột so sánh với `Country` là trục tung, `New Cases` là trục hoành.

```python
df_top = covid_df[['Country', 'New Cases']]
df_top = df_top.sort_values(by='New Cases', ascending=False).head(10)

fig = go.Figure()
fig.add_trace(go.Bar(x=df_top['Country'], y=df_top['New Cases']))
fig.update_layout(title='Top 10 quốc gia có ca mắc covid mới cao nhất',
                  xaxis_title='Quốc gia', yaxis_title='Số lượng')

fig.show()
```

Python

Hình 11: Code thể hiện quy trình trên

- Kết quả:



Hình 12: Top 10 quốc gia có số ca mắc covid trong ngày cao nhất

- ***Nhận xét biểu đồ:***
  - Nga là quốc gia có số lượng ca mắc covid mới nhiều nhất trong ngày 06/03/2023. Điều này có thể được giải thích là do tỷ lệ tiêm phòng vắc xin của công chúng Nga đang giảm dần và việc đó đã trở nên không còn bắt buộc, quảng bá với các chủng virus mới, các lệnh bắt buộc mang khẩu trang tại nơi đông người bị gỡ bỏ. Hậu quả là số ca mắc tăng do miễn dịch cộng đồng không hiệu quả.
  - Hầu hết các quốc gia có số ca nhiễm mới thuộc top đầu đều là các nước thuộc châu Á và châu Âu.
  - Quan hệ với các trường dữ liệu khác: Nhìn tổng quan biểu đồ, số ca nhiễm của các quốc gia khác không cao như Nga, phải chăng điều này có lẽ là do Nga có dân số đông hơn các quốc gia còn lại nên số ca nhiễm cũng từ đó mà cao vượt trội hơn hẳn? Từ đó, nhóm có một câu hỏi đặt ra là:

---

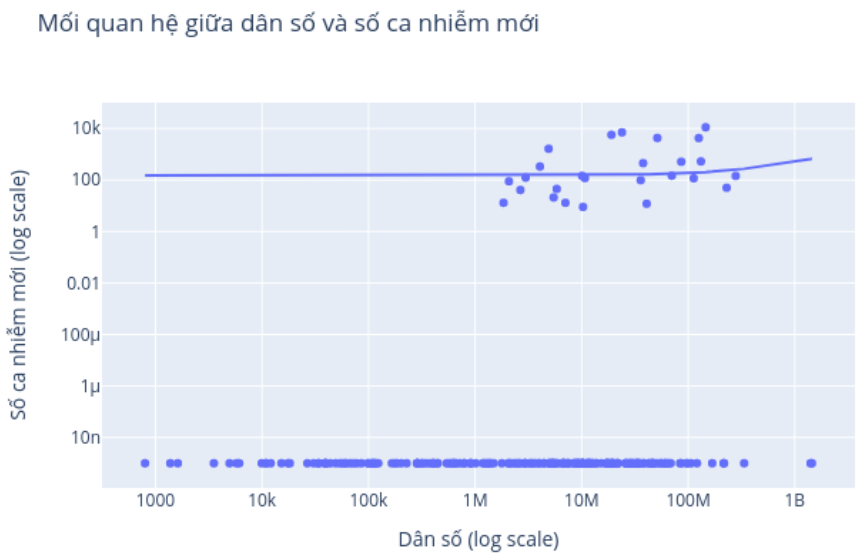
2. **Liệu số dân của một quốc gia có ảnh hưởng đến việc tăng các ca nhiễm mới?**
   - ***Trường dữ liệu được trực quan***: `New Cases`, `Population`.
   - ***Biểu đồ sử dụng***: Scatter chart.
   - ***Tính phù hợp của biểu đồ***: Scatter chart là biểu đồ thông dụng nhất khi quan sát mối quan hệ giữa các biến. Để trả lời cho câu hỏi được đặt ta, nhóm em cần xem xét có sự tương quan giữa hai biến `New Cases` và `Population` nên việc sử dụng scatter là hoàn toàn phù hợp.

- ***Mục đích của câu hỏi***: Nhằm xem xét mức độ ảnh hưởng của dân số đối với sự tăng ca nhiễm, chứng minh cho câu nói: "Nếu dân số đông thì việc lây lan dịch bệnh sẽ phủ rộng hơn, làm số ca nhiễm mới tăng, ngược lại thì mức lây lan hẹp, số ca nhiễm mới giảm".
- ***Giải thích cách làm/thuật toán:***
  - Thêm hệ số `epsilon = 1e-9` vào các giá trị của trường `New Cases` để tránh bị lỗi không xác định khi thực hiện log scale dữ liệu (log scale để thấy được rõ ràng mối quan hệ 2 trường `New Cases` và `Population`, tránh cho các giá trị bị phân tán quá rộng, dễ quan sát hơn).
  - Dùng thư viện `plotly` để tiến hành trực quan biểu đồ scatter. Truyền vào đối số `trendline=ols` để scatter hiện lên đường hồi quy, từ đó rút ra nhận xét về mối tương quan giữa 2 biến.

```
epsilon = 1e-9
covid_df['Population'] = covid_df['Population']
covid_df['New Cases'] = covid_df['New Cases'] + epsilon

fig = px.scatter(data_frame=covid_df, x='Population', y='New Cases',
                 log_x=True, log_y=True, trendline='ols',
                 labels={'Dân số': 'Population (log scale)',
                         'Số ca nhiễm mới': 'New Cases (log scale)'},
                 title='Mối quan hệ giữa dân số và số ca nhiễm mới' )

fig.show()
```

Hình 13: Code thể hiện quy trình trên

  - Kết quả:



- ***Nhận xét biểu đồ:***
  - Dựa vào biểu đồ với trendline (đường xu hướng), ta có một phương trình hồi quy là:

$$New\,Cases = 3.66312 * 10^{-7} * Population + 148.228$$

và hệ số xác định của nó là:

$$R^2 = 0.002435$$

Điều này cho ta biết thì với mỗi đơn vị tăng dân số, số ca mắc mới tăng khoảng $3.66312 * 10^{-7}$ đơn vị và giá trị cố định ban đầu của ca mắc mới là 148228. Với $R^2 = 0.002435$ cho ta biết chỉ khoảng 0.24% là sự biến thiên của số ca mắc mới có thể được giải thích bởi sự biến đổi của dân số. Như vậy, ta có thể khẳng định được rằng dân số không phải là một yếu tố quyết định trong việc giải thích sự biến động của ca mắc mới, độ tương quan của 2 biến này thấp.

  - Các điểm dữ liệu đa số tập trung ở gần trục hoành, điều này là dấu hiệu khả quan cho thấy số ca nhiễm mới rất ít dựa trên tình hình dân số. Điều này có thể được giải thích rằng các nước đông dân đã áp dụng tốt các quy định về phòng chống dịch covid hoặc miễn dịch cộng đồng cao làm cho số ca mắc mới giảm, cũng một phần củng cố cho 2 biến `Population` và `New Cases` ít có quan hệ với nhau.

---

3. **Tình hình hồi phục của các quốc gia với dịch bệnh qua mối quan hệ của 3 biến `Active Cases`, `New Cases`, `New Recovered` được thể hiện như thế nào?**
   - ***Trường dữ liệu được trực quan***: `Active Cases`, `New Cases`, `New Recovered`.
   - ***Biểu đồ sử dụng***: Bubble scatter chart.
   - ***Tính phù hợp của biểu đồ***: Bubble scatter chart cho phép vừa biểu thị mối quan hệ giữa hai biến `New Cases`, `New Recoverd` cũng vừa có thể cho phép biến thứ ba là `Active Cases` tham gia vào quá trình trực quan (thể hiện dưới dạng kích thước điểm dữ liệu) mà không cần sử dụng một biểu đồ 3D để làm việc đó.
   - ***Mục đích của câu hỏi***: Nhằm thể hiện tình hình hồi phục của các quốc gia có nhanh chóng hay không để cung cấp tốt các hoạt động chữa trị bệnh Covid-19, tránh sự quá tải về cơ sở y tế. Ví dụ nếu như số ca đang điều trị lớn, tình hình hồi phục nhỏ mà số ca mắc mới lại tăng lên sẽ làm cho tình trạng các cơ sở y tế bị quá tải, phải lập thêm nhiều trạm xá/bệnh viện hơn nữa để đáp ứng việc chữa trị.
   - ***Giải thích cách làm/thuật toán:***
     - Sử dụng thư viện `plotly` để 1 lần trực quan cả 3 biến. Trong đó trục tung và trục hoành đại diện cho lần lượt hai biến `New Recoverd` và `New Cases`, sử dụng biến thứ 3 là `Active Cases` để làm kích thước của bong bóng.

- Cài đặt việt di chuột lên các điểm dữ liệu, nhóm có thể thấy được chi tiết từng quốc gia.

```
fig = px.scatter(covid_df, x='New Cases', y='New Recovered', size='Active Cases',
                 color='Active Cases', hover_name='Country',
                 labels={'New Recovered': 'Số ca hồi phục mới',
                         'Active Cases': 'Số ca đang chữa trị',
                         'New Cases': 'Số ca nhiễm mới'},
                 title='Tình hình chịu tải/hồi phục của các quốc gia',
                 size_max=60)

fig.show()
```

Hình 14: Tình hình chịu tải/hồi phục của các quốc gia

- Kết quả:

Hình 15: Tình hình chịu tải y tế/phục hồi của các quốc gia

- ***Nhận xét biểu đồ:***
  - Tình hình y tế của đa số các nước không gặp vấn đề gì vì số ca nhiễm mới bị ít đi, số ca hồi phục cũng nhiều hơn đáng kể so với số ca đang điều trị hiện tại. Điều này cho thấy sự tích cực trong việc phòng và chữa bệnh của các quốc gia cũng như cơ sở y tế của họ không bị

quá tải.
- Nhật Bản là nơi có số ca nhiễm mới không quá cao, số ca hồi phục cũng ít nhưng lại có số ca đang chữa trị lớn nhất trên thế giới. Điều này chứng tỏ việc khả năng chữa bệnh của quốc gia này đang hạn chế so với phần còn lại nhưng có lẽ cũng sẽ không bị quá tải y tế.
- Nga là nước có số ca đang chữa trị thấp nhưng số ca nhiễm mới lại cao, tình hình hồi phục cũng thấp. Những thông số này cho thấy tình hình của Nga thật báo động vì nếu cứ tiếp tục theo xu hướng này, Nga sẽ bị quá tải về các cơ sở y tế.
- Ta có thể rút ra một điều: Nếu số ca nhiễm mới mà thấp, số ca đang chữa trị cao, số ca hồi phục cao thì quốc gia đó đang có tình hình y tế ổn định, kiểm soát được tình trạng lây lan dịch bệnh tốt và dấu hiệu chữa trị tích cực.

