TRỰC QUAN HÓA DỮ LIỆU

REPORT LAB01(NHÓM): MỐI QUAN HỆ TRONG DỮ LIỆU COVID

Giáo viên hướng dẫn: LÊ NGỌC THÀNH

Nhóm sinh viên thực hiện: Nhóm 03

Khoa Công nghệ thông tin

Đại học Khoa học tự nhiên TP HCM

MỤC LỤC

Yêu cầu của Report 01(nhóm): Mối quan hệ trong dữ liệu COVID

Project được thực hiện theo nhóm. Thời gian và cách thức nộp, xem trên Moodle.

Nội dung cần nộp:

- Báo cáo trình bày trong file .doc/.docx/pdf chứa:
  - o Thông tin nhóm: tên nhóm, mssv…
  - o Mức độ hoàn thành tổng thể của mỗi yêu cầu.
  - o Mức độ hoàn thành của từng thành viên.
  - o Chi tiết thuật toán, chạy ví dụ, nhận xét. Khuyến khích trình bày đơn giản, có hình minh họa.
- Souce code kèm hướng dẫn chạy nếu thực hiện trong môi trường khác Jupyter Notebook hoặc python gốc.
- Dataset được lấy gốc theo từng ngày, nếu có modify thì tạo file riêng.
- Ngôn ngữ lập trình bắt buộc: Python
  - o Cho phép sử dụng các thư viện đã được giới thiệu trong lý thuyết

1 Thông tin nhóm và phân công công việc

NHÓM 03

MSSV	Họ Tên	Đóng góp thực hiện project	%hoàn thành
18120374	Nguyễn Minh Hiếu	1./ Code các hàm crawl dữ liệu từ trang web Worldometer 2./ Lưu dữ liệu vào file .txt và chuyển dữ liệu từ file txt vào file csv 3./ Viết Code xử lý dữ liệu: - Viết code lấy dữ liệu. - Viết code vẽ tất cả các biểu đồ dữ liệu. 4./ Chọn trường dữ liệu để trực quan bằng biểu đồ và chọn các trường dữ liệu để thể hiện mối quan hệ: - Mối quan hệ giữa hai trường dữ liệu tổng số ca nhiễm bệnh (Total Cases) và tổng số ca tử vong vì covid (TotalDeaths) - Mối quan hệ giữa tổng số ca đã test (Total Tests) và tổng số ca mắc bệnh (Total Cases)	100%
1612654	Trần Minh Thiện	1./ Thực hiện báo cáo: - Phần 2: Thu thập thống kê số liệu từ trang Worldometer - Phần 3: Xử lý dữ liệu đã thu được - Phần 4: Chọn trường dữ liệu để thể hiện trực quan bằng biểu đồ và giải thích tính phù hợp với tính chất của trường dữ liệu của loại biểu đồ đã chọn.	100%

		• Phần 4.1: Trường dữ liệu: Tổng số ca tử vong (TotalDeaths) • Phần 4.2: Trường dữ liệu: Tổng số ca nhiễm bệnh tại mỗi quốc gia (total cases) - Phần 5: Mối quan hệ của các trường dữ liệu và Quan hệ nhân quả của các trường dữ liệu (chứng minh quan hệ nhân quả thông qua các phép trực quan hóa dữ liệu) • Phần 5.1: Mối quan hệ giữa hai trường dữ liệu tổng số ca nhiễm bệnh (Total Cases) và tổng số ca tử vong vì covid (TotalDeaths) • Phần 5.2: Mối quan hệ giữa tổng số ca đã test (Total Tests) và tổng số ca mắc bệnh (Total Cases) • Phần 5.3: Mối quan hệ nhân quả của tỉ lệ số ca tử vong trên tổng dân số của các quốc gia (TotalDeaths/Population) • Phần 5.4: Mối quan hệ nhân quả giữa số ca đang nhiễm bệnh (ActiveCases) và số ca tử vong mới (NewDeaths) 2./ Phân tích giải thích biểu đồ các trường dữ liệu và quan hệ: - Mối quan hệ giữa tổng số ca đã test (Total Tests) và tổng số ca mắc bệnh (Total Cases) - Mối quan hệ nhân quả của tỉ lệ số ca tử vong trên tổng dân số của các quốc gia (TotalDeaths/Population)	

		- Mối quan hệ nhân quả giữa số ca đang nhiễm bệnh (ActiveCases) và số ca tử vong mới (NewDeaths)	
1612642	Tống Thị Cam Thảo	Phân tích giải thích biểu đồ các quan hệ và viết báo cáo: - Mối quan hệ giữa Tỉ lệ các ca mắc bệnh (ActiveCases), chết (Deaths) và phục hồi (Recovered) giữa các châu lục - Mối quan hệ giữa Tỉ lệ các ca dương tính (Postive) và các ca âm tính (Negative) tại các châu lục - Mối quan hệ giữa Tỉ lệ phần trăm test trên 1 triệu người giữa các châu lục	100%
1612630	Hoàng Ngọc Kim Thanh	Phân tích giải thích biểu đồ các trường dữ liệu và quan hệ, viết báo cáo: - Tổng số ca tử vong (TotalDeaths) - Tổng số ca nhiễm bệnh tại mỗi quốc gia (total cases) - Mối quan hệ giữa hai trường dữ liệu tổng số ca nhiễm bệnh (Total Cases) và tổng số ca tử vong vì covid (TotalDeaths)	100%
1712575	Hoàng Xuân Long	- Tiền xử lý dữ liệu: xử lý thô - 4.3 Trường dữ liệu: New Cases - 4.4 Trường dữ liệu: New Deaths - 5.5 Mối quan hệ: New Cases and Population	100%

2 Thu thập thống kê số liệu từ trang Worldometer (www.worldometers.info)

2.1 Thực hiện thu thập số liệu thống kê ngày 19/04/2021

- Sử dụng code python chạy trên jupyter notebook để lấy dữ liệu từ trang Worldometer (http://www.worldometers.info)
- Khai báo các thư viện cần thiết để crawl dữ liệu:

  + **pandas:** thư viện của python được sử dụng để phân tích dữ liệu

  + **numpy:** NumPy là một gói Python viết tắt của Numerical Python. Đây là thư viện cốt lõi cho scientific computing, nó chứa một đối tượng mảng n chiều mạnh mẽ, cung cấp các công cụ để tích hợp C, C ++, v.v. Nó cũng hữu ích trong đại số tuyến tính, random number capability... NumPy Array cũng có thể được sử dụng như multi-dimensional container hiệu quả cho dữ liệu chung.

  + **bs4:** Beautiful Soup là một thư viện của Python để thực hiện việc lấy dữ liệu từ các trang web bằng cách đọc các file HTML hoặc XML.

  + **selenium:** Selenium là một công cụ tự động hóa dựa trên web mã nguồn mở. Python được sử dụng với Selenium để kiểm tra... Các API Python cho phép bạn kết nối với trình duyệt thông qua Selenium. Selenium có thể gửi các lệnh Python tiêu chuẩn đến các trình duyệt khác nhau, bất chấp sự thay đổi trong thiết kế của trình duyệt.

> Sử dụng Chorme nên cần kiểm tra phiên bản và download driver: https://chromedriver.chromium.org/downloads

In [1]:
```
import pandas as pd
import numpy as np
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
```

- Các hàm để crawl dữu liệu từ địa chỉ URL:

+ Hàm extract_row: hàm thực hiện lấy dữ liệu ở từng dòng

+ Hàm request_WEB: gửi request đến địa chỉ URL để tìm và lấy thông tin từ bảng dữ liệu.

