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

BÁO CÁO
MÔN HỌC: TRỰC QUAN HOÁ DỮ LIỆU
LAB01: MỐI QUAN HỆ CỦA DỮ LIỆU

Nhóm 5

1. 1712117 – Nguyễn Huỳnh Thảo Nhi
2. 1712710 – Lê Quang Quí
3. 1712713 – Lê Bá Quyền
4. 1712775 – Nguyễn Lê Trường Thành

GVHD: ..... TH.S Lê Ngọc Thành

Lớp: .....Trực quan hoá dữ liệu CQ2017/1

Năm học: 2019-2020

MỤC LỤC

I. Tổng quan đồ án:

1. Giới thiệu:

- Từ khoảng cuối năm 2019 và đầu năm 2020, một bệnh dịch hạch lan tràn khủng khiếp trên toàn thế giới. Mỗi ngày có hàng ngàn người bị nhiễm và hàng chục đến hàng trăm người chết. Tổ chức Worldometer (www.worldometers.info) đã thu thập dữ liệu thống kê từ nhiều nguồn và từ nhiều quốc gia báo cáo hàng ngày để tổng hợp thành một bảng trong Hình 1. Trong trang web, tổ chức Worldmeter cũng thực hiện vẽ biểu đồ để cho thấy sự thay đổi trực quan tình hình diễn biến dịch bệnh. Tuy nhiên chúng ta tạm thời không sử dụng nó.
- Bạn và nhóm của mình được quốc gia giao trọng trách để tìm hiểu dữ liệu này như giữa các trường dữ liệu có mối quan hệ gì không, liệu có bất thường trong dữ liệu hay không như báo cáo quốc gia khác với dữ liệu tổng hợp, sự bất bình thường trong việc nhảy số liệu, ...Nhiệm vụ này đòi hỏi nhiều kiến thức liên quan như Trực quan hóa dữ liệu, Phân tích dữ liệu thông minh, Học máy, ... Tuy nhiên, nhóm bạn là biệt đội chuyên làm trực quan hóa dữ liệu nên chúng ta sẽ tập trung hướng này trước.

2. Yêu cầu đồ án:

- Thu thập số liệu thống kê từng ngày từ trang Worldmeter.
  - Nhóm sinh viên (NSV) có thể chọn làm trên 1 ngày xác định. Do trang Worldometer chỉ thể hiện ngày hôm nay và ngày hôm qua nên nhóm cần thu thập nhiều ngày để có thể thực hiện tiếp cho các bài lab tiếp theo.
  - NSV có thể thủ công để chép dữ liệu và lưu trữ vào định dạng chuẩn .CSV hoặc sử dụng code để lấy dữ liệu (khuyến khích).
  - NSV có thể tiền xử lý dữ liệu trước khi chuyển sang pha tiếp theo nhưng cần báo cáo vấn đề này trong mục Tiền xử lý dữ liệu. Dữ liệu gốc và dữ liệu đã điều chỉnh cần lưu lại và nộp kèm trong bài nộp.
- Sử dụng nhận xét, code/thuật toán để thể hiện trực quan các mối quan hệ giữa các trường dữ liệu
  - NSV thảo luận và chọn ra các trường dữ liệu để thể hiện trực quan bằng các loại biểu đồ đã học.

- Việc chọn biểu đồ cần giải thích tính phù hợp với tính chất trường dữ liệu. Có thể sử dụng nhiều hơn 1 loại biểu đồ cho trường dữ liệu nhưng cần giải thích lí do.
- Việc thể hiện quan hệ phải tích hợp dần dần nghĩa là từ đơn giản đến phức tạp, từ một trường đơn đến quan hệ giữa nhiều trường, ...
- Ngoài quan hệ độc lập, NSV xem xét liệu trong dữ liệu có quan hệ nhân quả không (cause-effect). Ví dụ: liệu có thể có mối quan hệ giữa tỉ lệ ca nhiễm tăng với số ca chết không, ... Cần chứng minh thông qua các phép trực quan dữ liệu.
- NSV không cần phải làm hết tất cả các quan hệ nhưng nhiều nhất có thể và phủ được nhiều loại biểu đồ đã học.

- NSV giữ lại các dữ liệu để có thể thực hiện tiếp cho các bài sau.

II. Phân công công việc:

STT	Công việc	Thực hiện	Ghi chú
1	Lựa chọn, phân tích dữ liệu, vẽ biểu đồ ***Pie Chart.***	**Lê Bá Quyền**	
	Lựa chọn, phân tích dữ liệu, vẽ biểu đồ ***Stack Vertical Chart***.		
2	Lựa chọn, phân tích dữ liệu, vẽ biểu đồ ***Dot và Line Chart***.	**Lê Quang Quí**	
	Lựa chọn, phân tích dữ liệu, vẽ biểu đồ ***Scatterplot Matrix***.	**Lê Quang Quí**	
3	Lựa chọn, phân tích dữ liệu, vẽ biểu đồ ***Bar Chart***.	**Nguyễn Lê Trường Thành**	
4	Lựa chọn, phân tích dữ liệu, vẽ biểu đồ ***Cluster Heatmap***. Thu thập dữ liệu từ trang web Worldmeter.	**Nguyễn Huỳnh Thảo Nhi**	

III. Môi trường cài đặt:

- Ngôn ngữ lập trình: Python
- Môi trường lập trình: Jupyter Notebook
- Thư viện Python được sử dụng trong đồ án:
  - Requests
  - Pandas
  - Seaborn
  - Matplotlib
  - BeautifulSoup
  - lxml

IV. Tiền dữ lý dữ liệu:

- Đầu tiên, vì dữ liệu thống kê tồn tại trên trang web *www.worldometers.info* nên chúng ta phải dùng code bằng python để lấy dữ liệu về và lưu trữ dưới định dạng file csv.

