TRỰC QUAN VÀ PHÂN TÍCH TÌNH HÌNH DỊCH BỆNH COVID-19 TẠI VIỆT NAM

Thành viên nhóm

1753075 – Huỳnh Đoàn Minh Ngọc

1753086 – Tống Lê Thiên Phúc

1753134 – Nguyễn Ngọc Đăng Khanh

NỘI DUNG TRÌNH BÀY

Dataset

Dashboard

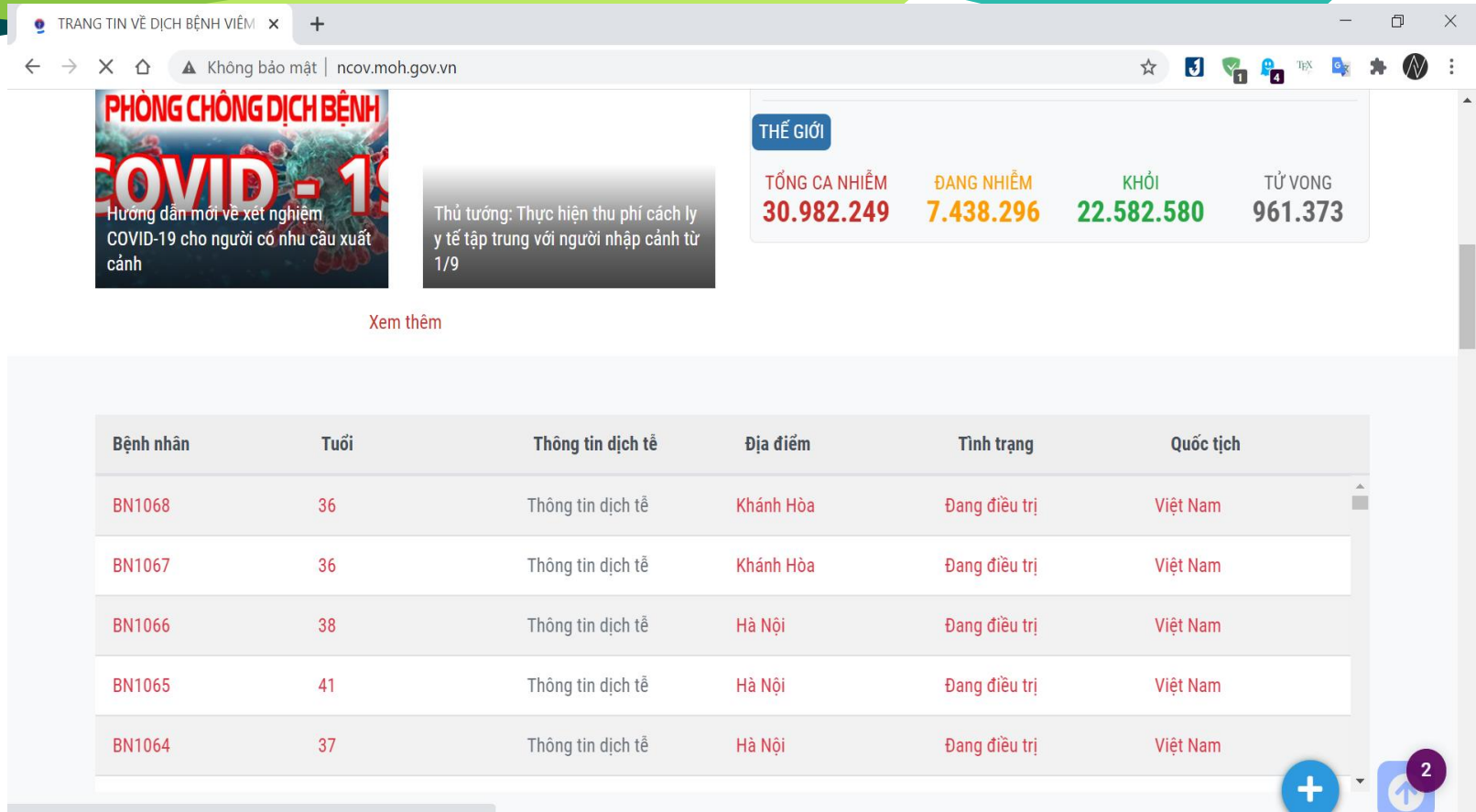
Machine learning

1.

