TRỰC QUAN HÓA DỮ LIỆU

ĐỒ ÁN CUỐI KÌ

TRỰC QUAN VÀ PHÂN TÍCH TÌNH HÌNH COVID TẠI VIỆT NAM

Nhóm

1753075 – Huỳnh Đoàn Minh Ngọc

1753086 – Tống Lê Thiên Phúc

1753134 – Nguyễn Ngọc Đăng Khanh

Khoa Công nghệ Thông tin

Đại học Khoa học Tự nhiên TP HCM

MỤC LỤC

1 Thông tin nhóm

MSSV	Họ và tên	Vai trò
1753075	Huỳnh Đoàn Minh Ngọc	Tải dữ liệu Trực quan dữ liệu Áp dụng mô hình học máy OLS và ARIMA (thư viện `statsmodels`) Kiểm định giả thuyết Viết báo cáo và bản trình bày
1753086	Tống Lê Thiên Phúc	Viết phần giới thiệu đề tài Phân tích dữ liệu từ biểu đồ
1753134	Nguyễn Ngọc Đăng Khanh	Áp dụng mô hình học máy (thư viện `sklearn`) Tiền xử lý dữ liệu

2 Thông tin đề tài

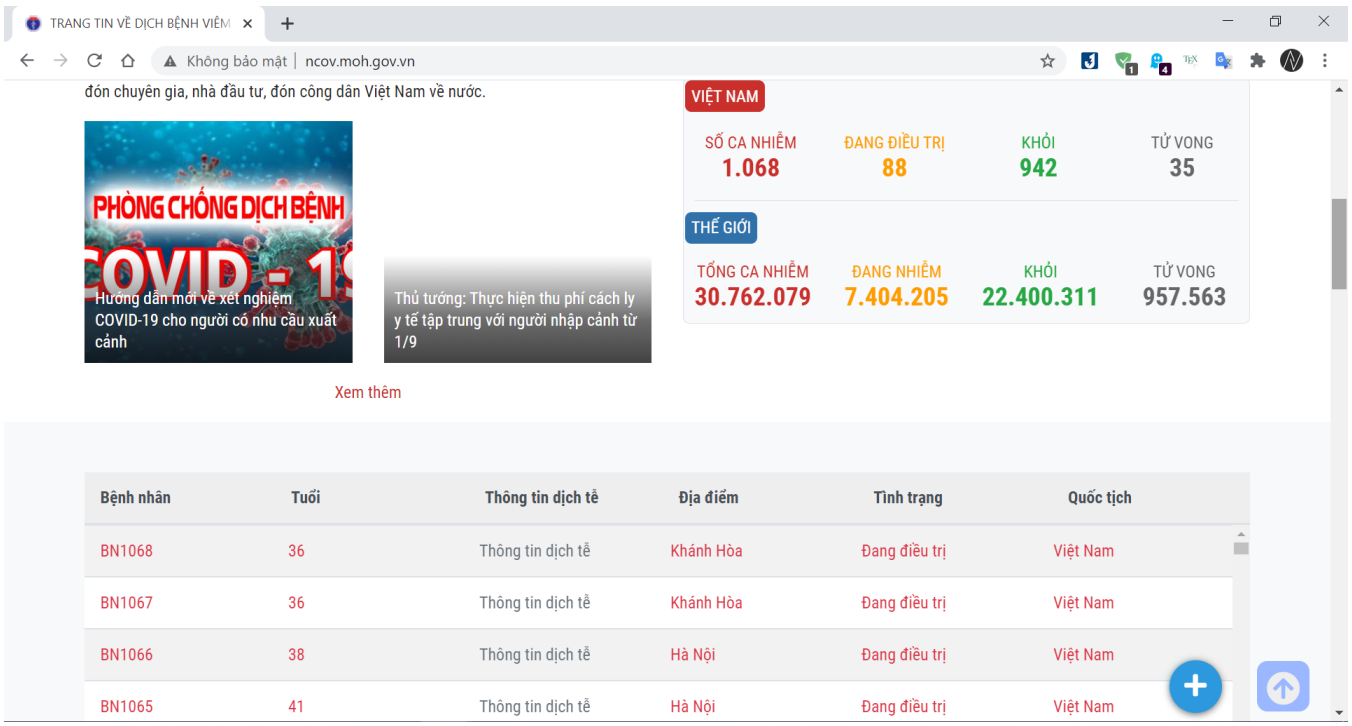
2.1 Tên đề tài

Trực quan và phân tích tình hình covid-19 tại Việt Nam.

2.2 Mô tả tổng quan đề tài

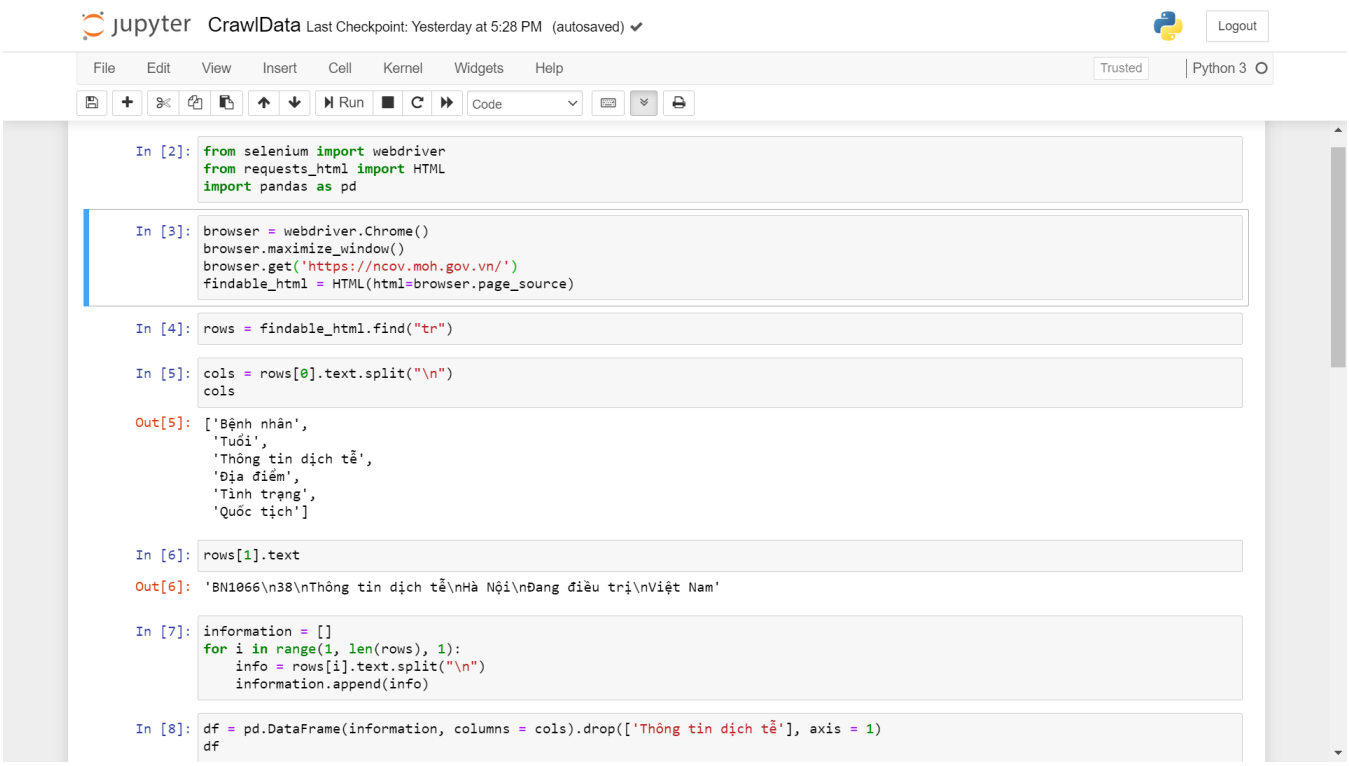
2.2.1. Mô tả dữ liệu

Dữ liệu về các bệnh nhân từ Bộ Y Tế: https://ncov.moh.gov.vn/



1. Trang web của Bộ Y Tế về dịch bệnh covid-19

Vì trang này không cho sẵn file dataset nên nhóm đã sử dụng thư viện `selenium` kết hợp với thư viện `requests` (python) để tiến hành lấy dữ liệu.



2. Source code để lấy dữ liệu

Out[8]:

	Bệnh nhân	Tuổi	Địa điểm	Tình trạng	Quốc tịch
0	BN1066	38	Hà Nội	Đang điều trị	Việt Nam
1	BN1065	41	Hà Nội	Đang điều trị	Việt Nam
2	BN1064	37	Hà Nội	Đang điều trị	Việt Nam
3	BN1063	26	Phú Yên	Đang điều trị	Việt Nam
4	BN1062	24	Phú Yên	Đang điều trị	Việt Nam
...	...	...	...	...	...
1061	BN5	23	Vĩnh Phúc	Khỏi	Việt Nam
1062	BN4	29	Vĩnh Phúc	Khỏi	Việt Nam
1063	BN3	25	Thanh Hóa	Khỏi	Việt Nam
1064	BN2	28	Hồ Chí Minh	Khỏi	Trung Quốc
1065	BN1	66	Hồ Chí Minh	Khỏi	Trung Quốc