Crawl Dữ liệu thống kê từng ngày ca nhiễm virus Covid-19 từ tổ chức Worldometer

Entrée [0]:
```
URL = 'https://www.worldometers.info/coronavirus/' #the website the data is extracted
page = requests.get(URL)
soup = BeautifulSoup(page.content, 'html.parser')
```

Entrée [0]:
```
table = soup.find(id='nav-tabContent')
table = table.find(id = 'nav-today')
table = table.find(id = '')
table = table.find(id = 'main_table_countries_today')
```

Entrée [0]:
```
table_rows = table.find_all('tr') #finds all the tables with the tag tr in html
l = []
for tr in table_rows:
    td = tr.find_all('td')
    row = [tr.text for tr in td]
    if len(row) == 0:
        continue
    row = row[:9]
    l.append(row)

dataset = pd.DataFrame(l, columns=["Ranking","Country","Total Cases","New Cases","Total Deaths
```

Lưu dữ liệu hàng ngày vào file .csv

Entrée [0]:
```
import datetime
datestr = datetime.date.today().strftime("%Y%m%d")
dataset.to_csv('CoronaData_{}.csv'.format(datestr))
```

- Dữ liệu được lưu trữ trong file csv với định dạng tên CoronaData_{Y/m/d} để đánh dấu phân biệt.
- Dữ liệu sau khi đã được lấy về từ trang web.

Tiền xử lý dữ liệu

Entrée [6]:
```
dataset
```

Out[6]:

	Ranking	Country	Total Cases	New Cases	Total Deaths	New Deaths	Total Recoverd	Active Cases	Serious Cases
0		\nNorth America\n	2,315,411	+12,075	137,287	+902	948,494	+5,217	1,229,630
1		\nSouth America\n	1,189,145	+12,869	51,889	+562	560,325	+364	576,931
2		\nEurope\n	2,124,125	+14,358	180,062	+702	1,104,935	+16,020	839,128
3		\nAsia\n	1,423,830	+31,348	36,275	+636	863,943	+21,029	523,612
4		\nAfrica\n	200,392	+1,402	5,426	+36	88,901	+689	106,065
...	...	...	...	...	...	...	...	...	...
226		Total:	1,423,830	+31,348	36,275	+636	863,943	+21,029	523,612
227		Total:	200,392	+1,402	5,426	+36	88,901	+689	106,065
228		Total:	8,877	+2	124		8,308		445
229		Total:	721		15		651		55
230		Total:	7,262,501	+72,054	411,078	+2,838	3,575,557	+43,333	3,275,866

231 rows × 9 columns

- Kiểu dữ liệu của bảng dữ liệu hiện tại.
- Qua 2 quan sát trên, ta nhận thấy dữ liệu hiện tại vẫn chưa thể sử dụng được, cần được xử lý phục vụ cho mục đích bài làm.
  - Kiểu dữ liệu các cột đều thuộc kiểu Object, không được định nghĩa rõ ràng.
  - Có các kí tự rác trong dữ liệu như: '\n\', '+', ','
  - Các giá trị không xác định như NaN.
- Hàm xử lý dữ liệu.

Entrée [0]:
```
# Function to Clean the DataSet
def dataframeCleaner(dataset):

    for columnname in dataset: #looping through titles of the table
        temp = []
        for column in dataset[columnname]:   #geting column elements for the each title
            column = str(column)
            column = column.replace(',','')# Removing unwanted data clutter
            column = column.replace('+','')#Removing unwanted '+'sign  \
            try:
                column = int(column)
            except:
                pass

            temp.append(column)
        dataset[columnname] = temp

#   dataset = dataset.drop(dataset.tail(5).index)          # Deleting the last row
    dataset = dataset.replace('N/A', '', regex=True)
    dataset = dataset.replace(r'^\s*$', 0, regex=True)     # Converting empty string to 0
    dataset.replace(['\n'], '', regex=True, inplace=True)
    dataset.replace([','], '', regex=True, inplace=True)
    return dataset
```

- Dữ liệu sau khi được xử lý.

Dữ liệu sau khi được xử lý

```
Entrée [0]: dataset = dataframeCleaner(dataset)
            dataset
```

Out[8]:

	Ranking	Country	Total Cases	New Cases	Total Deaths	New Deaths	Total Recoverd	Active Cases	Serious Cases
0	0	North America	2229176	5160	133972	692	898723	3345	1196481
1	0	South America	1072792	484	47463	12	520953	13513	504376
2	0	Europe	2062733	612	177978	19	1057051	1243	827704
3	0	Asia	1299214	7091	33792	117	786754	2565	478668
4	0	Africa	179337	0	4942	0	80324	0	94071
...	...	...	...	...	...	...	...	...	...
226	0	Total:	1299214	7091	33792	117	786754	2565	478668
227	0	Total:	179337	0	4942	0	80324	0	94071
228	0	Total:	8865	4	124	0	8275	0	466
229	0	Total:	721	0	15	0	651	0	55
230	0	Total:	6852838	13351	398286	840	3352731	20673	3101821

231 rows × 9 columns

- Dữ liệu sau khi xử lý được lưu trữ trong file csv với định dạng tên CoronaData_{Y/m/d}_HorseColic.csv để đánh dấu phân biệt.

V. Trực quan mối quan hệ giữa các trường dữ liệu:

1. Pie Chart:

a. Pie Chart là gì?