DATASET

Crawl and preprocessing

Dataset



https://ncov.moh.gov.vn/

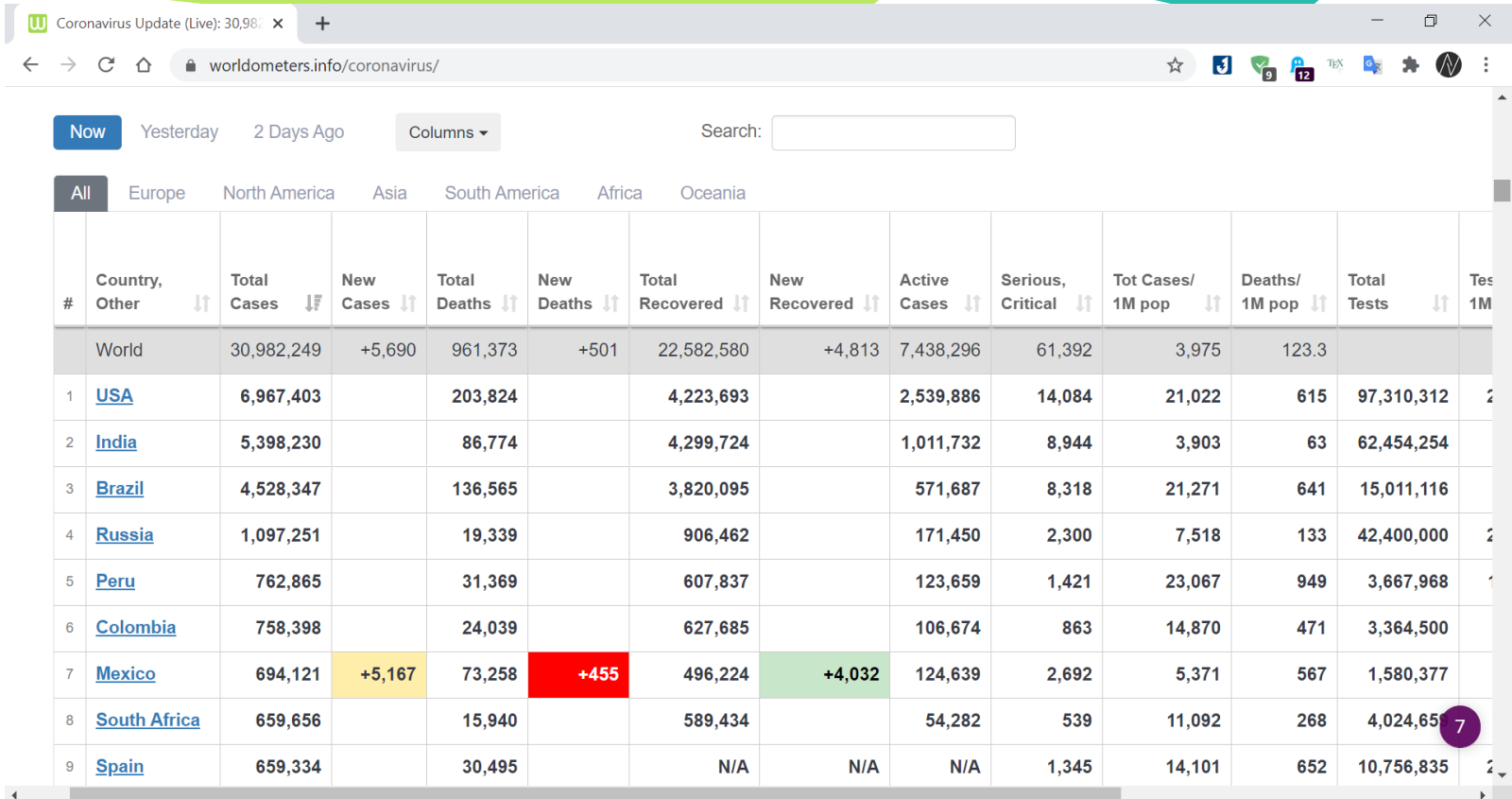
Dataset

	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

	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

Dataset



#	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	Tes 1M
	World	30,982,249	+5,690	961,373	+501	22,582,580	+4,813	7,438,296	61,392	3,975	123.3		
1	USA	6,967,403		203,824		4,223,693		2,539,886	14,084	21,022	615	97,310,312	2
2	India	5,398,230		86,774		4,299,724		1,011,732	8,944	3,903	63	62,454,254	
3	Brazil	4,528,347		136,565		3,820,095		571,687	8,318	21,271	641	15,011,116	
4	Russia	1,097,251		19,339		906,462		171,450	2,300	7,518	133	42,400,000	2
5	Peru	762,865		31,369		607,837		123,659	1,421	23,067	949	3,667,968	1
6	Colombia	758,398		24,039		627,685		106,674	863	14,870	471	3,364,500	
7	Mexico	694,121	+5,167	73,258	+455	496,224	+4,032	124,639	2,692	5,371	567	1,580,377	
8	South Africa	659,656		15,940		589,434		54,282	539	11,092	268	4,024,65	
9	Spain	659,334		30,495		N/A	N/A	N/A	1,345	14,101	652	10,756,835	2

https://www.worldometers.info/coronavirus/

Dataset

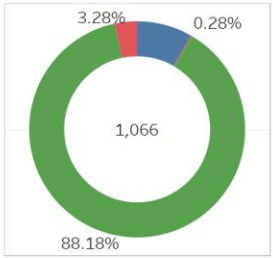
Date	Total Cases	New Cases	Total Deaths	New Deaths	Total Recovered	New Recovered	Active Cases	Serious, Critical	Tot Cases/1M pop	Deaths/1M pop	Total Tests	Tests/1M pop	1 Case every X ppl	1 Death every X ppl	1 Test every X ppl	Cas
Date2008	1007	13.0	25.0	0.0	542.0	9.0	440.0	0.0	10.0	3.0	817208.0	8385.0	96784	3898453.0	119.0	
Date0809	841	29.0	11.0	1.0	395.0	0.0	435.0	0.0	9.0	1.0	482456.0	4952.0	115850	8857296.0	202.0	
Date0810	841	0.0	13.0	2.0	399.0	4.0	429.0	0.0	9.0	0.0	482456.0	4952.0	115853	7494819.0	202.0	
Date0811	866	19.0	16.0	1.0	399.0	0.0	451.0	0.0	9.0	2.0	482456.0	4951.0	112517	6089988.0	202.0	
Date0812	883	17.0	17.0	1.0	409.0	10.0	457.0	0.0	9.0	2.0	621823.0	6382.0	110351	5731754.0	157.0	
Date0913	1063	3.0	35.0	0.0	918.0	8.0	110.0	0.0	11.0	4.0	1009145.0	10348.0	91737	2786180.0	97.0	
Date0914	1063	3.0	35.0	0.0	918.0	8.0	110.0	0.0	11.0	4.0	1009145.0	10348.0	91739	2786249.0	97.0	
Date0915	1063	0.0	35.0	0.0	931.0	5.0	97.0	0.0	11.0	4.0	1009145.0	10348.0	91744	2786385.0	97.0	
Date0916	1063	0.0	35.0	0.0	936.0	5.0	92.0	0.0	11.0	4.0	1009145.0	10348.0	91744	2786385.0	97.0	
Date0917	1066	3.0	35.0	0.0	940.0	4.0	91.0	0.0	11.0	4.0	1009145.0	10347.0	91488	2786453.0	97.0	

2.