1066 rows × 5 columns

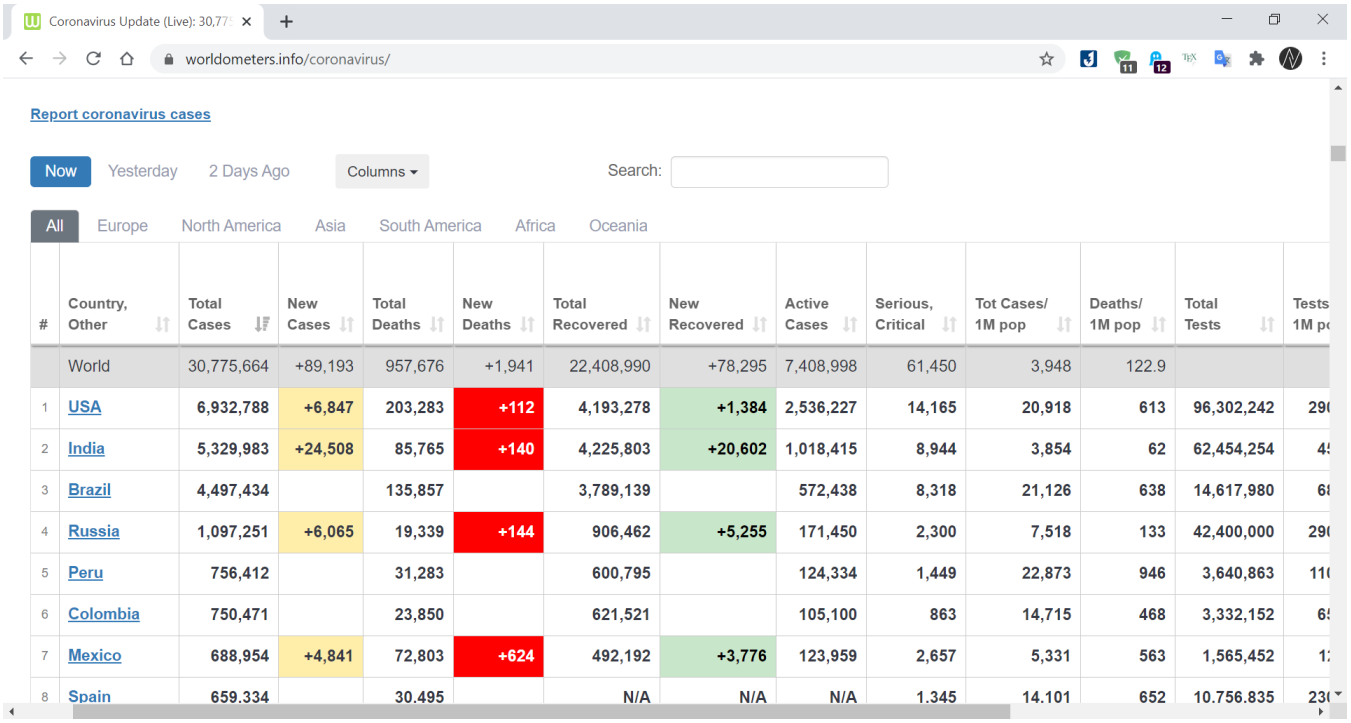
Out[9]:

	ID	Age	Province	Status	Nationality
0	BN1066	38	Hà Nội	Đang điều trị	Việt Nam
1	BN1065	41	Hà Nội	Đang điều trị	Việt Nam
2	BN1064	37	Hà Nội	Đang điều trị	Việt Nam
3	BN1063	26	Phú Yên	Đang điều trị	Việt Nam
4	BN1062	24	Phú Yên	Đang điều trị	Việt Nam
...	...	...	...	...	...
1061	BN5	23	Vĩnh Phúc	Khỏi	Việt Nam
1062	BN4	29	Vĩnh Phúc	Khỏi	Việt Nam
1063	BN3	25	Thanh Hóa	Khỏi	Việt Nam
1064	BN2	28	Hồ Chí Minh	Khỏi	Trung Quốc
1065	BN1	66	Hồ Chí Minh	Khỏi	Trung Quốc

1066 rows × 5 columns

3. Dữ liệu lấy được và lưu vào file

Dữ liệu về sự thay đổi các thông số về dịch bệnh từ trang web: https://www.worldometers.info/coronavirus/ và lấy riêng cho Việt Nam.



Report coronavirus cases

Now | Yesterday | 2 Days Ago | Columns | Search:

All | Europe | North America | Asia | South America | Africa | Oceania

#	Country, Other	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 pc
	World	30,775,664	+89,193	957,676	+1,941	22,408,990	+78,295	7,408,998	61,450	3,948	122.9		
1	USA	6,932,788	+6,847	203,283	+112	4,193,278	+1,384	2,536,227	14,165	20,918	613	96,302,242	29(
2	India	5,329,983	+24,508	85,765	+140	4,225,803	+20,602	1,018,415	8,944	3,854	62	62,454,254	4!
3	Brazil	4,497,434		135,857		3,789,139		572,438	8,318	21,126	638	14,617,980	6!
4	Russia	1,097,251	+6,065	19,339	+144	906,462	+5,255	171,450	2,300	7,518	133	42,400,000	29(
5	Peru	756,412		31,283		600,795		124,334	1,449	22,873	946	3,640,863	11(
6	Colombia	750,471		23,850		621,521		105,100	863	14,715	468	3,332,152	6!
7	Mexico	688,954	+4,841	72,803	+624	492,192	+3,776	123,959	2,657	5,331	563	1,565,452	1:
8	Spain	659,334		30,495		N/A	N/A	N/A	1,345	14,101	652	10,756,835	23(

4. Trang web thể hiện diễn biến dịch bệnh covid-19

Vì trang web chỉ hiển thị 3 ngày gần nhất nên nhóm lấy dữ liệu hằng ngày. Và để các con số cập nhật chính xác thì nhóm chỉ lấy dữ liệu sau khi ngày đó đã kết thúc (Lấy ở mục Yesterday hoặc 2 days ago).

Thư viện được sử dụng để lấy dữ liệu là `requests` và `beautifulsoup (bs4)` (python)

Sau đó, tiền xử lý dữ liệu lấy riêng dữ liệu cho Việt Nam và ghép các dữ liệu đó vào chung một file dataset.

jupyter Crawl Data Last Checkpoint: Yesterday at 5:24 PM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run | Markdown

In [1]:
```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import requests
import os
import datetime
import lxml
from bs4 import BeautifulSoup
```

In [2]:
```
def text_to_int(str):
    return str.replace('+','').replace(',','').replace('.','').replace(' ','').replace('N/A','')
def gettingData(date=0):
    url = 'https://www.worldometers.info/coronavirus/'
    req = requests.get(url)
    html_source = req.text

    # parsing it to beautiful soup
    soup = BeautifulSoup(html_source, 'html.parser')
    '''
    table = soup.find_all("tbody")
    table_rows = table[date*3].find_all('tr')
    '''
    table_rows = soup.findAll("table", {"id": date})[0].findAll('tr')[9:9+215]

    data = []
    another = []

    for i in range(len(table_rows)):
        row = []
        tag = table_rows[i].find(name = 'a', href = True)
        if tag is not None:
            row.append(tag.get_text()) #country
            td = table_rows[i].find_all(name='td')
            for j in range(2, 19, 1):
                if j != 15:
                    row.append(text_to_int(td[j].text.strip()))

            region = td[15].text.strip()
            if len(region): row.append(region)
            else: row.append("N/A")
```

5. Source code để lấy dữ liệu

AutoSave Off | CoronaVirusDate20200917.csv | Search | Huynh Doan Minh Ngoc HD

File Home Insert Page Layout Formulas Data Review View Help | Share | Comments

A1

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1		Country	Total Case	New Case	Total Deat	New Deati	Total Recc	New Reco	Active Cas	Serious, Ci	Tot Cases/	Deaths/1N	Total Test	Tests/1M	Populatior	1 Case eve	1 Death ev	1 Test eve	Region
2	0	USA	6874596	46295	202213	879	4155039	35881	2517344	14227	20743	610	95235022	287354	3.31E+08	48	1639	3	North Americ
3	1	India	5212686	96793	84404	1174	4109828	87779	1018454	8944	3769	61	60565728	43796	1.38E+09	265	16384	23	Asia
4	2	Brazil	4457443	35757	135031	857	3753082	32770	569330	8318	20938	634	14617980	68666	2.13E+08	48	1577	15	South Americ
5	3	Russia	1085281	5762	19061	144	895868	5754	170352	2300	7436	131	41748928	286053	1.46E+08	134	7657	3	Europe
6	4	Peru	750098	5698	31146	95	594513	6796	124439	1439	22683	942	3614738	109309	33069039	44	1062	9	South Americ
7	5	Colombia	743945	7568	23665	187	615457	5379	104823	863	14588	464	3298415	64677	50998462	69	2155	15	South Americ
8	6	Mexico	680931	4444	71978	300	485024	3956	123929	2747	5269	557	1545572	11961	1.29E+08	190	1795	84	North Americ
9	7	South Afri	655572	2128	15772	67	585303	1108	54497	539	11024	265	3983533	66987	59467369	91	3770	15	Africa
10	8	Spain	625651	11291	30405	162	0	0	0	1331	13380	650	10756835	230050	46758719	75	1538	4	Europe
11	9	Argentina	601713	12701	12460	344	456347	8084	132906	3108	13287	275	1653616	36516	45284429	75	3634	27	South Americ
12	10	Chile	441150	1863	12142	84	413928	2075	15080	895	23035	634	2942651	153653	19151322	43	1577	7	South Americ
13	11	France	415481	10593	31095	50	90840	505	293546	800	6362	476	10000000	153128	65304832	157	2100	7	Europe
14	12	Iran	413149	2815	23808	176	353848	1829	35493	3848	4906	283	3667551	43547	84220988	204	3538	23	Asia
15	13	UK	381614	3395	41705	21	0	0	0	124	5615	614	20521243	301948	67962867	178	1630	3	Europe
16	14	Banglades	344264	1593	4859	36	250412	2443	88993	0	2086	29	1783779	10808	1.65E+08	479	33966	93	Asia
17	15	Saudi Arat	328144	593	4399	30	307207	1203	16538	1180	9395	126	5917184	169410	34928178	106	7940	6	Asia
18	16	Iraq	307385	4326	8332	84	241100	3859	57953	586	7607	206	1998295	49451	40409852	131	4850	20	Asia
19	17	Pakistan	303634	545	6399	6	291169	409	6066	593	1369	29	3056795	13782	2.22E+08	730	34660	73	Asia
20	18	Turkey	298039	1648	7315	66	263745	1143	26979	1372	3526	87	8965165	106056	84532416	284	11556	9	Asia
21	19	Italy	293025	1585	35658	13	215954	689	41413	212	4848	590	10146324	167868	60442405	206	1695	6	Europe
22	20	Philippine	276269	3355	4783	51	208057	278	63429	1048	2514	44	3234856	29437	1.1E+08	398	22975	34	Asia
23	21	Germany	269042	2177	9457	8	241300	2200	18285	237	3209	113	14557136	173626	83841923	312	8866	6	Europe
24	22	Indonesia	232628	3635	9222	122	166686	2585	56720	0	849	34	2796924	10203	2.74E+08	1178	29727	98	Asia
25	23	Israel	175256	4791	1169	8	126329	3110	47758	573	19055	127	2931763	318753	9197590	52	7868	3	Asia
26	24	Ukraine	166244	3584	3400	60	73913	1589	88931	177	3806	78	1925482	44086	43675866	263	12846	23	Europe

CoronaVirusDate20200917

100%

6. Dữ liệu lấy được theo từng ngày

jupyter VietNam_CoronaWorldometer Last Checkpoint: 4 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [4]:
```
df = pd.read_csv(os.path.join(path, listfile[0]))

df = df.loc[df['Country']=='Vietnam']
df
```

Out[4]:

	Unnamed: 0	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	Cas ever X pp
154	154	Vietnam	1007	13.0	25.0	0.0	542.0	9.0	440.0	0.0	10	3.0	817208.0	8385.0	97461327	9678