---

4. **Ca nhiễm mới tăng lên theo thời gian thì số ca tử vong mới, ca nguy kịch mới liệu có cùng tăng hay không?**

   - ***Trường dữ liệu được trực quan***: `'New Cases'`, `'New Deaths'`, `'Serious, Critical'` và trường phụ `Date`.
   - ***Biểu đồ sử dụng***: Line chart.
   - ***Tính phù hợp của biểu đồ***: Line chart thể hiện xu hướng giữa các biến theo trình tự thời gian. Việc sử dụng line chart ở quan hệ này cho nhóm thấy được xu hướng thay đổi theo thời gian (1 tuần) giữa ba biến `New Cases`, `New Deaths` và `Serious`, `Critical`, từ đó đưa ra kết luận liệu 3 biến này có mối quan hệ với nhau (cụ thể là đồng biến, nghịch biến) hay không.
   - ***Mục đích của câu hỏi:*** Nhằm thể hiện tình hình dịch bệnh Covid-19 qua thời gian, đặc biệt là tình hình tăng cao của các ca nhiễm mới, ca tử vong mới, ca nguy kịch mới. Ví dụ nếu như số ca nhiễm mới tăng lên theo thời gian thì số ca tử vong mới, nguy kịch mới tăng cùng thì đây là dấu hiệu báo động về tình hình dịch bệnh, chính phủ cần đưa ra các biện pháp quyết liệt hơn để kiểm soát dịch bệnh.
   - ***Giải thích cách làm/thuật toán:***
     - Đọc dữ liệu từ 6 tệp tin csv chứa dữ liệu Covid-19 đã thu thập (trải dài từ ngày 6 đến ngày 13). Sau đó, nó chọn các cột liên quan đến số ca mắc mới, số ca tử vong mới và số ca bệnh nặng, tạo cột mới để lưu trữ ngày tương ứng của dữ liệu đó và ghép các tập dữ liệu lại thành một tập dữ liệu lớn.

```python
# Tiền xử lý dữ liệu cho mối liên hệ giữa các trường này:

def genDateDataframe(path, date):
    df = pd.read_csv(path)
    df = df[['Country', 'New Cases', 'New Deaths', 'Serious, Critical']]
    df['Date'] = date
    df['Date'] = pd.to_datetime(df['Date'])

    return df

dfCovid_0603 = genDateDataframe('../data/modified/Modified_March_06_2023_corona_data.csv', '2023-03-06')
dfCovid_0703 = genDateDataframe('../data/modified/Modified_March_07_2023_corona_data.csv', '2023-03-07')
dfCovid_0803 = genDateDataframe('../data/modified/Modified_March_08_2023_corona_data.csv', '2023-03-08')
dfCovid_0903 = genDateDataframe('../data/modified/Modified_March_09_2023_corona_data.csv', '2023-03-09')
dfCovid_1103 = genDateDataframe('../data/modified/Modified_March_11_2023_corona_data.csv', '2023-03-11')
dfCovid_1303 = genDateDataframe('../data/modified/Modified_March_13_2023_corona_data.csv', '2023-03-13')

dfCovid = pd.concat([dfCovid_0603, dfCovid_0703, dfCovid_0803,
                     dfCovid_0903, dfCovid_1103, dfCovid_1303])
```

Hình 16: Đọc và tiền xử lý dữ liệu trong 1 tuần

- Tiến hành trực quan hóa tình hình dịch bệnh tại 5 quốc gia có số ca mắc Covid-19 mới nhiều nhất bằng cách vẽ đồ thị biểu diễn sự thay đổi của số ca mắc mới, số ca tử vong và số ca nghiêm trọng tại các quốc gia đó theo thời gian. Sử dụng thư viện plotly và subplots để vẽ đồ thị và pandas để tiền xử lý dữ liệu (lọc 5 quốc gia có ca mắc cao).

```python
dfCovid_top = covid_df[['Country', 'New Cases']]
dfCovid_top = dfCovid_top.sort_values(by='New Cases', ascending=False).head(5)

fig = make_subplots(rows=5, cols=1, shared_xaxes=True,
                    subplot_titles=(dfCovid_top['Country'].values))

for i in range(5):
    country = dfCovid_top['Country'].values[i]
    dfCovid_country = dfCovid[dfCovid['Country'] == country]

    fig.add_trace(go.Line(x=dfCovid_country['Date'], y=dfCovid_country['New Cases'],
                          name='New Cases', line_color='blue', showlegend=i==0), row=i+1, col=1)
    fig.add_trace(go.Line(x=dfCovid_country['Date'], y=dfCovid_country['New Deaths'],
                          name='New Deaths', line_color='red', showlegend=i==0), row=i+1, col=1)
    fig.add_trace(go.Line(x=dfCovid_country['Date'], y=dfCovid_country['Serious, Critical'],
                          name='Serious, Critical', line_color='green', showlegend=i==0), row=i+1, col=1)


fig.update_layout(height=1000, width=800, title_text="Tình hình dịch bệnh tại 5 quốc gia có số ca mắc covid mới nhiều nhất");
fig.show();
```

Hình 17: Thực hiện trực quan lên biểu đồ các trường cần tìm

- Kết quả:

Hình 18: Kết quả thuật toán biểu diễn

- ***Nhận xét biểu đồ:***
  - Với 5 quốc gia được cho là có số ca nhiễm mới nhiều nhất tại thời điểm 06-03-2023 thì sau 1 tuần, số ca tử vong mới và số ca nguy kịch không có dấu hiệu tăng theo số ca nhiễm mới. Điều này làm rõ mối quan hệ của hai biến `New Deaths` và `Serious, Critical` với biến `New Cases` dường như không đồng biến với nhau, tức là "số ca nhiễm mới tăng lên thì sau một thời gian, số ca nguy kịch, tử vong mới không tăng lên theo số ca nhiễm mới".
  - Để giải thích cho điều trên thì có lẽ vì các quốc gia đã tiêm chủng vắc-xin đầy đủ, hệ miễn dịch cộng đồng mạnh mẽ nên không bị nguy kịch như lúc dịch mới bùng phát, chứng tỏ rằng covid-19 không phải là dịch bệnh nguy hiểm nữa ở thời điểm hiện tại.

2.3.2 Mối quan hệ giữa 2 trường dữ liệu `1 Test every X ppl` và `1 Case every X ppl`.

- Thư viện bổ sung:
  - `geopandas`: Giúp đỡ cho việc biểu diễn dữ liệu trên Choropleth Map.
- Các mối quan hệ được biểu diễn như sau:

1. **Tỉ lệ người mắc Covid và tỉ lệ người được xét nghiệm Covid có sự phân bố theo vị trí địa lý của quốc gia trên thế giới không?**
   - ***Trường dữ liệu được trực quan:*** `1 Test every X ppl`,`1 Case every X ppl`
   - ***Biểu đồ sử dụng:*** Choropleth Map.
   - ***Tính phù hợp của biểu đồ:*** Biểu đồ Choropleth Map cho phép chúng ta có thể nhìn thấy được thực tế tình trạng trên hình ảnh của toàn bộ thế giới, tuy rằng dữ liệu được trực quan lên sẽ không được hoàn toàn đầy đủ và chính xác như các biểu đồ số liệu khác, nhưng nó sẽ cho chúng ta một cái nhìn dễ dàng hơn trong việc so sánh sự ảnh hưởng của vị trí địa lý, điều này thì rất là khó để thể hiện trên các biểu đồ thông thường, và do thế chúng ra có thể nhận xét rõ hơn trên từng khu vực ta muốn, thay vì chỉ là từng quốc gia hay lục địa cụ thể.
   - ***Mục đích câu hỏi:*** Kiểm tra xem sự lây lan của dịch Covid có khác nhau khi nhận xét tổng quan trên toàn bộ thế giới hay không, cũng như là tình trạng kiểm tra sức khỏe y tế của các khu vực.
   - ***Giải thích cách làm/thuật toán:***
     - Đọc dữ liệu của tình trạng covid bằng thư viện `pandas`.
     - Đọc dữ liệu vị trí địa lý của các quốc gia bằng data có sẵn của thư viện geopandas.
     - Sửa lại tên của cột `name` của dữ liệu địa lý thành `Country` để tiện cho việc join bảng ở bước sau.
     - Sửa lại tên của các quốc gia không khớp nhau giữa 2 dữ liệu covid và địa lý (ta sẽ kiểm tra thủ công ở bước này).

```
# Lấy dữ liệu về các đồ thị quốc gia
df = geopandas.read_file(geopandas.datasets.get_path("naturalearth_lowres"))
```

