

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
Khoa CNTT CLC



BÁO CÁO

Bộ môn: NHẬP MÔN KHOA HỌC DỮ LIỆU

Đề án: Thống Kê Và Trục Quan Hoá Dữ Liệu

Lớp: 19KHDL

| GIẢNG VIÊN HƯỚNG DẪN |

ThS Lê Ngọc Thành

Họ và tên: **Lữ Thế Vỹ**

MSSV: **19127009**

THÀNH PHỐ HỒ CHÍ MINH - THÁNG 12 NĂM 2021

Thông tin sinh viên và lời nói đầu

Họ và Tên	MSSV	Lớp	% hoàn thành
Lữ Thế Vỹ	19127009	19KHDL	100

Em cam đoan báo cáo này em tự xây dựng và nghiên cứu không sao chép bất kỳ cá nhân nào.

STT	Công việc	Chi tiết	% hoàn thành	Ghi chú
1	Thu thập dữ liệu	- Thế giới (số liệu chung): ngày hôm nay, ngày hôm qua, hai ngày trước, các ngày trước (lastdays)	100	* Cột mốc: ngày 12/12/2021
		- Tất cả quốc gia: ngày hôm nay	100	* Cột mốc: ngày 12/12/2021
		- Tất cả châu lục (châu Á, châu Âu, châu Phi, châu Mỹ, châu Úc đại dương): ngày hôm nay, ngày hôm qua, hai ngày trước	100	* Cột mốc: ngày 12/12/2021
		- Việt Nam: ngày hôm nay, ngày hôm qua, hai ngày trước, các ngày trước (lastdays)	100	* Cột mốc: ngày 12/12/2021
2	Tiền xử lý dữ liệu	- Việt Nam: các ngày trước (lastdays)	100	
		- Thế giới (số liệu chung): các ngày trước (lastdays)	100	
		- Tất cả quốc gia: ngày hôm nay	100	
3	Trực quan hoá	- Pie chart	100	3 cái
		- Line chart	100	3 cái
		- Bar chart	100	1 cái
		- Radar chart	100	1 cái
		- Stacked Bar chart	100	1 cái
		- Scatter plot	100	1 cái
		- Bubble plot	100	1 cái
		- Stacked Area chart	100	1 cái
		- Choropleth map	100	3 cái

MỤC LỤC

Thông tin sinh viên và lời nói đầu	2
MỤC LỤC.....	3
Phần 1: Khai thác dữ liệu.....	4
1.1 Các thư viện Python cần thiết	4
1.2 Giải thích code	4
Phần 2: Tiền xử lý dữ liệu	6
Phần 3: Trực quan hoá dữ liệu	8
3.1 Các thư viện đồ hoạ Python cần thiết	8
3.2 Nhận định chung về API và đánh giá dataset	8
3.3 Giải thích code và nhận xét biểu đồ	9
3.3.1 Pie chart	9
3.3.2 Line chart.....	12
3.3.3 Radar chart.....	16
3.3.4 Stacked bar chart	17
3.3.4 Bar chart.....	17
3.3.5 Scatter plot.....	18
3.3.6 Bubble plot.....	20
3.3.7 Stacked area plot.....	21
3.3.8 Choropleth map	23

Phần 1: Khai thác dữ liệu

1.1 Các thư viện Python cần thiết

- import **requests** : hỗ trợ gửi HTTP/1.1 request, dùng để gọi API Worldometers
- import **json** : hỗ trợ thao tác với file định dạng JSON
- import **csv** : hỗ trợ thao tác với file định dạng CSV
- from **datetime** import datetime : hỗ trợ xử lý chuỗi thời gian (ngày, tháng, năm) trong dataset
- import **pandas** as pd : hỗ trợ đọc nội dung của file CSV
- from **pandas** import **json_normalize** : chuyển đổi Python Dictionary thành nội dung JSON
- from **pprint** import pprint : hỗ trợ in kết quả JSON ra console với định dạng chuẩn

1.2 Giải thích code

- Khai báo url là một đường dẫn API (*có chú thích thêm ở tài liệu API cá nhân đính kèm*) và sử dụng command request (phương thức GET) để gửi request HTTP

```
url = "https://disease.sh/v3/covid-19/all"
payload = {}
headers = {}
response = requests.request("GET", url, headers=headers, data=payload)
```

- Sau khi gửi request, client sẽ nhận được kết quả (response), sử dụng inline function .text của biến response để trích xuất nội dung của request. Hàm inline json.dumps để chuyển đổi Python Object (response.text) thành 1 chuỗi JSON

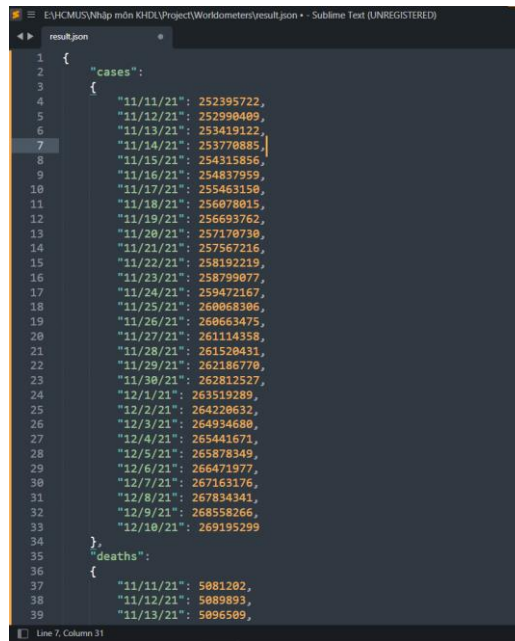
```
with open('result.json', 'w', encoding='utf-8') as f:
    f.write(json.dumps(response.text))
```

- Load nội dung trong result.json (ở dạng chuỗi JSON) và chuyển nó thành một Dictionary JSON bằng inline function loads

```
with open('result.json') as file_object:
    data = json.load(file_object)

d = json.loads(data)
with open('result.json', 'w', encoding='utf-8') as f:
    f.write(json.dumps(d))
```

* Minh họa:



- Sử dụng hàm `json_normalize` để biến JSON Dictionary thành một bảng phẳng (flat table, có thể truy cập dữ liệu dựa vào trị giá chứ không dựa vào key, thuận tiện cho việc truy xuất) gồm key và value để lưu vào file CSV

```
df = json_normalize(data)
df.to_csv('data/world/original/today.csv', index=False)
```

*** Minh họa:**

	updated	country	cases	todayCase	deaths	todayDeat	recovered	todayRecc	active	critical	casesPerO	deathsPerO	tests	testsPerO	population	continent	oneCasePer
1	1.64E+12	Vietnam	1398413	16141	27611	209	1053425	1084	317377	7558	14183	280	71152452	721633	98599197	Asia	71
2																	
3																	
4																	
5																	
6																	
7																	
8																	
9																	
10																	
11																	
12																	

Phần 2: Tiền xử lý dữ liệu

- Dữ liệu được lưu trong file định dạng CSV sẽ được truy xuất dựa vào các key (hay còn gọi là header), rất thuận tiện trong việc thao tác. Nhưng để có được các file CSV đã chắt lọc các thông tin cần thiết, ta cần phải trải qua bước tiền xử lý với dữ liệu thô (dữ liệu ban đầu thu được)