In [2]:
```
#Lấy văn bản ở dòng
def extract_row(row):
    result = []
    i = 0
    for td in row:
        i+=1
        if (i == 1):
            continue
        result.append(td.text.replace(',',''))
    return result
```

In [3]:
```
def request_WEB(browser, date = 0):
    '''browser là cái driver
       date mặc định là 0: hôm nay 1: hôm qua 2: hôm kia
    '''
    ID = "main_table_countries_today"
    if date == 1:
        ID = "main_table_countries_yesterday"
        browser.find_element_by_id('nav-yesterday-tab').click()
    if date == 2:
        ID = "main_table_countries_yesterday2"
        browser.find_element_by_id('nav-yesterday2-tab').click()

    soup=BeautifulSoup(browser.page_source,'html.parser')
    table= soup.find('table',{'id': ID})
    body = table.find('tbody')
    table_contents = body.find_all('tr')

    for row in table_contents:
        try:
            if row['style'] != 'display: none':
                value = row.find_all('td')
                result = ",".join(extract_row(value))
                the_file.write(result + '\n')
        except:
            continue
```

- Thực hiện lệnh vào trang tại địa chỉ URL:

```
In [4]:
#Vào trang
URL = 'https://www.worldometers.info/coronavirus/#countries'
browser = webdriver.Chrome(executable_path='./chromedriver.exe')
browser.maximize_window()
browser.get(URL)
```

- Thực hiện lệnh gọi mở file .txt để bắt đầu lưu lại dữ liệu.

```
In [5]:
name = "COVID_4-24.txt"
the_file = open(name,'a',encoding='utf-8')
```

- Lưu dữ liệu vào file name (file .txt)

Lưu dữ liệu vào name.txt

```
In [6]:
the_file.write("Country,Total Cases,New Cases,TotalDeaths,NewDeaths,TotalRecovered,ActiveCases,Serious,Tot Cases/1M pop,Death
```

- Gọi hàm request_WEB để crawl và lưu data vào file .txt

```
In [7]:
request_WEB(browser, date = 1)
```

2.2 Chuyển dữ liệu đã lưu từ file .txt vào file .csv

- Thực hiện chuyển dữ liệu từ file .txt

3 Xử lý dữ liệu đã thu thập được.

Ta có dữ liệu như sau:

	Country	Total Cases	New Cases	TotalDeaths	NewDeaths	TotalRecovered	ActiveCases	Serious	Iot Cases/1M pop	Deaths/1M pop	Total Tests	Tests/1M pop	Popu
0	USA	32476153	51653.0	581573	498.0	25043463	6851117	9815.0	97655.0	1749.0	430886619.0	1295672.0	3325
1	India	15314714	256947.0	180550	1757.0	13103220	2030944	8944.0	11011.0	130.0	267894549.0	192610.0	13908
2	Brazil	13977713	34642.0	375049	1607.0	12460712	1141952	8318.0	65387.0	1754.0	28600000.0	133789.0	2137
3	France	5296822	7296.0	101222	489.0	4151289	1044311	5970.0	81004.0	1548.0	72613536.0	1110478.0	653
4	Russia	4710690	8589.0	105928	346.0	4333598	271164	2300.0	32268.0	726.0	126000000.0	863104.0	1459

3.1 Tiền xử lý dữ liệu.

- Đầu tiên mục đích của nhóm là thực hiện công việc trên toán bộ các nước, vậy thì các dòng mà trường **Population** bị trống thì không phù hợp. Có 2 dòng là *MS Zaandam, Diamond Princess* sẽ bị bỏ.

Country	Total Cases	New Cases	TotalDeaths	NewDeaths	TotalRecovered	ActiveCases	Serious	Iot Cases/1M pop	Deaths/1M pop	Total Tests	Tests/1M pop	Population	Continent
Diamond Princess	712	NaN	13	NaN	699	0	NaN	NaN	NaN	NaN	NaN		NaN
MS Zaandam	9	NaN	2	NaN	7	0	NaN	NaN	NaN	NaN	NaN		NaN

- Một số **ô bị trống**, coi như giá trị tại đó bằng **0**.
- Cuối cùng kiểm tra kiểu dữ liệu trên các cột thì thấy có vấn đề với các cột kiểu **float** và **object** vì khi kiểm tra tất cả đều là **số nguyên**.

Thực hiện chuyển:

```
ds = ['Total Cases','New Cases','TotalDeaths','NewDeaths','TotalRecovered','ActiveCases','Serious','Tot Cases/1M pop','Deaths/1
data[ds] = data[ds].astype('int64')
```

```
[3]: data.dtypes

[3]: Country              object
     Total Cases           int64
     New Cases           float64
     TotalDeaths          object
     NewDeaths           float64
     TotalRecovered        int64
     ActiveCases           int64
     Serious             float64
     Tot Cases/1M pop    float64
     Deaths/1M pop       float64
     Total Tests         float64
     Tests/1M pop        float64
     Population           object
     Continent            object
     dtype: object
```

→

```
Country              object
Total Cases           int64
New Cases             int64
TotalDeaths           int64
NewDeaths             int64
TotalRecovered        int64
ActiveCases           int64
Serious               int64
Tot Cases/1M pop      int64
Deaths/1M pop         int64
Total Tests           int64
Tests/1M pop          int64
Population            int64
Continent            object
dtype: object
```

3.2 Sử dụng code python trên jupyter notebook để thực hiện xử lý dữ liệu và vẽ biểu để trực quan hóa dữ liệu

- Sử dụng các thư viện để thực hiện việc xử lý dữ liệu vẽ biểu đồ để trực quan hóa dữ liệu:

  + **matplotlib:** là một thư viện vẽ đồ thị rất mạnh mẽ hữu ích cho những người làm việc với Python và NumPy. **pyplot** là một module của matplotlib: cung cấp giao diện như MATLAB.

```
In [2]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
```

- Thực hiện đọc file .csv lưa data đã crawl về, đọc file .csv để phân tích và xử lý dữ liệu:

```
In [3]: data = pd.read_csv('COVID_4-19.csv')
        data.head()
```

4 Chọn trường dữ liệu để thể hiện trực quan bằng biểu đồ và giải thích tính phù hợp với tính chất của trường dữ liệu của loại biểu đồ đã chọn.

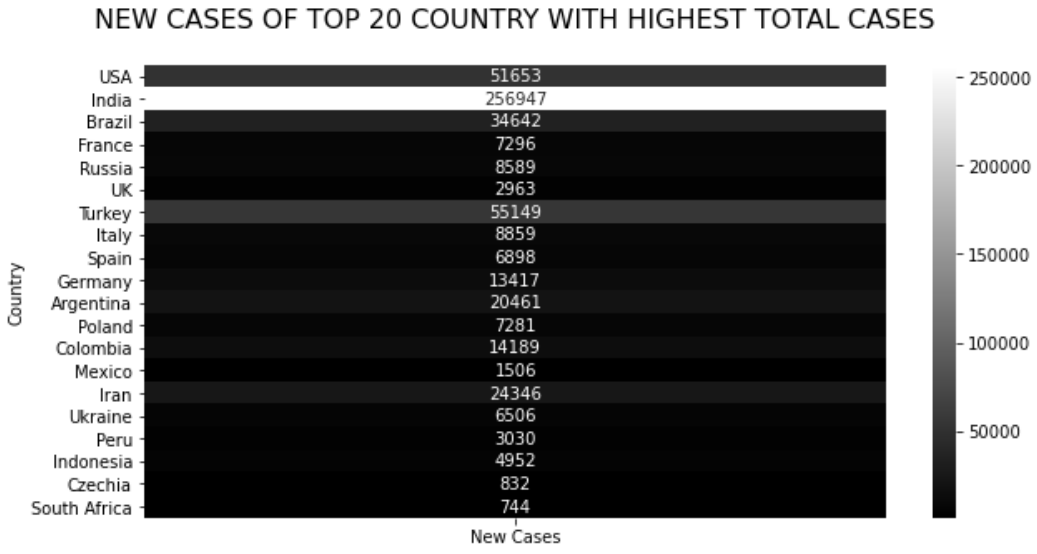
4.1 Trường dữ liệu: New Case:

- Data bao gồm các quốc gia sắp xếp theo tổng số ca bệnh và 1 trường số ca bệnh mới.