✓ 0.8s

```
#đổi tên cột cũng như đổi đổi tên các quốc qua cho đúng với data từ Geopandas
df = df.rename(columns={"name":"Country"})
covid_df =  covid_df.replace(['USA','UK','Bosnia and Herzegovina',
                              'S. Korea','Western Sahara','South Sudan',
                              'Dominican Republic','Solomon Islands',
                              'Equatorial Guinea','Falkland Islands',
                              'Eswatini','DPRK','DRC',
                              'UAE','CAR','Ivory Coast'],

                             ['United States of America','United Kingdom','Bosnia and Herz.',
                              'South Korea','W. Sahara', 'S. Sudan',
                              'Dominican Rep.', 'Solomon Is.',
                              'Eq. Guinea','Falkland Is.',
                              'eSwatini', 'North Korea','Dem. Rep. Congo',
                              'United Arab Emirates','Central African Rep.', "Côte d'Ivoire"])
```

✓ 0.0s

```
#Gộp dữ liệu 2 bảng
data = geopandas.GeoDataFrame(
    covid_df.set_index('Country').join(
        df[['Country','geometry']].set_index('Country'),
        lsuffix='_caller',
        rsuffix='_other',)
    )
```

✓ 0.0s

Hình 19: Code thực hiện công đoạn trên

- Tạo bảng data mới chứa dữ liệu của Covid cũng như vị trí quốc gia (kiểu dữ liệu `geopandas`).
- Viết hàm tinh chỉnh để có thể vẽ lên Choropleth map bằng heatmap.
- Vẽ 2 biểu đồ với 2 cột dữ liệu 1 `Test every X ppl`, 1 `Case every X ppl`, và scale theo log (do thế ta sẽ fix dữ liệu để tránh trường hợp số 0) để có cái nhìn dễ hơn.

```python
def MapPlot(Col, Norm = None,figsize =(30,18),fig = None,
            ax = None,data=data,cmap ='cool',vcenter = None,
            fontdict={'fontsize': '25', 'fontweight': '3'}
            ):
    col_plot = Col
    if fig == None and ax == None:
        fig, ax = plt.subplots(1, figsize=figsize)
    ax.axis('off')
    if type(Col) == type(""):
        col_plot = data[Col]
        ax.set_title(Col, fontdict=fontdict)

    if vcenter != None:
        colorNorm = colors.CenteredNorm(vcenter=vcenter)
    else:
        if Norm == None:
            vmin =  col_plot.min()
            vmax =  col_plot.max()
        else:
            vmin =  Norm(col_plot.min())
            vmax =  Norm(col_plot.max())
        colorNorm = plt.Normalize(vmin=vmin, vmax=vmax)

    if Norm == None:
        data.plot(col_plot,ax = ax,linewidth = 1, cmap = cmap , norm = colorNorm,edgecolor='black')
    else:
        data.plot(Norm(col_plot),ax = ax,linewidth = 1, cmap = cmap,edgecolor='black')
    ax.annotate('', xy=(0.1, .08), xycoords='figure fraction', horizontalalignment='left',
                verticalalignment='bottom', fontsize=10)

    sm = plt.cm.ScalarMappable(norm=colorNorm, cmap=cmap)
    sm._A = []
    # Add the colorbar to the figure
    cbaxes = ax.inset_axes([0.01, 0.02, 0.01, 1])
    cbar = fig.colorbar(sm, cax=cbaxes)
```

✓ 0.0s

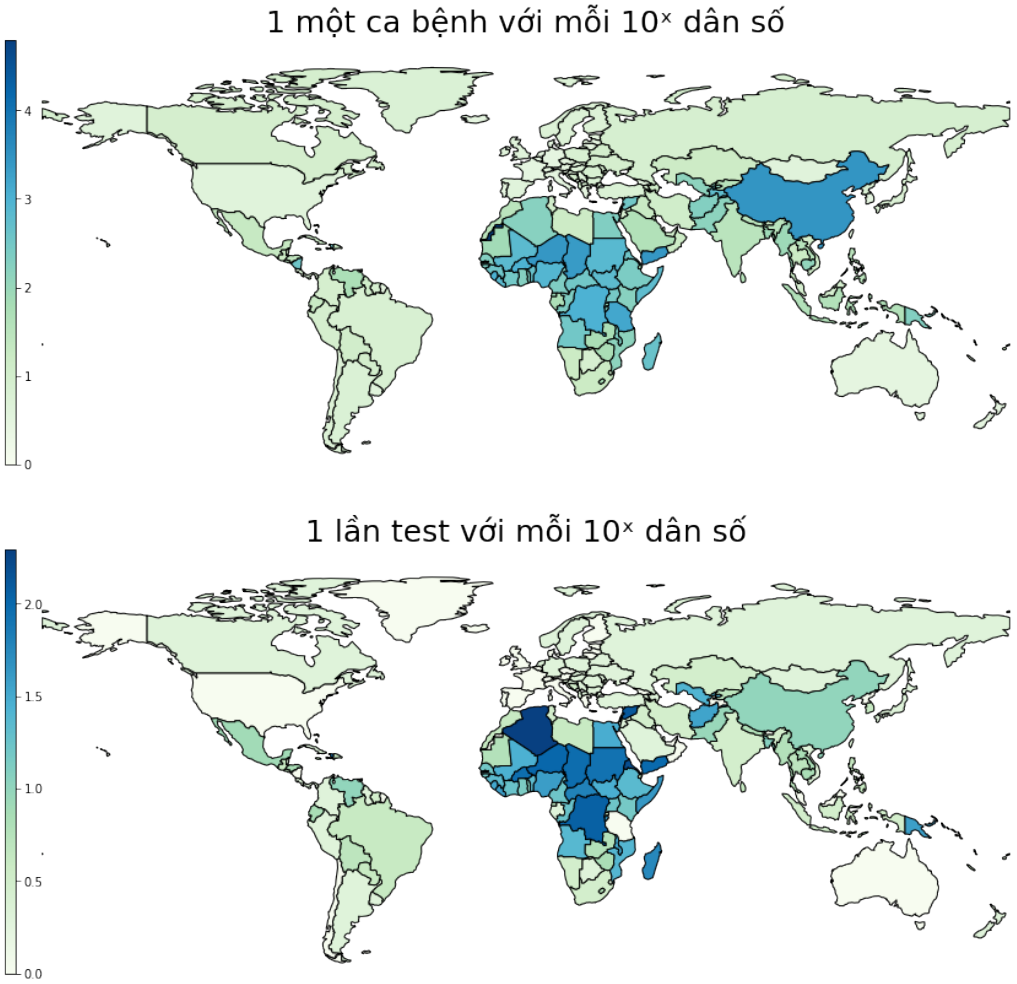
Hình 20: Hàm tinh chỉnh

```python
fig, (ax1,ax2) = plt.subplots(2, figsize=(30, 14))

MapPlot('CasePerPPL',
        data = data.assign(CasePerPPL = lambda x: x['1 Case every X ppl']+1),
        cmap = 'GnBu',
        ax=ax1,
        fig=fig,
        Norm = np.log10)
ax1.set_title('1 một ca bệnh với mỗi 10ˣ dân số', fontdict={'fontsize': '25', 'fontweight': '3'})

MapPlot('CasePerPPL',
        data = data.assign(CasePerPPL = lambda x: x['1 Test every X ppl']+1),
        cmap = 'GnBu',
        ax=ax2,
        fig=fig,
        Norm = np.log10)
ax2.set_title('1 lần test với mỗi 10ˣ dân số', fontdict={'fontsize': '25', 'fontweight': '3'})
```

✓ 0.5s

Hình 21: Vẽ biểu đồ

- ***Kết quả:***



Hình 22: Biểu đồ thể hiện 1 ca bệnh/test với mỗi 10^x dân số

- ***Nhận xét biểu đồ:***
  - Ta có thể thấy được có thể thấy được tỉ lệ mắc bệnh có vẻ được phân bố theo cách các nước có tỉ lệ nhiễm thấp lại ở gần nhau (Lục địa châu Phi) và phần nhỏ Châu Á. Để giải thích cho điều này thì đa phần các khu vực này có nhiệt độ và độ ẩm không thích hợp với Covid nên tỉ lệ bị nhiễm sẽ thấp hơn với phần còn lại của thế giới.
  - Nếu xem xét thêm về điều kiện kiểm tra bệnh, tại các khu vực này cũng có điều kiện kiểm tra thấp nhất trên toàn thế giới, vậy liệu điều này có phải là lí do giải thích cho việc khu vực này ghi nhận ít ca bệnh hơn khu vực khác hay còn nguyên nhân nào đó?

---

2. **Việc tỉ lệ kiểm tra y tế có ảnh hưởng đến số ca bệnh được ghi nhận trên hệ thống hay không?**

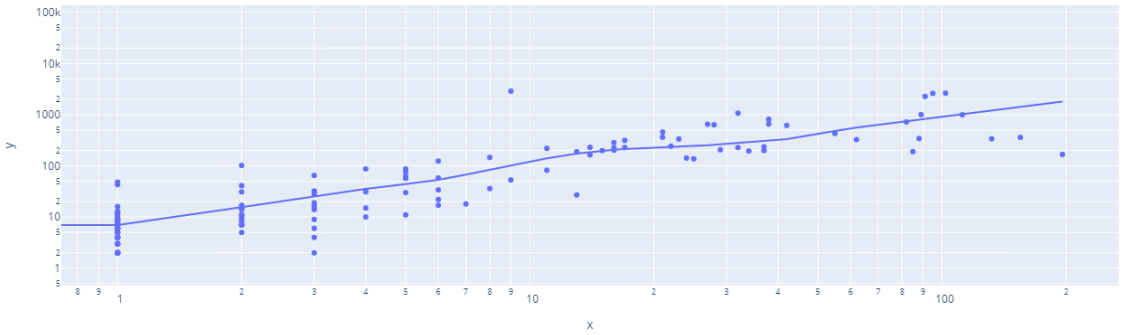
   - ***Trường dữ liệu được trực quan***: `1 Test every X ppl`,`1 Case every X ppl`
   - ***Biểu đồ sử dụng:*** Scatter chart.
   - ***Tính phù hợp của biểu đồ:*** Việc xem xét mối tương quan giữa 2 trường dữ liệu thì biểu đồ scatter vẫn cho ta một cái nhìn dễ nhất, khi mà tất cả điểm dữ liệu được trực quan trên một biểu đồ 1 cách rõ ràng, từ đó ta cũng có thể xem được phân bố của các dữ liệu này như thế nào.
   - ***Giải thích cách làm/thuật toán:***
     - Dùng thư viện `plotly` để thể hiện biểu đồ này, và ta scale 2 trục theo log.
     - Ta cũng bổ sung thêm một đường trendline (không phải linear), để có xem xét về mối tương quan giữa 2 biến.

```
px.scatter(
            x=covid_df['1 Test every X ppl'],
            y = covid_df['1 Case every X ppl'],
            log_y= True,log_x=True,
            trendline="lowess"
        )
```

✓ 0.1s

Hình 23: Code biểu đồ scatter bằng plotly

   - ***Kết quả:***



- **Nhận xét biểu đồ:**
  - Tại mỗi mức chỉ số 1 `Test every X ppl` thì ra thấy chỉ số 1 `Case every X ppl` được phân bố hầu như trong một khoản xác định và có xu hướng hơi tăng dần.
  - Từ biểu đồ này ta cũng thấy được rằng việc kiểm tra cũng phần nào phản ánh việc ghi nhận ca nhiễm bệnh trên thế giới, nên đây cũng là điều đáng phải quan tâm để có thể ghi nhận ca bệnh tốt nhất, giúp ích cho công cuộc theo dõi tình hình Covid trên toàn thế giới nói chung.