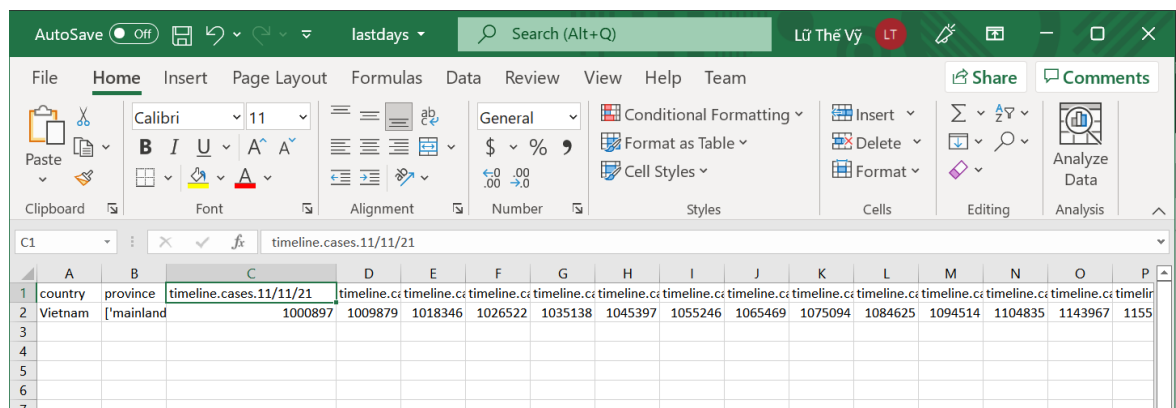
- Giả định sau khi request dữ liệu Covid-19 của Việt Nam trong 30 ngày bằng cách sử dụng API

```
url = "https://corona.lmao.ninja/v2/historical/Vietnam?lastdays=30"
payload = {}
headers = {}
response = requests.request("GET", url, headers=headers,
data=payload)

with open('result.json', 'w', encoding='utf-8') as f:
    f.write(json.dumps(response.text))

with open('result.json') as file_object:
    data = json.load(file_object)
```

- Ta thu được kết quả sau (trong file CSV):



country	province	11/11/21	11/11/21	11/11/21	11/11/21	11/11/21	11/11/21	11/11/21	11/11/21	11/11/21	11/11/21	11/11/21	11/11/21	11/11/21	11/11/21	11/11/21	11/11/21
Vietnam	mainland	1000897	1009879	1018346	1026522	1035138	1045397	1055246	1065469	1075094	1084625	1094514	1104835	11143967	11155		

- Lúc này, nếu ta muốn trích danh sách ca nhiễm (timeline.cases), chỉ cần gọi key (header) là ['timeline']['cases']. Lưu kết quả trích xuất vào 1 file CSV khác là cases.csv

```
csv_file = open("data/world/modified/lastdays/cases.csv", "w")
writer = csv.writer(csv_file)
writer.writerow(['timeline', 'cases'])
for key, value in data['cases'].items():
    writer.writerow([datetime.strptime(key,
'%m/%d/%y').strftime("%d/%m/%y"), value])
csv_file.close()
```

* Kết quả:

lume (E:) > HCMUS > Nhập môn KHDL > Project > Worldometers > data > vietnam > modified > lastdays >

Name	Date modified	Type	Size
.ipynb_checkpoints	07-Dec-21 11:24 AM	File folder	
cases	12-Dec-21 2:09 AM	Microsoft Excel ...	1 KB
deaths	12-Dec-21 2:09 AM	Microsoft Excel ...	1 KB
recovered	12-Dec-21 2:09 AM	Microsoft Excel ...	1 KB

AutoSave Off cases Search (Alt+)

File Home Insert Page Layout Formulas Data Review

Paste Font Alignment Number

A1 timeline

	A	B	C	D	E	F	G	H
1	timeline	cases						
2								
3	11-11-21	1000897						
4								
5	12-11-21	1009879						
6								
7	13-11-21	1018346						
8								
9	14-11-21	1026522						
10								
11	15-11-21	1035138						
12								
13	16-11-21	1045397						
14								
15	17-11-21	1055246						

Phần 3: Trực quan hoá dữ liệu

3.1 Các thư viện đồ hoạ Python cần thiết

- import **pandas** as pd : hỗ trợ đọc file CSV
- import **numpy** as np : hỗ trợ lập mảng phụ lưu trữ
- import **matplotlib.pyplot** as plt : thư viện đồ hoạ dùng để vẽ biểu đồ
- import **plotly.express** as px : thư viện đồ hoạ dùng để vẽ biểu đồ
- import **plotly** as py : thư viện đồ hoạ dùng để vẽ biểu đồ
- import **csv** : hỗ trợ xuất file CSV
- import **matplotlib.colors** as mc : chỉnh màu các đề mục chú thích
- import **plotly.graph_objs** as go : thư viện đồ hoạ dùng để vẽ biểu đồ

3.2 Nhận định chung về API và đánh giá dataset

- API nhìn chung cung cấp đủ các phương thức để lấy dữ liệu. Tuy nhiên cũng có 1 số lỗi hổng như sau:

- Không thể lấy đủ các trường dữ liệu (ví dụ: tỉ lệ tiêm vaccin,...)
- Không thể lấy dữ liệu cũ với đầy đủ thông tin (chỉ lấy được 3 trường cases,deaths,recovered)
- Trường recovered của dữ liệu cũ có thiếu sót ở một số quốc gia (ví dụ: Việt Nam, toàn bộ trường Recovered trong 30 ngày trước ngày mốc 12/12/2021)

	timeline	recovered
1	11/11/21	0
2	12/11/21	0
3	13/11/21	0
4	14/11/21	0
5	15/11/21	0
6	16/11/21	0
7	17/11/21	0
8	18/11/21	0
9	19/11/21	0
10	20/11/21	0
11	21/11/21	0
12	22/11/21	0
13	23/11/21	0
14	24/11/21	0
15	25/11/21	0
16	26/11/21	0
17	27/11/21	0
18	28/11/21	0
19	29/11/21	0
20	30/11/21	0

- Việc thiếu sót một số trường dữ liệu cũng như bộ dữ liệu sẽ gây bất lợi lớn cho việc trực quan hoá, so sánh các trường. Bởi lẽ số lượng trường dữ liệu còn khá ít (tận dụng tối đa 3 trường đã nêu)


- Giải pháp cho vấn đề này là sẽ lựa chọn dữ liệu của các phương thức khác và chọn loại biểu đồ thích hợp. Việc này sẽ làm hạn chế lại sự đa dạng của việc áp dụng các kiểu biểu đồ khác nhau.

3.3 Giải thích code và nhận xét biểu đồ

3.3.1 Pie chart

3.3.1.1 Tình hình dịch Covid-19 của thế giới (ngày hôm nay)

- Khi dùng API để request content, tín hiệu response sẽ bao gồm nhiều trường thông tin, số liệu về tình hình dịch Covid-19 của thế giới (như hình mô tả)



	updated	cases	todayCases	deaths	todayDeaths	recovered	todayRec
1	1639248984682	269758712	345595	5316182	4454	242625329	

- Lấy cột mốc là ngày 12/12/2021, ta sẽ dùng 4 trường dữ liệu: cases, deaths, recovered, active để thể hiện cơ cấu tỉ lệ phần trăm các thành phần trong bối cảnh dịch Covid-19 trên thế giới.

```
world = pd.read_csv('data/world/original/today.csv')

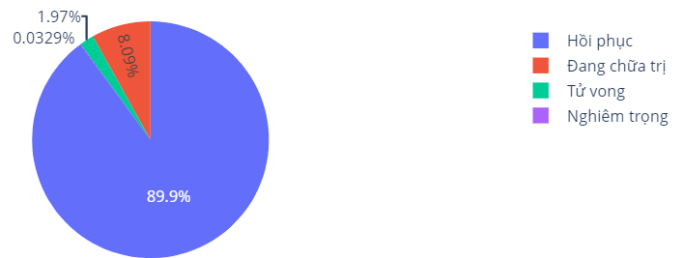
deathNum = (world['deaths'].values/world['cases'].values)*100
recoveredNum = (world['recovered'].values/world['cases'].values)*100
activeNum = (world['active'].values/world['cases'].values)*100
criticalNum = (world['critical'].values/world['cases'].values)*100

myvalues = np.array([deathNum.item(), recoveredNum.item(),
activeNum.item(), criticalNum.item()])
```

- deathNum, recoveredNum, activeNum, criticalNum là tỉ lệ phần trăm chiếm giữ của các trường tương ứng trong một bộ data lớn là cases (tính phần trăm bằng cách lấy giá trị thành phần chia tổng rồi nhân 100)

- **Kết quả:**

Biểu đồ tròn thể hiện phần trăm cơ cấu dịch Covid-19 của thế giới vào ngày 12/12/2021



- Nhận xét:

- Trường recovered (Hồi phục) chiếm tỉ trọng cao nhất do có giá trị lớn nhất

cases	todayCases	deaths	todayDeaths	recovered
269758712	345595	5316182	4454	242625329

