Data preprocessing

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import plotly.express as px
```

```
from warnings import filterwarnings
filterwarnings(action='ignore')
```

```
covid = pd.read_csv('/content/StatewiseTestingDetails.csv')
covid.head()
```

	Date	State	TotalSamples	Negative	Positive
0	2020-04-17	Andaman and Nicobar Islands	1403.0	1210	12.0
1	2020-04-24	Andaman and Nicobar Islands	2679.0	NaN	27.0
2	2020-04-27	Andaman and Nicobar Islands	2848.0	NaN	33.0
3	2020-05-01	Andaman and Nicobar Islands	3754.0	NaN	33.0
4	2020-05-16	Andaman and Nicobar Islands	6677.0	NaN	33.0

```
covid.isnull().sum()
```

```
Date               0
State              0
TotalSamples       0
Negative        9367
Positive       10674
dtype: int64
```

```
covid['Negative'] = covid['Negative'].replace(np.nan, 0)
```

```
covid['Positive'] = covid['Positive'].replace(np.nan,0)
```

```
covid.head()
```

	Date	State	TotalSamples	Negative	Positive
0	2020-04-17	Andaman and Nicobar Islands	1403.0	1210	12.0

```
covid.isnull().sum()
```

```
Date            0
State           0
TotalSamples    0
Negative        0
Positive        0
dtype: int64
```

```
covid
```

	Date	State	TotalSamples	Negative	Positive
0	2020-04-17	Andaman and Nicobar Islands	1403.0	1210	12.0
1	2020-04-24	Andaman and Nicobar Islands	2679.0	0	27.0
2	2020-04-27	Andaman and Nicobar Islands	2848.0	0	33.0
3	2020-05-01	Andaman and Nicobar Islands	3754.0	0	33.0
4	2020-05-16	Andaman and Nicobar Islands	6677.0	0	33.0
...	...	...	...	...	...
16331	2021-08-06	West Bengal	15999961.0	0	0.0
16332	2021-08-07	West Bengal	16045662.0	0	0.0
16333	2021-08-08	West Bengal	16092192.0	0	0.0
16334	2021-08-09	West Bengal	16122345.0	0	0.0
16335	2021-08-10	West Bengal	16162814.0	0	0.0

16336 rows × 5 columns

```
covid_new = covid.groupby(['State'])['Positive'].sum().reset_index()
```

```
covid_new = covid.groupby(['State'])['TotalSamples','Negative','Positive'].sum().reset_inc
```

```
covid_new.head(10)
```

	State	TotalSamples	Positive
0	Andaman and Nicobar Islands	8.747008e+07	1763591.0
1	Andhra Pradesh	4.967773e+09	3859260.0
2	Arunachal Pradesh	1.636096e+08	51245.0
3	Assam	2.853509e+09	2065991.0
4	Bihar	7.392796e+09	1859345.0