2.3.3 Mối quan hệ xoay quanh trường dữ liệu `Total Cases` (Tổng số ca nhiễm):

- Thư viện bổ sung:
  - `matplotlib`: Thư viện trực quan cơ bản cho dữ liệu.
- Các mối quan hệ được biểu diễn như sau:

1. **Tỉ lệ số ca nhiễm và tỉ lệ số ca hồi phục của các Châu lục như thế nào?**
   - ***Trường dữ liệu được trực quan:*** `Total Cases`, `Total Recovered`, `Continent`.
   - ***Biểu đồ sử dụng:*** Nested Pie chart (Nested Donut chart).
   - ***Tính phù hợp của biểu đồ:*** Nested Pie chart thường được sử dụng để hiển thị các tỷ lệ phần trăm của các phân loại dữ liệu con bên trong một phân loại dữ liệu lớn hơn. Trong trường hợp này, nhóm em sử dụng cho mục đích so sánh tỉ lệ số ca hồi phục so với số ca nhiễm khi cái trước là 1 phần nhỏ của cái sau.
   - ***Mục đích của câu hỏi:*** Để xem tình hình và kết quả khám chữa bệnh các bệnh nhân nhiễm Covid-19 ở các châu lục đã và đang diễn ra như thế nào, và phần nào chứng tỏ mức độ hiện đại và hiệu quả của nền y học của các châu lục.
   - ***Giải thích cách làm/thuật toán:***
     - Sử dụng `pivot_table` để tính tổng số ca mắc và số ca hồi phục trên từng lục địa từ tập dữ liệu `covid_df`.

```
continent_total_cases = pd.pivot_table(data=covid_df,
                                       values=['Total Cases', 'Total Recovered'],
                                       index='Continent',
                                       aggfunc='sum',
                                      ).sort_values(by='Total Cases')
```

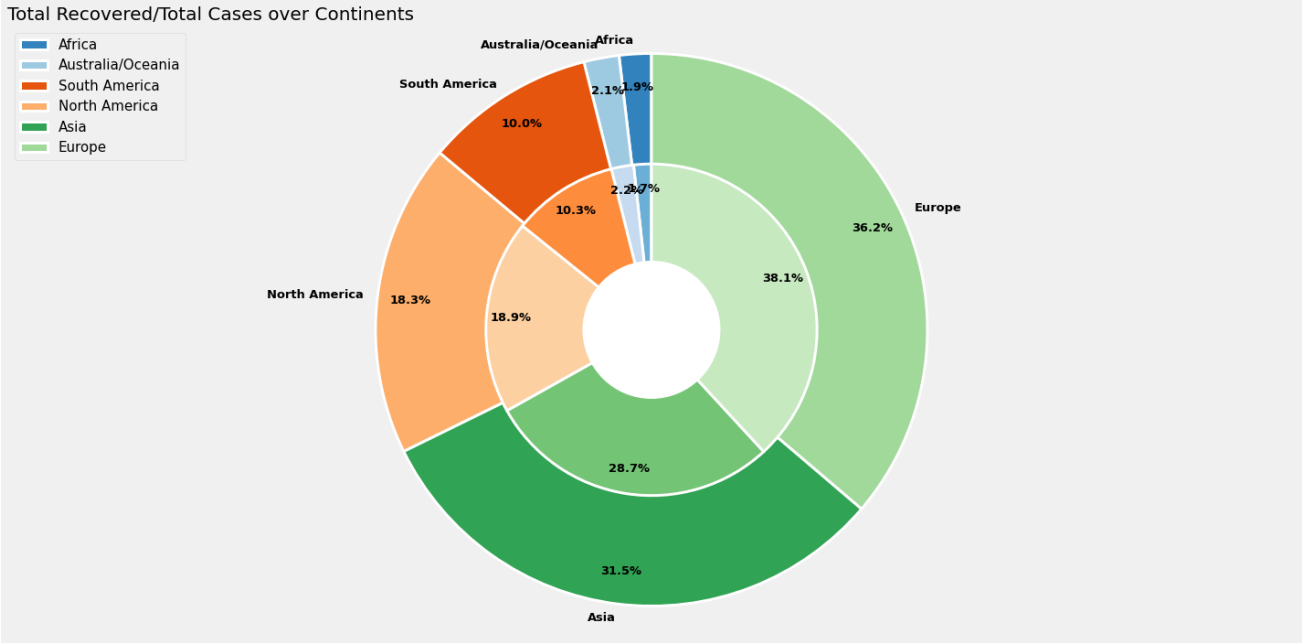
Hình 24: Sử dụng pivot table để tính tổng ca mắc và số ca hồi phục từng lục địa

- Sử dụng `plt.pie` để vẽ biểu đồ Pie Chart lồng vào nhau, với Pie Chart ngoài thể hiện tỷ lệ tổng số ca mắc Covid trên các lục địa, và Pie Chart trong thể hiện tỷ lệ tổng số ca hồi phục Covid trên các lục địa. Các thông số của biểu đồ được điều chỉnh như màu sắc, phần trăm, tỉ lệ bán kính, khoảng cách giữa các vòng Pie Chart. Cuối cùng, đoạn code thêm một đường tròn trắng ở giữa biểu đồ.

```
cmap = plt.colormaps["tab20c"]
outer_colors = cmap(np.arange(6)*2)
inner_colors = cmap([1, 3, 5, 7, 9, 11])

plt.figure(figsize=(20,10))

plt.pie (
  continent_total_cases['Total Cases'],
  labels=continent_total_cases.index,
  startangle=90, pctdistance =0.88,
  colors=outer_colors,
  autopct = '%1.1f%%', radius= 1.0, labeldistance=1.05,
  textprops ={ 'fontweight': 'bold','fontsize':13},
  wedgeprops = {'linewidth' : 3, 'edgecolor' : "w"}
)

plt.pie (
  continent_total_cases['Total Recovered'],
  startangle=90, pctdistance =0.85,
  colors=inner_colors,
  autopct = '%1.1f%%', radius= 0.6, labeldistance=1.05,
  textprops ={ 'fontweight': 'bold','fontsize':13},
  wedgeprops = {'linewidth' : 3, 'edgecolor' : "w"}
)

centre_circle = plt.Circle((0,0), 0.25, fc='white')
fig= plt.gcf()
fig.gca().add_artist(centre_circle)

plt.title('Total Recovered/Total Cases over Continents', fontsize=20, loc='left')
plt.axis('equal')
plt.legend(loc=2, fontsize=15)
plt.tight_layout()
plt.show()
```

Hình 25: Thực hiện vẽ biểu đồ

- ***Kết quả:***



Hình 26: Biểu đồ thể hiện tỉ lệ Recoverd/Total Cases từng châu lục

- ***Nhận xét biểu đồ***:
  - Châu Âu, Châu Á là hai châu lục có số ca nhiễm nhiều nhất thế giới, theo sau đó là Châu Mỹ (gồm Bắc Mỹ và Nam Mỹ)
  - Các Châu lục kể trên với Châu Đại dương và Châu Phi, ngoại trừ Châu Á và Châu Phi, có tỉ lệ số ca hồi phục cao hơn tỉ lệ số ca nhiễm (so với toàn thế giới), phần nào chứng tỏ hiệu quả của công tác khám chữa bệnh ở các châu lục này.

