Covid Varient EDA using Python

Importing required libraries. For Analysis we are using libreries to visualize inforamtion.

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
import datetime as datetime
import plotly.express as px
import plotly.graph_objs as go
import plotly.figure_factory as ff
from plotly import tools
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)
import warnings
warnings.filterwarnings("ignore")
```

Reading the file using read.csv using pandas

In [3]:
```
dt=pd.read_csv("C:\\Users\\Admin\\Downloads\\covid-variants.csv")
```

1. Finding Basic information about data

In [4]:
```
dt.columns
```

Out[4]:
```
Index(['location', 'date', 'variant', 'num_sequences', 'perc_sequences',
       'num_sequences_total'],
      dtype='object')
```

In [5]:
```
dt.head(10)
```

Out[5]:

	location	date	variant	num_sequences	perc_sequences	num_sequences_total
0	Angola	2020-07-06	Alpha	0	0.0	3
1	Angola	2020-07-06	B.1.1.277	0	0.0	3
2	Angola	2020-07-06	B.1.1.302	0	0.0	3
3	Angola	2020-07-06	B.1.1.519	0	0.0	3
4	Angola	2020-07-06	B.1.160	0	0.0	3
5	Angola	2020-07-06	B.1.177	0	0.0	3
6	Angola	2020-07-06	B.1.221	0	0.0	3
7	Angola	2020-07-06	B.1.258	0	0.0	3
8	Angola	2020-07-06	B.1.367	0	0.0	3
9	Angola	2020-07-06	B.1.620	0	0.0	3

In [6]:
```
dt.tail()
```

Out[6]:

	location	date	variant	num_sequences	perc_sequences	num_sequences_total
100411	Zimbabwe	2021-11-01	Omicron	0	0.0	6
100412	Zimbabwe	2021-11-01	S:677H.Robin1	0	0.0	6
100413	Zimbabwe	2021-11-01	S:677P.Pelican	0	0.0	6
100414	Zimbabwe	2021-11-01	others	0	0.0	6
100415	Zimbabwe	2021-11-01	non_who	0	0.0	6

In [7]:
```
dt.shape
```

Out[7]:
```
(100416, 6)
```

from the above details of file we found, The data (COVID-19 Variants) contains the following information:

location- this is the country for which the variants information is provided.

date - date for the data entry.

variant - this is the variant corresponding to this data entry.

num_sequences - the number of sequences processed (for the country, variant and date).

perc_sequences - percentage of sequences from the total number of sequences (for the country, variant and date).

num_sequences_total - total number of sequences (for the country, variant and date).

In [8]:
```
dt.describe()
```

Out[8]:

	num_sequences	perc_sequences	num_sequences_total
count	100416.000000	100416.000000	100416.000000
mean	72.171676	6.154355	1509.582457
std	1669.262169	21.898989	8445.291772
min	0.000000	-0.010000	1.000000
25%	0.000000	0.000000	12.000000
50%	0.000000	0.000000	59.000000
75%	0.000000	0.000000	394.000000
max	142280.000000	100.000000	146170.000000

2.showing data types of columns if required then we can change the data type.