4.1.1 Trực quan dữ liệu bằng biểu đồ

```
[8]: data_newcases = data.head(20)
     data_newcases = data_newcases.set_index('Country')
     plt.figure(figsize = (10,5))
     plt.title('NEW CASES OF TOP 20 COUNTRY WITH HIGHEST TOTAL CASES',y = 1.06,fontsize = '16')
     sns.heatmap(data_newcases[['New Cases']],annot=True,cmap = 'gray',fmt='d')
```

```
[8]: <AxesSubplot:title={'center':'NEW CASES OF TOP 20 COUNTRY WITH HIGHEST TOTAL CASES'}, ylabel='Country'>
```



4.1.2 Giải thích:

- *Màu càng trắng thì tại quốc gia đó có số lượng ca nhiễm mới càng cao. Việc lựa chọn màu trắng - đen muốn thể hiện sự tang tóc của tình trạng nước đó.*

4.1.3 Ý nghĩa hợp lý sau khi dữ liệu được trực quan

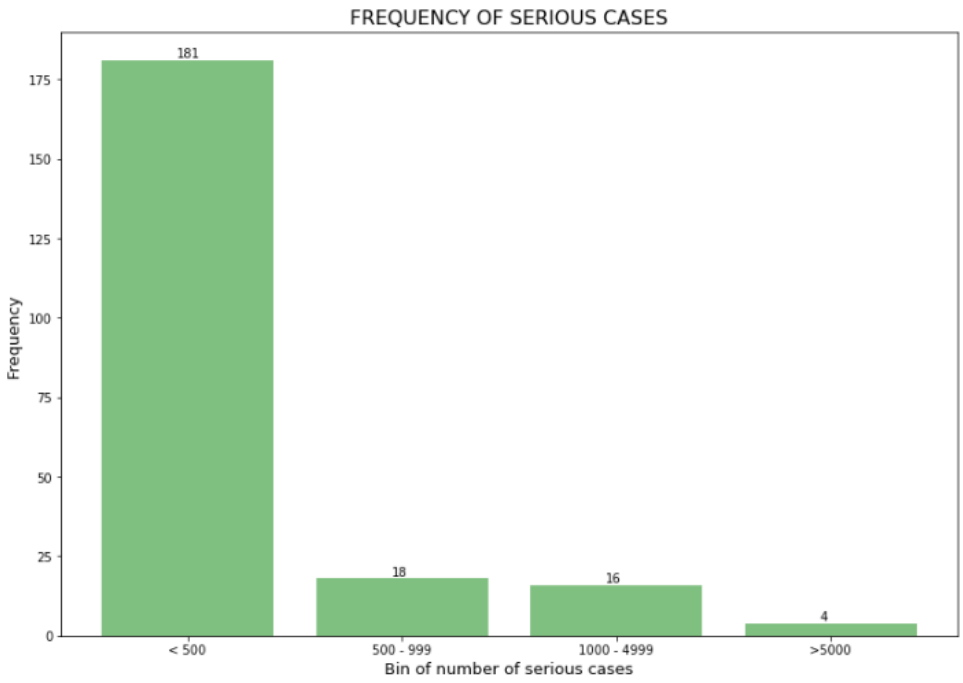
- *Nhanh chóng kiểm tra xem hiện giờ các nước có tổng số ca nhiễm lớn có thể khống chế được dịch chưa. (VD: Màu trắng ở Ấn Độ cho thấy chính quyền đang gặp khó khan bởi dịch bệnh).*

4.2 Trường dữ liệu: Serious

- Data bao số ca nghiêm trọng và các nước.

4.2.1 Trực quan dữ liệu bằng biểu đồ

```
df = data[['Serious']]

data_fre = df.apply(pd.Series.value_counts, bins=[0,500,1000,5000,10000])
x = ['< 500','500 - 999','1000 - 4999','>5000']
y = list(data_fre['Serious'].values)

plt.figure(figsize = (13,9))
plt.bar(x,y,color = 'green',alpha = 0.5)

x_pos = -0.05
for value in y:
    plt.text(x_pos, value + 1, value)
    x_pos +=1

plt.xlabel("Bin of number of serious cases",fontsize = 13)
plt.ylabel("Frequency ",fontsize = 13)
plt.title("FREQUENCY OF SERIOUS CASES",fontsize = 16)
plt.show()
```



4.2.2 Giải thích:

- Chia giá trị của Serious thành các bins tượng trưng cho ít, trung bình, hơi cao, cao. Rồi thực hiện đếm theo bin chia sẵn đó.

4.2.3 Ý nghĩa hợp lý sau khi dữ liệu được trực quan

- Xem qua tình trạng các ca nghiêm trọng trên các nước. Đa số các nước có số lượng chỉ dưới 500 ca, rất ít nước có số ca nghiêm trọng ở mức cao (hơn 5000 ca).

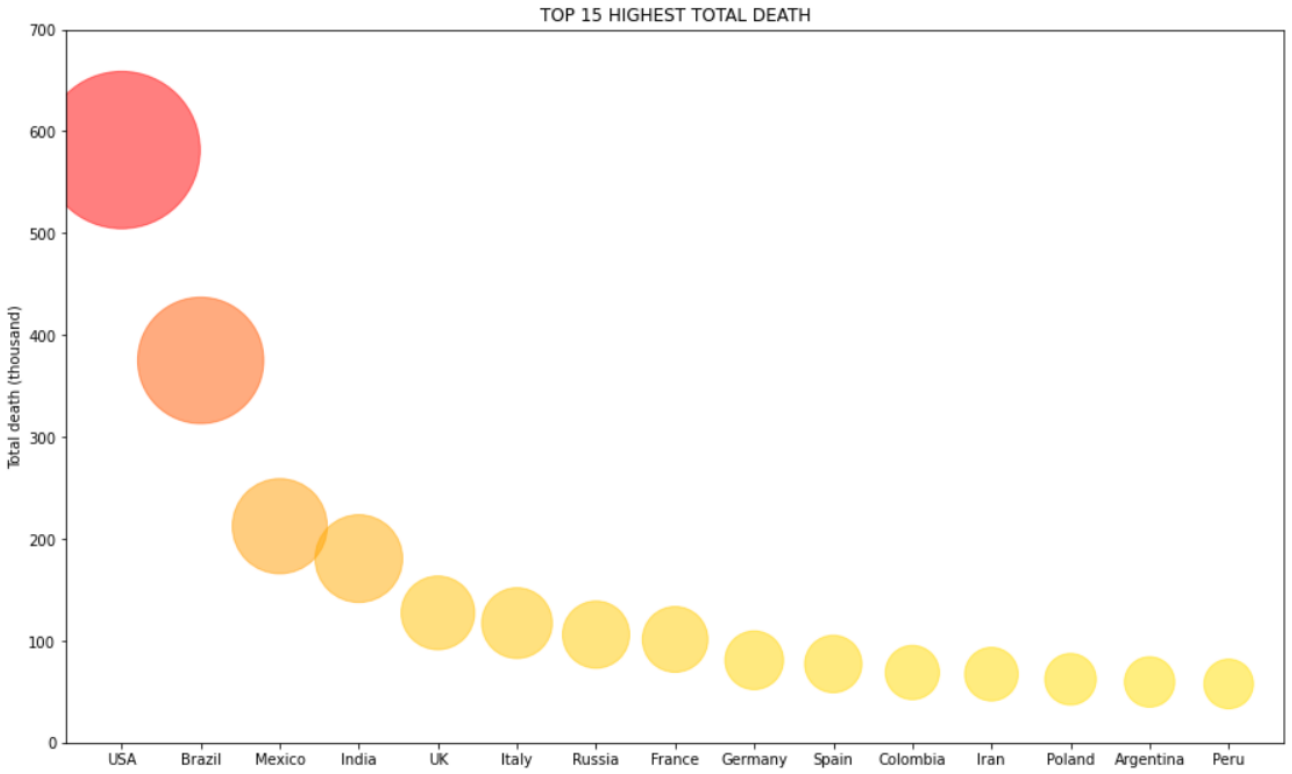
4.3 Trường dữ liệu: Tổng số ca tử vong (TotalDeaths)