In [5]:
```
df_new = df
df_new.insert(2, 'Date', listrow[0])
```

In [6]:
```
df_new
```

Out[6]:

	Unnamed: 0	Country	Date	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	Popula
154	154	Vietnam	Date2008	1007	13.0	25.0	0.0	542.0	9.0	440.0	0.0	10	3.0	817208.0	8385.0	9746

In [7]:
```
#df_new
```

In [8]:
```
for i in range(1,len(listfile)):
    df_temp = pd.read_csv(os.path.join(path, listfile[i]))
    selected_row = df_temp.loc[df_temp['Country']=='Vietnam']
    selected_row.insert(2, 'Date', listrow[i])
    #print(selected_row)
    df_new = df_new.append(selected_row)
```

In [9]:
```
df_new
```

7. Source code lọc dữ liệu

AutoSave Off | Vietnam_Corona_Worldometer.csv | Search | Huynh Doan Minh Ngoc HD

File Home Insert Page Layout Formulas Data Review View Help Share Comments

A1

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1		Date	Total Case	New Case	Total Deat	New Deat	Total Reco	New Reco	Active Cas	Serious, C	Tot Cases/	Deaths/1M	Total Test	Tests/1M	1 Case eve	1 Death ev	1 Test eve	Tot Cases/1M	ppl
2	0	Date0809	841	29	11	1	395	0	435	0	9	1	482456	4952	115850	8857296	202	0	
3	1	Date0810	841	0	13	2	399	4	429	0	9	0	482456	4952	115853	7494819	202	0	
4	2	Date0811	866	19	16	1	399	0	451	0	9	2	482456	4951	112517	6089988	202	0	
5	3	Date0812	883	17	17	1	409	10	457	0	9	2	621823	6382	110351	5731754	157	0	
6	4	Date0813	911	28	21	4	425	16	465	0	9	2	621823	6381	106962	4640105	157	0	
7	5	Date0814	930	19	22	1	437	12	471	0	10	2	621823	6381	104779	4429300	157	0	
8	6	Date0815	951	21	23	1	447	10	481	0	10	2	621823	6381	102468	4236825	157	0	
9	7	Date0816	962	11	24	1	456	9	482	0	0	0	723596	7425	101299	4060391	135	10	
10	8	Date0817	962	11	24	1	456	9	482	0	10	2	723596	7425	101304	4060590	135	0	
11	9	Date0818	983	21	24	0	467	11	492	0	10	2	723596	7425	99140	4060590	135	0	
12	10	Date0819	994	5	25	0	533	7	436	0	10	3	723596	7425	98045	3898262	135	0	
13	11	Date0820	1007	13	25	0	542	9	440	0	10	3	817208	8385	96784	3898453	119	0	
14	12	Date0821	1009	2	25	0	545	3	439	0	10	3	817208	8385	96592	3898453	119	0	
15	13	Date0822	1014	5	26	1	563	18	425	0	10	3	817208	8385	96118	3748605	119	0	
16	14	Date0823	1016	2	27	1	563	0	426	0	10	3	817208	8385	95931	3609856	119	0	
17	15	Date0824	1022	6	27	0	587	24	408	0	10	3	817208	8384	95370	3609944	119	0	
18	16	Date0825	1029	7	27	0	592	5	410	0	11	3	1009145	10353	94724	3610033	97	0	
19	17	Date0826	1034	5	29	2	632	40	373	0	11	3	1009145	10353	94268	3361148	97	0	
20	18	Date0827	1036	2	30	1	637	5	369	0	11	3	1009145	10353	94091	3249269	97	0	
21	19	Date0828	1038	2	30	0	663	26	345	0	11	3	1009145	10353	93910	3249269	97	0	
22	20	Date0829	1040	2	32	2	677	14	331	0	11	3	1009145	10352	93734	3046339	97	0	
23	21	Date0830	1040	0	32	0	695	18	313	0	11	3	1009145	10352	93734	3046339	97	0	
24	22	Date0831	1044	4	34	2	707	12	303	0	11	3	1009145	10352	93377	2867213	97	0	
25	23	Date0901	1044	0	34	0	735	28	275	0	11	3	1009145	10351	93381	2867353	97	0	
26	24	Date0902	1046	2	34	0	746	11	266	0	11	3	1009145	10351	93203	2867353	97	0	

Vietnam_Corona_Worldometer

100%

8. Dữ liệu thu thập được sau khi lọc

2.2.2. Đôi nét về đề tài

Về dịch SARS-COV-2 (Covid-19)

Virus Corona là một họ virus lớn thường lây nhiễm cho động vật nhưng đôi khi chúng có thể tiến hóa và lây sang người. Khi virus xâm nhập vào cơ thể, nó xâm nhập vào một số tế bào và chiếm lấy bộ máy tế bào (gây tổn thương viêm đặc hiệu ở đường hô hấp), đồng thời virus chuyển hướng bộ máy đó để phục vụ cho nó, tạo ra virus mới và nhiễm tiếp người khác.

Có 7 loại virus Corona, trong đó, 4 loại không nguy hiểm là 229E, NL63, OC43 và HKU1; hai loại khác là Hội chứng hô hấp Trung Đông (MERS) và Hội chứng hô hấp cấp tính nặng (SARS), nguy hiểm hơn và từng gây ra đại dịch toàn cầu. Bên cạnh đó, còn một loại virus Corona thuộc chủng mới (ký hiệu SARS-CoV-2, còn được gọi với cái tên "Virus Vũ Hán") đang "tung hoành" trong những ngày này. SARS-CoV-2 là tác nhân gây ra bệnh viêm phổi cấp, khiến hàng nghìn người nhiễm bệnh và làm số ca tử vong không ngừng tăng lên từng ngày.

Novel Coronavirus 2019 hay SARS-CoV-2 – chủng virus corona gây dịch tại Vũ Hán hay virus Vũ Hán là một loại virus đường hô hấp mới thuộc "gia đình" vi rút Corona gây bệnh viêm đường hô hấp cấp ở người và cho thấy có sự lây lan từ người sang người. Virus này là chủng virus mới chưa được xác định trước đó. SARS-CoV-2 lần đầu tiên được xác định ở Vũ Hán, tỉnh Hồ Bắc, Trung Quốc vào tháng 12 năm 2019.

Hầu hết các loại SARS-CoV-2 có con đường lây truyền giống như những loại virus gây cảm lạnh khác, đó là:

- Người bệnh ho và hắt hơi mà không che miệng, dẫn tới phát tán các giọt nước vào không khí, làm lây lan virus sang người khỏe mạnh.
- Người khỏe mạnh chạm hoặc bắt tay với người có SARS-CoV-2 khiến virus truyền từ người này sang người khác.
- Người khỏe mạnh tiếp xúc với một bề mặt hoặc vật thể có virus, sau đó đưa tay lên mũi, mắt hoặc miệng của mình.

Trong những trường hợp hiếm hoi, virus Corona có thể lây lan qua tiếp xúc với phân.

Virus này ban đầu có thể xuất hiện từ nguồn động vật nhưng hiện nay đã lây lan từ người sang người. Các nhà khoa học Trung Quốc cho biết, trung bình một bệnh nhân nhiễm SARS-CoV-2 sẽ lây lan sang 5,5 người khác. Chính vì SARS-CoV-2 có khả năng

lan truyền rất nhanh từ người sang người, nên nếu người dân không được trang bị kiến thức về phòng chống bệnh, đại dịch rất dễ xảy ra.

Về nội dung đồ án môn học

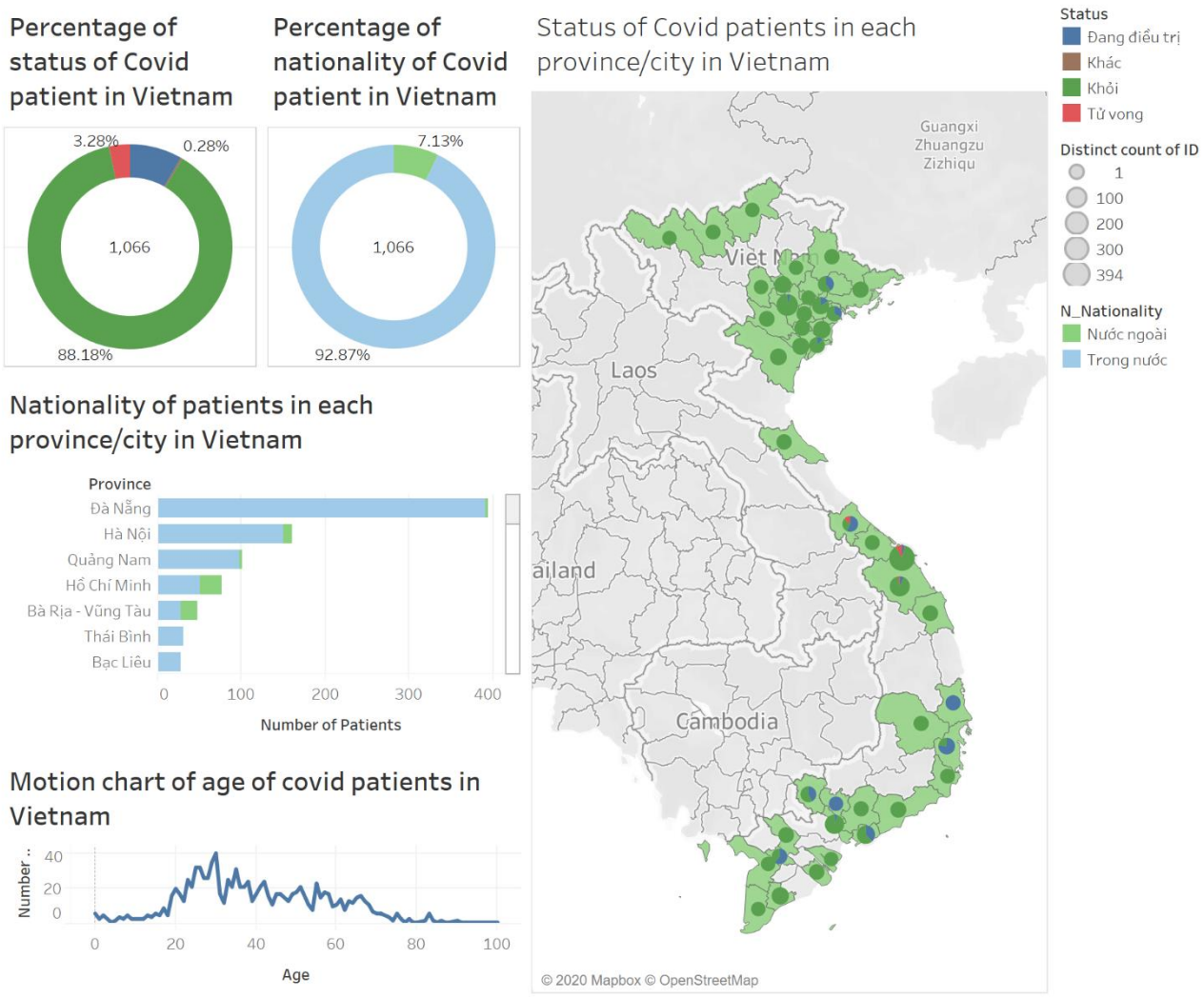
Trong quá trình học môn học này, nhóm đã học được các kỹ thuật để trực quan hóa dữ liệu khác nhau, đặc biệt là khả năng có thể đọc và phân tích những gì đã được trực quan. Thông qua môn học, nhóm sinh viên đã có được một cái nhìn khách quan hơn và học được về tầm quan trọng của một lĩnh vực trước đây chưa từng nghĩ đến, và hiểu sự liên kết của việc trực quan hóa dữ liệu đối với một bài toán thực tế.