- Pie chart là biểu đồ dạng hình tròn thể hiện mối quan hệ theo phần trăm giữa các phần so với tổng thể.

b. Lựa chọn dữ liệu:

- Đối với loại biểu đồ này, và dựa vào bảng dữ liệu. Nhóm quyết định chọn 3 trường dữ liệu là Total Recoverd, Total Deaths, Active Cases từ dòng có "Country, Other" = "World" (là tổng hợp từ tất cả các nước trên thế giới) để trực quan hóa cũng như quan sát mối liên hệ giữa các trường với nhau

c. Code Python:

- Để vẽ biểu đồ Pie Chart, nhóm sử dụng thư viện pandas.

- Lọc ra các cột dữ liệu, dòng dữ liệu cần thiết đối với biểu đồ, loại bỏ những giá trị thừa, không cần thiết, làm biểu đồ không được chính xác.

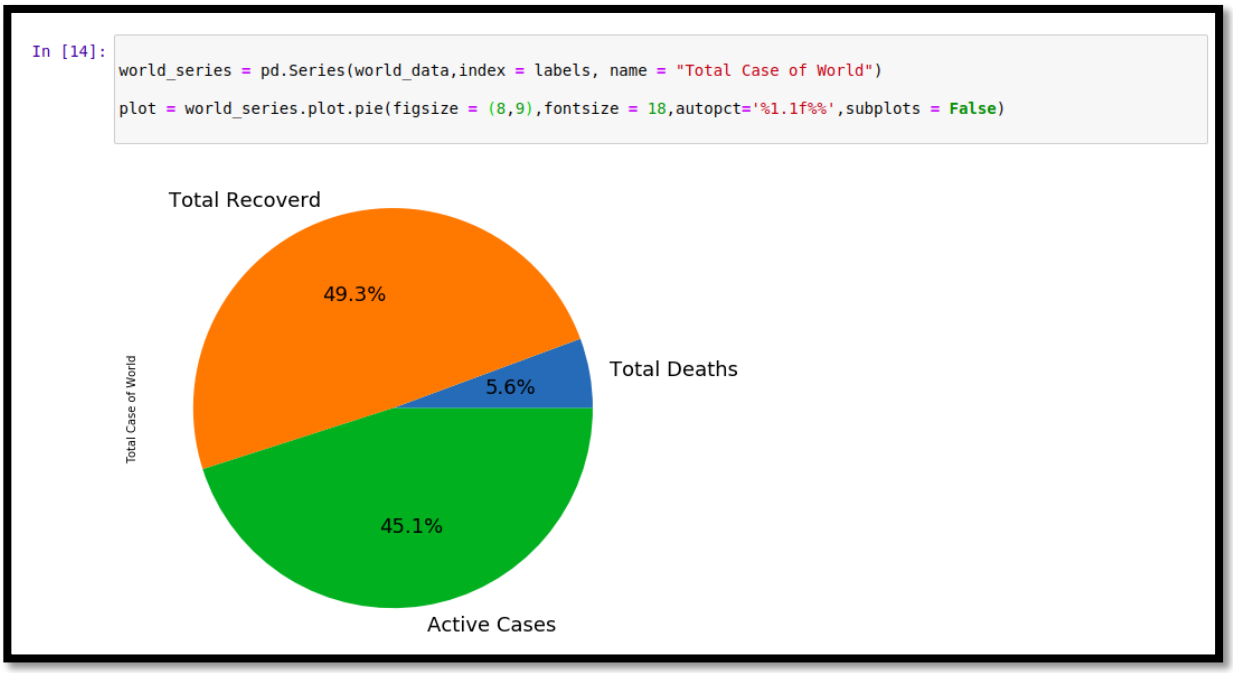
```
In [13]: labels = ["Total Deaths","Total Recoverd","Active Cases"]
         world_total = data[data["Country"] == "World"][labels]
         world_data = world_total[labels].values[0].tolist()  # convert ndarray to  list
         print(world_total)

            Total Deaths  Total Recoverd  Active Cases
         0        414126         3620284       3308652

In [14]: world_series = pd.Series(world_data,index = labels, name = "Total Case of World")

         plot = world_series.plot.pie(figsize = (8,9),fontsize = 18,autopct='%1.1f%%',subplots = False)
```

d. Biểu đồ:



e. Nhận xét mối quan hệ:

- Từ biểu đồ trên ta nhận thấy đã chưa khỏi được gần một nửa số ca mắc bệnh trên toàn cầu. Còn lại khoảng 45.1% ca mắc bệnh vẫn chưa được chữa khỏi.
- Số ca tử vong chỉ chiếm 5.6% tổng số ca mắc bệnh trên toàn cầu cho thấy bệnh này có tỷ lệ tử vong khá thấp.

2. Dot and Line Chart:

a. Dot and Line Chart là gì?

- Biểu đồ kết hợp giữ biểu đồ đường và biểu đồ chấm.

b. Lựa chọn dữ liệu:

- Nhóm quyết định lấy dữ liệu của top 10 quốc gia có số lượng ca mất cao nhất.