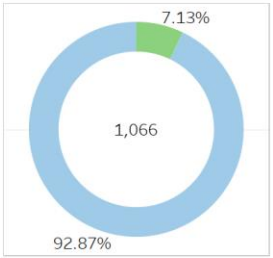
DASHBOARD

Visualization and analysis

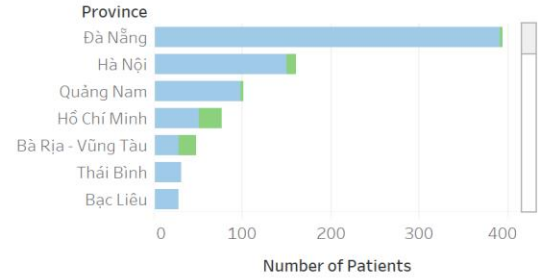
Percentage of status of Covid patient in Vietnam



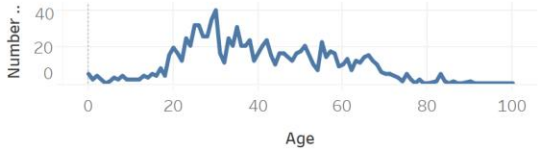
Percentage of nationality of Covid patient in Vietnam



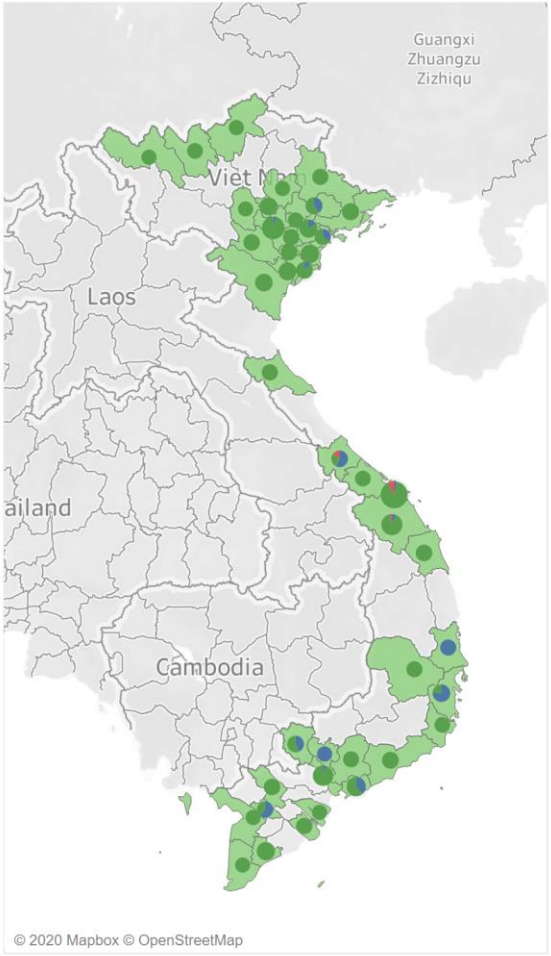
Nationality of patients in each province/city in Vietnam



Motion chart of age of covid patients in Vietnam



Status of Covid patients in each province/city in Vietnam

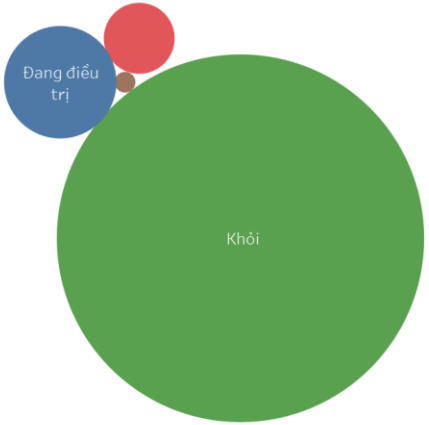


Status
- Đang điều trị
- Khác
- Khỏi
- Tử vong

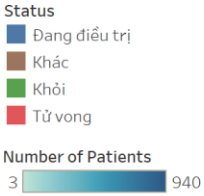
Distinct count of ID
- 1
- 100
- 200
- 300
- 394

N_Nationality
- Nước ngoài
- Trong nước

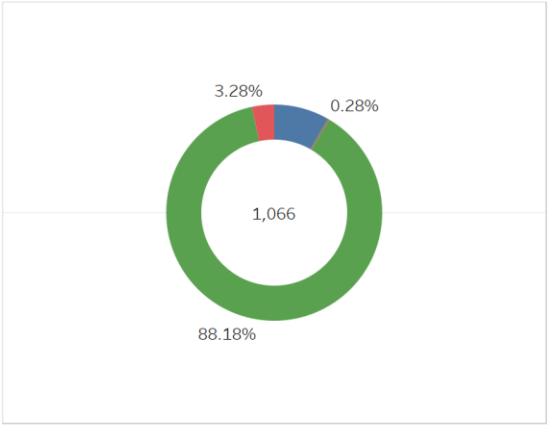
Packaged bubbles of status of patients in Vietnam