- Trường dữ liệu cần trực quan là tổng số ca tử vong của 15 quốc gia có nhiều ca tử vong nhất.

4.3.1 Trực quan dữ liệu bằng biểu đồ

In [10]:
```
top15_deaths = data1[['Country','TotalDeaths']]
top15_deaths.sort_values(by = 'TotalDeaths', ascending = False,ignore_index=True, inplace = True)
top15_deaths = top15_deaths.head(15)

names = top15_deaths['Country']
values = top15_deaths['TotalDeaths']

color = []
s = 0
for i in range(len(names)-1):
        color.append((1,s/600000,0))
        s += (values[i] - values[i+1])
color.append((1,s/600000,0))
plt.figure(figsize=(15, 9))
plt.ylim([0,700])
plt.scatter( x = names, y = values/1000,
            s = values/50, c = color, alpha=0.5)
plt.ylabel('Total death (thousand)' )
plt.title('TOP 15 HIGHEST TOTAL DEATH')
plt.show()
```

```
<ipython-input-10-502278918aa2>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  top15_deaths.sort_values(by = 'TotalDeaths', ascending = False,ignore_index=True, inplace = True)
```



- *Trực quan dữ liệu tổng số ca tử vong tại 15 quốc gia có số ca tử vong cao nhất bằng biểu đồ phân tán với các điểm có kích thước dựa trên định lượng số ca tử vong, và màu của các điểm được thể hiện bằng màu của biểu đồ nhiệt.*

4.3.2 Giải thích:

- *Lựa chọn biểu đồ điểm để thể hiện được ví trí cao thấp của số liệu ca tử vong trong không gian với trục y là số lượng ca tử vong và trục x là các quốc gia. Điểm nào cao thì tại quốc gia đó có số lượng ca tử vong cao hơn các quốc gia còn lại.*
- *Lựa chọn việc thể hiện kích thước của các điểm dựa trên định lượng số lượng ca tử vong để có thể thể hiện rõ việc nơi nào có kích thước lớn hơn thì số ca tử vong cao hơn.*
- *Do đây là số liệu về tổng các ca tử vong vì vậy để biểu thị sự báo động ta lựa chọn việc tô màu các điểm dữ liệu theo biểu đồ nhiệt để dễ nhận thấy được nơi có màu càng đỏ càng đậm thì nơi đó có số ca tử vong cao hơn.*

4.3.3 Ý nghĩa hợp lý sau khi dữ liệu được trực quan

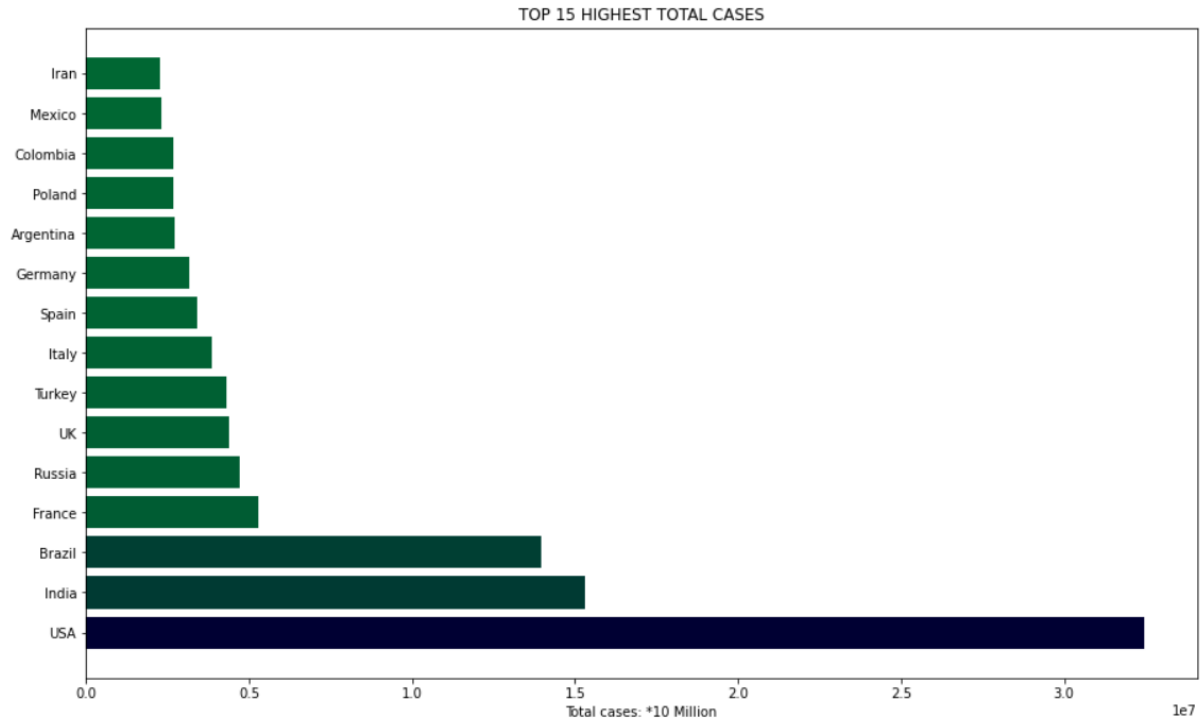
- *Việc kết hợp cả ba yếu tố biểu đồ phân tán với các điểm có kích thước tùy theo định lượng số ca tử vong và có màu theo màu biểu đồ nhiệt nhằm để thể hiện biểu đồ tổng số ca tử vong của từng nước được rõ ràng dễ nhìn dễ nhận thấy hơn. Có thể dễ dàng phán đoán ngay khi nhìn vào biểu đồ được nơi nào có số ca tử vong cao hơn.*
- *Chỉ cần nhìn vào biểu đồ ta có thể thấy ngay được màu sắc của nơi tử vong cao với màu đậm kích thước chấm lớn và ở trên cao hơn so với những quốc gia khác.*

4.4 Trường dữ liệu: Tổng số ca nhiễm bệnh tại mỗi quốc gia (total cases)

- Trực quan bằng biểu đồ trường dữ liệu tổng số ca nhiễm bệnh tại 15 quốc gia có số ca nhiễm bệnh cao nhất.