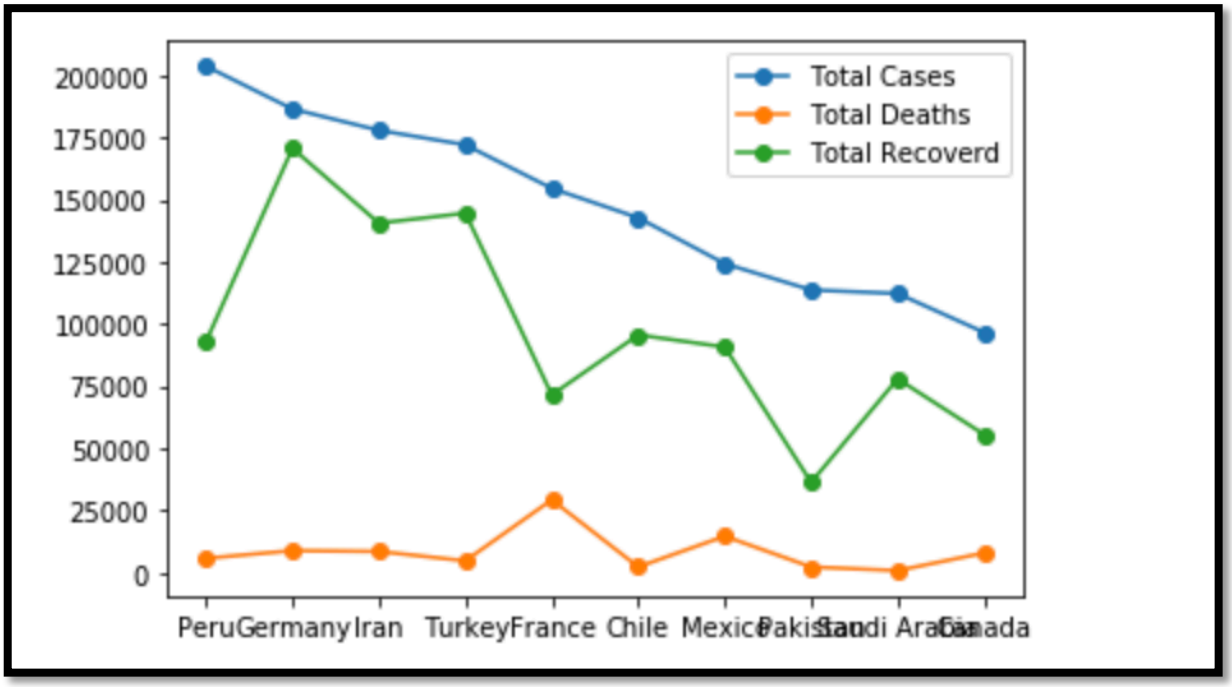
c. Code Python:

```
Entrée [11]: total_case_death_reco=df
features = ["Country","Total Cases","Total Deaths","Total Recoverd","Active Cases","Serious
df_total_case_death_reco=total_case_death_reco[features]
df_total_case_death_reco = df_total_case_death_reco.drop(df.head(1).index)
df_total_case_death_reco=df_total_case_death_reco.head(10)
df_total_case_death_reco
```

Out[11]:

	Country	Total Cases	Total Deaths	Total Recoverd	Active Cases	Serious Cases
8	Peru	203736	5738	92929	105069	1077
9	Germany	186525	8831	170700	6994	533
10	Iran	177938	8506	140590	28842	2639
11	Turkey	172114	4729	144598	22787	642
12	France	154591	29296	71506	53789	955
13	Chile	142759	2283	95631	44845	1689
14	Mexico	124301	14649	90748	18904	378
15	Pakistan	113702	2255	36308	75139	111
16	Saudi Arabia	112288	819	77954	33515	1693
17	Canada	96653	7897	55572	33184	1864

d. Biểu đồ:



e. Nhận xét:

- Từ biểu đồ ta thấy Germany và Turkey có tỉ lệ hồi phục khá cao, cho thấy 2 quốc gia này có khả năng khắc phục dịch bệnh tốt.
- Tình hình dịch bệnh ở Peru không có vẻ khả quan lắm khi sô người hồi phục không được cao lắm.

- France có tỉ lệ người chết vì bệnh cao nhất và tỉ lệ phục hồi khá thấp, dường như nước này đang gặp vấn đề trong việc khắc chế dịch bệnh.

3. Scatterplot Matrix:

a. Scatterplot Matrix là gì?

- Scatterplot Matrix được viết tắt là SPLOM, là một công cụ đồ họa tương đối hiếm gặp, sử dụng nhiều biểu đồ phân tán để xác định mối tương quan (nếu có) giữa một loạt các biến.

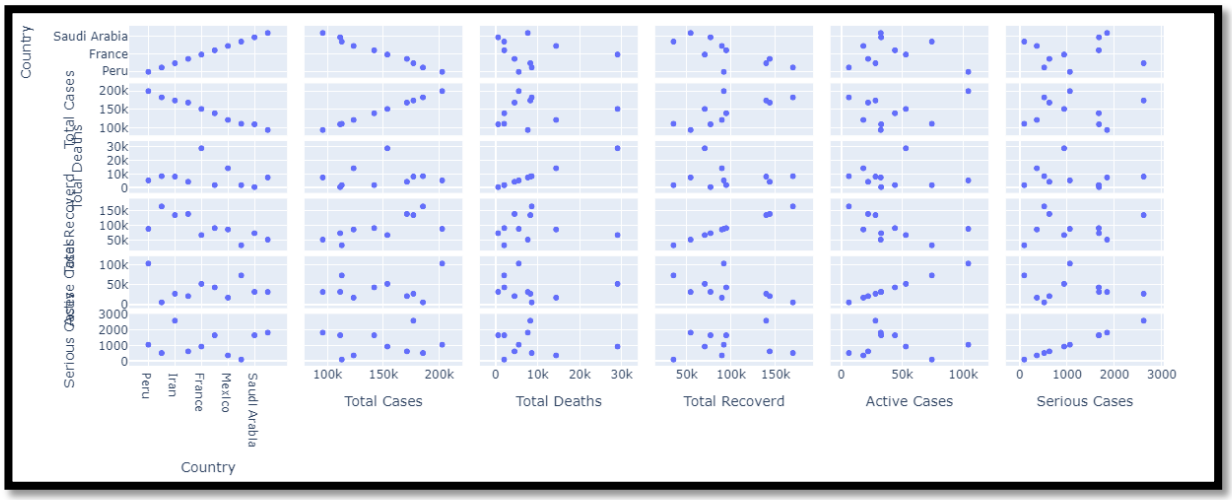
b. Lựa chọn dữ liệu:

- Nhóm quyết định chọn bộ dữ liệu của top 10 quốc gia có số lượng ca cao nhất.

c. Code python:

```
import plotly.express as px
df = px.data.iris()
fig = px.scatter_matrix(df_total_case_death_reco)
fig.show()
```

d. Biểu đồ:



e. Ý nghĩa rút ra.

- Ở cột Serious Cases và Total Cases các dấu chấm phân bố gần như điều cho thấy tỉ lệ các ca mắt nghiêm trọng so với các thuộc tính khác khá đồng đều với nhau, tỉ lệ của tổng ca mất cũng tương tự như vậy.
- Còn các cột khác thì tỉ lệ của các thuộc tính khác so với các thuộc tính còn lại nằm lệch về bên trái nhưng điều này không chứng tỏ các tỉ lệ này thấp.

4. Bar Chart:

a. Bar Chart là gì?

- Bar Chart là biểu đồ thể hiện dữ liệu phân loại với các thanh hình chữ nhật có chiều cao hoặc chiều dài tỷ lệ với các giá trị mà chúng đại diện. Các thanh có thể được vẽ theo chiều dọc hoặc chiều ngang. Biểu đồ thanh dọc đôi khi được gọi là biểu đồ cột.

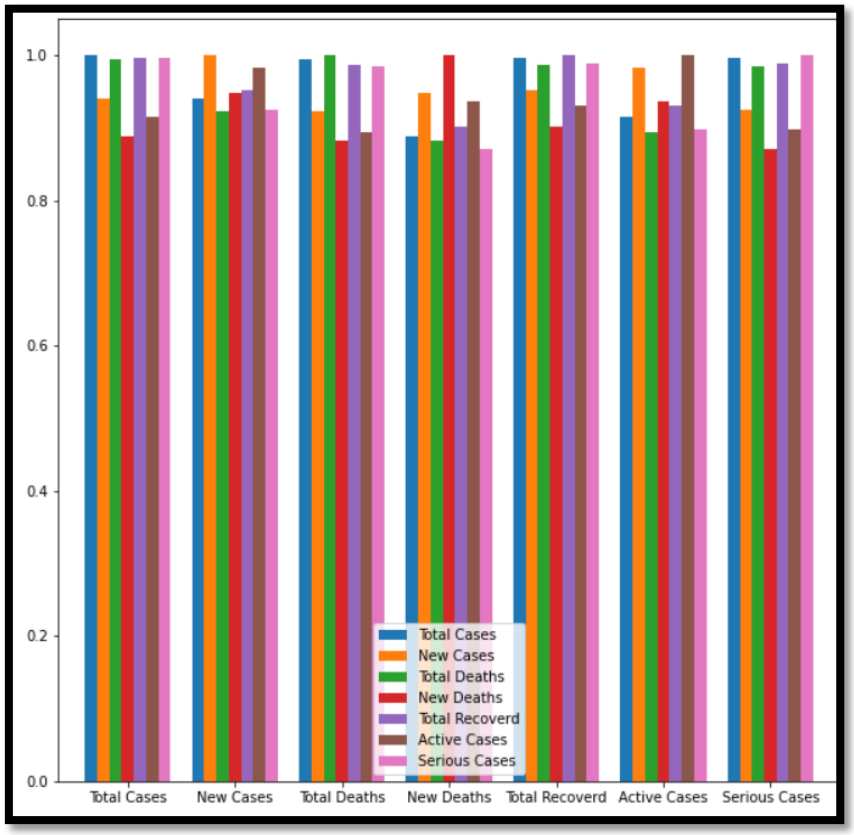
b. Lựa chọn dữ liệu:

- Nhóm lựa chọn tất cả các trường dữ liệu của bảng dữ liệu.

c. Code Python:

```
Entrée [12]: %matplotlib inline
             plt.figure(figsize=(100,50))
             corr_data.corr().plot.bar(rot=0, width = 0.8, figsize=(10,10))
```

d. Biểu đồ:



e. Nhận xét:

- Mỗi cột hiển thị mối tương quan giữa các biến trên mỗi trục.
- Các giá trị gần bằng 0 hơn có nghĩa là không có xu hướng tuyến tính giữa hai biến. Gần 1 tương quan là chúng có mối tương quan tích cực hơn; đó là khi một cái tăng và cái kia cũng vậy và càng gần 1 thì mối quan hệ này càng bền chặt. Cột có giá trị 1 vì các cột đó tương quan với từng biến của chính nó (ta gọi là một mối tương quan hoàn hảo). Ví dụ, tương quan r = 0,9 cho thấy mối liên hệ tích cực, mạnh mẽ giữa hai biến. Một mối tương quan gần bằng 0 cho thấy không có mối liên hệ tuyến tính giữa hai biến liên tục.
- Cột ngắn hơn cho ta thấy sự ít tương quan vào nhau hơn. Ví dụ như nhìn vào hình ta thấy giữa Serious Cases và New Cases có mối tương quan rất

mật thiết với nhau bởi vì r = 0.99 và bên cạnh đó Total Cases và New Death sẽ ít tương quan với nhau hơn với r = 0.87.