In [9]:
```
dt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100416 entries, 0 to 100415
Data columns (total 6 columns):
 #   Column               Non-Null Count   Dtype 
---  ------               --------------   ----- 
 0   location             100416 non-null  object
 1   date                 100416 non-null  object
 2   variant              100416 non-null  object
 3   num_sequences        100416 non-null  int64 
 4   perc_sequences       100416 non-null  float64
 5   num_sequences_total  100416 non-null  int64 
dtypes: float64(1), int64(2), object(3)
memory usage: 4.6+ MB
```

In [10]:
```
#converting date Dtype oject to Dtype date
dt["date"]=dt["date"].apply(pd.to_datetime, dayfirst=True)
dt=dt.fillna(0)
```

In [11]:
```
dt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100416 entries, 0 to 100415
Data columns (total 6 columns):
 #   Column               Non-Null Count   Dtype 
---  ------               --------------   ----- 
 0   location             100416 non-null  object
 1   date                 100416 non-null  datetime64[ns]
 2   variant              100416 non-null  object
 3   num_sequences        100416 non-null  int64 
 4   perc_sequences       100416 non-null  float64
 5   num_sequences_total  100416 non-null  int64 
dtypes: datetime64[ns](1), float64(1), int64(2), object(2)
memory usage: 4.6+ MB
```

3.Finding the Null values

In [12]:
```
#count or check any missing values
dt.isnull().sum()
```

Out[12]:
```
location              0
date                  0
variant               0
num_sequences         0
perc_sequences        0
num_sequences_total   0
dtype: int64
```

4.Finding Duplicate values

In [13]:
```
# find any duplicate
dt.duplicated().sum()
```

Out[13]:
```
0
```

3.Unique values in the data

In [14]:
```
# countries
countries=dt['location'].unique()
countrie=pd.Series(countries)
countrie
```

Out[14]:
```
0             Angola
1          Argentina
2              Aruba
3          Australia
4            Austria
           ...
116    United States
117          Uruguay
118          Vietnam
119           Zambia
120         Zimbabwe
Length: 121, dtype: object
```

In [15]:
```
# types of varients
var=dt['variant'].unique()
varients=pd.Series(var)
varients
```

Out[15]:
```
0              Alpha
1          B.1.1.277
2          B.1.1.302
3          B.1.1.519
4            B.1.160
5            B.1.177
6            B.1.221
7            B.1.258
8            B.1.367
9            B.1.620
10              Beta
11             Delta
12           Epsilon
13               Eta
14             Gamma
15              Iota
16             Kappa
17            Lambda
18                Mu
19           Omicron
20     S:677H.Robin1
21    S:677P.Pelican
22            others
23           non_who
dtype: object
```

5.varient wise number of sequence occured

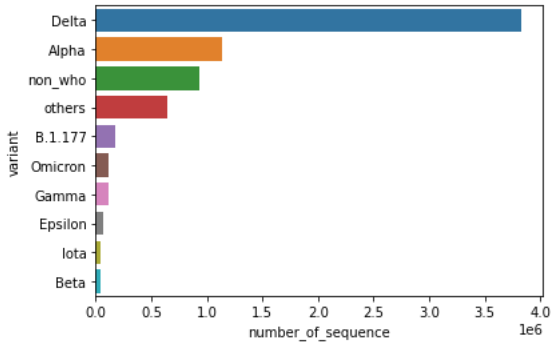
In [16]:
```
variants=dt.variant.unique()
variant_num_seq=[]
for i in variants:
    x=dt[dt.variant.values==i]
    num_seq=sum(x.num_sequences)
    variant_num_seq.append(num_seq)


variant_set=pd.DataFrame({"variant":variants,"number_of_sequence":variant_num_seq})
var_index=variant_set.number_of_sequence.sort_values(ascending=False).index.values
variant_set=variant_set.reindex(var_index)
variant_set
```

Out[16]:

	variant	number_of_sequence
11	Delta	3834100
0	Alpha	1132595
23	non_who	931098
22	others	642603
5	B.1.177	170457
19	Omicron	115538
14	Gamma	115156
12	Epsilon	66127
15	Iota	42905
10	Beta	40514
4	B.1.160	34019
7	B.1.258	30787
3	B.1.1.519	22825
6	B.1.221	15377
18	Mu	14248
17	Lambda	9411
16	Kappa	7477
13	Eta	6924
20	S:677H.Robin1	6547
21	S:677P.Pelican	4837
1	B.1.1.277	1183
9	B.1.620	1016
8	B.1.367	961
2	B.1.1.302	486

In [17]:
```
sb.barplot(variant_set.number_of_sequence.head(10),variant_set.variant.head(10))
plt.show()
```



6.Last date analysis of covid variant data

In [21]:
```
last_date_data_df = sample.groupby(["variant", "Location"])["date"].max().reset_index()
print(last_date_data_df.shape)
last_date_data_df
```

```
(2904, 3)
```

Out[21]:

	variant	Location	date
0	Alpha	Angola	2021-10-04
1	Alpha	Argentina	2021-12-27
2	Alpha	Aruba	2021-12-13
3	Alpha	Australia	2021-12-27
4	Alpha	Austria	2021-12-13
...	...	...	...