4.4.1 Trực quan dữ liệu bằng biểu đồ

In [11]:
```
top15_case = data[['Country','Total Cases']]

top15_case = top15_case.head(15)

names = top15_case['Country']
values = top15_case['Total Cases']

c = []
s = 0
for i in range(len(names)-1):
        c.append((0,s/75_000_000,0.2))
        s += (values[i] - values[i+1])
c.append((0,s/75_000_000,0.2))

c
plt.figure(figsize=(15, 9))

plt.title('TOP 15 HIGHEST TOTAL CASES')
plt.xlabel('Total cases: *10 Million')

plt.barh(names, values,color = c)
```



- *Dữ liệu tổng số ca nhiễm bệnh của 15 quốc gia có số ca nhiễm bệnh cao nhất được trực quan bằng biểu đồ cột ngang. Và tô màu cho mỗi cột theo dạng nhiệt nơi có màu đậm hơn thì số ca nhiễm cao hơn.*

4.4.2 Giải thích

- *Đây là dữ liệu của 15 quốc gia có tổng số ca nhiễm cao nhất. vì dữ liệu này là dữ liệu riêng lẻ của từng quốc gia vì thế chọn biểu đồ cột để thể hiện dữ liệu.*
- *Chọn biểu đồ dạng cột ngang để có thể dễ dàng nhận thấy được sự khác biệt giữa các quốc gia. Do dữ liệu cột ngang khi so sánh ta sẽ nhìn từ trên xuống và mắt người dễ nhận thấy sự khác biệt của những cột được xếp chồng lên nhau hơn những cột thẳng đứng (dữ liệu cột dọc khi so sánh ta phải nhìn ngang qua)*
- *Tô màu cho từng cột theo dạng đâm nhạt cho những nơi có nhiều ca sẽ màu đậm hơn những nơi khác.*

4.4.3 Ý nghĩa hợp lý sau khi dữ liệu được trực quan

- *Khi trực quan dữ liệu tổng số ca nhiễm với dạng biểu đồ cột nằm ngang và tô màu đậm nhạt cho dữ liệu thì dữ liệu sẽ được thể hiện rõ nét dễ dàng nhận thấy và dễ dàng so sánh.*

4.4.4 Giải thích tính phù hợp

- *So sánh dữ liệu được phân loại cho một bộ dữ liệu*
- *Sử dụng biểu đồ cột để thể hiện mức độ tăng trưởng các ca tử vong do lây nhiễm Covid mới của các nước qua từng ngày*
- *Dễ dàng nhận thấy các nước có các ca tử vong mới tăng cao, mức độ càng lớn màu càng đậm -> cảnh báo nguy hiểm*

5 Mối quan hệ của các trường dữ liệu và Quan hệ nhân quả của các trường dữ liệu (chứng minh quan hệ nhân quả thông qua các phép trực quan hóa dữ liệu)

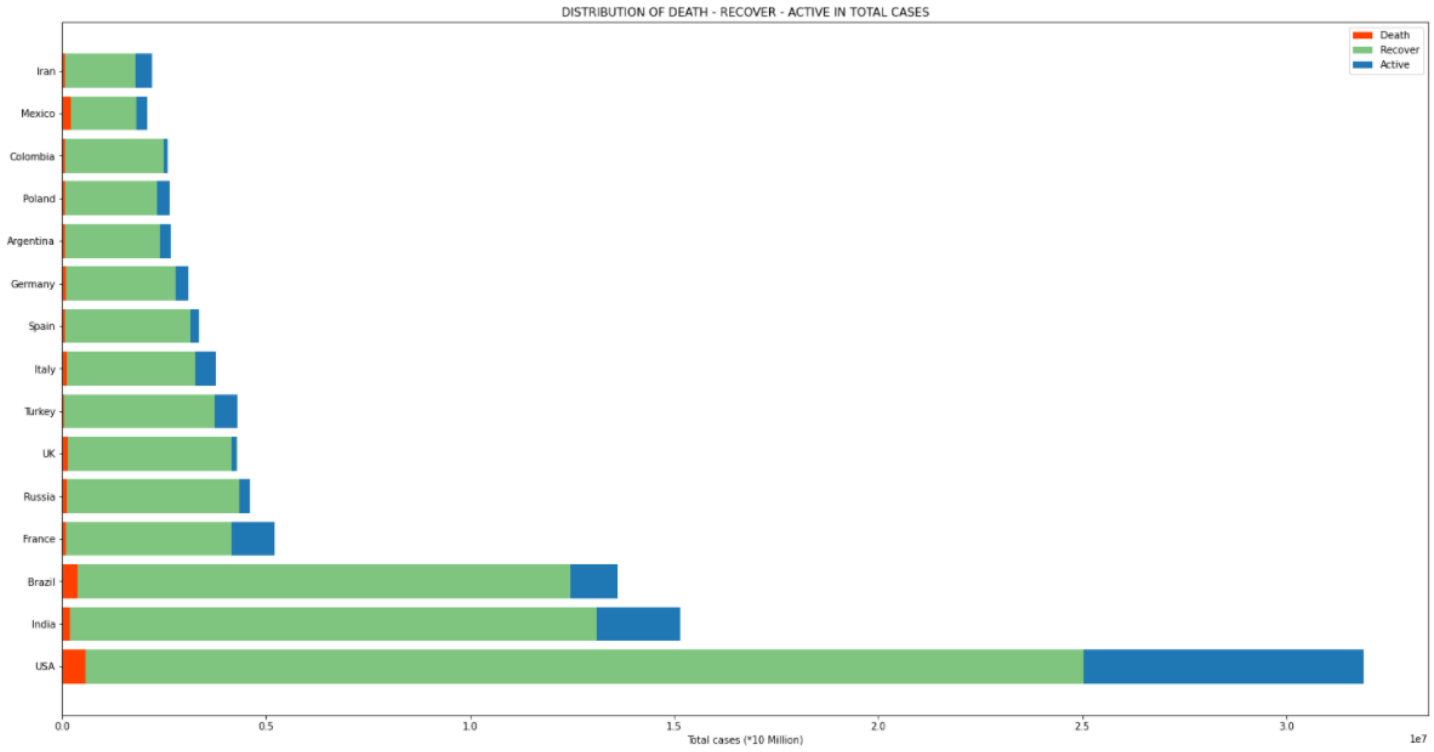
5.1 Mối quan hệ giữa hai trường dữ liệu tổng số ca nhiễm bệnh (Total Cases) và tổng số ca tử vong vì covid (TotalDeaths)

5.1.1 Giải thích mối quan hệ giữa hai trường dữ liệu

- *Sau khi thống kê 15 quốc gia có tổng số ca nhiễm bệnh cao nhất và 15 quốc gia có tổng số ca tử vong vì covid cao nhất để trực quan bằng biểu đồ ở phần 4.1 và 4.2 thì ta nhận thấy rằng trong dữ liệu thống kê được có đến 14 quốc gia nằm ở cả hai bảng thống kê. Đó là 14 quốc gia: USA, India, Brazil, France, Russia, UK, Italy, Spain, Germany, Argentina, Poland, Colombia, Mexico, Iran.*
- *Từ điều đó ta có thể suy ra được tổng số ca nhiễm sẽ có mối quan hệ tuyến tính với tổng số ca tử vong. Khi số ca nhiễm bệnh càng cao thì số ca tử vong càng cao. Quốc gia càng có nhiều ca nhiễm bệnh sẽ càng có nhiều ca tử vong.*
- *Ta còn có quan hệ:* ***Total case = Total Death + Total Recovered + Active Case***