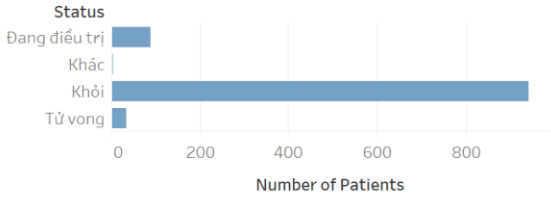
Treemaps of status of covid patients in Vietnam



Percentage of status of Covid patient in Vietnam

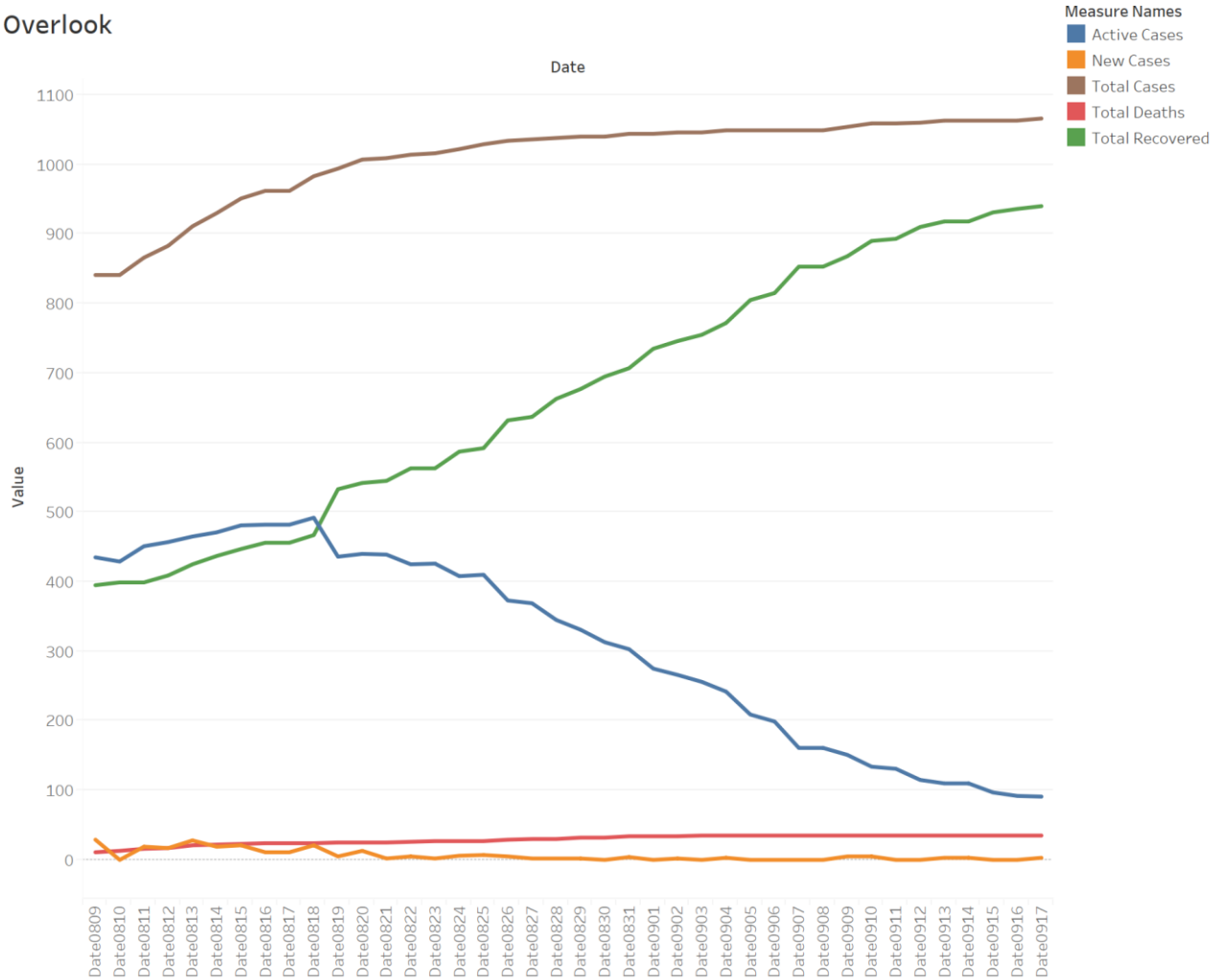


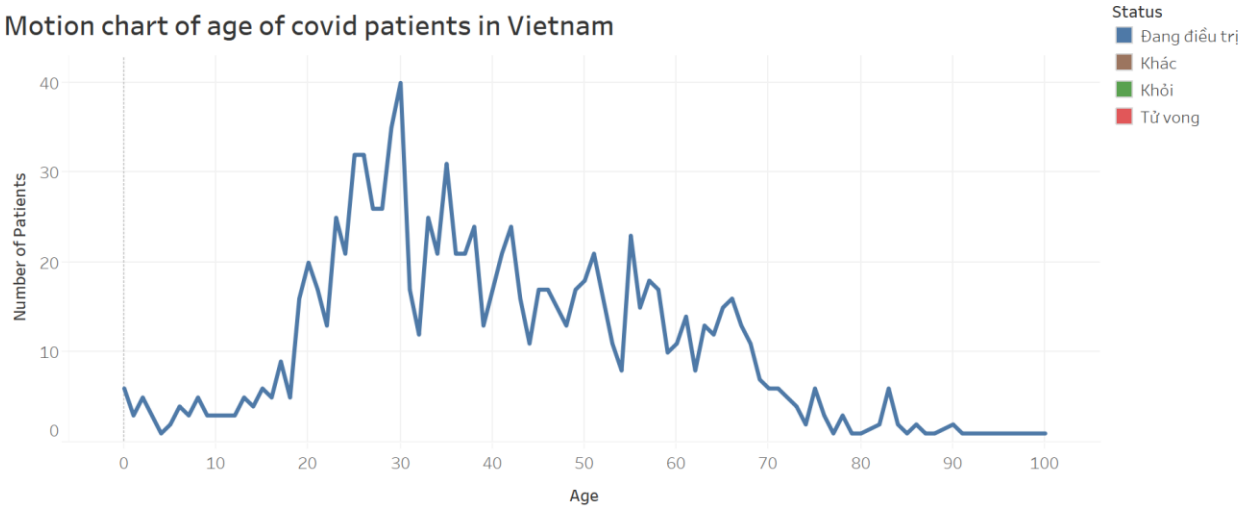
Bar chart about status of covid patients in Vietnam