- Từ đó giúp ta có thể có thêm một số ý nghĩa về mặt dự đoán như: để dự đoán số ca nghiệm trọng(Serious Cases) sắp tới thì có thể dựa vào số ca mới (New Cases), để dự đoán số ca tử vong sắp tới(New Death) thì không nên dựa vào tổng số ca nhiễm(Total Cases).

5. Stack Vertical Bar Chart:

a. Stack Vertical Bar Chart là gì?

- Biểu đồ thanh xếp chồng cho phép bạn hiển thị và so sánh các mối quan hệ một phần với toàn bộ

b. Lựa chọn dữ liệu

- Đối với loại biểu đồ này, và dựa vào bảng dữ liệu. Nhóm quyết định chọn 3 trường dữ liệu là Total Recoverd, Total Deaths, Active Cases từ top 4 nước có số ca tử vong ngày hôm nay cao nhất để trực quan hóa cũng như quan sát mối liên hệ giữa các trường với nhau.

c. Code python:

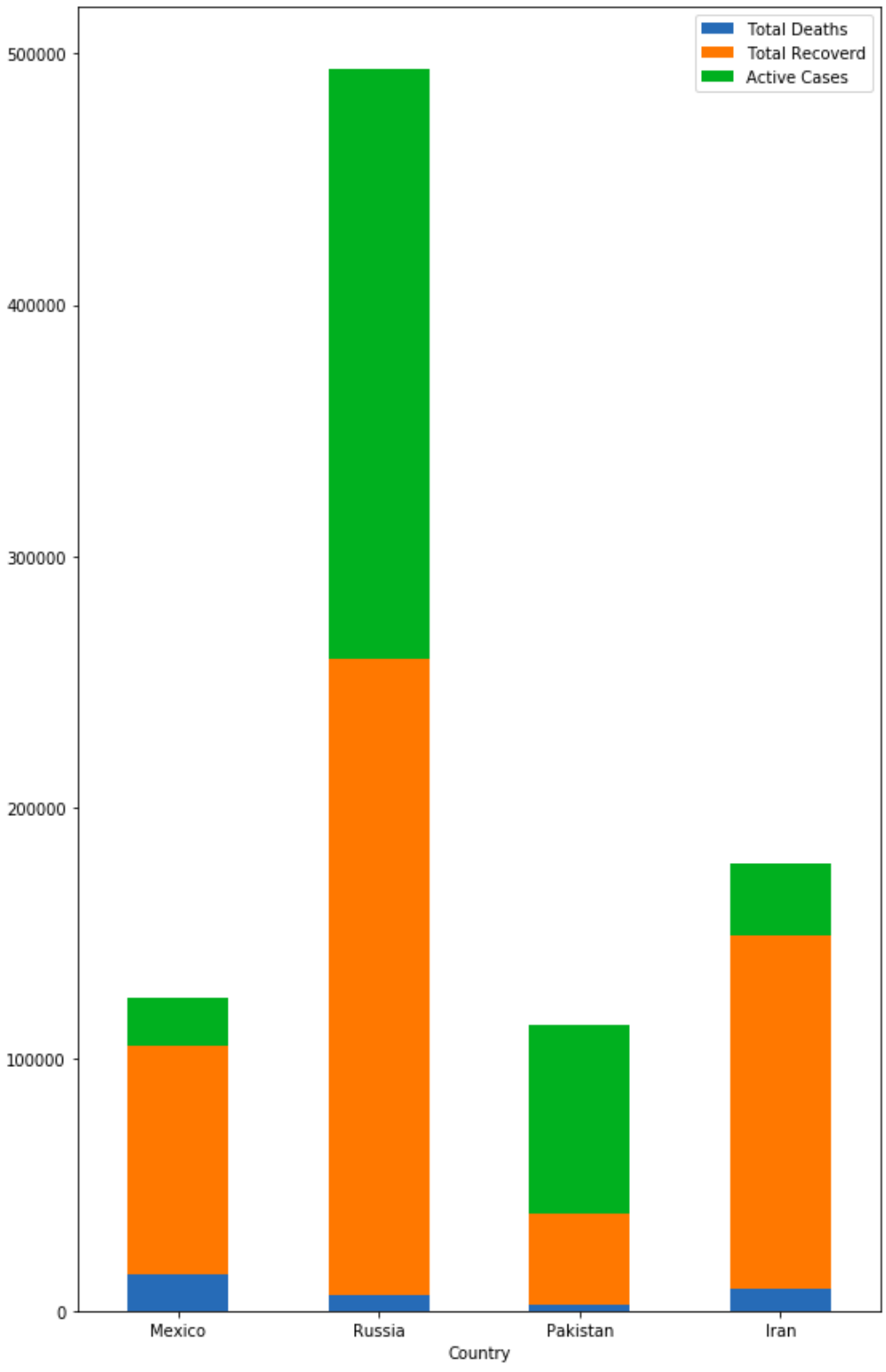
- Để vẽ biểu đồ Pie Chart, nhóm sử dụng thư viện pandas.
- Lấy ra 4 nước có số ca tử vong ngày hôm nay cao nhất.
- Lọc ra các cột dữ liệu cần thiết đối với biểu đồ, loại bỏ những giá trị thừa, không cần thiết, làm biểu đồ không được chính xác.

```
In [15]: large5 = data[data['Country'] != "World"].nlargest(4,"New Deaths")  #get top 4 country have largest New Deaths
         indexs = ["Country","Total Deaths", "Total Recoverd", "Active Cases"]
         large_filter = large5[indexs].set_index("Country")
         print(large_filter)

                  Total Deaths  Total Recoverd  Active Cases
         Country
         Mexico          14649           90748         18904
         Russia           6358          252783        234516
         Pakistan         2255           36308         75139
         Iran             8506          140590         28842

In [17]: plot = large_filter.plot.bar(stacked=True,figsize = (9,15),rot=0)
```

d. Biểu đồ:



e. Nhận xét:

- Từ biểu đồ ta thấy 5 nước có số ca tử vong cao nhất hôm nay lần lượt là Mexico, Russia, Pakistan, Iran.