2899	others	United States	2022-01-05
2900	others	Uruguay	2021-05-03
2901	others	Vietnam	2021-12-27
2902	others	Zambia	2021-12-27
2903	others	Zimbabwe	2021-11-01

2904 rows × 3 columns

In [22]:
```
last_date_data_df = last_date_data_df.merge(sample, how="left")#merging data using left join
print(last_date_data_df.shape)
last_date_data_df.head()
```

```
(2904, 6)
```

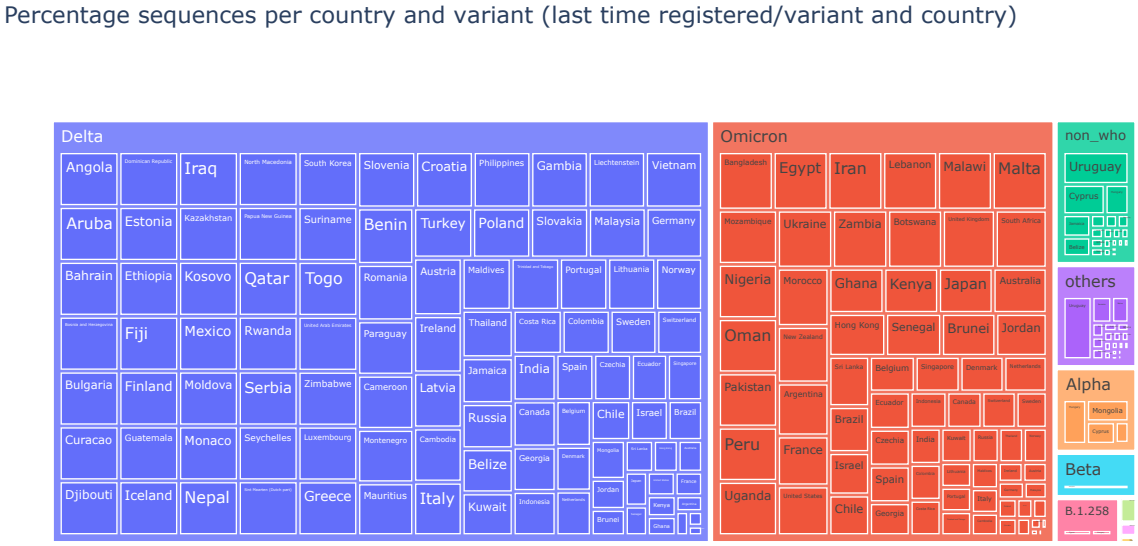
Out[22]:

	variant	Location	date	num_sequences	perc_sequences	Number of Case
0	Alpha	Angola	2021-10-04	0	0.0	33
1	Alpha	Argentina	2021-12-27	0	0.0	94
2	Alpha	Aruba	2021-12-13	0	0.0	61
3	Alpha	Australia	2021-12-27	0	0.0	1726
4	Alpha	Austria	2021-12-13	0	0.0	183

In [23]:
```
print(f"Countries number: {last_date_data_df.Location.nunique()}")
print(f"Date number: {last_date_data_df.date.nunique()}")
print(f"Variants number: {last_date_data_df.variant.nunique()}")
print(f"Variants names: {last_date_data_df.variant.unique()}")
```

```
Countries number: 121
Date number: 17
Variants number: 24
Variants names: ['Alpha' 'B.1.1.277' 'B.1.1.302' 'B.1.1.519' 'B.1.160' 'B.1.177' 'B.1.221'
 'B.1.258' 'B.1.367' 'B.1.620' 'Beta' 'Delta' 'Epsilon' 'Eta' 'Gamma'
 'Iota' 'Kappa' 'Lambda' 'Mu' 'Omicron' 'S:677H.Robin1' 'S:677P.Pelican'
 'non_who' 'others']
```

In [24]:
```
fig = px.treemap(last_date_data_df, path = ['variant', 'Location'], values = 'perc_sequences',
                 title="Percentage sequences per country and variant (last time registered/variant and country)")
fig.show()
```



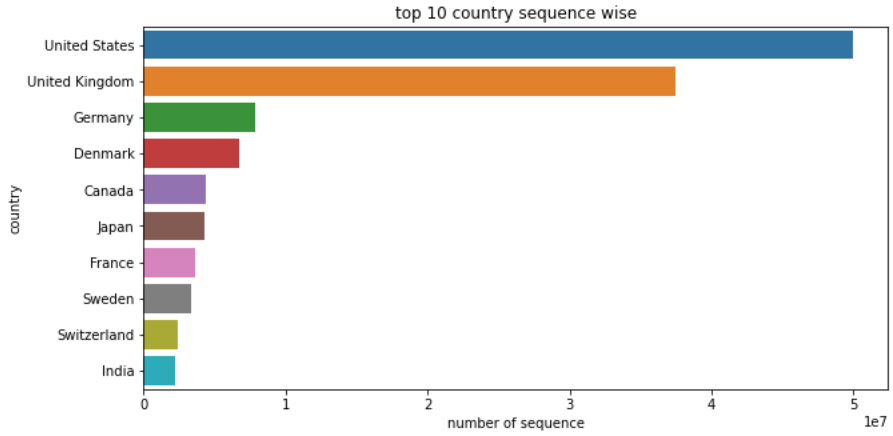