Hiện tại, vấn đề nóng bỏng và đạt được nhiều sự quan tâm của xã hội nhất vào lúc này không gì khác ngoài tình hình của đại dịch Sars-Cov-2 hay COVID-19. Nhưng ngoài tổng số ca nhiễm và số ca nhiễm mới hằng ngày, rất khó để có thể theo dõi sát sao và một cách trực quan về những yếu tố khác như phân bố các vùng dịch, độ tuổi những người mắc bệnh,..

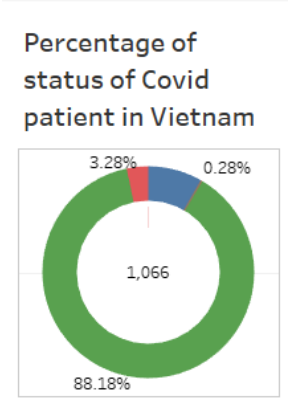
Thông qua những gì đã được học trong môn học, nhóm sinh viên muốn có thể đem đến cho mọi người một cái nhìn trực quan và dễ hiểu về các dữ liệu mà mọi người có thể dễ dàng theo dõi diễn biến dịch bệnh tại Việt Nam. Bên cạnh đó là phân tích các số liệu thống kê và dự đoán tình hình sắp tới của dịch, để từ đó có thể biết được khả năng chống dịch của nước ta hiện tại là như thế nào.

3 Trực quan hóa và phân tích dữ liệu

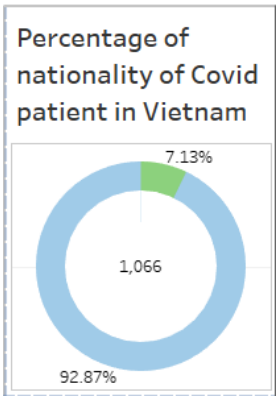
3.1 Dashboard



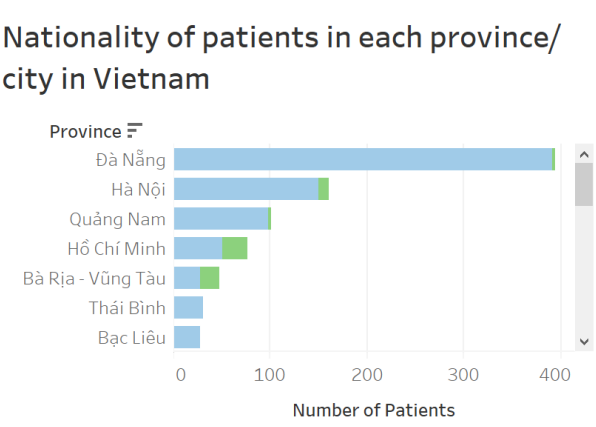
9. Dashboard



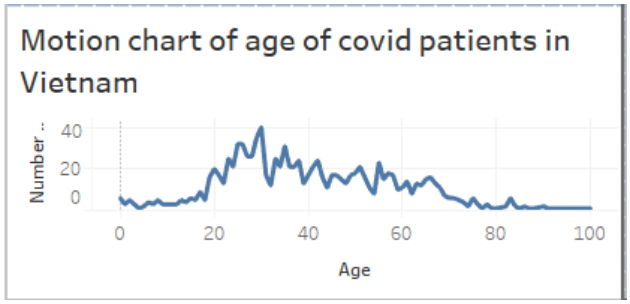
Biểu đồ tròn thể hiện tỷ lệ của các tình trạng bệnh nhân Covid-19 tại Việt Nam, trong đấy chiếm tỷ lệ cao nhất là các bệnh nhân đã được chữa khỏi với tỷ lệ rất cao là 88.18% hay cụ thể là 940 bệnh nhân đã được chữa khỏi, tiếp đến là các bệnh nhân đang điều trị chiếm tỷ lệ 8.26%, số lượng bệnh nhận tử vong cả do COVID-19 lẫn không do virus này đều ở mức khá thấp, lần lượt là 3.28% và 0.28%.



Biểu đồ tròn thể hiện phân bố về quốc tịch của các bệnh nhân tại Việt Nam. Khá rõ ràng với việc chúng ta cố gắng đóng của biên giới và kiểm soát dịch từ ngoài nước vào nên tỷ lệ số người bệnh là người nước ngoài rất thấp so với người VN, cụ thể chỉ chiếm 7.13% trong tổng số 1066 người bệnh tính đến thời điểm được lấy dữ liệu.

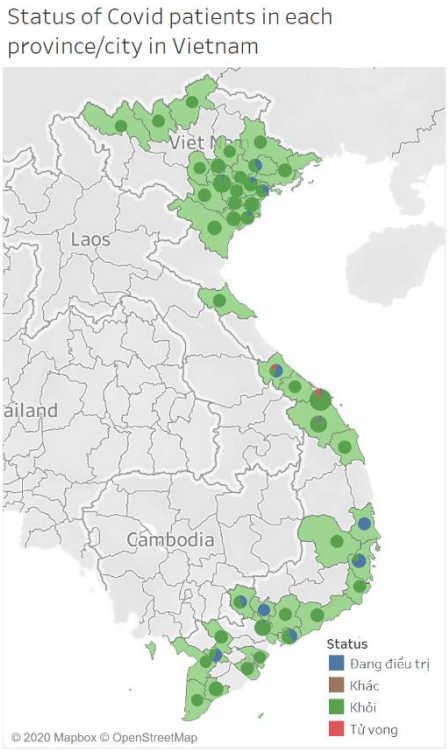


Biểu đồ thống kê quốc tịch của các bệnh nhân ở Việt Nam, theo biểu đồ này có thể thấy rõ, những nơi có lượng bệnh nhân có quốc tịch không phải người Việt Nam thường là các thành phố lớn như Đà Nẵng, Hà Nội, Thành phố Hồ Chí Minh hoặc địa điểm du lịch nổi tiếng và có nhiều du khách nước ngoài như Vũng Tàu và tỷ lệ thường không chiếm đa số. Thành phố có nhiều người nước ngoài bị mắc bệnh nhất là Thành phố Hồ Chí Minh với 26 ca. Có một trường hợp đặc biệt đó là tỉnh Quảng Ninh, tỉnh này có số người mắc bệnh là người nước ngoài cao hơn người trong nước, cụ thể có 6 ca mắc bệnh ở Quảng Ninh là người nước ngoài trong khi người Việt Nam mắc bệnh ở đây là 5 ca.

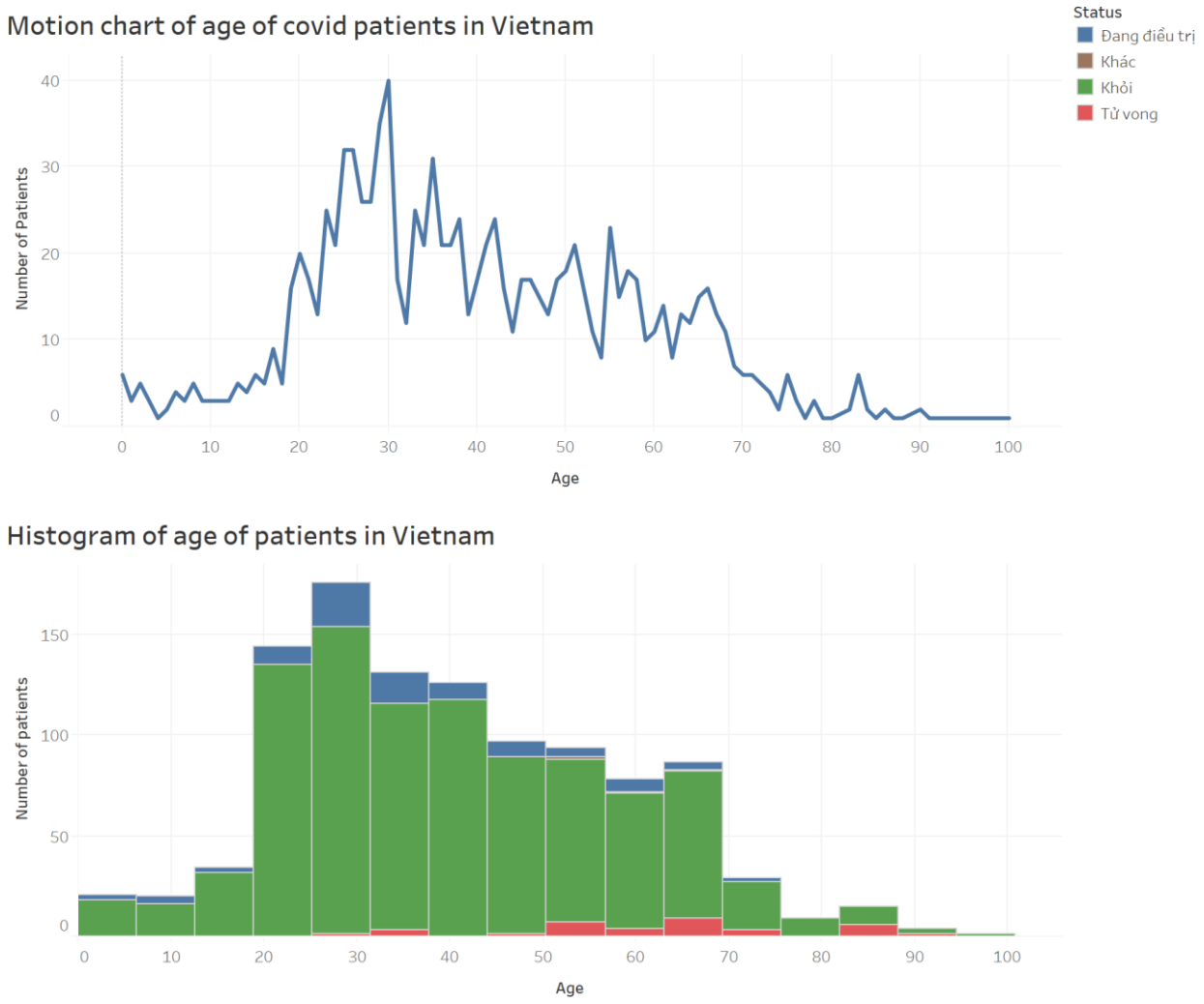


Biểu đồ thể hiện phân bố về độ tuổi của các bệnh nhân mắc COVID-19 ở Việt Nam, có thể thấy rõ ràng rằng bệnh từ trẻ em vài tháng tuổi cho đến các cụ già gần 100 tuổi đều có ca mắc bệnh, nên COVID-19 không phải bệnh dành cho một 1 độ tuổi nhất định, và khác với nhận định ban đầu của mọi người, tỷ lệ người trẻ tuổi (10-30 tuổi) cũng có tỷ lệ mắc bệnh cao không khác gì những lứa tuổi lớn hơn,