- Một cách tổng thể ta thấy mặc dù Mexico có số ca tử vong hôm nay là cao nhất nhưng lại tỷ lệ hồi phục lại khá cao nên có thể dự đoán rằng ở nước này có nhiều ca nhiễm nghiêm trọng hơn so với các nước còn lại.
- Ta cũng có thể thấy rằng Pakistan hồi phục khá là chậm chạp vì số ca nhiễm chưa được chữa khỏi còn lớn hơn nhiều so với số ca đã được chữa khỏi của nước này.

6. Cluster Heatmap:

a. Cluster Heatmap là gì?

- ***Heatmap*** là một kỹ thuật trực quan hóa dữ liệu cho thấy mức độ của một hiện tượng là màu sắc ở hai chiều. Sự thay đổi màu sắc có thể là do màu sắc hoặc cường độ, mang lại tín hiệu thị giác rõ ràng cho người đọc về cách hiện tượng được nhóm lại hoặc thay đổi theo không gian.
- Có hai loại ***heatmap*** cơ bản khác nhau: ***cluster heat map*** và ***spatial heat map***. Trong ***cluster heat map***, các cường độ được đặt trong một ma trận có kích thước ô cố định có các hàng và cột là các hiện tượng và thể loại riêng biệt, và việc sắp xếp các hàng và cột là có chủ ý và có phần tùy ý, với mục tiêu đề xuất các cụm hoặc mô tả chúng như là được phát hiện thông qua phân tích thống kê.

b. Lựa chọn dữ liệu:

- Đối với loại biểu đồ này, và dựa vào bảng dữ liệu. Nhóm quyết định chọn 7 trường dữ liệu để trực quan hoá cũng như quan sát mối quan hệ giữa tất cả các trường dữ liệu với nhau.

c. Code python:

- Để vẽ biểu đồ Cluster Heatmap, nhóm sử dụng thư viện seaborn và matplotlib.

- Lọc ra các cột dữ liệu, dòng dữ liệu cần thiết đối với biểu đồ, loại bỏ những giá trị thừa, không cần thiết, làm biểu đồ không được chính xác.

Entrée [0]:
```
df = dataset
df = dataframeCleaner(df)
df = df.drop(df.head(7).index)
df = df.drop(df.tail(8).index)
df
```

Out[24]:

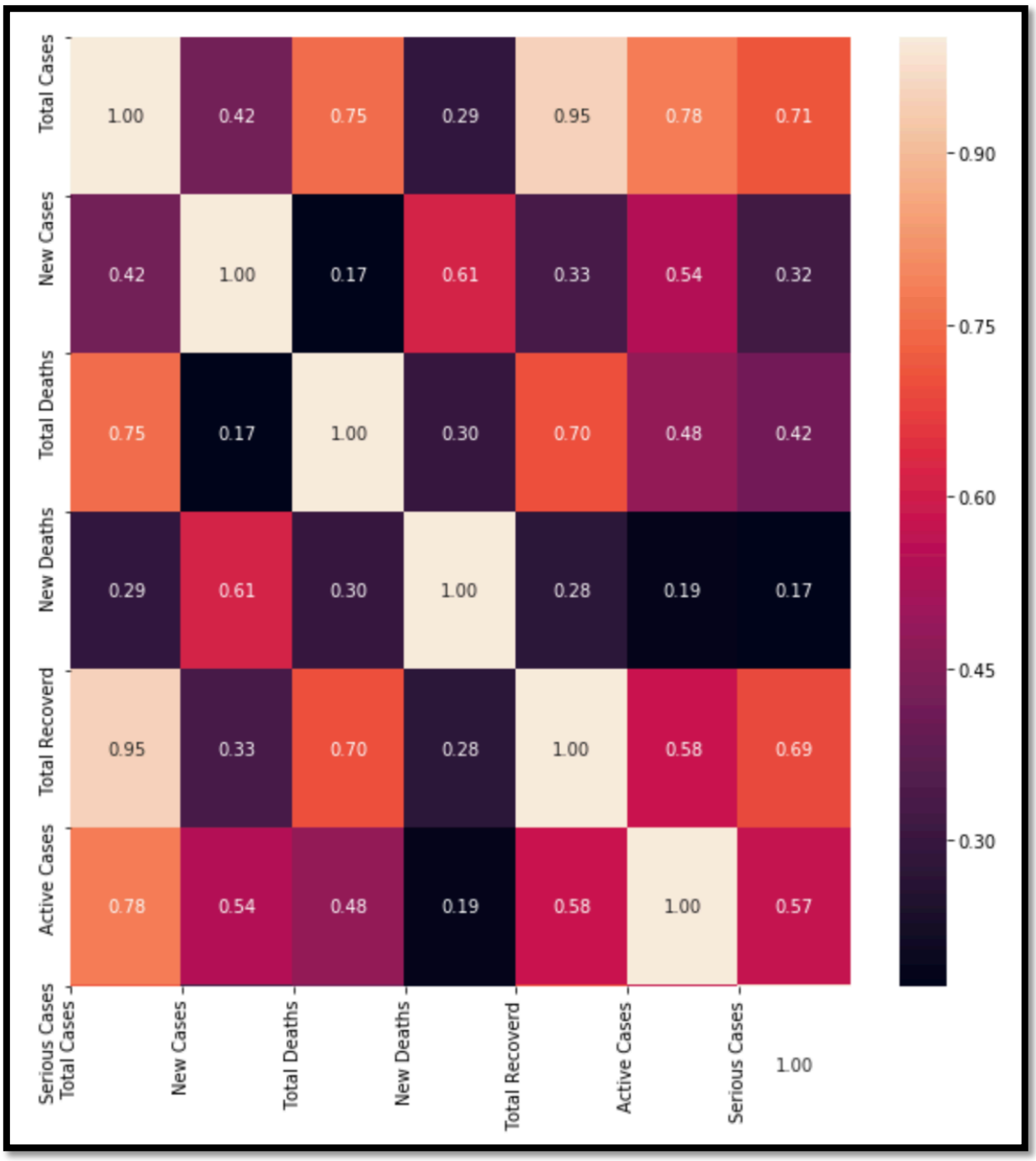
	Ranking	Country	Total Cases	New Cases	Total Deaths	New Deaths	Total Recoverd	Active Cases	Serious Cases
7	0	World	6852838	13351	398286	840	3352731	20673	3101821
8	1	USA	1965912	204	111394	4	738729	83	1115789
9	2	Brazil	646006	0	35047	0	302084	13432	308875
10	3	Russia	449834	0	5528	0	212680	0	231626
11	4	Spain	288058	0	27134	0	0	0	0
...	...	...	...	...	...	...	...	...	...
218	211	St. Barth	6	0	0	0	6	0	0
219	212	Lesotho	4	0	0	0	2	0	2
220	213	Anguilla	3	0	0	0	3	0	0
221	214	Saint Pierre Miquelon	1	0	0	0	1	0	0
222	215	China	83030	3	4634	0	78329	2	67