```
covid_new
```

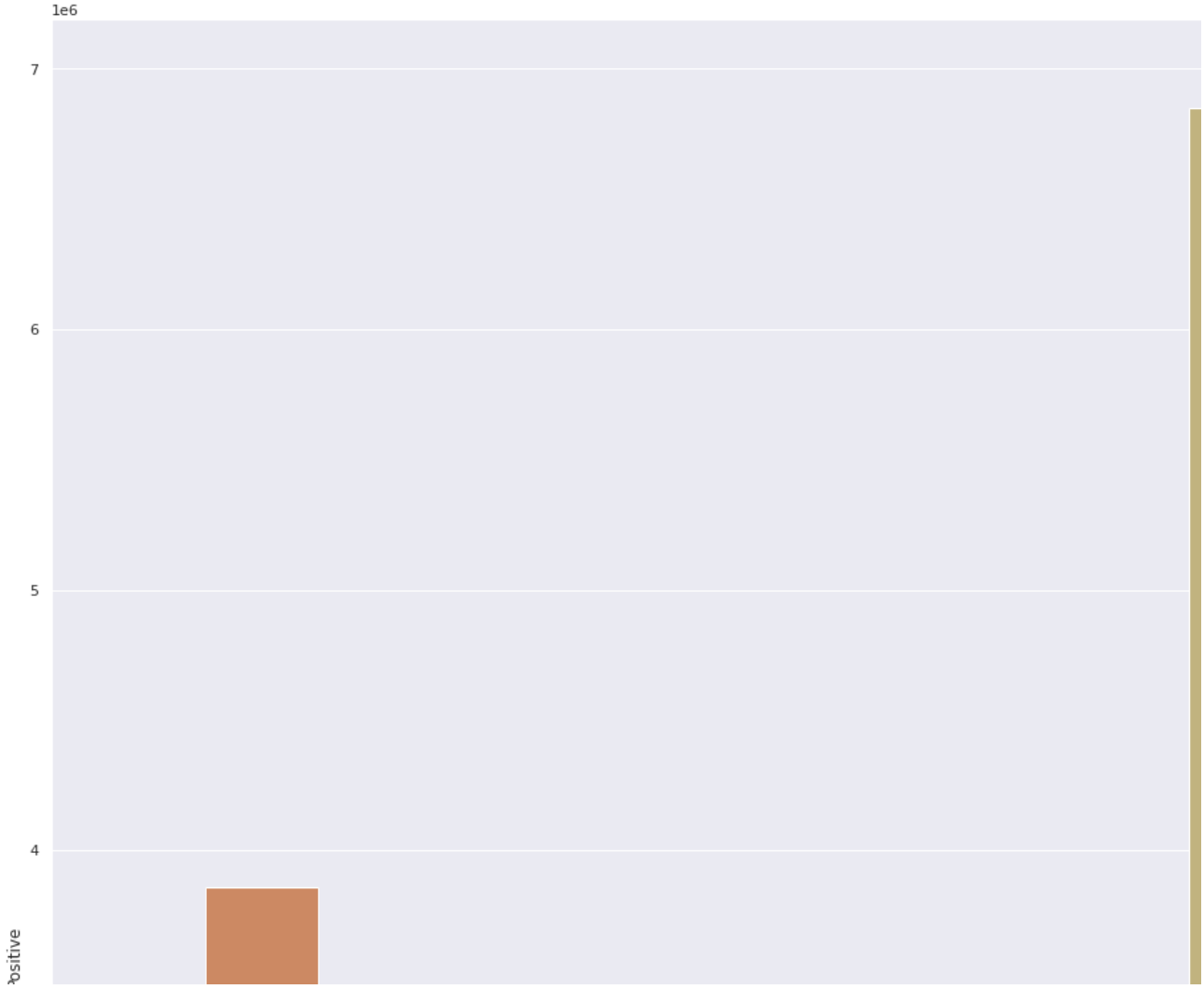
	State	TotalSamples	Positive
0	Andaman and Nicobar Islands	8.747008e+07	1763591.0
1	Andhra Pradesh	4.967773e+09	3859260.0
2	Arunachal Pradesh	1.636096e+08	51245.0
3	Assam	2.853509e+09	2065991.0
4	Bihar	7.392796e+09	1859345.0
5	Chandigarh	9.974705e+07	59195.0
6	Chhattisgarh	1.863129e+09	467857.0
7	Dadra and Nagar Haveli and Daman and Diu	6.324267e+06	169010.0
8	Delhi	4.310596e+09	6848173.0
9	Goa	1.979067e+08	266181.0
10	Gujarat	4.623914e+09	8009517.0

Data Visualization

```
fig = px.scatter(covid_new,x='TotalSamples',y='Positive',color='State')
fig.show()
```

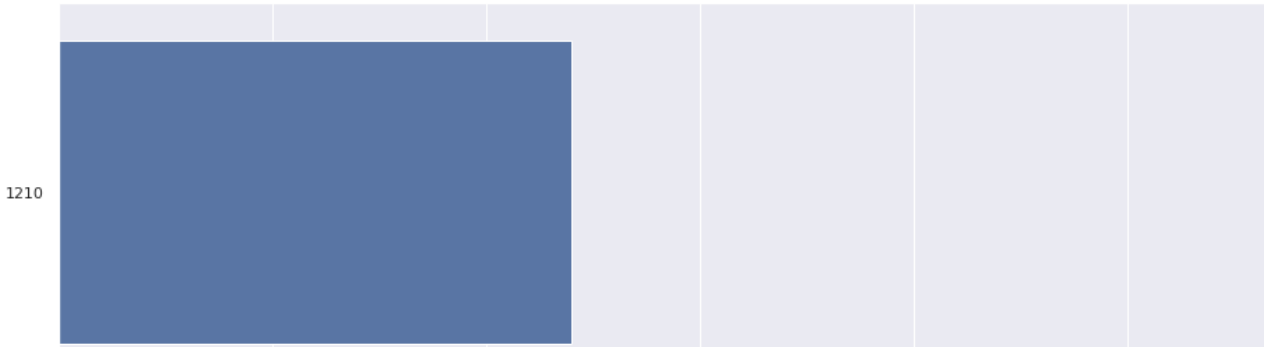
```
import seaborn as sns
sns.set(rc={'figure.figsize':(19,26)})
sns.barplot(x='State',y='Positive',data=covid_new[:10])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f980a0fd790>
```




```
sns.set(rc={'figure.figsize':(19,10)})
sns.barplot(x='Positive',y='Negative',data=covid[:10])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9809e8c390>
```