5.1.2 *Trực quan bằng biểu đồ:*

```
In [12]: top15_distribute = data[['Country','TotalDeaths','TotalRecovered','ActiveCases']].head(15)


labels = top15_distribute['Country']
death = top15_distribute['TotalDeaths']
recover = top15_distribute['TotalRecovered']
active = top15_distribute['ActiveCases']

       # the width of the bars: can also be len(x) sequence

fig, ax = plt.subplots(figsize=(25,13))

ax.barh(labels, death , label='Death',color = (1,0.25,0))
ax.barh(labels, recover, left = death,label='Recover',color = (0,0.55,0),alpha = 0.5)
ax.barh(labels, active,  left = recover,label='Active')

ax.set_ylabel('Country')
ax.set_xlabel('Total cases (*10 Million)')
ax.set_title('DISTRIBUTION OF DEATH - RECOVER - ACTIVE IN TOTAL CASES')
ax.legend()

plt.show()
```



- *Sử dụng biểu đồ cột ngang xếp chồng để trực quan dữ liệu tổng số ca nhiễm bệnh của 15 quốc gia có số ca nhiễm bệnh cao nhất. Trong tổng số ca nhiễm bệnh thể hiện tỉ lệ số ca đã tử vong bằng màu đỏ, số ca đã bình phục bằng màu xanh lá, số ca còn đang mắc bệnh bằng màu xanh dương.*

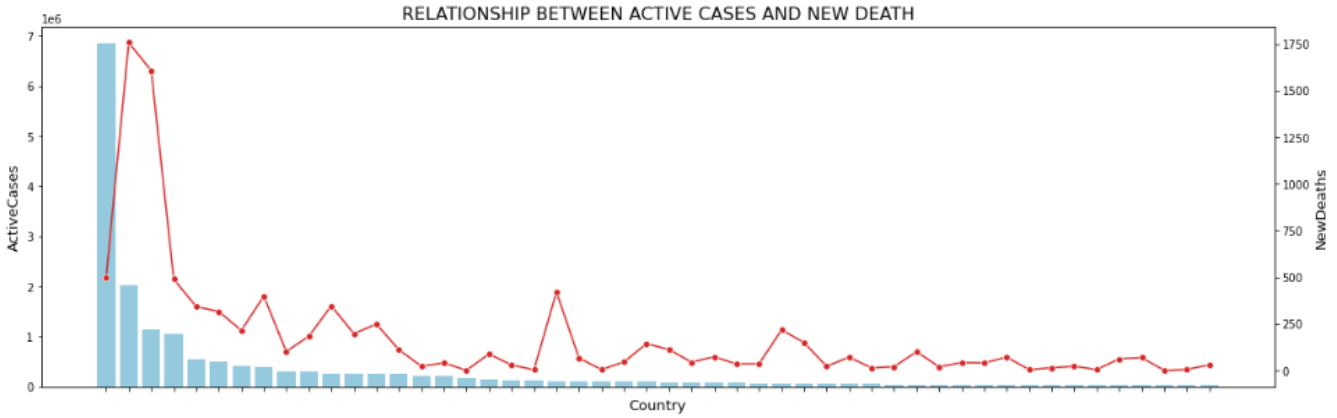
5.1.3 Ý nghĩa hợp lý sau khi dữ liệu được trực quan

- *Sau khi trực quan dữ liệu tổng số ca nhiễm bệnh và tổng số ca tử vong thì ta có thể nhận thấy được rằng quốc gia nào có số ca nhiễm bệnh càng nhiều thì số ca tử vong càng lớn, phần màu đỏ càng lớn.*
- *Trực quan hóa dữ liệu của tổng số ca tử vong và tổng số ca nhiễm trên cùng một biểu đồ giúp ta dễ nhận thấy được mối quan hệ tuyến tính giữa hai trường dữ liệu trên.*

5.2 Mối quan hệ nhân quả giữa số ca đang nhiễm bệnh (ActiveCases) và số ca tử vong mới (NewDeaths)

- Chọn 15 quốc gia có số ca đang nhiễm bệnh cao nhất (ActiveCases) để vẽ biểu đồ.

5.2.1 Trực quan dữ liệu bằng biểu đồ



- *Trực quan dữ liệu số ca đang nhiễm bệnh của nhiều quốc gia có số ca đang nhiễm bệnh cao nhất bằng biểu đồ cột và tô màu xanh dương cho số ca nhiễm bệnh*
- *Với dữ liệu số ca tử vong mới của các quốc gia trên, trực quan bằng biểu đồ đường với các điểm là các quốc gia.*

5.2.2 Giải thích mối quan hệ nhân quả giữa hai trường dữ liệu trên

- *Sau khi thống kê dữ liệu ta nhận thấy được rằng những quốc gia đang có số ca mắc bệnh cao nhất cũng là những quốc có số ca tử vong mới cao nhất.*

- *Dựa vào biểu đồ sau khi trực quan ta có thể dễ dàng nhận thấy rằng số ca tử vong mới tỉ lệ với số ca đang mắc bệnh. Hai trường dữ liệu này có quan hệ tỉ lệ thuận với nhau (mặc dù có 1 vài ngoại lệ) nghĩa là nơi nào càng có nhiều ca đang mắc bệnh thì nơi đó sẽ có càng nhiều ca tử vong mới.*

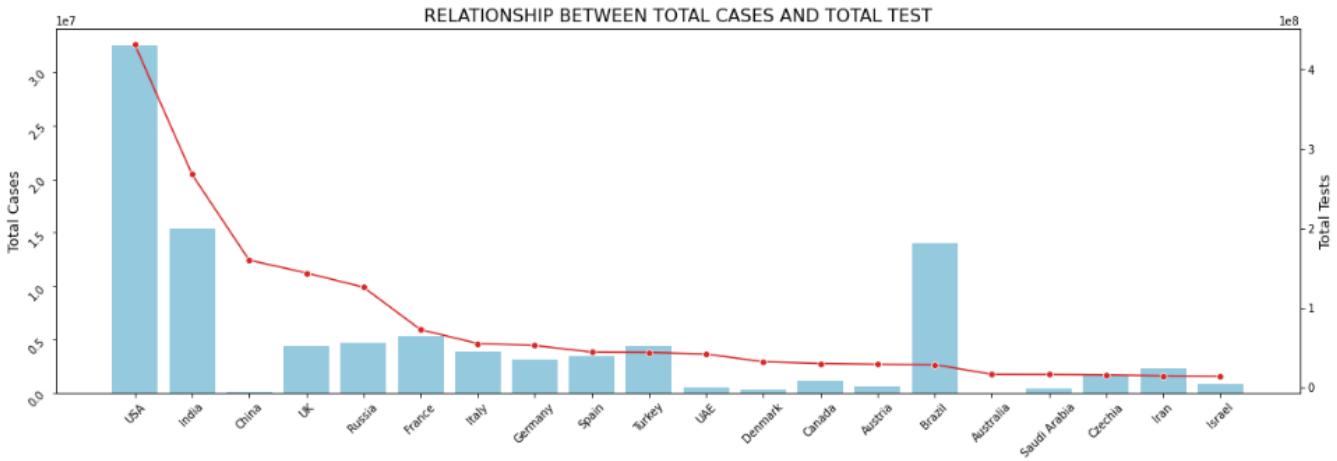
⇨ *Điều này giúp ta nhận thấy rằng đây là mối quan hệ nhân quả. Số ca tử vong mới chính là số bệnh nhân đang mắc bệnh nhưng không qua khỏi. Số ca tử vong mới (NewDeaths) là kết quả của số ca đang mắc bệnh (ActiveCases). Và hai trường dữ liệu này tỉ lệ thuận với nhau.*