2. **Tổng số ca hồi phục trên toàn thế giới có phụ thuộc vào tổng số ca nhiễm hay không?**

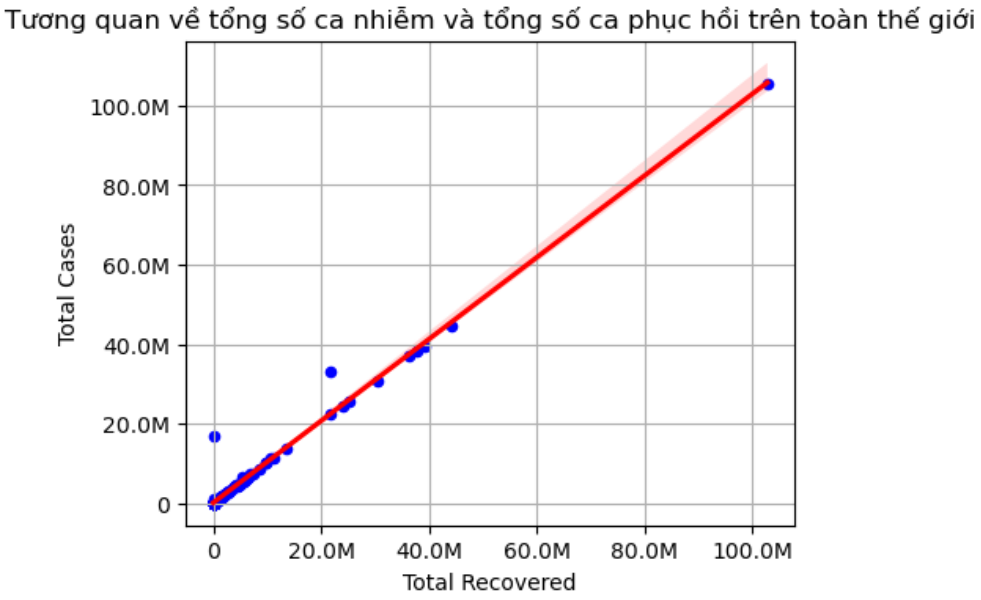
- ***Trường dữ liệu được trực quan:*** `Total Cases`, `Total Recovered`
- ***Biểu đồ sử dụng:*** Scatter chart.
- ***Tính phù hợp của biểu đồ:*** Để thấy được mối tương quan giữa 2 trường dữ liệu để xem rằng 2 trường dữ liệu này có mối quan hệ tác động, phụ thuộc hoặc độc lập với nhau không.
- ***Mục đích của câu hỏi:*** Để nắm rõ được tình hình rằng các ca nhiễm covid có ảnh hưởng như thế nào đến số ca hồi phục trên toàn thế giới để phần nào hiểu rõ được tình hình phòng chống dịch.
- ***Giải thích cách làm/thuật toán***:
  - Đọc dữ liệu tình trạng covid bằng thư viện `pandas`
  - Lấy ra 2 cột dữ liệu `Total Cases` và `Total Recovered`

- Sử dụng biểu đồ scatter để trực quan dữ liệu với trục x là `Total Recovered` và trục y là `Total Cases`.
- Cuối cùng, scale lại trục y và trục x theo hàm `log`.
- Ở biểu đồ này, để vẽ được thêm đường hồi quy, ta dùng hàm `regplot` của seaborn và được kết quả:

```
fig, ax = plt.subplots()
ax.set_title("Tương quan về tổng số ca nhiễm và tổng số ca phục hồi trên toàn thế giới"
covid_df[['Total Cases', 'Total Recovered']].plot.scatter(x='Total Recovered', y='Total

ax.grid()
ax.yaxis.set_major_formatter(ticker.FuncFormatter(format_y_axis))
ax.xaxis.set_major_formatter(ticker.FuncFormatter(format_y_axis))

sns.regplot(x='Total Recovered', y='Total Cases', data=covid_df, ax=ax, scatter=False,

plt.show()
```

Hình 27: Code thuật toán trực quan

- Kết quả:



Hình 28: Biểu đồ tương quan về tổng số ca nhiễm và tổng số ca hồi phục trên toàn thế giới

- ***Nhận xét biểu đồ***:
  - Mối tương quan giữa 2 trường dữ liệu này gần như phụ thuộc nhau hoàn toàn khi tất cả các điểm dữ liệu tập trung thành 1 đường thẳng, điều này cho thấy số ca nhiễm và số ca hồi phục trên toàn thế giới gần như tuyến tính với nhau.
  - Từ đó, ta hiểu được rằng cứ số ca nhiễm trên thế giới ở từng quốc gia tăng lên thì số lượng người hồi phục cũng tăng theo, cho thấy được khả năng phòng chống dịch của các quốc gia đang được nâng cao và người dân có ý thức được tinh thần chống dịch bệnh.
  - Chính vì vậy, việc xây dựng mô hình hồi quy tuyến tính sẽ giúp ta theo dõi được ngay số lượng người được hồi phục ngay tương ứng với số ca nhiễm, nhưng bên cạnh đó cũng phải dùng các phương pháp thống kê để đánh giá mô hình, đảm bảo được độ chính xác cao nhất.

3. **Tổng số ca test trên toàn thế giới có mối quan hệ với tổng số ca nhiễm hay không?**

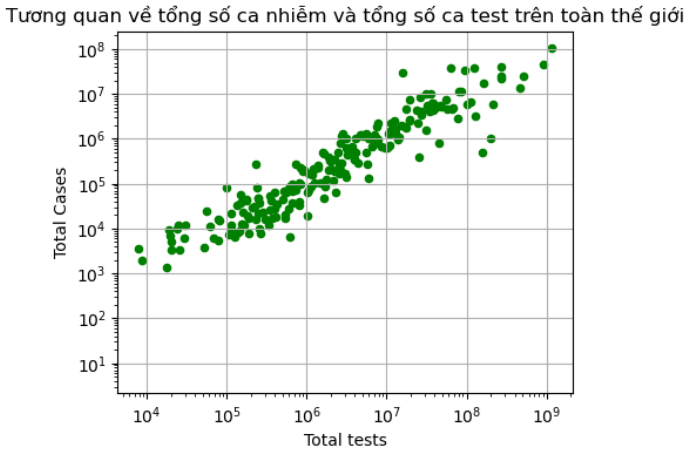
   - ***Trường dữ liệu được trực quan***: `Total Cases`, `Total Tests`
   - ***Biểu đồ sử dụng***: Scatter plot (Biểu đồ phân tán)
   - ***Tính phù hợp của biểu đồ***: Để thấy được mối tương quan giữa 2 trường dữ liệu để xem rằng 2 trường dữ liệu này có mối quan hệ tác động, phụ thuộc hoặc độc lập với nhau không.
   - ***Mục đích của câu hỏi:*** Để nắm rõ được tình hình các ca test trên thế giới được phân bố như thế nào trước tổng số ca nhiễm trên mỗi quốc gia.
   - ***Giải thích cách làm/thuật toán***:
     - Đọc dữ liệu tình trạng covid bằng thư viện `pandas`.
     - Lấy ra 2 cột dữ liệu `Total Cases` và `Total Tests`.
     - Sử dụng biểu đồ scatter để trực quan dữ liệu với trục x là `Total Tests` và trục y là `Total Cases`.
     - Cuối cùng, scale lại trục y và trục x theo hàm `log`.

```python
fig, ax = plt.subplots()
ax.set_title("Tương quan về tổng số ca nhiễm và tổng số ca test trên toàn
covid_df[['Total Cases', 'Total tests']].plot.scatter(
    x='Total tests',
    y='Total Cases',
    ax=ax,
    color='green',
    figsize=(5, 4)
)

ax.grid()
ax.yaxis.set_major_formatter(ticker.FuncFormatter(format_y_axis))
ax.xaxis.set_major_formatter(ticker.FuncFormatter(format_y_axis))

plt.xscale('log')
plt.yscale('log')
plt.show()
```

MagicPython

Hình 29: Code thuật toán

– Kết quả:



Hình 30: Biểu đồ tổng số ca nhiễm và tổng số ca test trên toàn thế giới

- ***Nhận xét biểu đồ***:
  - Số ca nhiễm và số ca test gần như hồi quy, 2 trường dữ liệu này có sự tương quan với nhau khi các điểm dữ liệu tạo thành 1 đường thẳng.
  - Từ đó, ta có thể rút ra được kết luận rằng số ca test covid sẽ phụ thuộc vào số ca nhiễm covid, khi số ca nhiễm tăng thì số ca test sẽ tăng.
  - Xét trường dữ liệu liên quan: Để tìm hiểu kỹ hơn, ta hãy theo dõi mối quan hệ giữa số ca nhiễm và số ca chết gây ra bởi dịch covid trong mục 2.3.4.

2.3.4 Mối quan hệ xoay quanh trường dữ liệu `Tests/1M pop` (Tỷ lệ test trên 1 triệu dân):

1. **Tổng số ca dương tính, âm tính và số lần test trên 1 triệu dân của top 10 quốc gia sắp xếp tăng dần theo tổng số ca nhiễm.**

   - ***Trường dữ liệu được trực quan***: `Country`, `Tests/1M pop`, `Tot Cases/1M pop`
   - ***Biểu đồ sử dụng***: Nested & Stacked Bar chart.
   - ***Tính phù hợp của biểu đồ***: Nested & Stacked Bar chart giúp hiển thị các phân loại và giá trị liên quan đến trường dữ liệu một cách rõ ràng và trực quan, giúp dễ dàng so sánh và phân tích dữ liệu, đồng thời cho phép hiển thị nhiều mức độ thông tin khác nhau trong cùng một biểu đồ. Nhờ vào những lợi ích đó, áp dụng vào quan hệ này để dễ dàng so sánh giữa số ca dương tính so với số ca âm tính và so sánh với số lần test trên 1 triệu dân.
   - ***Mục đích của câu hỏi***: Để xem tình hình, mức độ kiểm soát, phát hiện và ngăn ngừa dịch bệnh ở các quốc gia đứng đầu về số ca nhiễm.
   - ***Giải thích cách làm/thuật toán:***
     - Chọn ra 10 quốc gia có số ca nhiễm nhiều nhất từ dataframe `covid_df`, sau đó chọn ra các cột cần thiết là `Country`, `Tests/1M pop`, `Tot Cases/1M pop` và tính toán thêm cột `Neg./1M pop` (số ca âm tính trên 1 triệu dân) bằng cách trừ 10^6 với số ca dương tính trên 1 triệu dân.

```
case_test_1m_pop = covid_df.nlargest(columns='Total Cases', n=10) \
                   .loc[:, ['Country', 'Tests/1M pop', 'Tot Cases/1M pop']] \
                   .sort_values('Tot Cases/1M pop')

case_test_1m_pop['Neg./1M pop'] = 10**6 - case_test_1m_pop['Tot Cases/1M pop']
```

Hình 31: Chọn 10 quốc gia có số ca nhiễm và tính toán thêm cột

     - Vẽ biểu đồ bằng `matplotlib` và `pandas`, sử dụng bar chart với 3 cột dữ liệu, tương ứng với số lần test trên 1 triệu dân, tổng số ca dương tính trên 1 triệu dân và tổng số ca âm tính trên 1 triệu dân cho 10 quốc gia được chọn. Biểu đồ sử dụng nested và stacked để hiển thị 2 loại dữ liệu tổng số ca và số ca âm tính trên 1 triệu dân dưới dạng stacked và dữ liệu số lần test trên 1 triệu dân dưới dạng nested.