The tree map of last day data shows the delta and omicron varients are more active in lots of countries as compair to other varients.

7.country wise sequence

In [25]:
```
# Using groupby() and sum() to  check country wise sequence in desc
data3 = dt.groupby(['location'])['num_sequences_total'].sum().sort_values(ascending=False)
data3
```

Out[25]:
```
location
United States     49960248
United Kingdom    37427568
Germany            7851432
Denmark            6728880
Canada             4365240
                    ...
Belize                7536
Iraq                  4008
Moldova               3648
Mongolia              3600
Monaco                2016
Name: num_sequences_total, Length: 121, dtype: int64
```

In [26]:
```
country=data3.head(10)
plt.figure(figsize=(10,5))
sb.barplot(data3.head(10).values,data3.head(10).index)
plt.title("top 10 country sequence wise")
plt.ylabel("country",fontsize=10)
plt.xlabel("number of sequence",fontsize=10)
plt.show()
```



In given data set number of sequence occures is higher in US follwed by UK.so we can conclude that US and UK are most affected areas in covid.

In [20]:
```
sample  = dt.rename(columns={"location":"Location","num_sequences_total":"Number of Case"})
sample
```

Out[20]:

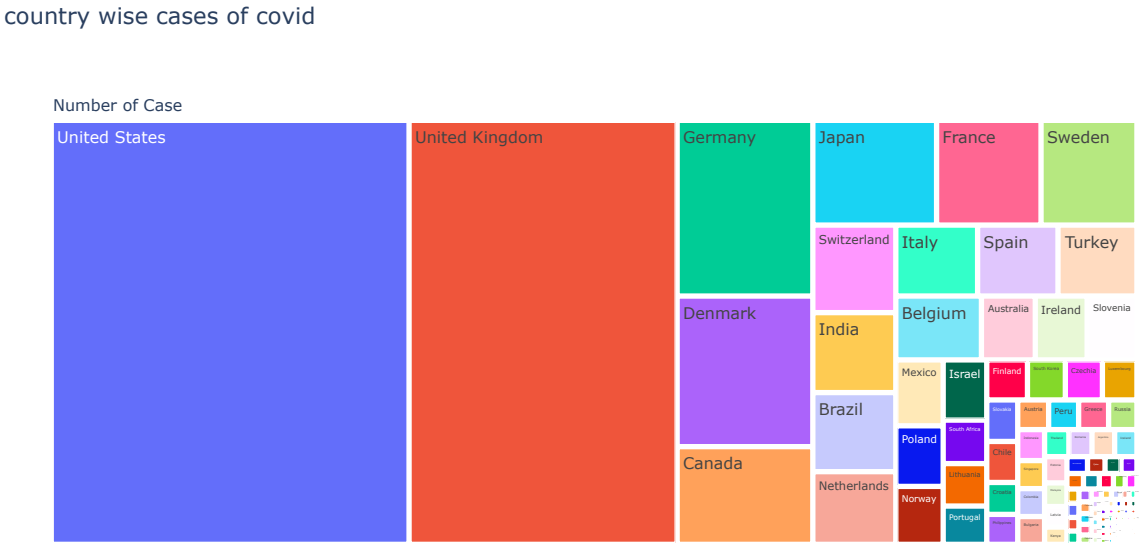
	Location	date	variant	num_sequences	perc_sequences	Number of Case
0	Angola	2020-07-06	Alpha	0	0.0	3
1	Angola	2020-07-06	B.1.1.277	0	0.0	3
2	Angola	2020-07-06	B.1.1.302	0	0.0	3
3	Angola	2020-07-06	B.1.1.519	0	0.0	3
4	Angola	2020-07-06	B.1.160	0	0.0	3
...	...	...	...	...	...	...