5.2.3 Ý nghĩa hợp lý sau khi dữ liệu được trực quan

- *Thể hiện dữ liệu số ca tử vong bằng màu cam giúp người nhìn có thể nhận biết được ngay đâu là biểu diễn cảu số ca tử vong vì màu cam biểu thị nguy hiểm cảnh báo.*
- *Trực quan dữ liệu của các quốc gia có số ca đang mắc bệnh cao nhất cùng với dữ liệu số ca tử vong mới của quốc gia đó giúp ta thấy được sự tăng và tỉ lệ thuận của hai số liệu. Từ đó có thể thấy được mối quan hệ nhân quả của chúng.*

5.3 Mối quan hệ giữa tổng số ca đã test (Total Tests) và tổng số ca mắc bệnh (Total Cases)

5.3.1 Trực quan bằng biểu đồ:



- *Sử dụng biểu đồ cột để trực quan dữ liệu tổng số ca nhiễm bệnh của 20 quốc gia có số ca nhiễm bệnh cao nhất và sử dụng biểu đồ đường với các các điểm là các*

quốc gia để thể hiện tổng số ca đã thực hiện test. Số ca nhiễm bệnh được tô màu đỏ, các điểm thể hiện số ca đã test được tô màu xanh dương

5.3.2 Giải thích mối quan hệ giữa hai trường dữ liệu tổng số ca đã test và tổng số ca mắc bệnh

- *Sau khi thống kê và trực quan dữ liệu 20 quốc gia có số ca test cao nhất, cùng với dữ liệu số ca nhiễm bệnh của 20 quốc gia đó thì ta nhận thấy rằng dữ liệu tổng số ca test và dữ liệu tổng số ca nhiễm bệnh có quan hệ tuyến tính với nhau, khi số ca test nhiều thì số ca nhiễm bệnh cũng tăng theo.*

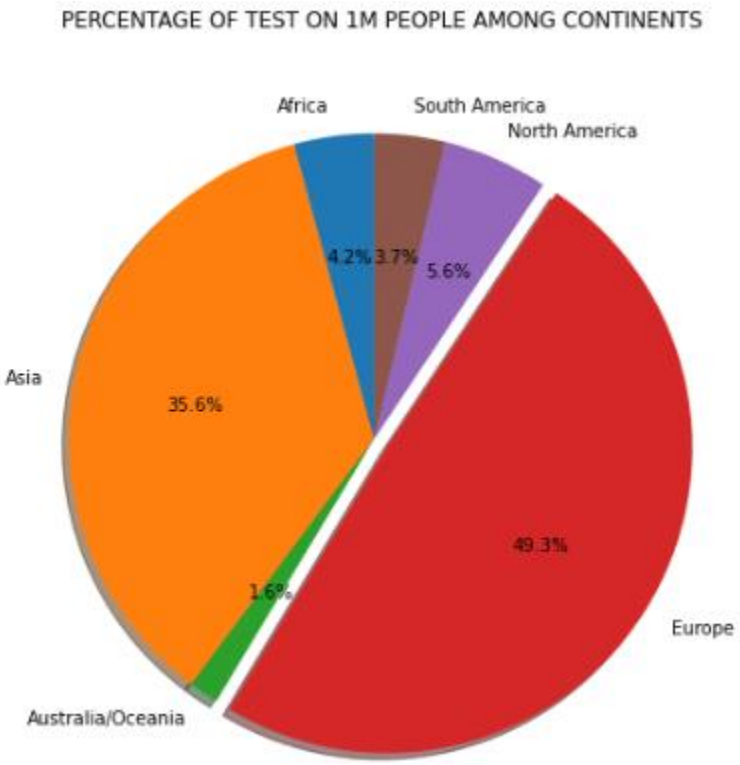
5.3.3 Ý nghĩa hợp lý sau khi dữ liệu được trực quan

- *Trực quan hóa dữ liệu của tổng số ca đã test và tổng số ca nhiễm trên cùng một biểu đồ, tô màu cam cho tổng số ca nhiễm giúp ta có thể nhìn ngay được đâu là dữ liệu tổng số ca nhiễm vì màu cam báo hiệu sự cảnh báo.*
- *Dữ liệu sau khi trực quan thành biểu đồ như trên giúp cho ta dễ dàng nhận thấy được sự tăng của cả hai trường dữ liệu dù rằng có một vài trường hợp ngoại lệ nhưng số liệu chung vẫn là cả hai cùng tăng số ca đước test tăng thì số ca nhiễm bệnh cũng tăng theo.*

5.4 Kiểm tra xem châu lục nào có tốc độ xét nghiệm (Test/1M) nhanh nhất

- Data bao gồm tổng của các trường dữ liệu mà trong đó chỉ tính của các nước có **dân số (Population) từ 1tr người trở lên.**

5.4.1 Trực quan bằng biểu đồ:



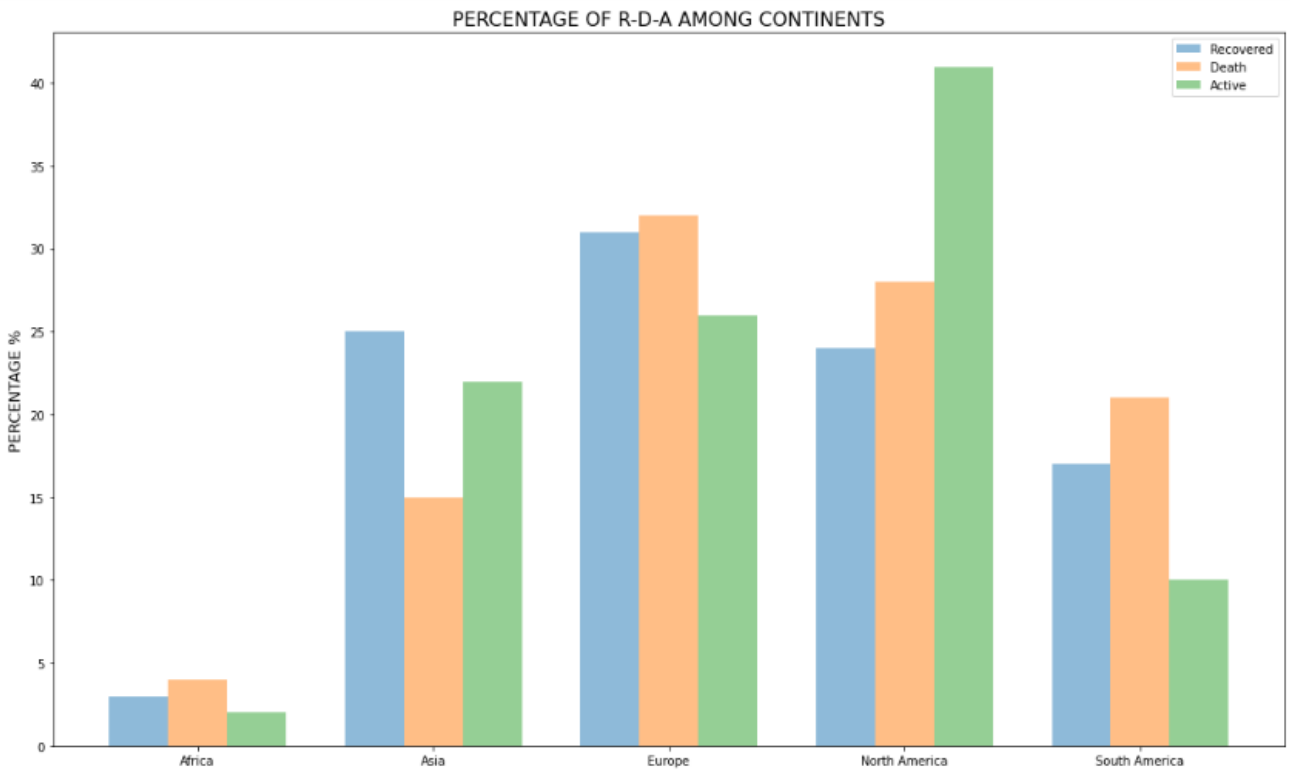
- *Trực quan dữ liệu bằng biểu đồ cột để cho thấy sự chênh lệch về tỉ lệ giữa các châu lục với nhau.*

5.4.2 Ý nghĩa hợp lý sau khi dữ liệu được trực quan

- Dễ dàng thấy Châu Âu là châu lục có tốc độ xét nghiệm rất nhanh.