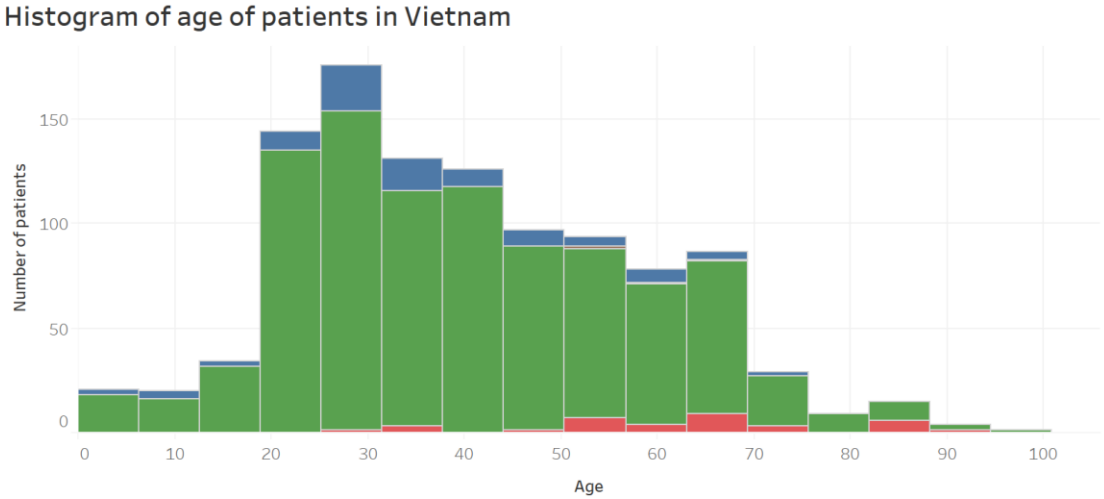


Dashboard of Status

Overlook

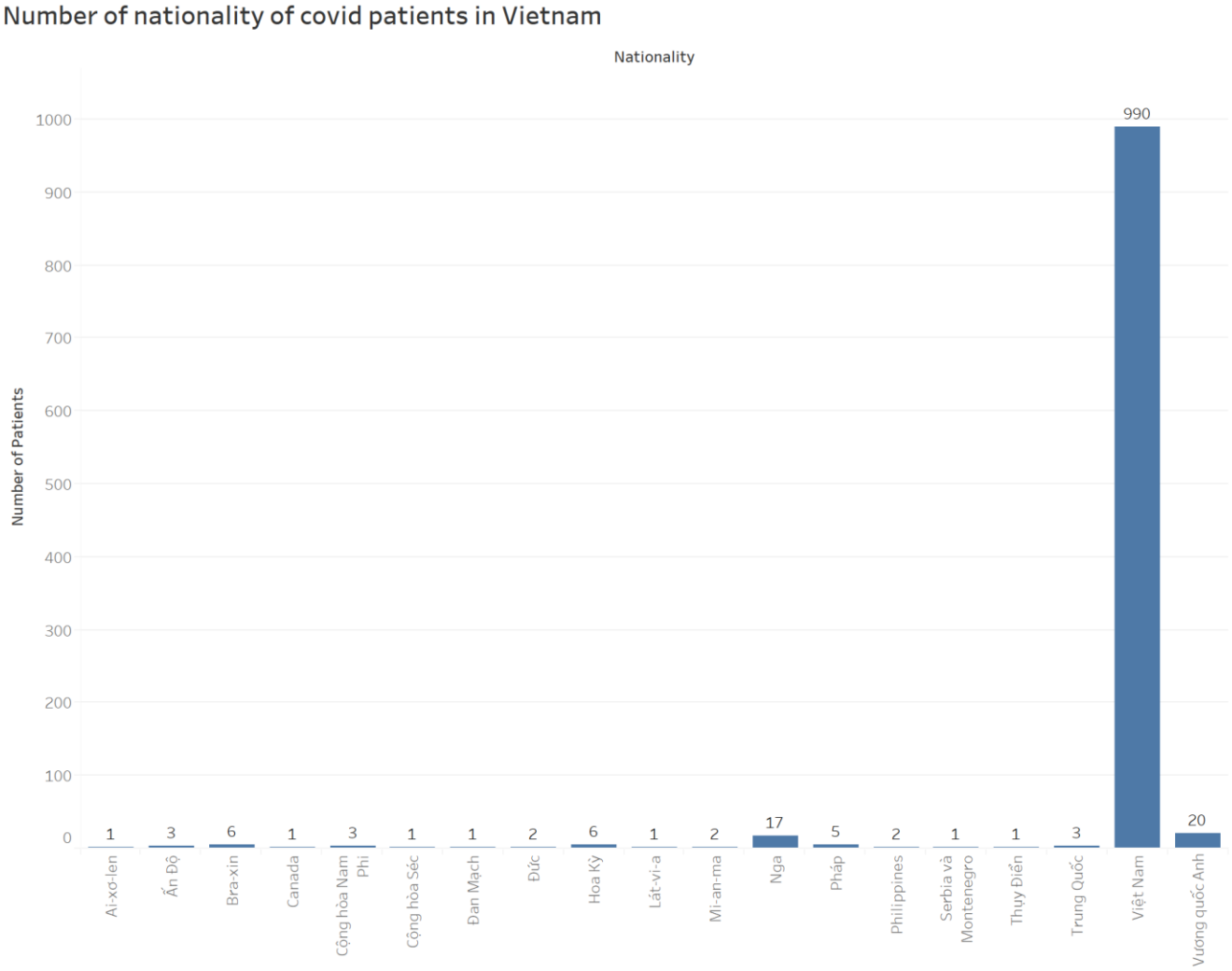

Measure Names
Active Cases
New Cases
Total Cases
Total Deaths
Total Recovered
Date
Value
1100
1000
900
800
700
600
500
400
300
200
100
0
Date0809
Date0810
Date0811
Date0812
Date0813
Date0814
Date0815
Date0816
Date0817
Date0818
Date0819
Date0820
Date0821
Date0822
Date0823
Date0824
Date0825
Date0826
Date0827
Date0828
Date0829
Date0830
Date0831
Date0901
Date0902
Date0903
Date0904
Date0905
Date0906
Date0907
Date0908
Date0909
Date0910
Date0911
Date0912
Date0913
Date0914
Date0915
Date0916
Date0917