Bản đồ phân bố tình hình bệnh của Việt Nam, có thể thấy rõ ổ dịch lớn nhất nước là nằm ở Đà Nẵng với số lượng tử vong đủ để thể hiện rõ ràng trên bản đồ, bên cạnh đó, các tỉnh, khu vực nằm xung quanh các thành phố lớn như Hà Nội, Thành phố Hồ Chí Minh đều bị ảnh hưởng nặng nề. Bên cạnh tâm dịch Đà Nặng, rõ ràng nhận thấy khu vực đồng bằng Sông Hồng, đồng bằng Sông Cửu Long, vừng Đông Nam Bộ và vùng Tây Bắc là những nơi có lượng mắc bệnh tập trung thành khu vực đáng chú ý nhất.



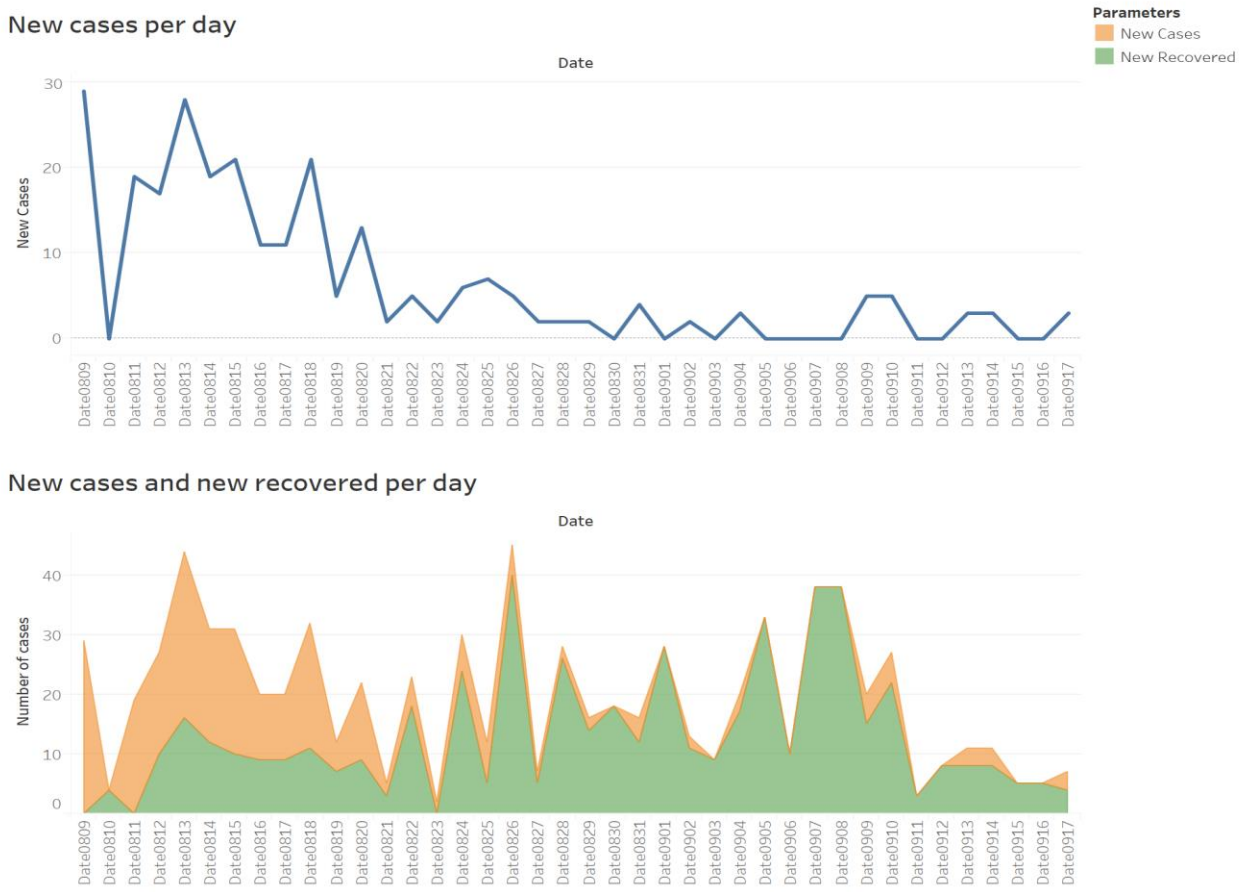
3.2 Dashboard of Age



10. Dashboard of Age

Histogram dưới đây thể hiện mức độ phân bố của độ tuổi cũng như tình trạng của các bệnh nhân. Một lần nữa ta thấy được, bệnh được phủ rộng qua các độ tuổi, không nhất thiết phải là người lớn tuổi mới là đối tượng bị bệnh. Ngược lại, số lượng những người bị bệnh từ 20 đến 30 tuổi chiếm số lượng vượt trội so với các lứa tuổi khác. Thế nhưng, đa phần các trường hợp tử vòng thì lại rơi vào những người có độ tuổi trên 50 tuổi, với một ngoại lệ duy nhất. Và như các thông số đã nêu từ trước, số ca được chữa khỏi vẫn chiếm đa số trong tất cả các độ tuổi

3.3 Dashboard of New cases

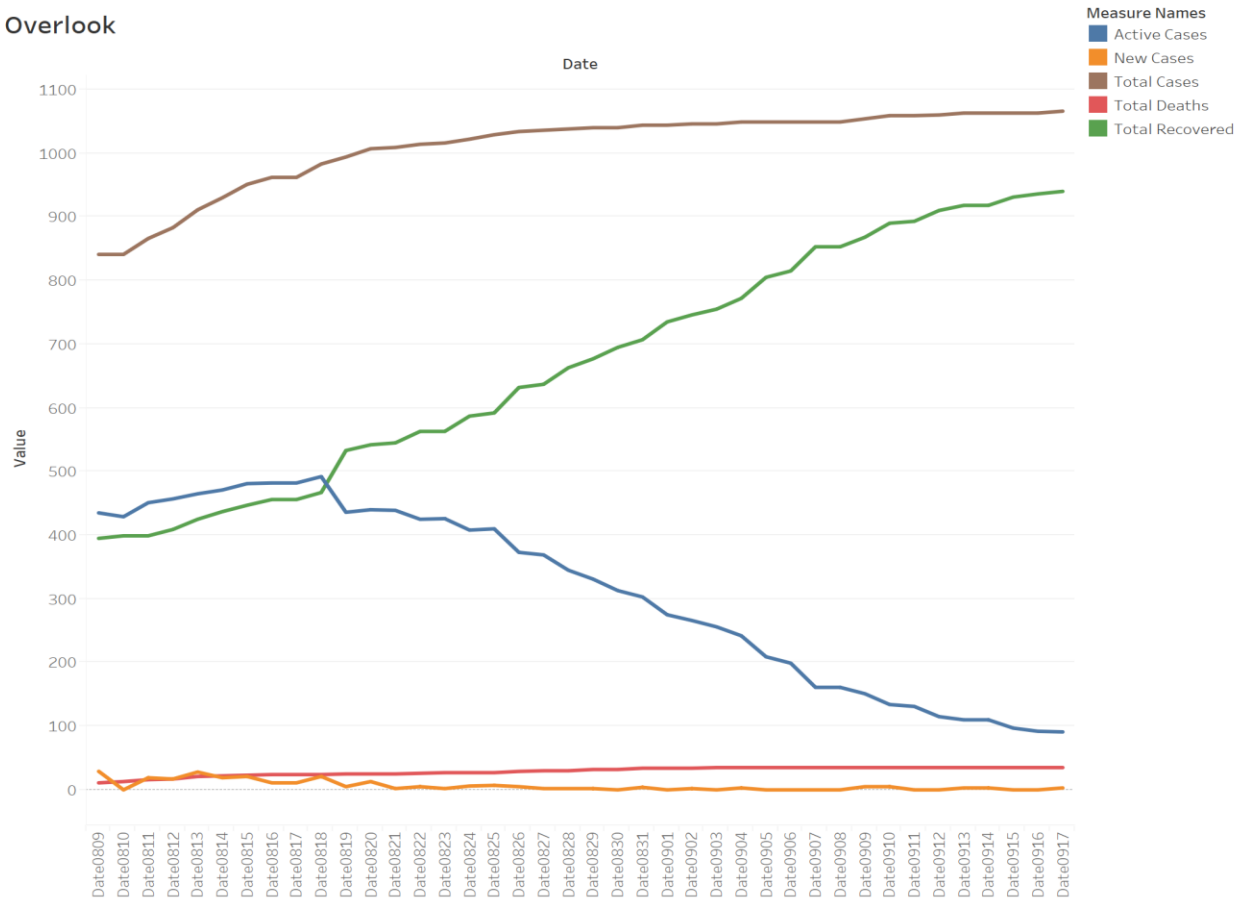


11. Dashboard of New Cases

Biểu đồ đường thể hiện biến đông trong số ca nhiễm mới mỗi ngày trong hơn 1 tháng từ ngày 9/8 đến ngày 17/9. Có thể thấy được rằng trong khoảng 10 ngày đầu tiên số ca nhiễm mới tăng một cách đột biến, số lượng là rất lớn, sau đấy số lượng tăng theo ngày giảm dần, đến những ngày cuối thì rất ít, có nhiều ngày liền không có ca mắc mới.