100411	Zimbabwe	2021-11-01	Omicron	0	0.0	6
100412	Zimbabwe	2021-11-01	S:677H.Robin1	0	0.0	6
100413	Zimbabwe	2021-11-01	S:677P.Pelican	0	0.0	6
100414	Zimbabwe	2021-11-01	others	0	0.0	6
100415	Zimbabwe	2021-11-01	non_who	0	0.0	6

100416 rows × 6 columns

8.Tree map of a all data set to visualize the affection of covid varients country wise.

In [27]:
```
fig = px.treemap(sample, path=[px.Constant('Number of Case'),'Location'],values='Number of Case',hover_data=['Location'],
                 title='country wise cases of covid')
```

In [28]:
```
fig.show()
```




```
9.Yearwise total cases of cove from given data set
```

In [49]:
```
#get year from corresponding date column
dt['year']=pd.DatetimeIndex(dt['date']).year
```

In [50]:
```
#yearwise sequences occured..
data2 = dt.groupby(['year'])['num_sequences_total'].sum()
data2
```

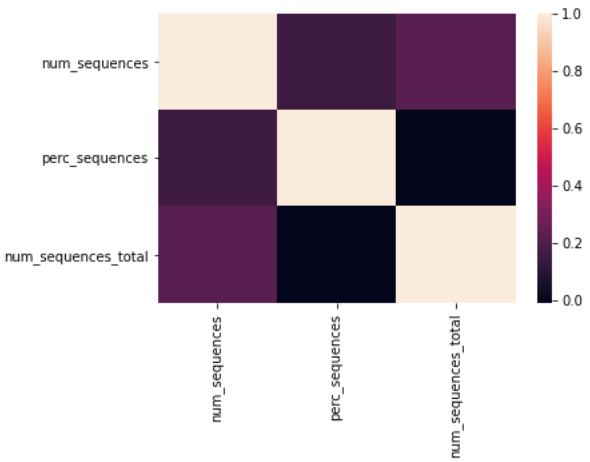
Out[50]:
```
year
2020     10942512
2021    140620224
2022        23496
Name: num_sequences_total, dtype: int64
```

In [34]:
```
#correlation between data
dt.corr()
```

Out[34]:

	num_sequences	perc_sequences	num_sequences_total
num_sequences	1.000000	0.147368	0.219677
perc_sequences	0.147368	1.000000	-0.011211
num_sequences_total	0.219677	-0.011211	1.000000

In [253]:
```
#Correlation Plot
Correlation_Plot=sb.heatmap(dt.corr())
```



Conclusion

from the above analysis of file we found, The data (COVID-19 Variants) contains the following information:
1.From barplot of varients vs occurence we can conclude the most number of sequences occure varient is delta. also we can see the top 10 varient in geven data set.

2. from the tree map of Percentage sequences per country and variant (last time registered/variant and country) we found the affection od varients country wise.delta and omicron afected in more countries.

3.In bar chart of location and sequences we can see that number of sequence occures is higher in US follwed by UK.so we can conclude that US and UK are most affected areas in covid.

4. from given data also found that highest number of occurence in year 2021.