- Trường deaths do giá trị chiếm phần quá nhỏ so với tổng nên phần hiển thị khó nhận biết nhất

cases	todayCases	deaths
269758712	345595	5316182

3.3.1.2 Tình hình dịch Covid-19 của thế giới (ngày hôm nay)

- Cũng như khi dùng API request số liệu tình hình dịch Covid-19 của thế giới, API request dùng cho quốc gia cũng trả về các trường tương tự

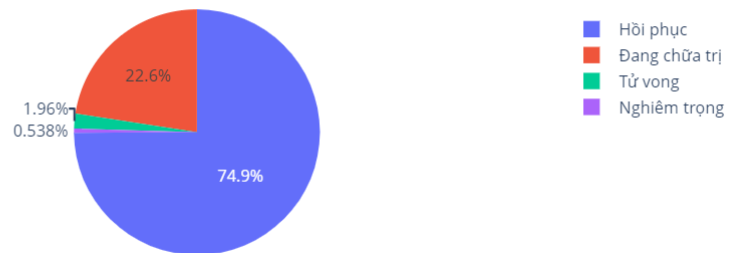
```
việtnam = pd.read_csv('data/vietnam/original/today.csv')

deathNum = (việtnam['deaths'].values/việtnam['cases'].values)*100
recoveredNum =
(việtnam['recovered'].values/việtnam['cases'].values)*100
activeNum = (việtnam['active'].values/việtnam['cases'].values)*100
criticalNum =
(việtnam['critical'].values/việtnam['cases'].values)*100

myvalues = np.array([deathNum.item(), recoveredNum.item(),
activeNum.item(), criticalNum.item()])
mylabels = ["Tử vong", "Hồi phục", "Đang chữa trị", "Nghiêm trọng"]
```

- Kết quả:

Biểu đồ tròn thể hiện phần trăm cơ cấu dịch Covid-19 của Việt Nam vào ngày 12/12/2021



- Nhận xét:

- Trường recovered (Hồi phục) chiếm tỉ trọng cao nhất do có giá trị lớn nhất

cases	todayCases	deaths	todayDeaths	recovered
1398413	16141	27611	209	1053425

- Trường deaths do giá trị chiếm phần quá nhỏ so với tổng nên phần hiển thị khó nhận biết nhất

cases	todayCases	deaths
1398413	16141	27611

3.3.1.3 Tình hình dịch Covid-19 của các châu lục (ngày hôm nay)

- Cũng như khi dùng API request số liệu tình hình dịch Covid-19 của thế giới, API request dùng cho các châu lục cũng trả về các trường tương tự

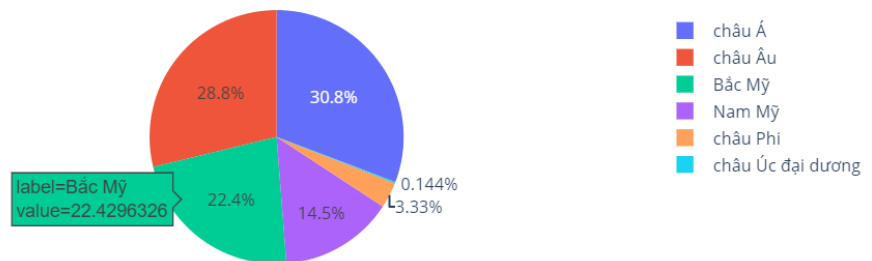
```
world = pd.read_csv('data/world/original/today.csv')

africaNum = (africa['cases'].values/world['cases'].values)*100
asiaNum = (asia['cases'].values/world['cases'].values)*100
europeNum = (europe['cases'].values/world['cases'].values)*100
naNum = (na['cases'].values/world['cases'].values)*100
oceaniaNum = (oceania['cases'].values/world['cases'].values)*100
saNum = (sa['cases'].values/world['cases'].values)*100

myvalues = np.array([africaNum.item(), asiaNum.item(),
europeNum.item(), naNum.item(), oceaniaNum.item(), saNum.item()])
mylabels = ['châu Phi', 'châu Á', 'châu Âu', 'Bắc Mỹ', 'châu Úc đại
duong', 'Nam Mỹ']
```

- Kết quả:

Biểu đồ tròn thể hiện phần trăm cơ cấu dịch Covid-19 của 6 châu lục vào ngày 12/12/2021



- Nhận xét:

- Khi xét cơ cấu tỉ lệ phần trăm tổng số ca nhiễm của mỗi châu lục góp vào tổng, ta dùng trường cases
- Châu Á có số ca nhiễm (đã xác nhận) cao nhất, chiếm tỉ trọng lớn nhất trong biểu đồ tròn.

cases	todayCases
83001797	47342

- Châu Úc đại dương có số ca nhiễm (đã xác nhận) thấp nhất, chiếm tỉ trọng nhỏ nhất trong biểu đồ tròn (387781 ca trên tổng 269758712 ca)

cases	todayCases
387781	1825

3.3.2 Line chart

3.3.2.1 Biểu thị sự thay đổi về số lượng ca nhiễm của Việt Nam trong 30 ngày gần đây

- Sau khi tiền xử lý dữ liệu file lastdays.csv chứa thông tin số liệu dịch bệnh của Việt Nam trong 30 ngày, ta chọn biểu đồ đường để thể hiện xu hướng thay đổi của dữ liệu, cụ thể ở trường cases và deaths.

	timeline	cases
1	11/11/21	1000897
2	12/11/21	1009879
3	13/11/21	1018346
4	14/11/21	1026522
5	15/11/21	1035138
6	16/11/21	1045397
7	17/11/21	1055246
8	18/11/21	1065469
9	19/11/21	1075094
10	20/11/21	1084625
11	21/11/21	1094514
12	22/11/21	1104835
13	23/11/21	11143067

- Vì có trường timeline dùng để biểu thị ngày tháng trên x-axis nên trong bước tiền xử lý, ta phải chuyển đổi các chuỗi ngày tháng này về đúng định dạng ngày tháng (sử dụng inline function `strptime`), kết quả xuất file riêng hiển thị như hình trên

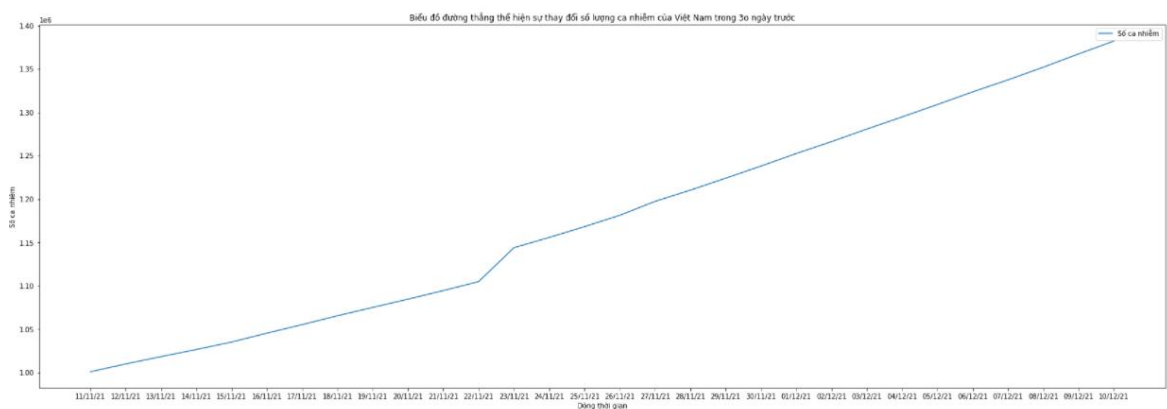
```
for key, value in data['timeline']['cases'].items():
    writer.writerow([datetime.strptime(key, '%m/%d/%y').strftime("%d/%m/%y"), value])
csv_file.close()
```

```
case = pd.read_csv('data/vietnam/modified/lastdays/cases.csv')
```

```
fig = plt.figure()
x = case['timeline']
y = case['cases']

plt.plot(x, y, label='Số ca nhiễm')
plt.xlabel('Dòng thời gian')
plt.ylabel('Số ca nhiễm')
```