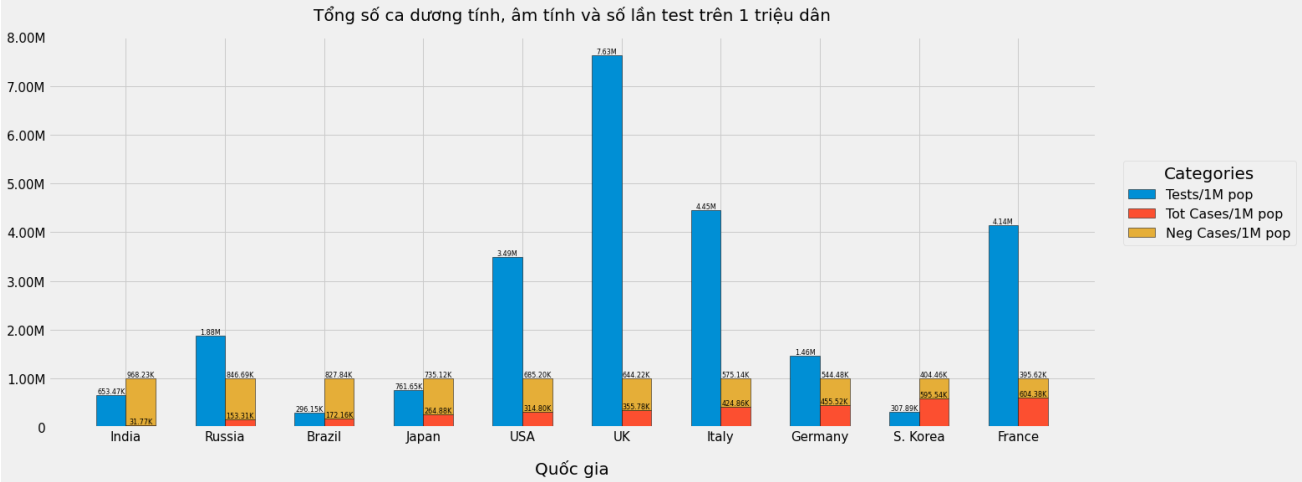
```
plt.style.use("fivethirtyeight")

fig, ax = plt.subplots(1,1, figsize = (20,8))
label = case_test_1m_pop['Country']

x = np.arange(len(label))
width = 0.3

rect1 = ax.bar(x - width/2,
               case_test_1m_pop['Tests/1M pop'],
               width = width,
                label = 'Tests/1M pop',
                edgecolor = "black"
)

rect2 = ax.bar(x + width/2,
               case_test_1m_pop['Tot Cases/1M pop'],
               width = width,
                label = 'Tot Cases/1M pop',
                edgecolor = "black"
)
rect3 = ax.bar(x + width/2,
               case_test_1m_pop['Neg./1M pop'],
               width = width,
                label = 'Neg Cases/1M pop',
                edgecolor = "black",
                bottom=case_test_1m_pop['Tot Cases/1M pop']
)
ax.bar_label(rect1, fontsize=8, labels=[format_y_axis(x) for x in case_test_1m
ax.bar_label(rect2, fontsize=8, labels=[format_y_axis(x) for x in case_test_1m
ax.bar_label(rect3, fontsize=8, labels=[format_y_axis(x) for x in case_test_1m

# ax.set_ylabel("Số lượng", fontsize = 20, labelpad = 20)
ax.set_xlabel("Quốc gia", fontsize = 20, labelpad =20)
ax.set_title("Tổng số ca dương tính, âm tính và số lần test trên 1 triệu dân",
#set the ticks
ax.set_xticks(x)
```

Hình 32: Thể hiện thuật toán bằng code 1

```
ax.set_xticklabels(label)
#add the legends
#using the labels of the bars
ax.legend(title = "Categories", fontsize = 16, title_fontsize = 20, bbox_to_an
#adjust the tick paramaters
ax.tick_params(axis = "x", which = "both", labelrotation = 90, labelsize = 15)
ax.tick_params(axis = "y", which = "both", labelsize = 15)
ax.yaxis.set_major_formatter(ticker.FuncFormatter(format_y_axis))
ax.xaxis.set_tick_params(rotation=0)


plt.show()
```

Hình 33: Thể hiện thuật toán bằng code 2

- ***Kết quả:***



Hình 34: Biểu đồ thể hiện tổng số ca dương tính, âm tính và số lần test trên 1 triệu dân

- ***Nhận xét biểu đồ***:
  - Quốc gia lớn của Châu Âu gồm Anh, Ý và Pháp lần lượt là 3 nước dẫn đầu về mức độ kiểm tra phòng chống dịch bệnh, theo sau đó là 2 siêu cường quốc Mỹ và Nga.
  - Hàn Quốc là nước có mức độ kiểm tra phòng chống dịch bệnh thấp thứ 2 (chỉ sau Brazil), nhưng lại có số ca dương tính trên 1 triệu người cao thứ 2 (chỉ sau Pháp).
  - Ấn Độ là nước có số ca dương tính trên 1 triệu dân thấp nhất, thấp hơn đáng kể so với các nước còn lại trong top 10.

2.3.5 Mối quan hệ xoay quanh trường dữ liệu `Total Deaths` (Tổng số ca tử vong) và `Total Cases` (Tổng số ca nhiễm):

1. **Top 3 quốc gia có tổng số ca tử vong nhiều nhất và trong đó quốc gia nào bị ảnh hưởng sâu sắc nhất:**
   - ***Trường dữ liệu được trực quan***: `Total Deaths`
   - ***Biểu đồ sử dụng***: Pie Chart.
   - ***Tính phù hợp của biểu đồ***: Để dễ dàng so sánh top 3 quốc gia có tổng ca tử vong nhiều nhất, pie chart là một biểu đồ phù hợp vì việc so sánh được thể hiện rõ tỉ lệ phần trăm trên mỗi phần biểu đồ.
   - ***Mục đích của câu hỏi:*** Để xem top 3 quốc gia có số ca tử vong nhiều nhất và trong đó nước nào bị ảnh hưởng sâu sắc nhất trong tình hình hiện tại:
   - ***Giải thích cách làm/thuật toán:***

- Ở trường dữ liệu `Total Deaths`, nhóm các dữ liệu trong bảng `covid_df` theo tên quốc gia và tính tổng số lượng ca tử vong của từng quốc gia. Sau đó, sắp xếp các quốc gia theo thứ tự giảm dần về số lượng ca tử vong và lấy 3 quốc gia có số lượng ca tử vong cao nhất.
- Ở trường dữ liệu `Total Cases`, làm tương tự với trường `Total Deaths`, lấy 5 quốc gia có số lượng nhiễm cao nhất.

```
deaths_by_country = covid_df.groupby('Country')['Total Deaths'].sum()
sorted_deaths_by_country = deaths_by_country.sort_values(ascending=False)
top_3_deaths_by_country = sorted_deaths_by_country[:3]
```

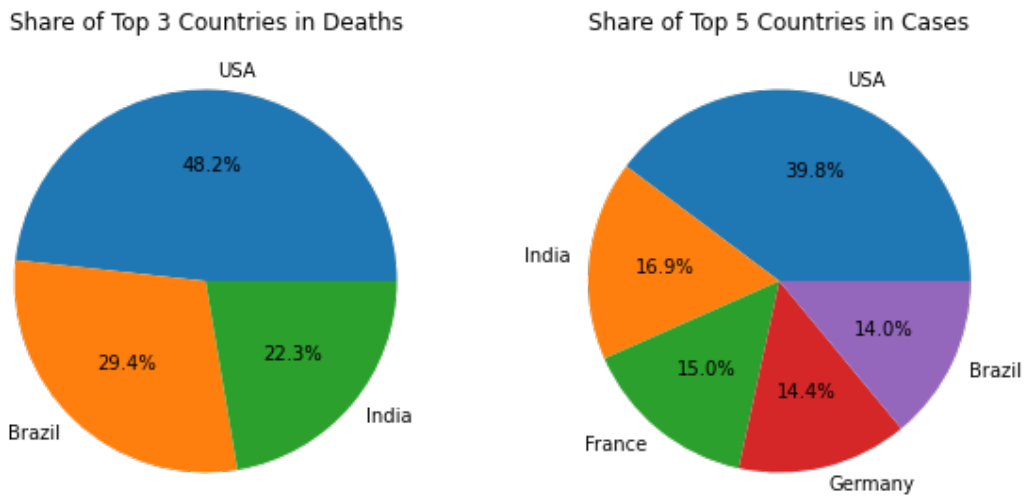
✓ 0.1s MagicPython

```
cases_by_country = covid_df.groupby('Country')['Total Cases'].sum()
sorted_cases_by_country = cases_by_country.sort_values(ascending=False)
top_5_cases_by_country = sorted_cases_by_country[:5]
```

✓ 0.0s MagicPython

Hình 35: Thực hiện thuật toán bằng code

- Thực hiện dùng `matplotlib` để vẽ biểu đồ và Kết quả:



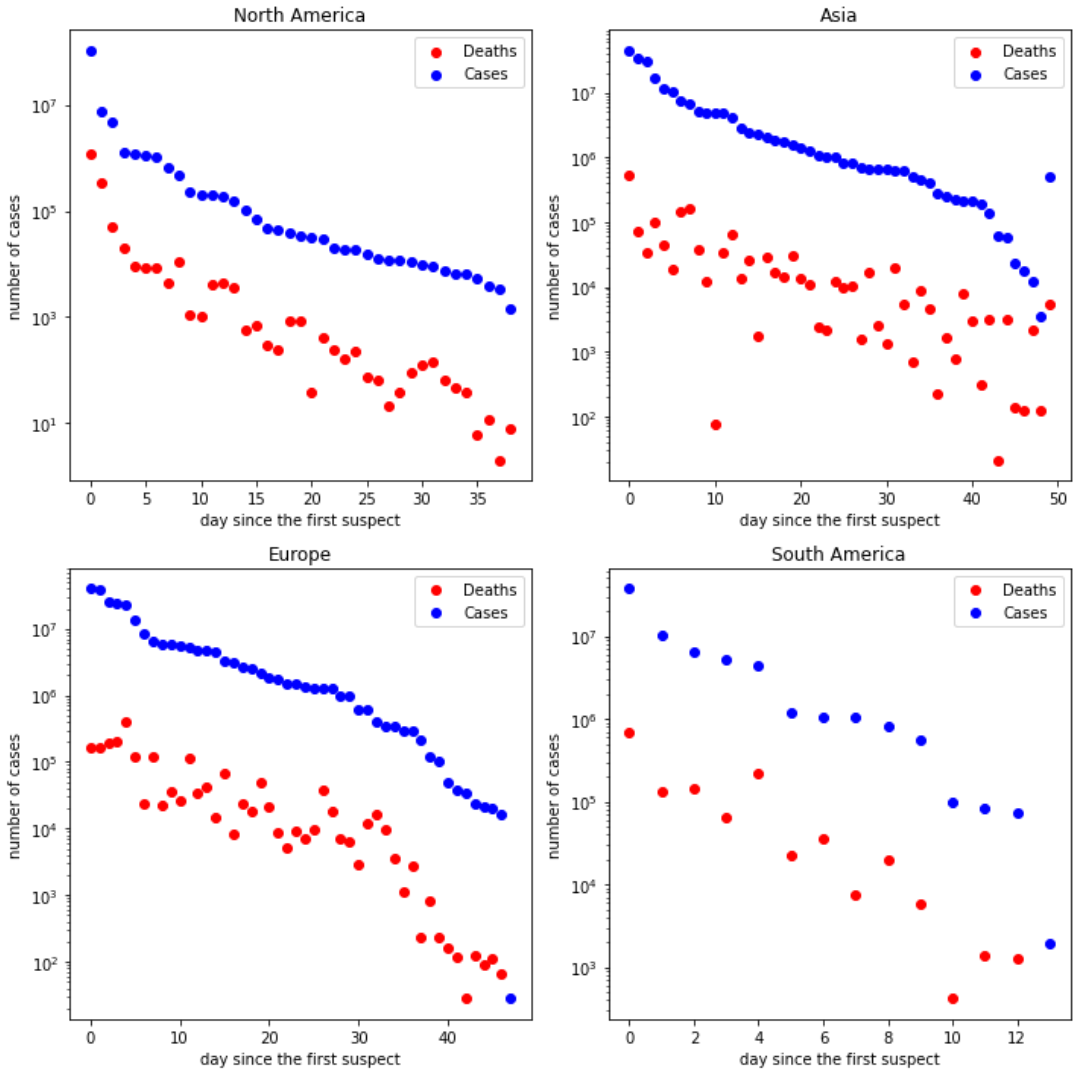
Hình 36: 2 biểu đồ pie để so sánh 2 trường dữ liệu

- ***Nhận xét biểu đồ:***
  - USA là quốc gia có số lượng tử vong nhiều nhất trong top 3. Điều này có thể được giải thích là do ban đầu Mỹ khá chủ quan trong việc phòng chống dịch bệnh Covid, Mỹ

không bắt buộc đeo khẩu trang ở nơi công cộng thời đầu dịch. Hậu quả là số ca mắc tăng do miễn dịch cộng đồng không hiệu quả.
  - Quan hệ với các trường dữ liệu khác: Nhìn tổng quan biểu đồ, số ca nhiễm của Mỹ, Ấn Độ và Brazil cũng có tổng số ca nhiễm cao trong top 5 nên cũng dễ hiểu tại sao 3 nước lại có tỷ lệ tử vong cao có nhiều yếu tố nữa nhưng đa phần là do quá tải của hệ thống y tế.

2. **Vậy hậu đỉnh dịch thì bây giờ tổng số ca tử vong có tỉ lệ thuận với tổng số ca nhiễm không?**

   - ***Trường dữ liệu được trực quan***: `Total Cases`, `Total Deaths`.
   - ***Biểu đồ sử dụng***: Scatter chart.
   - ***Tính phù hợp của biểu đồ:*** Scatter là biểu đồ thông dụng nhất khi quan sát mối quan hệ của hai biến, đặc biệt là xem xét sự tương quan giữa hai biến đó. Với lý do như vậy nên sử dụng scatter plot cho câu hỏi xem xét liệu 2 biến `Total Cases` và `Total Deaths` có sự tương quan với nhau hay không là phù hợp.
   - ***Mục đích của câu hỏi:*** Nhằm xem xét sự tương quan giữa tổng số ca nhiễm với tổng số ca tử vong theo châu lục. Nếu ca nhiễm giảm thì ca tử vong phải giảm, và ngược lại. Phân biệt theo châu lục vì mỗi châu lục có những điểm khác nhau về tập tính văn hóa nên phân biệt châu lục sẽ thấy được tương quan rõ nhất.
   - ***Giải thích cách làm/thuật toán:***
     - Tạo ra một mảng chứa các giá trị duy nhất của cột `Continent` trong dataframe `covid_df` bằng đoạn code: `df = covid_df["Continent"].unique()`.
     - Thực hiện vòng lặp plot từng biểu đồ bằng `matplotlib` và kết quả:
   - ***Nhận xét các biểu đồ:***
     - Qua biểu đồ, nhóm nhận thấy đa số tổng ca nhiễm ở các châu lục để giảm dần theo một đường đi xuống và tổng số ca tử vong cũng giảm thấy rõ ràng. Chứng minh rằng tình hình dịch đã được kiểm soát.
     - Nhìn chung, các quốc gia trên thế giới kiểm soát tình hình dịch rất tốt và ổn định, số ca mắc mới giảm dần, số ca phục hồi tăng lên.
     - Mối quan hệ nhân quả: Mối quan hệ suy ra ở đây là nếu các biện pháp phòng chống dịch bệnh được hiệu quả thì số ca tử vong giảm khi số ca mắc bệnh giảm.



Hình 37: Phân bố dữ liệu giữa tổng số ca nhiễm và tổng ca tử vong

3. Số ca nhiễm, chết và hồi phục được phân bố như thế nào theo từng châu lục

   - ***Trường dữ liệu được trực quan:*** `Total Cases`, `Total Deaths`, `Total Recovered`
   - ***Biểu đồ sử dụng:*** Multiple bar chart.
   - ***Tính phù hợp của biểu đồ:*** Để có thể theo dõi được nhiều trường dữ liệu trên các châu lục cùng 1 lúc, dùng multiple bar chart để có cái nhìn tổng quan, dễ đối chiếu các số liệu với nhau.
   - ***Mục đích của câu hỏi:*** Kiểm tra tình hình phân bố số ca nhiễm, chết và hồi phục theo từng châu lục.
   - ***Giải thích cách làm/thuật toán:***
     - Gom nhóm các quốc gia trong cùng châu lục rồi lấy tổng số ca phục hồi.
     - Làm tương tự với tổng số ca nhiễm và tổng số ca chết bằng việc gom nhóm.
     - Sau đó ứng với từng trường dữ liệu, plot ra biểu đồ cột tương ứng với châu lục rồi show giá trị chính xác của dữ liệu cho từng cột
     - Cuối cùng, scale lại các giá trị trên trục y bằng hàm `log`

```
fig, ax = plt.subplots(1,1, figsize = (20, 8))
width = 0.3

recover = pd.DataFrame(covid_df.groupby(["Continent"])["Total Recovered"].agg(sum))
cases = pd.DataFrame(covid_df.groupby(["Continent"])["Total Cases"].agg(sum))
deaths = pd.DataFrame(covid_df.groupby(["Continent"])["Total Deaths"].agg(sum))

continent = cases["Total Cases"].keys()
x = np.arange(len(continent))
ax.set_xticks(x)
ax.set_xticklabels(continent)

ax.set_title("Tổng số ca nhiễm, chết và hồi phục theo châu lục")
rect1 = ax.bar(
    x - width,
    recover["Total Recovered"].values,
    width = width,
    label = 'Total Recovered',
    edgecolor = "black",
    color = "#1fe074",
)

ax.bar_label(
    ax.containers[0],
    fontsize=10,
    labels=[format_y_axis(x) for x in recover["Total Recovered"].values],
    label_type="edge",
)
```

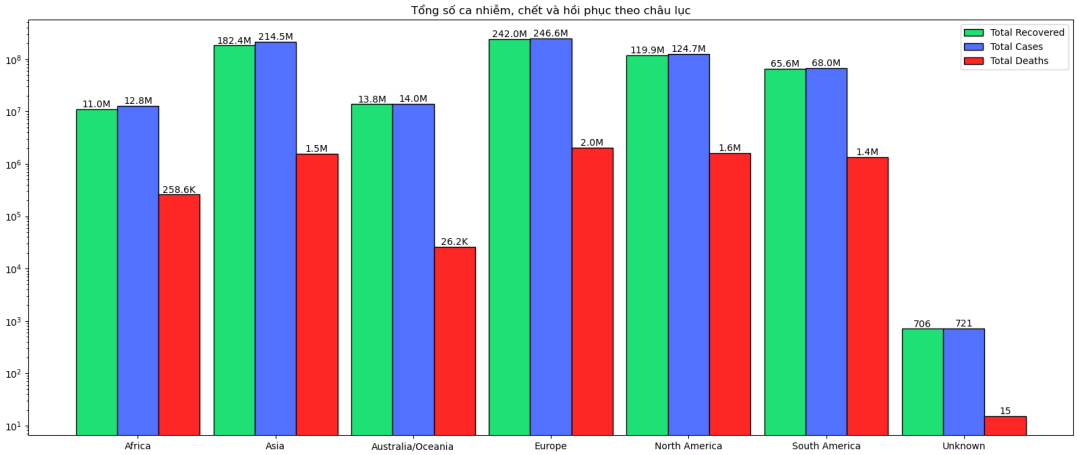
Hình 38: Code thuật toán trực quan

- Để scale các giá trị lại trên trục y thì ta dùng hàm `yscale` và scale theo hàm `log`:

```
62  ax.legend(["Total Recovered", "Total Cases", "Total Deaths"])
63  ax.yaxis.set_major_formatter(ticker.FuncFormatter(format_y_axis))
64  plt.yscale('log')
```