216 rows × 9 columns

Entrée [0]:
```
def heatMap(df):
    corr = df.corr()
    fig, ax = plt.subplots(figsize=(10, 10))
    sns.heatmap(corr,  annot=True, fmt=".2f")
    plt.xticks(range(len(corr.columns)), corr.columns)
    plt.yticks(range(len(corr.columns)), corr.columns)
    plt.show()
```

Entrée [0]:
```
data = df
features = ["Total Cases","New Cases","Total Deaths","New Deaths","Total Recoverd","Active Cas
corr_data = data[features]
heatMap(corr_data)
```

d. Biểu đồ:



e. Nhận xét mối quan hệ:

- Đối với những ô có màu càng đậm, chúng tỏ sự phụ thuộc giữa các trường càng ít, mối liên quan lẫn nhau càng ít.
- Đối với những ô có màu càng nhạt, sự phụ thuộc, liên quan và tác động với nhau càng lớn.

- Nhìn vào biểu đồ ta có thể thấy các trường hợp tổng số ca nhiễm và tổng số ca phục có tỉ lệ 0.95 chứng tỏ số lượng ca nhiễm phục rất cao, có thể thấy tình hình dịch bệnh có xu hướng diễn biến tốt.
- Tỉ lệ ca hồi phục so với ca nhiễm mới là 0.33, tương đương với việc số ca hồi phục lớn hơn so với số ca nhiễm mới. Tiến độ điều trị và tình hình hồi phục khả quan.

VI. Mức độ hoàn thành đồ án:

STT	Công việc	Mức độ hoàn thành	Ghi chú
1	Thu thập và tiền xử lý dữ liệu từ trang web.	**100%**	
2	Lựa chọn, phân tích dữ liệu, vẽ biểu đồ ***Pie Chart.***	**100%**	
3	Lựa chọn, phân tích dữ liệu, vẽ biểu đồ ***Dot và Line Chart***.	**100%**	
4	Lựa chọn, phân tích dữ liệu, vẽ biểu đồ ***Stack Vertical Chart.***	**100%**	
5	Lựa chọn, phân tích dữ liệu, vẽ biểu đồ ***Scatterplot Matrix***.	**100%**	
6	Lựa chọn, phân tích dữ liệu, vẽ biểu đồ ***Bar Chart***.	**100%**	
7	Lựa chọn, phân tích dữ liệu, vẽ biểu đồ ***Cluster Heatmap***.	**100%**	

VII. Mức độ thực hiện đồ án của các thành viên:

STT	Thành viên	Mức độ hoàn thành	Ghi chú
1	**Nguyễn Huỳnh Thảo Nhi**	**100%**	
2	**Lê Quang Quí**	**100%**	
3	**Lê Bá Quyền**	**100%**	
4	**Nguyễn Lê Trường Thành**	**100%**	

VIII. Tham khảo:

[1]. https://www.datafied.world/web-scraping-live-covid-19-data-and-its-analysis-190

[2].https://towardsdatascience.com/tracking-corona-covid-19-spread-in-india-using-python-40ef8ffa7e31

[3]. https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html

[4]. https://opensource.com/article/20/4/python-data-covid-19

[5]. https://scipython.com/blog/plotting-covid-19-cases/

[6]. https://nextjournal.com/mpd/covid-19-exploratory-data-analysis