- Kết quả:



- Nhận xét:

- Cột hoành (x-axis) thể hiện dòng thời gian ngày (có thể tăng số ngày lên nhưng biểu đồ sẽ bị kéo dãn đến mức không nhìn thấy)

- Cột tung (y-axis) thể hiện số ca nhiễm, theo mức độ tăng dần

3.3.2.2 Biểu thị sự thay đổi về số lượng ca tử vong của Việt Nam trong 30 ngày gần đây

- Tương tự như biểu thị số ca nhiễm, số lượng ca tử vong cũng được biểu thị theo cột gồm hai trường timeline và deaths (số ca)

	timeline	deaths
1	11/11/21	22849
2	12/11/21	22930
3	13/11/21	23018
4	14/11/21	23082
5	15/11/21	23183
6	16/11/21	23270
7	17/11/21	23337
8	18/11/21	23476
9	19/11/21	23578
10	20/11/21	23685
11	21/11/21	23761
12	22/11/21	23951
13	23/11/21	24118

- Vì có trường timeline dùng để biểu thị ngày tháng trên x-axis nên trong bước tiền xử lý, ta phải chuyển đổi các chuỗi ngày tháng này về đúng định dạng ngày tháng (sử dụng inline function `strftime`), kết quả xuất file riêng hiển thị như hình trên

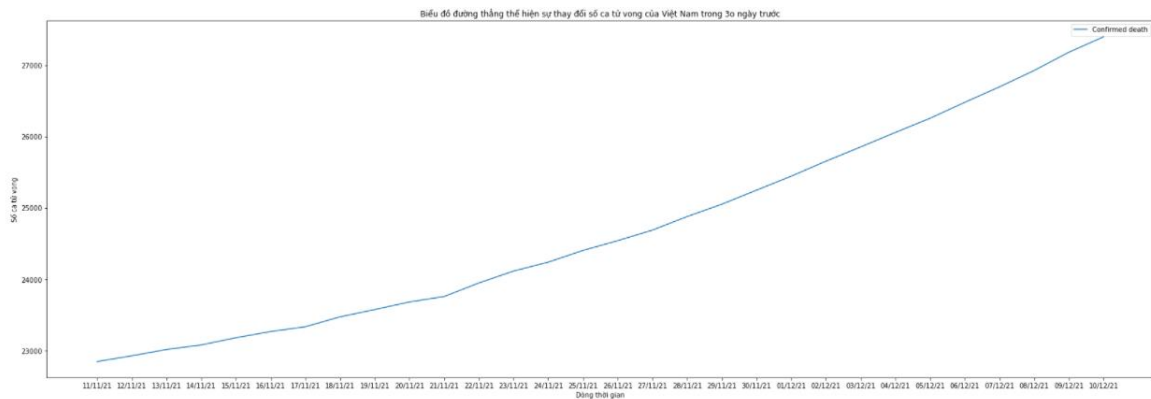
```
csv_file = open("data/vietnam/modified/lastdays/deaths.csv", "w")
writer = csv.writer(csv_file)
writer.writerow(['timeline', 'deaths'])
for key, value in data['timeline']['deaths'].items():
    writer.writerow([datetime.strptime(key, '%m/%d/%y').strftime("%d/%m/%y"), value])
csv_file.close()
```

```
death = pd.read_csv('data/vietnam/modified/lastdays/deaths.csv')

fig = plt.figure()
x = death['timeline']
y = death['deaths']

plt.plot(x, y, label = "Confirmed death")
plt.xlabel('Dòng thời gian')
plt.ylabel('Số ca tử vong')
```

- Kết quả:



- Nhận xét:

- Cột hoành (x-axis) thể hiện dòng thời gian 30 ngày (có thể tăng số ngày lên nhưng biểu đồ sẽ bị kéo giãn đến mức không nhìn thấy)
- Cột tung (y-axis) thể hiện số ca tử vong, theo mức độ tăng dần

3.3.2.3 Biểu thị sự thay đổi về số lượng ca nhiễm lẫn ca tử vong của Việt Nam trong 30 ngày gần đây

- Ta dùng dữ liệu hai trường cases và deaths để so sánh sự thay đổi của 2 trường này trong thời gian 30 ngày. Chọn biểu đồ đường để phác họa là thích hợp đối với loại dữ liệu thay đổi gắn liền với thời gian.

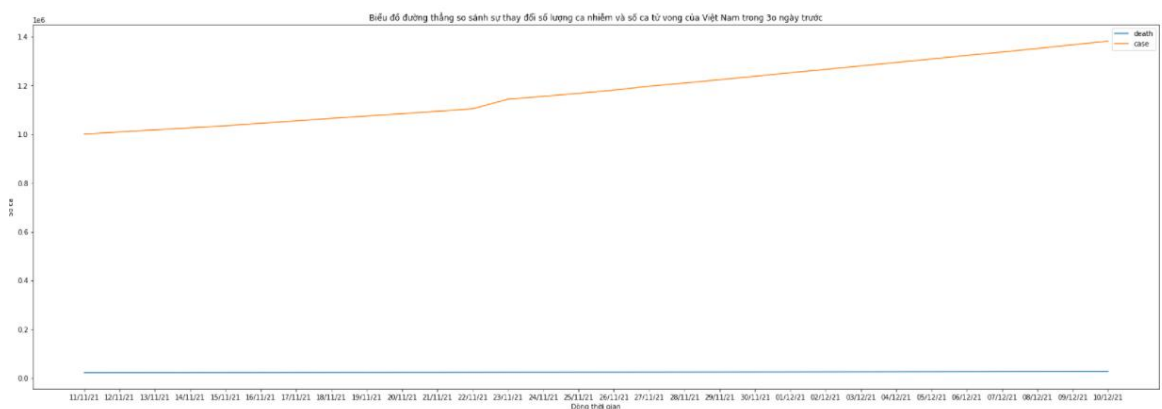
- Đọc dữ liệu từ 2 file csv và chọn 2 trường death, case

```
case = pd.read_csv('data/vietnam/modified/lastdays/cases.csv')
death = pd.read_csv('data/vietnam/modified/lastdays/deaths.csv')

fig = plt.figure()
x1 = death['timeline']
y1 = death['deaths']
plt.plot(x1, y1, label = "death")

x2 = case['timeline']
y2 = case['cases']
plt.plot(x2, y2, label = "case")
```

- Kết quả:



- Nhận xét:

- Cột hoành (x-axis) thể hiện dòng thời gian 30 ngày (có thể tăng số ngày lên nhưng biểu đồ sẽ bị kéo dãn đến mức không nhìn thấy)
- Cột tung (y-axis) thể hiện số ca tử vong, theo mức độ tăng dần
- Do số liệu ca tử vong thấp hơn nhiều so với ca nhiễm nên đường thẳng biểu thị ca tử vong xem như nằm ngang và bên dưới đường thẳng số ca nhiễm
- Dù số ca nhiễm tăng khá nhanh theo thời gian, nhưng biểu đồ cho thấy tỉ lệ tử vong không thay đổi đáng kể và quá nhỏ theo thời gian (trong 30 ngày)

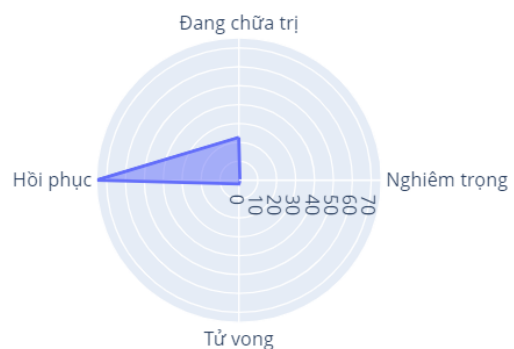
3.3.3 Radar chart

- Sử dụng 4 trường death, recovered, active, critical, ta có thể tạo ra biểu đồ mạng nhện để đánh giá tình hình dịch bệnh ở Việt Nam
- Vì giá trị dữ liệu của 4 trường nêu trên khá lớn (ở hàng trăm nghìn) nên việc biểu thị bằng giá trị phần trăm (đã chia cho tổng số) không làm thay đổi kết quả chung, ngược lại giúp biểu thị dễ hơn vì số đã được làm nhỏ đi

```
việtnam = pd.read_csv('data/vietnam/original/today.csv')  
  
deathNum = (việtnam['deaths'].values/việtnam['cases'].values)*100  
recoveredNum =  
(việtnam['recovered'].values/việtnam['cases'].values)*100  
activeNum = (việtnam['active'].values/việtnam['cases'].values)*100  
criticalNum =  
(việtnam['critical'].values/việtnam['cases'].values)*100
```