Biểu đồ thứ hai: thể hiện sự tăng giảm của số nhiễm mới và số ca được trị khỏi trong khoảng thời gian xét dữ liệu. Có thể chi biểu đồ thành 2 giai đoạn, giải đoạn trước 21/8 và sau mốc thời gian trên. Trước ngày 21/8 có thể thấy qua từng ngày dù có biến động lớn giữa 2 giá trị nhưng chênh lệch vẫn là rất lớn và đáng kể. Thế nhưng sau mốc thời gian này, gần như không có sự khác biệt giữa 2 giá trị trên, gần như là tương đồng, trong đó về số ca nhiễm mới vẫn ít nhiều cao hơn số ca hồi phục mỗi ngày, dù chênh lệch là không còn lớn nữa.

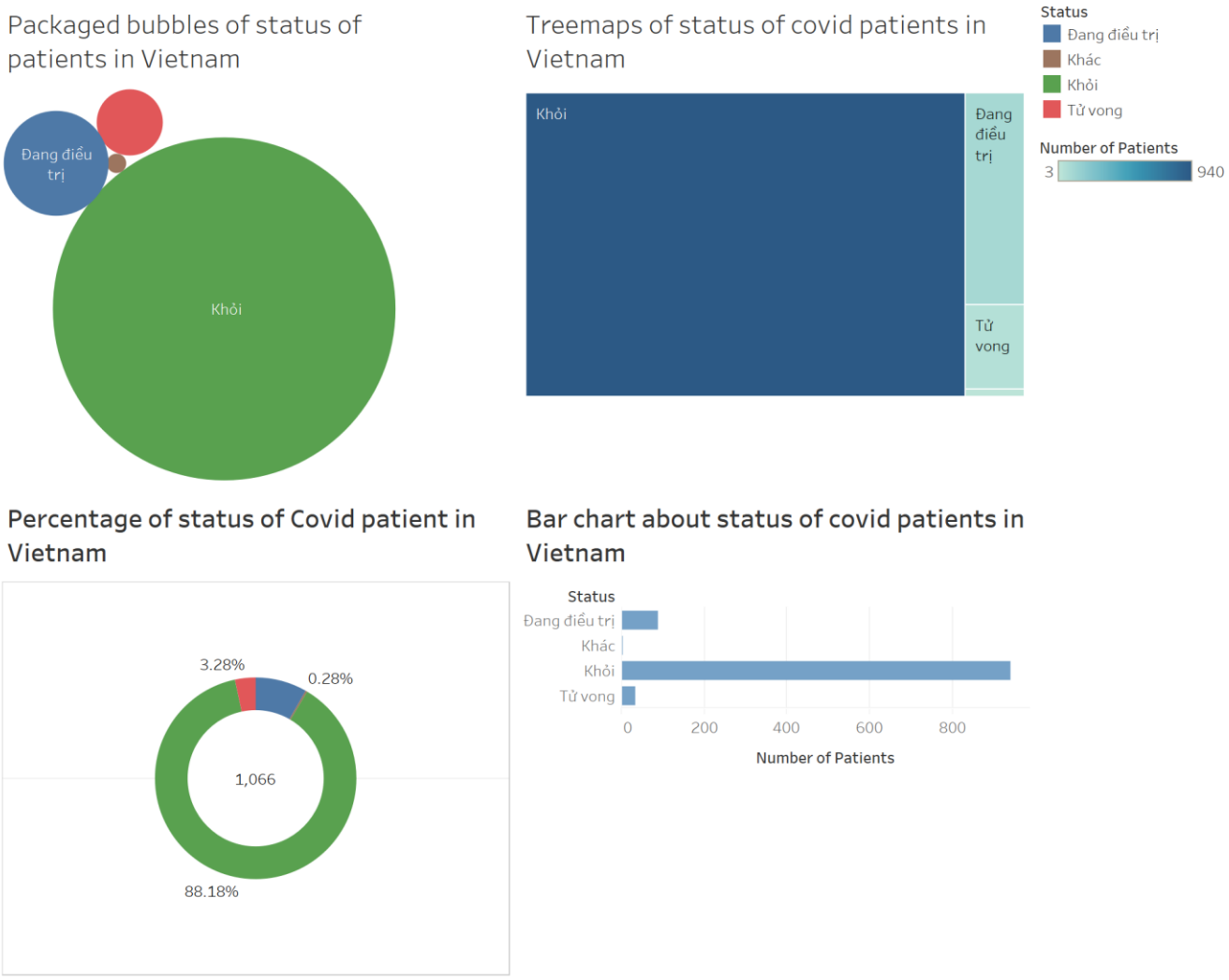
3.4 Dashboard of Overlook



12. Dashboard of Overlook

Biểu đồ biểu diễn tổng quan các chỉ số quan trọng trong việc theo dõi COVID-19. Đầu tiên, số ca nhiễm đã có thể coi là khá ổn định và không quá cao, có thể xem là ít nếu so với các thông số khác. Tổng người chết cũng không có sự tăng đột biến và số lượng cũng là khá ít. Tổng số ca hồi phục và số ca còn đang điều trị có xu hướng ngược chiều nhau là điều hiển nhiên. Nhưng nhìn vào xu hướng này ta có thể thấy tình hình điều trị bệnh của Việt Nam đang theo chiều hướng tốt dần lên. Số người được điều trị đã giảm mạnh từ khoảng cuối tháng 8 đến nay, trong khi trước đó không có dấu hiệu giảm. Việc tổng số ca tăng đều là điều có thể hiểu được, nhưng đáng nói là ngoài việc giai đoạn từ 9/8 đến 17/8, số ca tăng có hướng tăng nhanh thì trong khoảng thời gian sau đó, số ca nhiễm có tăng nhưng tốc độ không quá nhanh. Điều này thể hiện rõ khả năng kiểm soát dịch của Việt Nam.

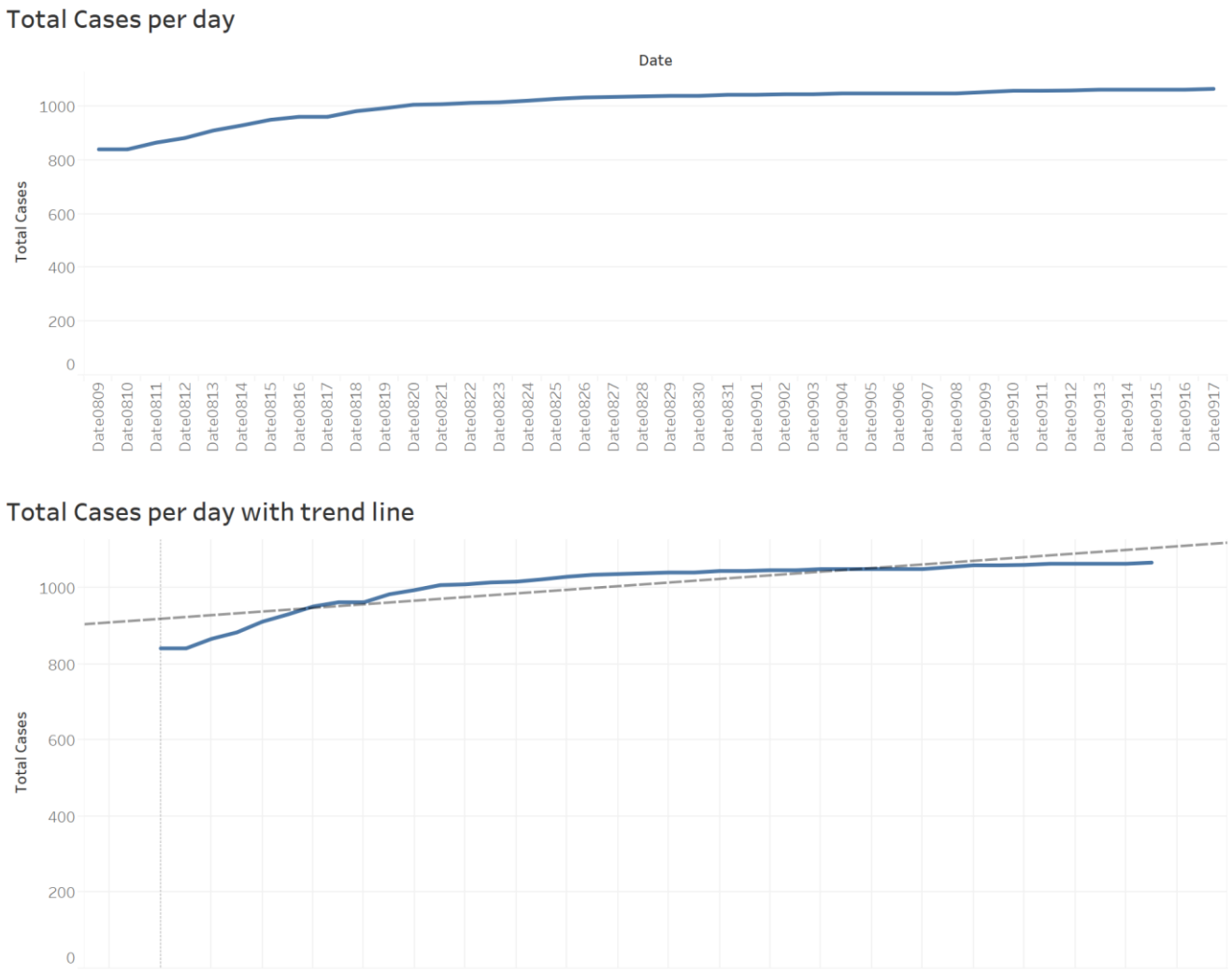
3.5 Dashboard of Status



13. Dashboard of Status

Các biểu đồ khác nhau thể hiện tỷ lệ tình trạng các bệnh nhân mắc Covid-19 tại Việt Nam. Tuy cách thể hiện là khác nhau nhưng đều thể hiện chung một ý chính. Các bệnh nhân đã được chữa khỏi vẫn chiếm tỷ lệ cao nhất và bỏ xa các tình trạng khác, Số lượng bệnh nhận đang điều trị tuy không cao bằng số người được chữa khỏi, nhưng vẫn không phải là quá lớn. Tỷ lệ tử vong do virus lẫn không do Virus đều không quá cao.