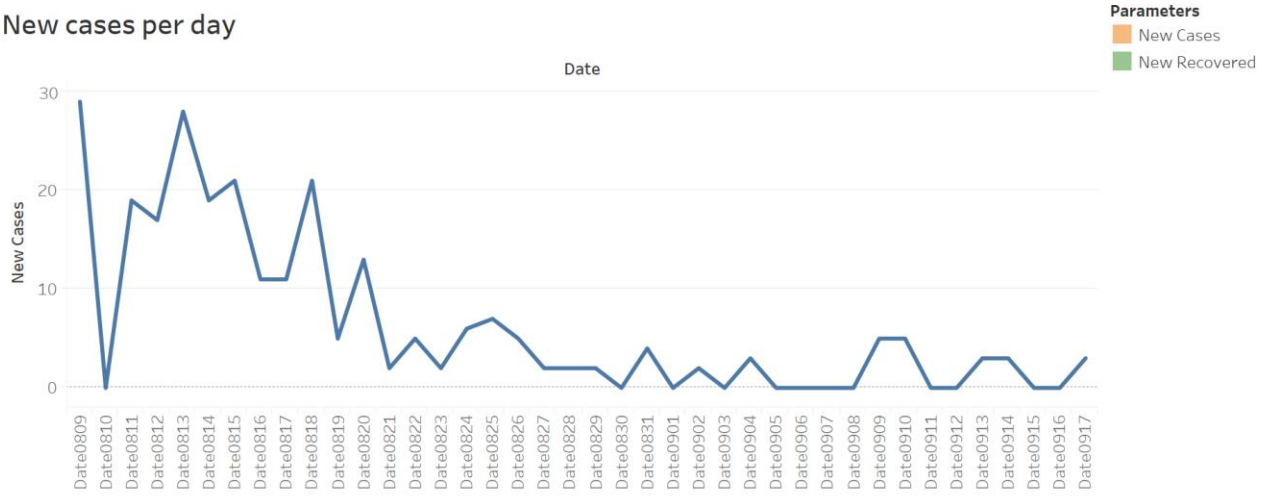


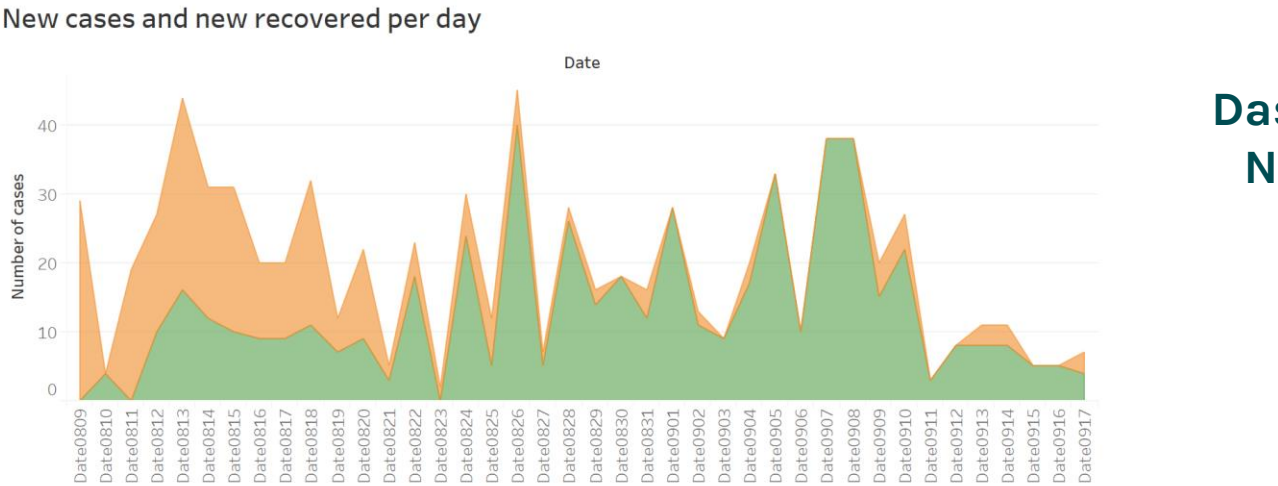


Dashboard of Age

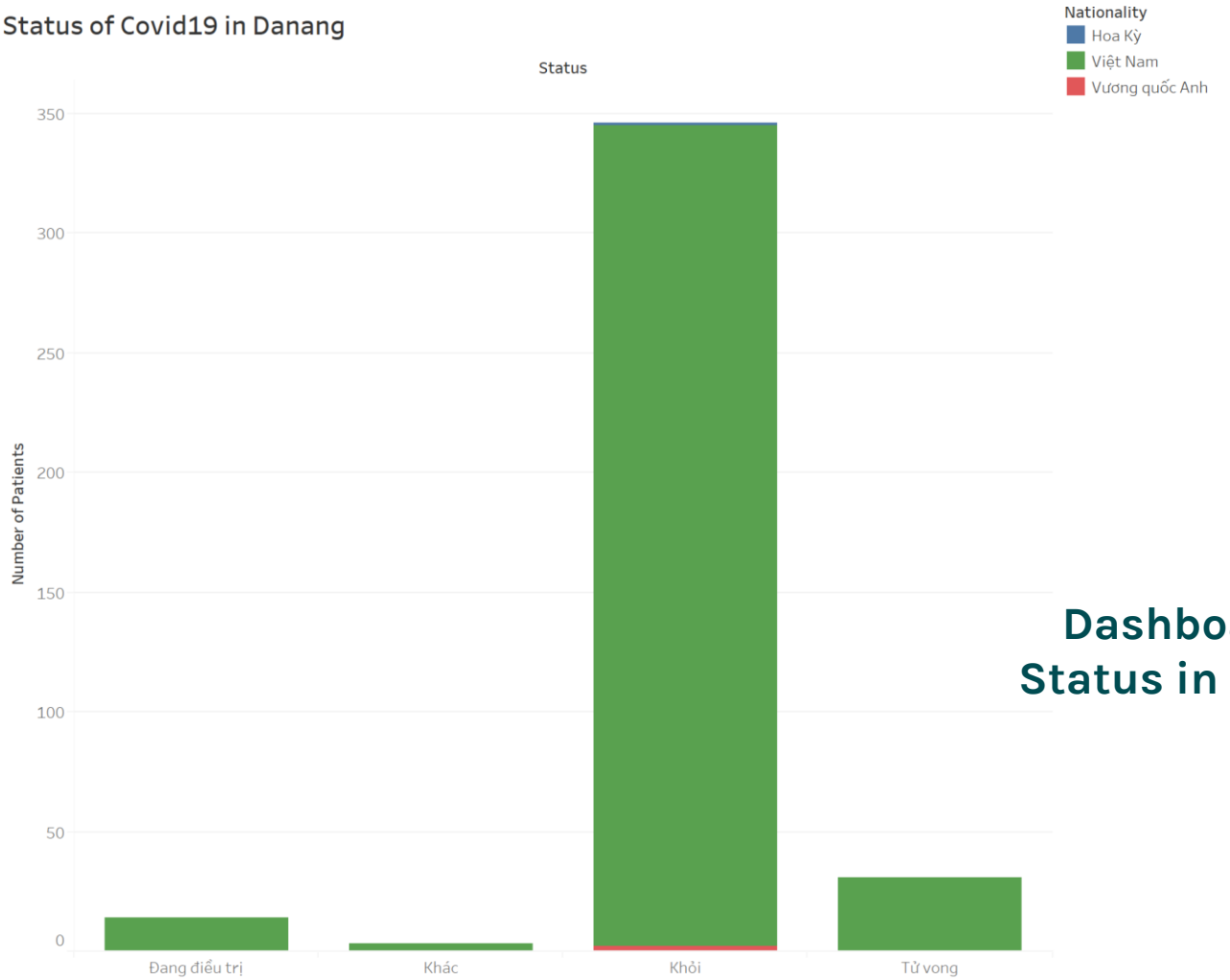
Dashboard of Nationality





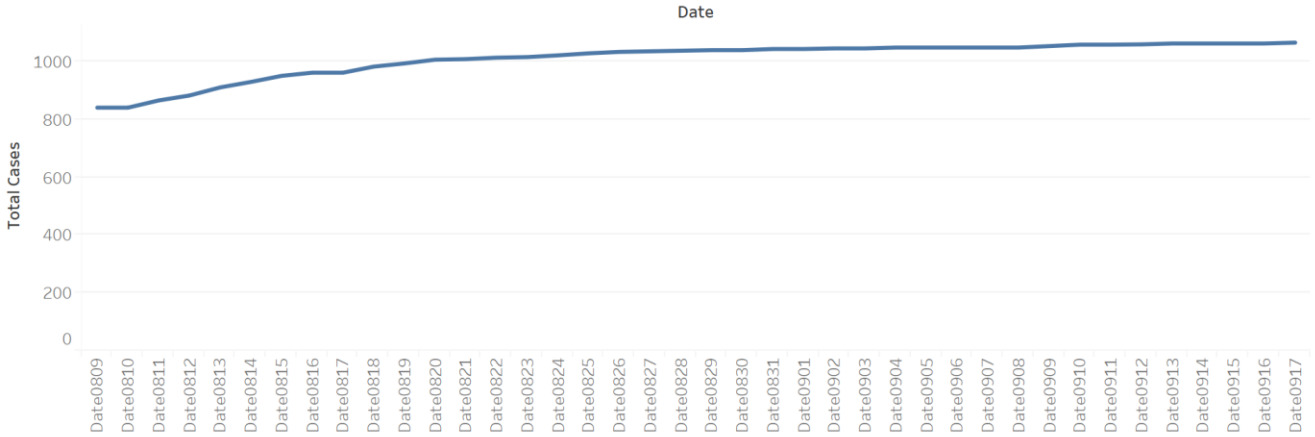


Dashboard of New Cases

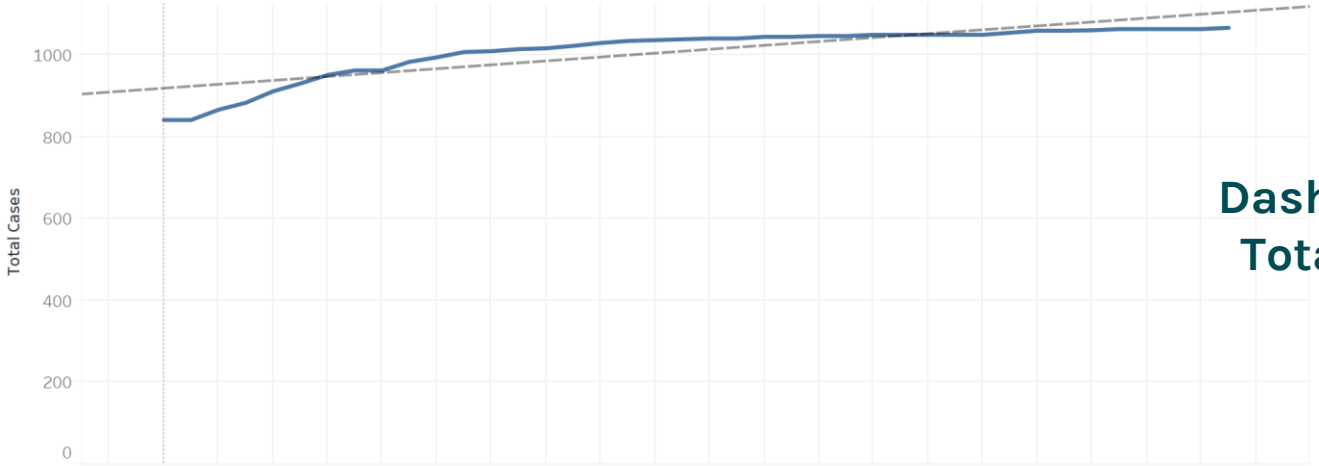


Dashboard of Status in Danang

Total Cases per day



Total Cases per day with trend line



Dashboard of Total Cases

3.

MACHINE LEARNING

Linear regression and hypothesis tesing

Train & test dataset

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)
```

```
[[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]]
```

x_train

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]