- Kết quả:

Biểu đồ mạng nhện thể hiện tình hình dịch Covid-19 ở Việt Nam trong ngày 12/12/2021



- Nhận xét:

- Dữ liệu được lấy trong ngày mốc 12/12/2021, cho thấy rằng sự khả quan của tình hình dịch bệnh Việt Nam, khi mà số ca đã hồi phục chiếm tỉ lệ rất cao (lean hẳn về trường recovered) và trường active (đang chữa trị)
- Ngược lại, ta thấy rằng tỉ lệ tử vong và nghiêm trọng đang ở mức nhỏ, không đáng kể, cho thấy rằng tình hình dịch bệnh ở Việt Nam và công tác phòng dịch vẫn đang chuyển biến khá tích cực

3.3.4 Stacked bar chart

- Biểu đồ cột chồng thích hợp cho việc so sánh nhiều trường dữ liệu (cụ thể là hai trường)
- Tại đây, ta tiếp tục sử dụng hai trường cases và deaths để biểu thị xu hướng thay đổi của hai trường này theo dòng thời gian (timeline 30 ngày)

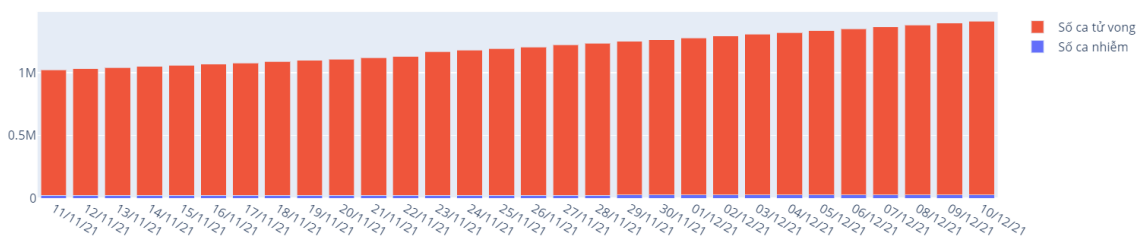
```
case = pd.read_csv('data/vietnam/modified/lastdays/cases.csv')
death = pd.read_csv('data/vietnam/modified/lastdays/deaths.csv')

x_axis=death['timeline']

fig = go.Figure(data=[
    go.Bar(name='Số ca nhiễm', x=x_axis, y=death['deaths']),
    go.Bar(name='Số ca tử vong', x=x_axis, y=case['cases']),
])
```

- Kết quả:

Biểu đồ cột chồng so sánh số lượng ca nhiễm và tử vong qua dịch Covid-19 ở Việt Nam trong 30 ngày qua



- Nhận xét:

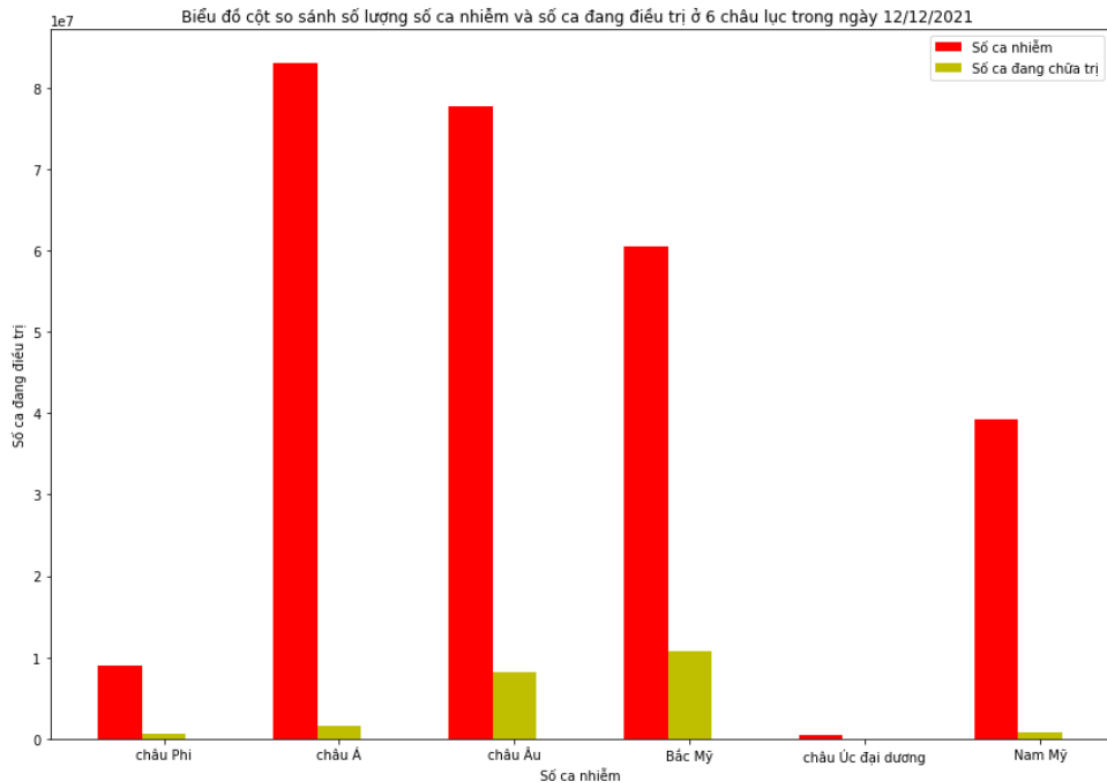
- Dữ liệu được lấy trong 30 trước ngày mốc 12/12/2021, ta thấy rằng số ca nhiễm vẫn tăng lũy tiến đều, số ca tử vong vẫn tăng nhưng ở mức rất thấp nhiều lần, không đáng kể, cho thấy rằng tình hình dịch bệnh ở Việt Nam và công tác phòng dịch vẫn đang chuyển biến khá tích cực

3.3.4 Bar chart

- Biểu đồ cột thích hợp cho việc so sánh nhiều trường dữ liệu (cụ thể là hai trường)
- Tại đây, ta tiếp tục sử dụng hai trường cases và active để biểu thị xu hướng thay đổi của hai trường này theo dòng thời gian (timeline 30 ngày)

- Dữ liệu sau đây là dữ liệu dịch bệnh của 6 châu lục trên thế giới.

- Kết quả:



- Nhận xét:

- Dữ liệu được lấy trong ngày mốc 12/12/2021, ta thấy rằng số ca nhiễm của châu Á chiếm tỉ lệ cao nhất trong tổng số ca nhiễm toàn cầu. Lý giải cho điều này là vì châu Á (cụ thể Trung Quốc) là nơi phát hiện F0 đầu tiên trên thế giới, rồi sau một thời gian dịch bệnh mới lây lan sang các châu lục khác. Cho nên số lượng ca nhiễm cao nhất cũng không phải điều gì bất thường
- Châu Âu, Bắc Mỹ có số ca nhiễm ít hơn, nhưng số ca đang điều trị và số ca tử vong cao hơn châu Á (sẽ làm rõ trong Scatter Plot) cho thấy mức độ nghiêm trọng của dịch bệnh ở các châu lục này

3.3.5 Scatter plot

- Biểu đồ phân tán sử dụng giá trị dữ liệu của các trường (cụ thể là 2 trường) để chuyển đổi thành tọa độ trên biểu đồ