5.5 Mối quan hệ giữa Tỉ lệ các ca mắc bệnh (ActiveCases), chết (Deaths) và phục hồi (Recovered) giữa các châu lục

5.5.1 Trực quan bằng biểu đồ:



- *Trực quan dữ liệu tỉ lệ số ca mắc bệnh bằng màu xanh lá, số ca chết bằng màu cam, số ca hồi phục bằng màu xanh dương.*
- *Trực quan dữ liệu bằng biểu đồ cột để cho thấy sự chênh lệch về tỉ lệ phần trăm giữa các châu lục với nhau.*

5.5.2 Ý nghĩa hợp lý sau khi dữ liệu được trực quan

- *Tính đến thời điểm thu thập dữ liệu thì Châu Âu có số người chết nhiều nhất, và cũng có số ca hồi phục nhiều nhất. Đúng thực tế khi Châu Âu đã từng là nơi chịu ảnh hưởng nặng nề nhất nhưng giờ đã đỡ hơn và có đầy đủ trang thiết bị để chữa trị.*
- *Số ca nhiễm hiện tại ở Bắc Mĩ cao đột biến nghĩa là cần kha khá thời gian để rất lâu nơi này khống chế dịch thành công.*

5.6 Mối quan hệ giữa Tỉ lệ các ca dương tính (Postive) và các ca âm tính (Negative) tại các châu lục

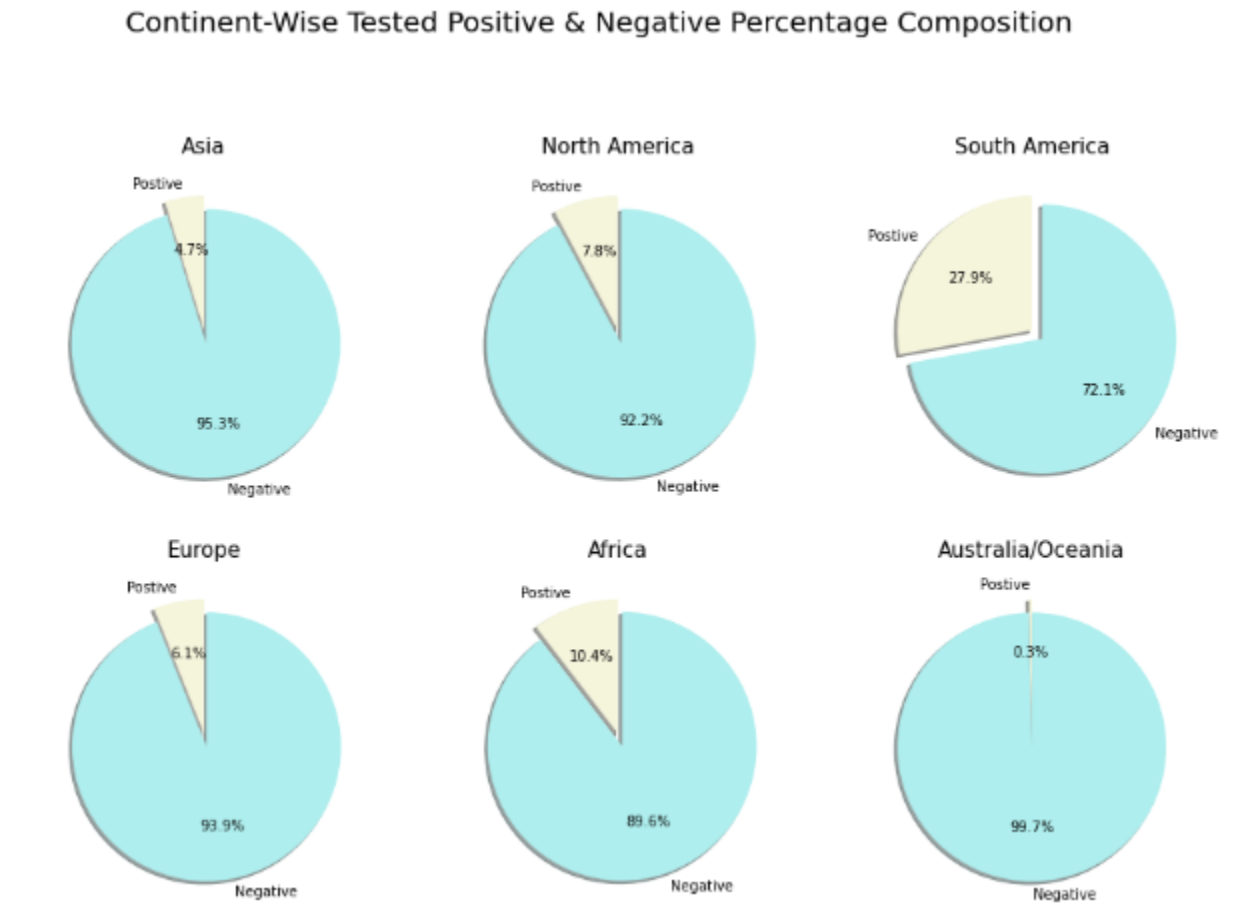
5.6.1 Giải thích mối quan hệ giữa hai trường dữ liệu

```
In [25]: positive = (continent_data['Total Cases']/continent_data['Total Tests']*100).round(2)
         negative = ((continent_data['Total Tests'] - continent_data['Total Cases']) / continent_data['Total Tests']*100).round(
         negative_positive = pd.DataFrame({
                                           'Pos':positive,
                                           'Nega':negative})
         negative_positive
```

Out[25]:

Continent	Pos	Nega
Africa	10.42	89.58
Asia	4.70	95.30
Australia/Oceania	0.33	99.67
Europe	6.06	93.94
North America	7.81	92.19
South America	27.87	72.13

5.6.2 Trực quan bằng biểu đồ:



- *Trực quan hóa dữ liệu tỉ lệ ca dương tính và âm tính của từng châu lục theo từng biểu đồ hình tròn. Tỉ lệ dương tính được thể hiện bằng màu xám, tỉ lệ âm tính được thể hiện bằng màu xanh dương.*

5.6.3 Ý nghĩa hợp lý sau khi dữ liệu được trực quan

- *Xét trên các trường hợp đã tests, Nam Mỹ là nơi có tỉ lệ dương tính cao nhất 27,9%, tỉ lệ âm tính 72,1%.*
- *Kiểm tra tỉ lệ nhằm đoán xem tỉ lệ lượng người đã từng dương tính có cao quá không. Nếu cao quá thì những người hiện giờ an toàn (chưa từng nhiễm bệnh) sẽ không còn "an toàn" nữa.*
- *Châu Đại Dương là châu lục có tỉ lệ dương tính thấp nhất.*