```

```
[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]
```

y_train

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

x_test

y_test [1068, 1068]

Scikit-learn

from sklearn.linear_model import LinearRegression as LR

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.])
```

Statsmodels

```
import statsmodels.api as sm
from statsmodels.tsa.arima_model
import ARIMA
```

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.])
```

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

ARIMA statsmodels

In [5]:
```
df = pd.Series(data["Total Cases"].values, index = data["Date"])
df.head()
```

Out[5]:
```
Date
2020-08-09    841
2020-08-10    841
2020-08-11    866
2020-08-12    883
2020-08-13    911
dtype: int64
```

In [6]:
```
model = ARIMA(df, order=(5,1,0))
model_fit = model.fit(disp=0)
print(model_fit.summary())
```

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:	23:00:19	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

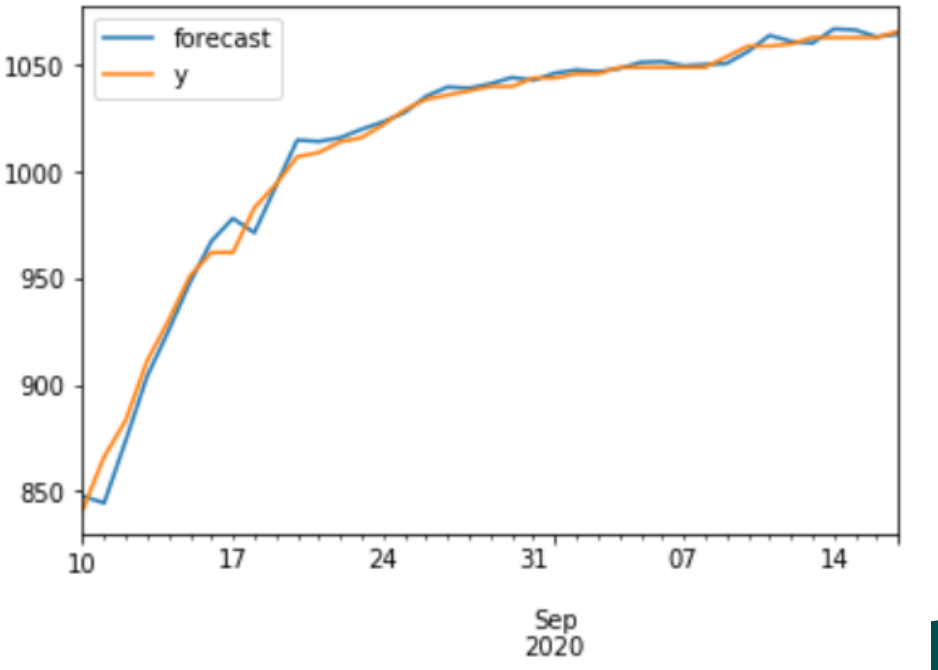
Roots

	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

ARIMA statsmodels

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

Out[7]:



ARIMA statsmodels

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
```

Thanks!

Any questions?

1753075 – Huynh Doan Minh Ngoc

1753086 – Tong Le Thien Phuc

1753134 – Nguyen Ngoc Dang Khanh