- Sử dụng hai trường cases và deaths để thể hiện mức độ nghiêm trọng của dịch bệnh ở 6 châu lục, chuyển đổi các giá trị dữ liệu thành tọa độ (x,y) và phác lên biểu đồ

```
africa = pd.read_csv('data/continents/africa/original/today.csv')
asia = pd.read_csv('data/continents/asia/original/today.csv')
europe = pd.read_csv('data/continents/europe/original/today.csv')
na = pd.read_csv('data/continents/north america/original/today.csv')
oceania = pd.read_csv('data/continents/oceania/original/today.csv')
sa = pd.read_csv('data/continents/south america/original/today.csv')

x1 = africa['cases'].values
y1 = africa['deaths'].values

x2 = asia['cases'].values
y2 = asia['deaths'].values
```

```
x3 = europe['cases'].values
y3 = europe['deaths'].values

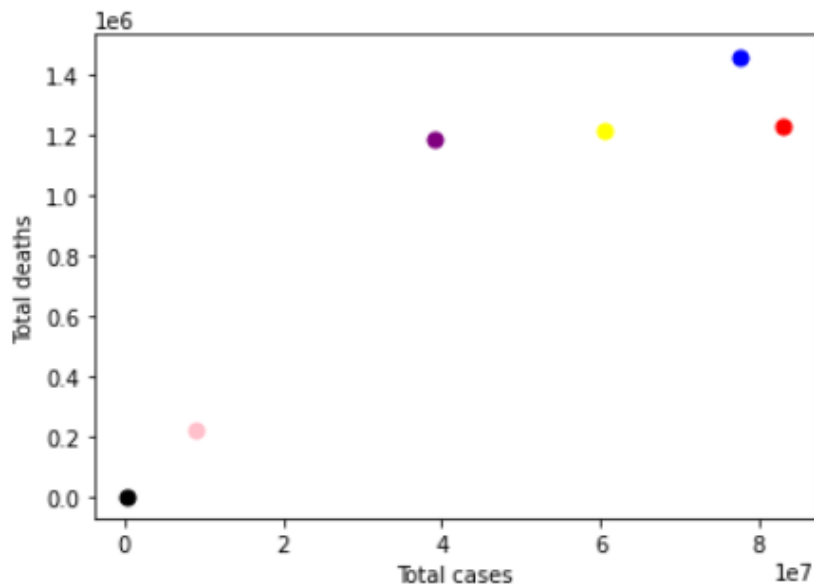
x4 = na['cases'].values
y4 = na['deaths'].values

x5 = oceania['cases'].values
y5 = oceania['deaths'].values

x6 = sa['cases'].values
y6 = sa['deaths'].values

plt.subplots_adjust(bottom = 0.1)
lb_africa=plt.scatter(x1, y1, marker='o',
c="pink",s=50,cmap=plt.get_cmap('Spectral'))
lb_asia=plt.scatter(x2, y2, marker='o',
c="red",s=50,cmap=plt.get_cmap('Spectral'))
lb_europe=plt.scatter(x3, y3, marker='o',
c="blue",s=50,cmap=plt.get_cmap('Spectral'))
lb_na=plt.scatter(x4, y4, marker='o',
c="yellow",s=50,cmap=plt.get_cmap('Spectral'))
lb_oce=plt.scatter(x5, y5, marker='o',
c="black",s=50,cmap=plt.get_cmap('Spectral'))
lb_sa=plt.scatter(x6, y6, marker='o',
c="purple",s=50,cmap=plt.get_cmap('Spectral'))
```

- Kết quả:



Biểu đồ phân tán thể hiện mối quan hệ giữa tổng số ca tử vong và tổng số ca nhiễm của 6 châu lục

● châu Phi	● châu Âu	● châu Úc đại dương
● châu Á	● Bắc Mỹ	● Nam Mỹ

- Nhận xét:

- Ở biểu đồ trước (Bar chart) thể hiện tổng số ca nhiễm và đang điều trị của 6 châu lục tính đến ngày 12/12/2021, ta thấy châu Âu và Bắc Mỹ có số ca nhiễm ít hơn và số ca đang điều trị nhiều hơn châu Á.
- Tuy nhiên, không thể kết luận tình hình dịch ở hai châu lục này chuyển hướng tích cực hơn khi biểu đồ phân tán thể hiện số ca tử vong của hai châu lục này cao kỉ lục (châu Âu vị trí đầu, theo sau là Bắc Mỹ và châu Á)
- Điều này khá nghịch lý là dù tỉ lệ đang chữa trị bệnh nhân ở mức cao nhưng đồng thời châu Âu và Bắc Mỹ cũng dẫn đầu về tỉ lệ tử vong

3.3.6 Bubble plot

- Biểu đồ bong bóng có thể thể hiện cùng lúc 3 trường dữ liệu, rất thuận tiện cho việc phân tích biểu đồ và đánh giá tình hình dịch bệnh trực quan hơn.
- Ta sử dụng 3 trường dữ liệu cases, deaths, tests (số ca nhiễm, số ca tử vong, số lượt test).
- Giống như biểu đồ phân tán, biểu đồ bong bóng cần chuyển đổi các giá trị dữ liệu thành tọa độ (x,y)

```
africa = pd.read_csv('data/continents/africa/original/today.csv')
asia = pd.read_csv('data/continents/asia/original/today.csv')
europe = pd.read_csv('data/continents/europe/original/today.csv')
na = pd.read_csv('data/continents/north america/original/today.csv')
oceania = pd.read_csv('data/continents/oceania/original/today.csv')
sa = pd.read_csv('data/continents/south america/original/today.csv')

fig = plt.figure()
ax = plt.subplots(figsize = (9, 9))

x = africa['cases'].values
y = africa['deaths'].values
z = africa['tests'].values

x1 = asia['cases'].values
y1 = asia['deaths'].values
z1 = asia['tests'].values

x2 = europe['cases'].values
y2 = europe['deaths'].values
z2 = europe['tests'].values

x3 = na['cases'].values
y3 = na['deaths'].values
z3 = na['tests'].values

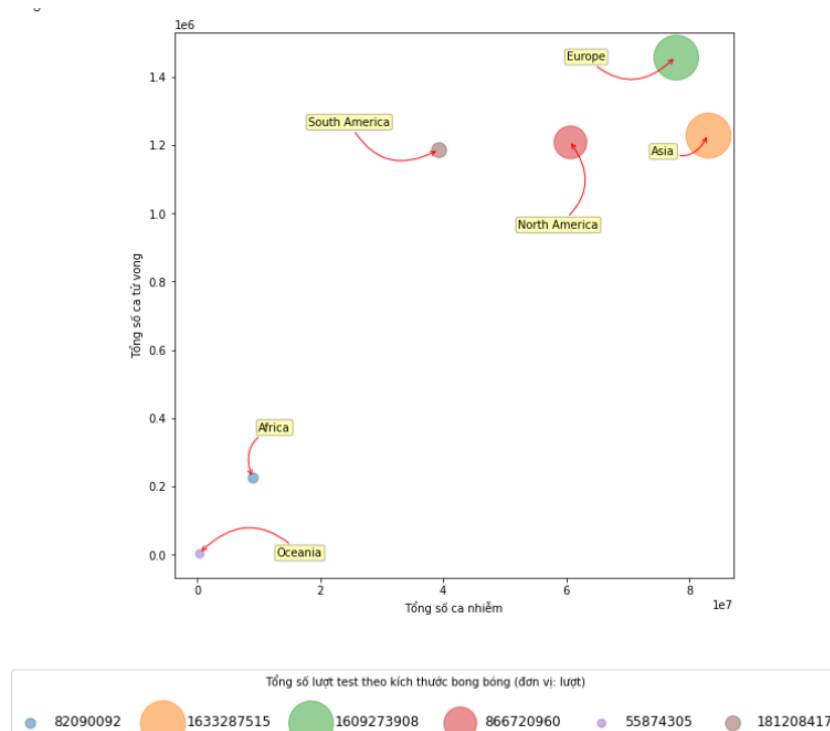
x4 = oceania['cases'].values
y4 = oceania['deaths'].values
z4 = oceania['tests'].values

x5 = sa['cases'].values
y5 = sa['deaths'].values
z5 = sa['tests'].values

# s là độ lớn của một bong bóng, tùy theo giá trị của test
lb_africa = plt.scatter(x, y, s=z/1000000, alpha=0.5)
lb_asia = plt.scatter(x1, y1, s=z1/1000000, alpha=0.5)
lb_europe = plt.scatter(x2, y2, s=z2/1000000, alpha=0.5)
lb_na = plt.scatter(x3, y3, s=z3/1000000, alpha=0.5)
```

```
lb_oce = plt.scatter(x4, y4, s=z4/1000000, alpha=0.5)
lb_sa = plt.scatter(x5, y5, s=z5/1000000, alpha=0.5)
```