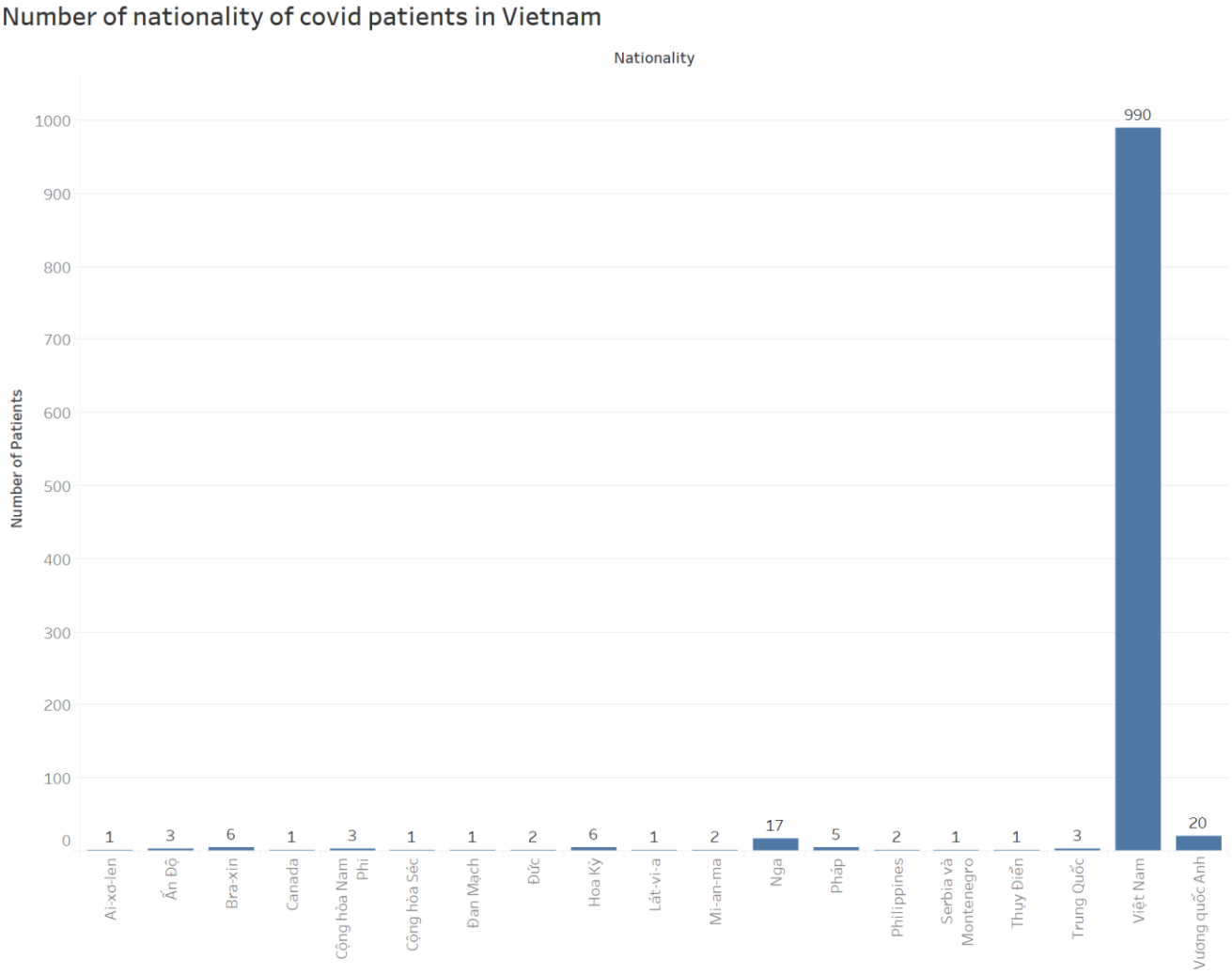
3.6 Dashboard of Total Cases



14. Dashboard of Total Cases

Biểu đồ đường thể hiện diễn biến theo tổng số ca nhiễm của bệnh COVID-19 kèm đường dự đoán. Như đã đề cập ở trên, việc chỉ số này tăng theo thời gian là khá hiển nhiên tuy nhiên điều đáng nói ở đây là so với dự đoán trên lý thuyết thì những ngày sau Việt Nam đã làm khá tốt công tác kiểm soát dịch, không để đại dịch tăng thêm

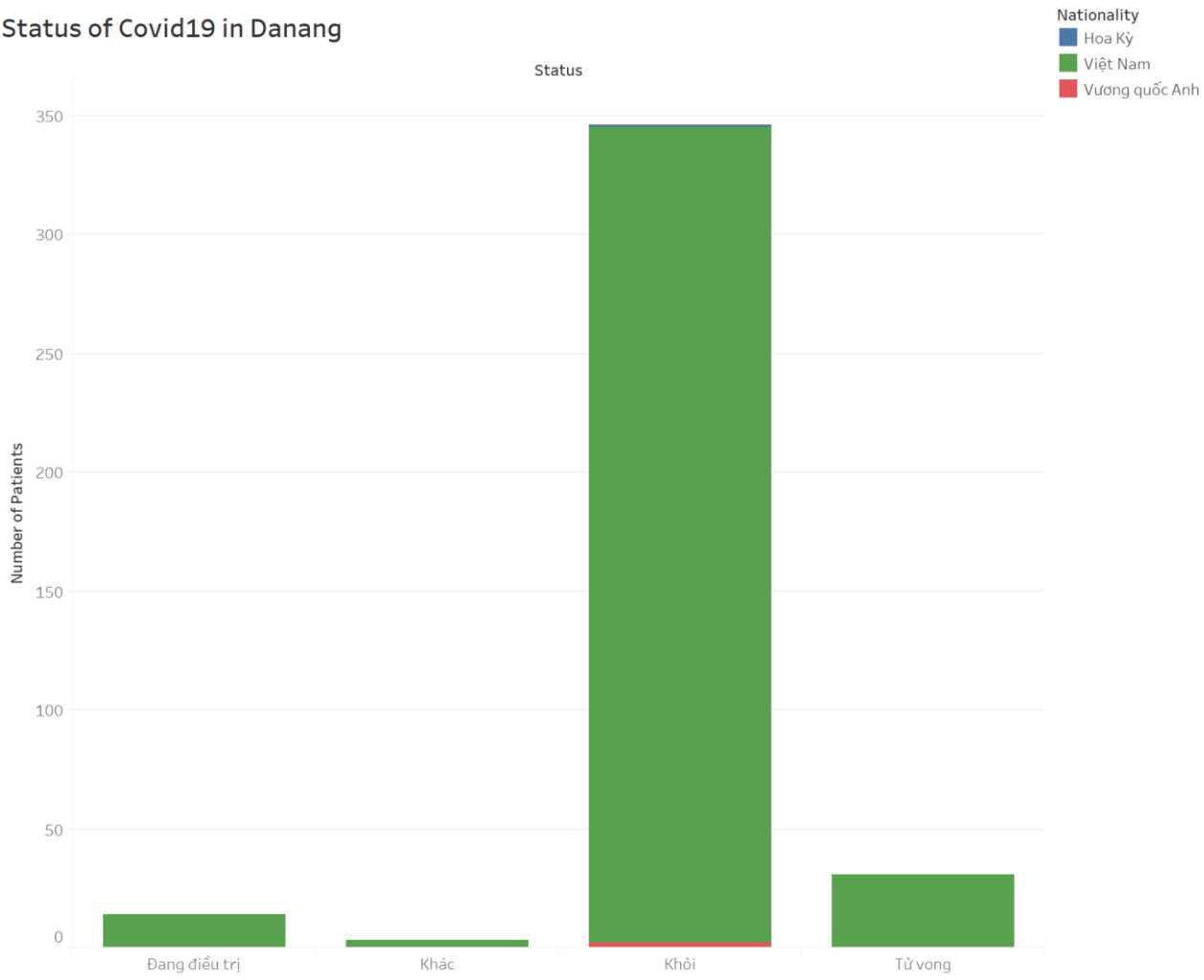
3.7 Dashboard of Nationality



15. Dashboard of Nationality

Biểu đồ cột cho biết số lượng về quốc tịch của các bệnh nhân mắc virus tại Việt Nam. 3 nước có số lượng bệnh nhân vượt qua 2 con số lần lượt là Việt Nam, Vương Quốc Anh và Nga. Trong đó, số bệnh nhân là người Việt Nam là đông nhất, không có gì quá ngạc nhiên về thông số như vậy. Điều đáng lưu ý ở đây là số người nước ngoài mắc bệnh ở Việt Nam đến từ rất nhiều châu lục khác nhau trên thế giới, trong đó các nước Châu Âu chiếm phần đông, các châu lục khác đều không quá đáng kể.

3.8 Dashboard of Status in Danang



16. Dashboard of status of cases in Danang

Biểu đồ thể hiện tình hình các bệnh nhân ở tâm dịch Đà Nẵng, không có gì quá đặc biệt về tỷ lệ được chữa khỏi vẫn chiếm tỷ lệ áp đảo so với các tình trạng khác, đặc biệt trong đó có 2 bệnh nhân người Anh. Điểm khác biệt lớn nhất của Đà Nẵng so với tình trạng chung của cả nước đó là tỷ lệ bệnh nhân của Đà Nẵng cao hơn thấy rõ so với các bệnh đang được điều trị, cao hơn gần gấp 2 lần.

4 Áp dụng mô hình học máy vào dữ liệu

4.1 Trước khi áp dụng các mô hình học máy

Ý tưởng: Dựa vào tổng số ca nhiễm của 7 ngày trước đó để dự đoán tổng số ca nhiễm ngày kế tiếp. Nếu kết quả dự đoán là một số nhỏ hơn số ca nhiễm của ngày gần nhất thì lấy kết quả của số ca nhiễm ngày gần nhất (vì tổng số ca nhiễm chỉ có thể giữ nguyên hoặc tăng).

In [4]:
```
X = []
y = []
d = 7

for i in range(0, 30, 1):
    x_tmp = []
    for j in range(0, d, 1):
        x_tmp.append(data[i + j])
    y.append(data[i + d])
    X.append(x_tmp)
```

In [5]:
```
X_test = []
X_test.append(np.array(data[-d:]))
X_test.append([1060, 1063, 1063, 1063, 1063, 1066, 1068])
print(X_test)

y_test = [1068, 1068]
```

17. Source code lấy tập train và test

Dữ liệu train:

```
[[841, 841, 866, 883, 911, 930, 951], [841, 866, 883, 911, 930, 951, 962],
 [866, 883, 911, 930, 951, 962, 962], [883, 911, 930, 951, 962, 962, 983],
 [911, 930, 951, 962, 962, 983, 994], [930, 951, 962, 962, 983, 994, 1007],
 [951, 962, 962, 983, 994, 1007, 1009], [962, 962, 983, 994, 1007, 1009, 1014],
 [962, 983, 994, 1007, 1009, 1014, 1016], [983, 994, 1007, 1009, 1014, 1016, 1022],
 [994, 1007, 1009, 1014, 1016, 1022, 1029], [1007, 1009, 1014, 1016, 1022, 1029, 1034],
 [1009, 1014, 1016, 1022, 1029, 1034, 1036], [1014, 1016, 1022, 1029, 1034, 1036, 1038],
 [1016, 1022, 1029, 1034, 1036, 1038, 1040], [1022, 1029, 1034, 1036, 1038, 1040, 1040],
 [1029, 1034, 1036, 1038, 1040, 1040, 1044], [1034, 1036, 1038, 1040, 1040, 1044, 1044],
 [1036, 1038, 1040, 1040, 1044, 1044, 1046], [1038, 1040, 1040, 1044, 1044, 1046, 1046],
 [1040, 1040, 1044, 1044, 1046, 1046, 1049], [1040, 1044, 1044, 1046, 1046, 1049, 1049],
 [1044, 1044, 1046, 1046, 1049, 1049, 1049], [1044, 1046, 1046, 1049, 1049, 1049, 1049],
 [1046, 1046, 1049, 1049, 1049, 1049, 1049], [1046, 1049, 1049, 1049, 1049, 1049, 1054],
 [1049, 1049, 1049, 1049, 1049, 1054, 1059], [1049, 1049, 1049, 1049, 1054, 1059, 1059],
 [1049, 1049, 1049, 1054, 1059, 1059, 1060], [1049, 1049, 1054, 1059, 1059, 1060, 1063]]
```