Hình 39: Scale giá trị

- Cuối cùng ta được biểu đồ như sau:



Hình 40: Biểu đồ tổng số ca nhiễm, chết và hồi phục theo châu lục

- **Nhận xét biểu đồ**:
  - Từ biểu đồ, ta có thể thấy được Châu Âu, dùng là lục địa có diện tích tương đối nhỏ và số dân tương đối ít, nhưng lại cao nhất về số người mắc covid và số người chết bởi covid (với gần 247 triệu ca nhiễm và 2 triệu người chết)
  - Tuy nhiên Châu Âu cũng dẫn đầu về số người hồi phục cao nhất với 242 triệu người, mặt khác, Châu Á có số ca nhiễm cao hơn Bắc Mỹ tuy nhiên số người chết ở Châu Á lại ít hơn Bắc Mỹ (1,6 triệu người chết ở Bắc Mỹ và 1,5 triệu người chết ở Châu Á)
  - Tóm lại, biểu đồ này giúp ta có cái nhìn khách quan hơn về tình hình dịch ở từng châu lục từ đó đánh giá được tình hình phòng chống dịch covid ở mỗi châu lục.

2.3.6 Mối quan hệ xoay quanh các trường dữ liệu khác:

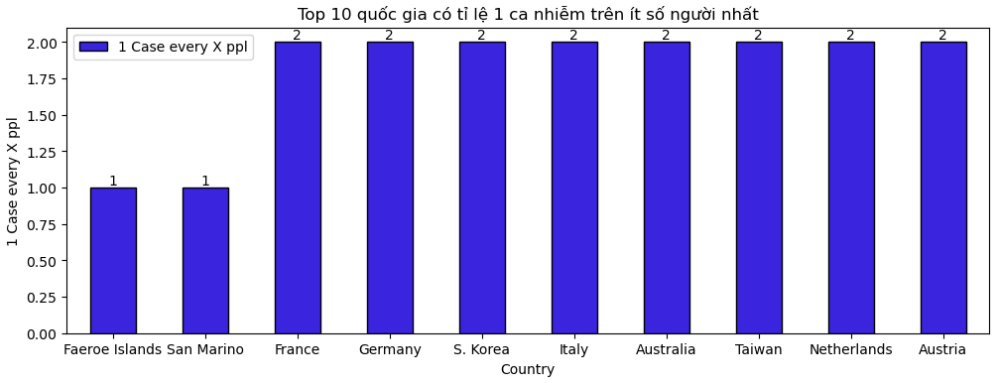
1. Các quốc gia có tỉ lệ 1 ca nhiễm trên ít số người nhất:
   - ***Trường dữ liệu được trực quan***: `1 Case every X ppl`
   - ***Biểu đồ sử dụng***: Bar chart.
   - ***Tính phù hợp của biểu đồ***: Để dễ dàng so sánh top 10 quốc gia có tỉ lệ 1 ca nhiễm trên ít số người nhất, bar chart là một biểu đồ phù hợp vì việc so sánh được thể hiện rõ ở chiều cao các cột tương ứng với các quốc gia (các giá trị rời rạc).

- ***Mục đích của câu hỏi:*** Kiểm tra tỉ lệ nhiễm bệnh cao nhất của các quốc gia để rút ra được lý do và cảnh báo cho các quốc gia này.
- ***Giải thích cách làm/thuật toán***:
  - Loại bỏ những quốc gia có trường dữ liệu `1 Case every X ppl` bằng 0 (do không có số liệu) rồi lấy ra top 10 quốc gia có giá trị thấp nhất xong dùng thư viện `matplotlib` để trực quan biểu đồ.

```
fig, ax = plt.subplots()
ax.set_title("Top 10 quốc gia có tỉ lệ 1 ca nhiễm trên ít số người nhất")
covid_df.drop(covid_df[covid_df['1 Case every X ppl'] == 0].index)[['Country',
    n=10,
    columns=['1 Case every X ppl']
).plot(
    kind="bar",
    color=(0.23, 0.14, 0.87),
    figsize=(12, 4),
    x='Country',
    xlabel="Country",
    ylabel='1 Case every X ppl',
    fontsize=10, ax=ax,
    edgecolor="black"
)

ax.bar_label(
    ax.containers[0],
    fontsize=10,
)

ax.xaxis.set_tick_params(rotation=0)
plt.show()
```

Hình 41: Code thuật toán trực quan

  - Kết quả:



Hình 42: Top 10 quốc gia có tỷ lệ ca nhiễm trên ít số người nhất

- ***Nhận xét biểu đồ***:
  - Faeroe Islands và San Marino là 2 quốc gia có tỉ lệ nhiễm bệnh cao nhất khi trung bình cứ 1 người thì người đó sẽ nhiễm covid.
  - Ngoài ra, các nước nằm trong top 10 tỉ lệ nhiễm bệnh cao nhất này cũng không khá hơn khi trung bình cứ 2 người thì có 1 người nhiễm covid.
  - Đặc biệt hơn, các quốc gia này là các nước phát triển và như Đức, Pháp, Hàn Quốc, Ý, ...
  - Điều này cho thấy đa số các nước phát triển sẽ có mật độ dân cư cao, dễ tập trung đông người, từ đó gây ra lây lan dịch bệnh nhanh.
  - Ta sẽ xem các quốc gia có tỉ lệ chết trên số người ít nhất để xem nó có mối quan hệ gì đến với tỉ lệ nhiễm bệnh hay không.

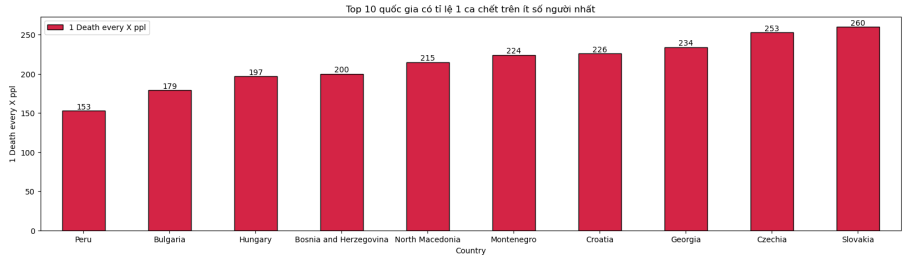
2. Top 10 quốc gia có tỉ lệ 1 ca chết trên ít số người nhất

   - ***Trường dữ liệu được trực quan:*** `1 Death every X ppl`
   - ***Biểu đồ sử dụng:*** Bar chart.
   - ***Tính phù hợp của biểu đồ:*** Để dễ dàng so sánh top 10 quốc gia có tỉ lệ 1 ca chết trên ít số người nhất, bar chart là một biểu đồ phù hợp vì việc so sánh được thể hiện rõ ở chiều cao các cột tương ứng với các quốc gia (các giá trị rời rạc).
   - ***Mục đích của câu hỏi:*** Kiểm tra tỉ lệ chết cao nhất để xem xét mối quan hệ với tỉ lệ nhiễm bệnh.
   - ***Giải thích cách làm/thuật toán***:
     - Loại bỏ những quốc gia có trường dữ liệu `1 Death every X ppl` bằng 0 (do không có số liệu) xong lấy ra top 10 quốc gia có giá trị thấp nhất rồi dùng `matplotlib` để trực quan:

```python
fig, ax = plt.subplots()
ax.set_title("Top 10 quốc gia có tỉ lệ 1 ca chết trên ít số người nhất")
covid_df.drop(covid_df[covid_df['1 Death every X ppl'] == 0].index)[['Country', '1 Death every X ppl']].nsmallest(
    n=10,
    columns=['1 Death every X ppl']
).plot(
    kind="bar",
    color=(0.83, 0.14, 0.27),
    figsize=(20, 5),
    x='Country',
    xlabel="Country",
    ylabel='1 Death every X ppl',
    fontsize=10, ax=ax,
    edgecolor="black"
)

ax.bar_label(
    ax.containers[0],
    fontsize=10,
)

ax.xaxis.set_tick_params(rotation=0)
plt.show()
```

Hình 43: Code trực quan biểu đồ

- Kết quả:



Hình 44: Biểu đồ thể hiện top 10 quốc gia có tỉ lệ chết trên ít số người nhất

- ***Nhận xét biểu đồ***:
  - Từ biểu đồ, ta có thể thấy được tỉ lệ chết do covid trên số người ở các quốc gia này khá thấp, trải dài từ khoảng 150 người đến hơn 250 người.
  - Tuy nhiên, các quốc gia nằm trong top tỉ lệ chết trên số người lại không phải là các quốc gia phát triển, mà đó là các nước như Peru, Bulgaria, Montenegro,… Nhìn chung là các quốc gia đang phát triển.
  - Điều đó cho thấy được rằng các quốc gia phát triển thì sẽ có tỉ lệ mắc bệnh cao hơn nhưng ở các quốc gia kém phát triển hơn, do điều kiện phòng chống dịch chưa được tốt, nên tỉ lệ chết do covid của những quốc gia này cao hơn hẳn.