- Kết quả:



- Nhận xét:

- Châu Âu dẫn đầu về giá trị của 2 trường tests, deaths. Nhưng bị tụt hạng ở trường cases (tổng số ca), đứng sau châu Á
- Số lượt test của châu Á và châu Âu xấp xỉ gần bằng nhau, dù dịch bùng ở châu Âu trễ hơn – quy mô dân số thấp hơn châu Á nhiều lần, cho thấy công tác sàng lọc bệnh nhân của châu Âu diễn ra rất tích cực và năng động (tính đến ngày mốc hiện tại là 12/12/2021)

3.3.7 Stacked area plot

- Biểu đồ cột chồng đánh giá dữ liệu nhiều trường trên các miền giá trị xếp chồng lên nhau
- Ta sử dụng tất cả 6 trường dữ liệu gồm cases, deaths, recovered, active, tests, critical (số lượng bệnh nhân nghiêm trọng)

```
africa = pd.read_csv('data/continents/africa/original/today.csv')
asia = pd.read_csv('data/continents/asia/original/today.csv')
europe = pd.read_csv('data/continents/europe/original/today.csv')
na = pd.read_csv('data/continents/north america/original/today.csv')
oceania = pd.read_csv('data/continents/oceania/original/today.csv')
sa = pd.read_csv('data/continents/south america/original/today.csv')

x=['tests','cases','recovered','active','critical','death']
```

```
y1=[africa['tests'][0],africa['cases'][0],africa['recovered'][0],africa['active'][0],africa['critical'][0],africa['deaths'][0]]

y2=[asia['tests'][0],asia['cases'][0],asia['recovered'][0],asia['active'][0],asia['critical'][0],asia['deaths'][0]]

y3=[europe['tests'][0],europe['cases'][0],europe['recovered'][0],europe['active'][0],europe['critical'][0],europe['deaths'][0]]

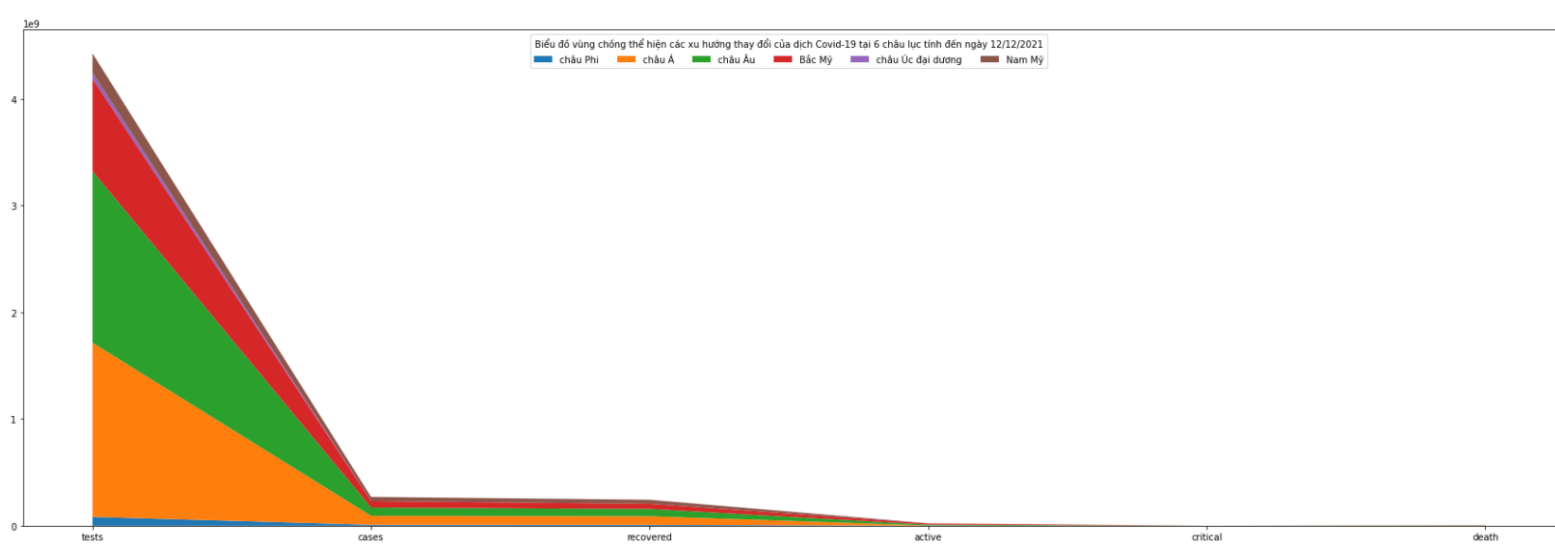
y4=[na['tests'][0],na['cases'][0],na['recovered'][0],na['active'][0],na['critical'][0],na['deaths'][0]]

y5=[oceania['tests'][0],oceania['cases'][0],oceania['recovered'][0],oceania['active'][0],oceania['critical'][0],oceania['deaths'][0]]

y6=[sa['tests'][0],sa['cases'][0],sa['recovered'][0],sa['active'][0],sa['critical'][0],sa['deaths'][0]]
```

- Trục x-axis sẽ khai báo các tên trường dữ liệu tham gia, trong khi đó y-axis sẽ là 1 danh sách (list) các giá trị, sẽ được xếp theo tầng

- **Kết quả:**



- Nhận xét:

- Sử dụng biểu đồ cột chồng là một lựa chọn đúng đắn để thể hiện trực quan nhiều trường dữ liệu
- Nhược điểm của loại biểu đồ này là phải sắp xếp vị trí các trường hợp lý để biểu thị thẩm mỹ hơn
- Ta thấy tỉ lệ nghiêm trọng (critical) và tử vong (death) của các châu lục nhìn chung là ở mức thấp, không đáng quan ngại. Tỉ lệ test sàng lọc bệnh nhân cao nhờ đó có thể ngăn chặn được các tình trạng biến chuyển nặng, góp phần giảm tỉ lệ tử vong

3.3.8 Choropleth map

- Choropleth map là một loại biểu đồ địa hình rất hữu ích cho việc phân dữ liệu theo cụm, vị trí địa lý, v.v..