18. X_train

```
[962, 962, 983, 994, 1007, 1009, 1014, 1016, 1022, 1029, 1034,
1036, 1038, 1040, 1040, 1044, 1044, 1046, 1046, 1049, 1049,
1049, 1049, 1049, 1054, 1059, 1059, 1060, 1063, 1063]
```

19. y_train

Dữ liệu test:

```
[array([1059, 1060, 1063, 1063, 1063, 1063, 1066], dtype=int64),
 [1060, 1063, 1063, 1063, 1063, 1066, 1068]]
```

20. X_test

```
[1068, 1068]
```

21. y_test

4.2 Thư viện sklearn

```
In [7]: #sklearn
        model1 = LR().fit(X, y)

In [8]: print("Formula of model")
        print(printFormula(model1.coef_, model1.intercept_))

        Formula of model
        217.78 + 0.02*Day_7 + -0.2*Day_6 + 0.36*Day_5 + 0.11*Day_4 + -0.15*Day_3 + 0.06*Day_2 + 0.6*Day_1

In [9]: y_pred1 = predictions(model1, X)
        mse1 = MSE(y_pred1, y)
        mse1

Out[9]: 7.6

In [10]: predictions1 = predictions(model1, X_test)
         predictions1

Out[10]: array([1066., 1068.])
```

22. Source code train và test sử dụng thư viện sklearn

4.3 Mô hình OLS (statsmodels)

```
In [11]: #statsmodels.api
         model2 = sm.OLS(y, X).fit()

In [12]: print("Formula of model")
         print(printFormula(model2.params, 0))

         Formula of model
         0 + -0.09*Day_7 + -0.36*Day_6 + 0.43*Day_5 + 0.09*Day_4 + -0.12*Day_3 + 0.17*Day_2 + 0.87*Day_1

In [13]: y_pred2 = predictions(model2, X)
         mse2 = MSE(y_pred2, y)
         mse2

Out[13]: 10.666666666666666

In [14]: predictions2 = predictions(model2, X_test)
         predictions2

Out[14]: array([1068., 1069.])
```

23. Source code train và test sử dụng thư viện statsmodels

4.4 Nhận xét kết quả thu được

Mặc dù với mô hình được sử dụng thư viện `statsmodels` có MSE cao hơn so với `sklearn` nhưng kết quả được dự đoán từ mô hình `statsmodels` có vẻ tốt hơn so với `sklearn`.

Hàm hồi quy của 2 mô hình:

- `Statsmodels`:
  `0 + -0.09*Day_7 + -0.36*Day_6 + 0.43*Day_5 + 0.09*Day_4 + -0.12*Day_3 + 0.17*Day_2 + 0.87*Day_1`
- `Sklearn`:
  `217.78 + 0.02*Day_7 + -0.2*Day_6 + 0.36*Day_5 + 0.11*Day_4 + -0.15*Day_3 + 0.06*Day_2 + 0.6*Day_1`

Dù mô hình `statsmodels` không có hệ số tự do nhưng bộ trọng số của 2 mô hình lệch nhau không nhiều.

4.5 Kiểm định mô hình OLS (statsmodels)

```
In [15]: model2.summary()
```

Out[15]:

OLS Regression Results

Dep. Variable:	y	**R-squared (uncentered):**	1.000
Model:	OLS	**Adj. R-squared (uncentered):**	1.000
Method:	Least Squares	**F-statistic:**	3.456e+05
Date:	Sat, 19 Sep 2020	**Prob (F-statistic):**	3.39e-56
Time:	22:54:22	**Log-Likelihood:**	-77.315
No. Observations:	30	**AIC:**	168.6
Df Residuals:	23	**BIC:**	178.4
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	-0.0857	0.112	-0.768	0.450	-0.317	0.145
x2	-0.3595	0.185	-1.942	0.065	-0.742	0.023
x3	0.4302	0.216	1.995	0.058	-0.016	0.876
x4	0.0929	0.231	0.402	0.691	-0.385	0.571
x5	-0.1191	0.235	-0.507	0.617	-0.605	0.367
x6	0.1702	0.235	0.724	0.476	-0.316	0.656
x7	0.8718	0.165	5.298	0.000	0.531	1.212

Omnibus:	3.441	**Durbin-Watson:**	2.545
Prob(Omnibus):	0.179	**Jarque-Bera (JB):**	1.993
Skew:	-0.427	**Prob(JB):**	0.369
Kurtosis:	3.930	**Cond. No.**	1.25e+03

24. Summary của mô hình sử dụng thư viện statsmodels

Dù kết quả thu được lúc dự đoán khá tốt nhưng các giá trị `p-value` của mô hình `OLS (statsmodels)` đều lớn hơn 0.05 nên **không có ý nghĩa về mặt thống kê**.

Do đó nhóm chuyển sang sử dụng mô hình ARIMA.

4.6 Mô hình ARIMA (statsmodels)

```
df = pd.Series(data["Total Cases"].values, index = data["Date"])
df.head()
```

```
Date
2020-08-09    841
2020-08-10    841
2020-08-11    866
2020-08-12    883
2020-08-13    911
dtype: int64
```

```
model = ARIMA(df, order=(5,1,0))
model_fit = model.fit(disp=0)
print(model_fit.summary())
```

25. Dữ liệu và cài đặt tham số cho mô hình ARIMA

```
                coef
--------------------
const          6.5902
ar.L1.D.y      0.2755
ar.L2.D.y      0.5115
ar.L3.D.y     -0.2764
ar.L4.D.y      0.0799
ar.L5.D.y      0.2449
```

Công thức thu được từ mô hình:

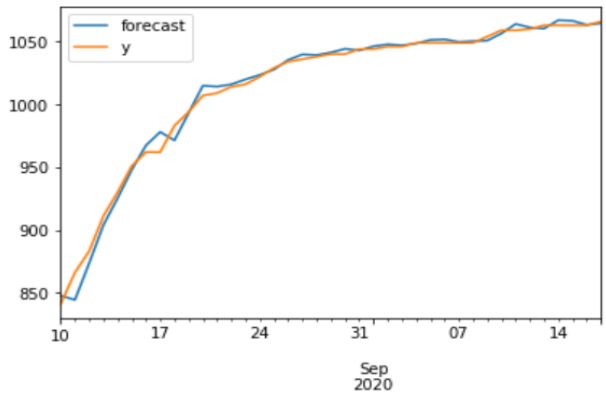
```
6.5902 + 0.2755*Day1 + 0.5115*Day2 - 0.2764*Day3
+ 0.0799*Day4 + 0.2449*Day5
```

26. Các tham số thu được

Với mô hình ARIMA, số ngày được chọn để dự đoán cho ngày kế tiếp là 5.

```
In [7]: model_fit.plot_predict(dynamic=False)
```

Out[7]:



27. Kết quả dự đoán thu được của mô hình

Nhìn qua thì ta thấy được mô hình học khá tốt với dữ liệu train. Sử dụng mô hình dự đoán cho 5 ngày kế tiếp, ta thu được kết quả:

```
In [8]: predicts = list()
        his = list(df.values)
        nextDays = ["2020/9/18", "2020/9/19", "2020/9/20", "2020/9/21", "2020/9/22"]
        actuals = [1068, 1068, 1068, 1068, np.NaN]
        for t in range(len(nextDays)):
            model = ARIMA(his, order=(5,1,0))
            model_fit = model.fit(disp=0)
            output = round(model_fit.forecast()[0][0], 0)
            predicts.append(output)
            his.append(actuals[t])
        for i in range(len(nextDays)):
            print("Date " + str(nextDays[i]) + "\t Predict: " + str(predicts[i])
                  + "\t Actual: " + str(actuals[i]))
```

```
Date 2020/9/18   Predict: 1069.0         Actual: 1068
Date 2020/9/19   Predict: 1071.0         Actual: 1068
Date 2020/9/20   Predict: 1069.0         Actual: 1068
Date 2020/9/21   Predict: 1069.0         Actual: 1068
Date 2020/9/22   Predict: 1070.0         Actual: nan
```

28. Kết quả dự đoán 5 ngày kế tiếp của mô hình ARIMA

Nhìn kết quả dự đoán được, ta có thể thấy mô hình dự đoán chênh lệch không nhiều so với kết quả thực tế.

4.7 Kiểm định mô hình ARIMA (statsmodels)

ARIMA Model Results

Dep. Variable:	D.y	No. Observations:	39
Model:	ARIMA(5, 1, 0)	Log Likelihood	-122.340
Method:	css-mle	S.D. of innovations	5.474
Date:	Mon, 21 Sep 2020	AIC	258.681
Time:	22:12:14	BIC	270.326
Sample:	08-10-2020 - 09-17-2020	HQIC	262.859

	coef	std err	z	P>|z|	[0.025	0.975]
const	6.5902	4.278	1.540	0.123	-1.795	14.975
ar.L1.D.y	0.2755	0.178	1.549	0.121	-0.073	0.624
ar.L2.D.y	0.5115	0.183	2.796	0.005	0.153	0.870
ar.L3.D.y	-0.2764	0.201	-1.378	0.168	-0.670	0.117
ar.L4.D.y	0.0799	0.199	0.401	0.688	-0.311	0.470
ar.L5.D.y	0.2449	0.193	1.267	0.205	-0.134	0.624

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	1.0741	-0.0000j	1.0741	-0.0000
AR.2	0.6519	-1.2108j	1.3751	-0.1714
AR.3	0.6519	+1.2108j	1.3751	0.1714
AR.4	-1.3522	-0.4269j	1.4180	-0.4513
AR.5	-1.3522	+0.4269j	1.4180	0.4513

29. Summary của mô hình ARIMA

Trong các giá trị `p-value` thu được, chỉ có tham số `ar.L3.D.y` (ngày cách ngày dự đoán 3 ngày) là có ý nghĩa về mặt thống kê ($0.005 \leq 0.05$).