- Ta sử dụng 3 trường cases, deaths, tests để biểu thị dữ liệu. Mỗi bản đồ chỉ nhận biểu thị 1 loại dữ liệu

```
datas = pd.read_csv('data/world/modified/today/cases.csv')
data = dict (
    type = 'choropleth',
    locations = datas['country'],
    locationmode='country names',
    colorscale = 'portland',
    z=datas['cases'])

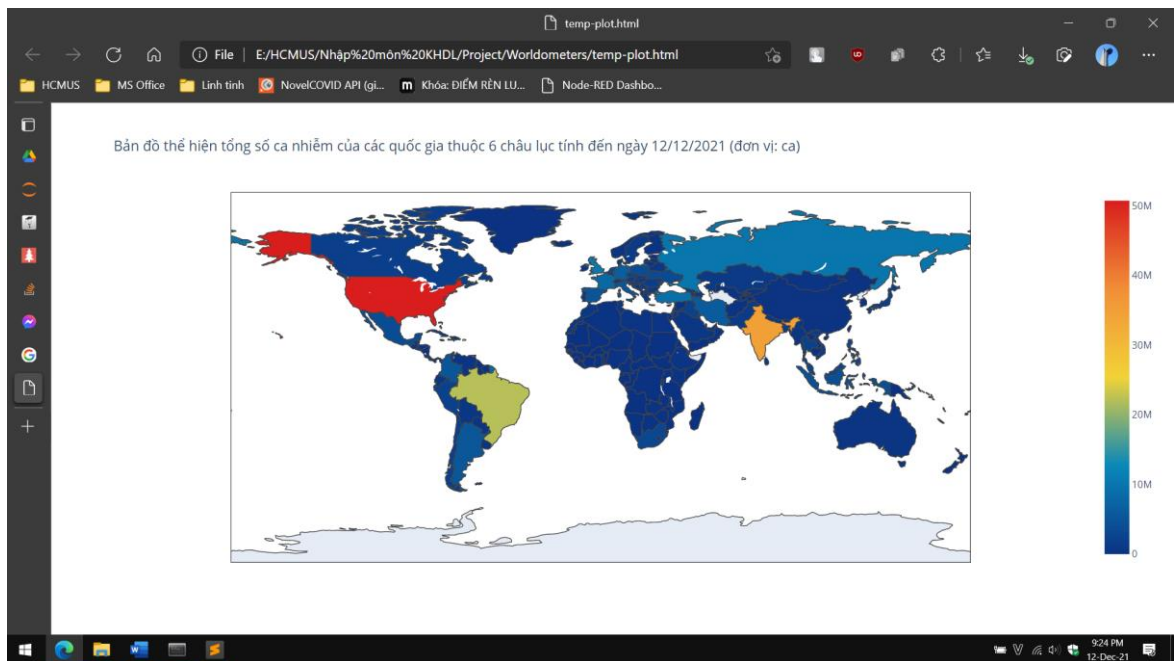
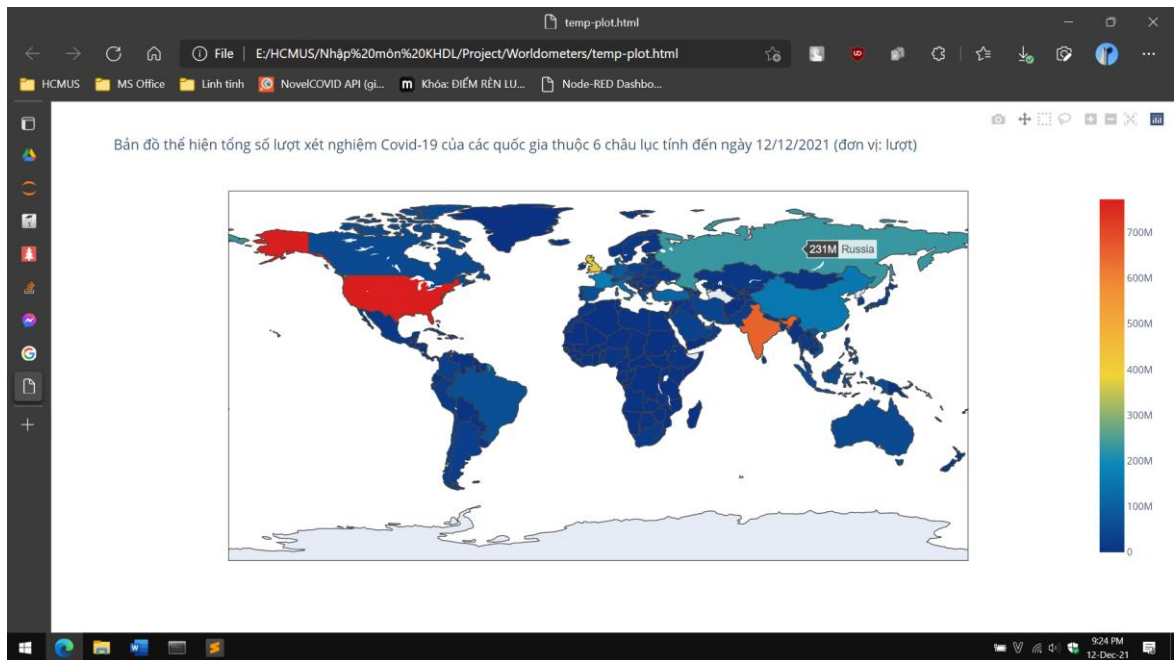
layout = go.Layout(title='Bản đồ thể hiện tổng số ca nhiễm của các
quốc gia thuộc 6 châu lục tính đến ngày 12/12/2021 (đơn vị: ca)',
    hovermode='closest', xaxis=dict(title='src freq',
type='log', autorange=True),
    yaxis=dict(title='trg freq', type='log',
autorange=True))
map = go.Figure(data=[data], layout=layout)
py.offline.plot(map)
```

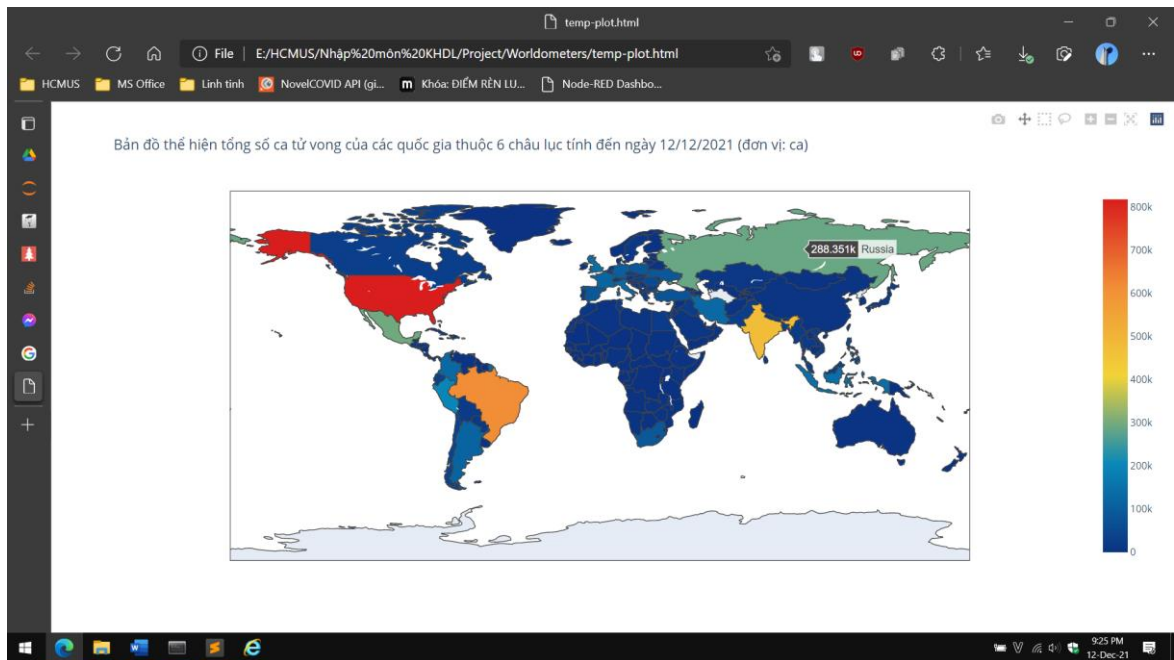
- Trong thư viện **plotly.graph_objs** có hàm Figure hỗ trợ trong việc phân bản đồ và dữ liệu, việc cần làm là khai báo 1 Python Dictionary gồm các thông tin như: **type (loại biểu đồ)**, **locations (sẽ lấy trường dữ liệu tên các nước)**, **colorscale (màu hiển thị)**, **z (dữ liệu cần phân họa)**.

- Nếu khởi chạy thành công, chương trình sẽ tạo ra 1 file html (temp-plot.html) có thể sử dụng offline, trong này là phân họa bản đồ kèm dữ liệu có thể tương tác được

 data	20 hours ago
 temp-plot.html	8 hours ago
 result icon	6 hours ago

- Kết quả:





- Nhận xét:

- Đây là dạng biểu đồ địa hình mà chúng ta có thể tương tác được. Bằng cách đưa chuột vào (cụ thể đây là 1 quốc gia) thì thông tin dữ liệu ta nạp vào trước đó cũng sẽ hiển thị
- Mức độ tùy theo giá trị của tập dữ liệu sẽ được biểu thị bằng màu sắc, có kèm thanh màu bên cạnh để tiện quan sát

Tài liệu tham khảo

[*] API Covid-19 to get data from Worldometers.info:

[NovelCOVID API \(github.com/NovelCOVID/API\)](https://github.com/NovelCOVID/API) (getpostman.com) (API version 2)

[disease.sh Docs](https://disease.sh/docs) (API version 3)