```
df_Maharashtra = covid[covid["State"]=='Maharashtra']
```

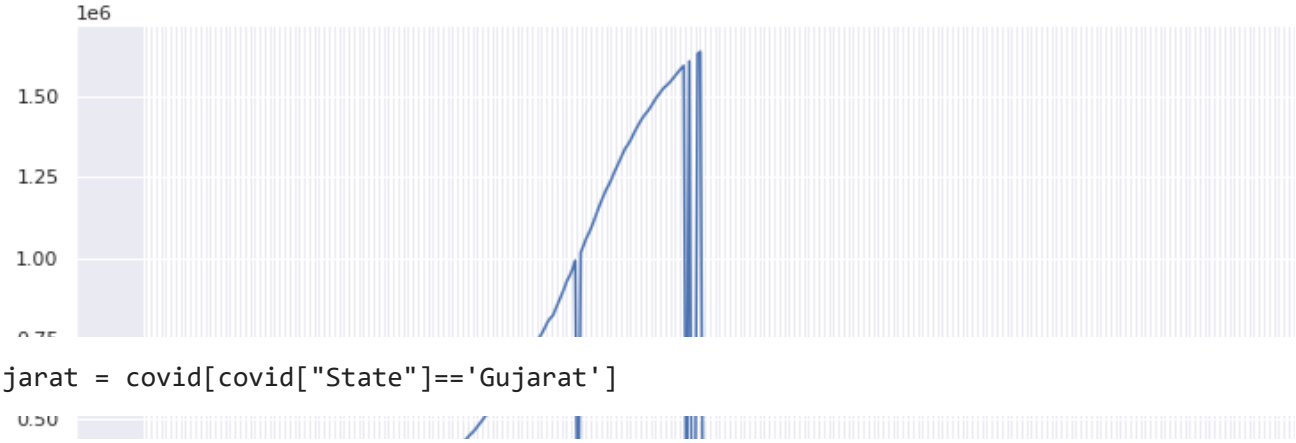
```
df_Maharashtra
```

	Date	State	TotalSamples	Negative	Positive
8888	2020-04-05	Maharashtra	16008.0	14837	0.0
8889	2020-04-06	Maharashtra	17563.0	15808	868.0
8890	2020-04-07	Maharashtra	20877.0	19290	1018.0
8891	2020-04-09	Maharashtra	20877.0	19290	868.0
8892	2020-04-10	Maharashtra	30000.0	28865	1135.0
...	...	...	...	...	...
9371	2021-08-06	Maharashtra	49172531.0	0	0.0
9372	2021-08-07	Maharashtra	49372212.0	0	0.0
9373	2021-08-08	Maharashtra	49568519.0	0	0.0
9374	2021-08-09	Maharashtra	49725694.0	0	0.0
9375	2021-08-10	Maharashtra	49905065.0	0	0.0

488 rows × 5 columns

```
fig, ax = plt.subplots()
fig.set_figwidth(15)
fig.set_figheight(6)
ax.plot(df_Maharashtra["Date"],df_Maharashtra["Positive"])
```

```
[<matplotlib.lines.Line2D at 0x7f97f931b690>]
```




```
df_Gujarat = covid[covid["State"]=='Gujarat']
```

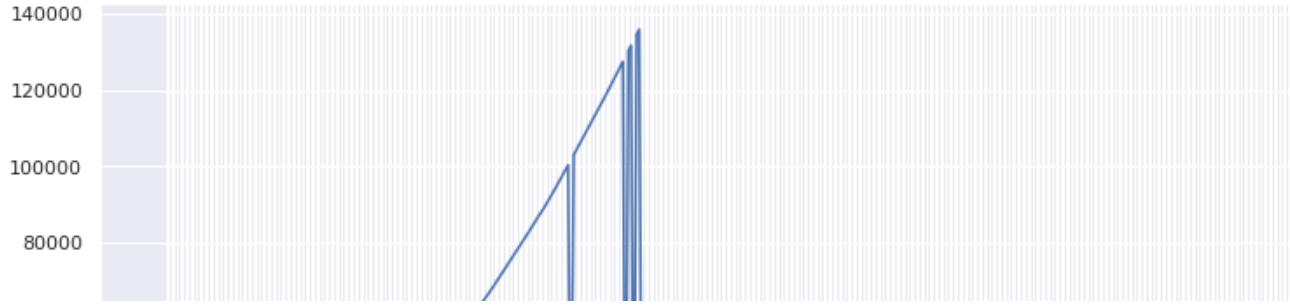
```
df_Gujarat
```

	Date	State	TotalSamples	Negative	Positive
4479	2020-04-08	Gujarat	4224.0	3905	186.0
4480	2020-04-10	Gujarat	7718.0	7237	378.0
4481	2020-04-11	Gujarat	9763.0	8888	468.0
4482	2020-04-12	Gujarat	11715.0	10867	516.0
4483	2020-04-13	Gujarat	14251.0	12970	572.0
...	...	...	...	...	...
4961	2021-08-06	Gujarat	25936650.0	0	0.0
4962	2021-08-07	Gujarat	26001986.0	0	0.0
4963	2021-08-08	Gujarat	26070279.0	0	0.0
4964	2021-08-09	Gujarat	26131219.0	0	0.0
4965	2021-08-10	Gujarat	26192626.0	0	0.0

487 rows × 5 columns

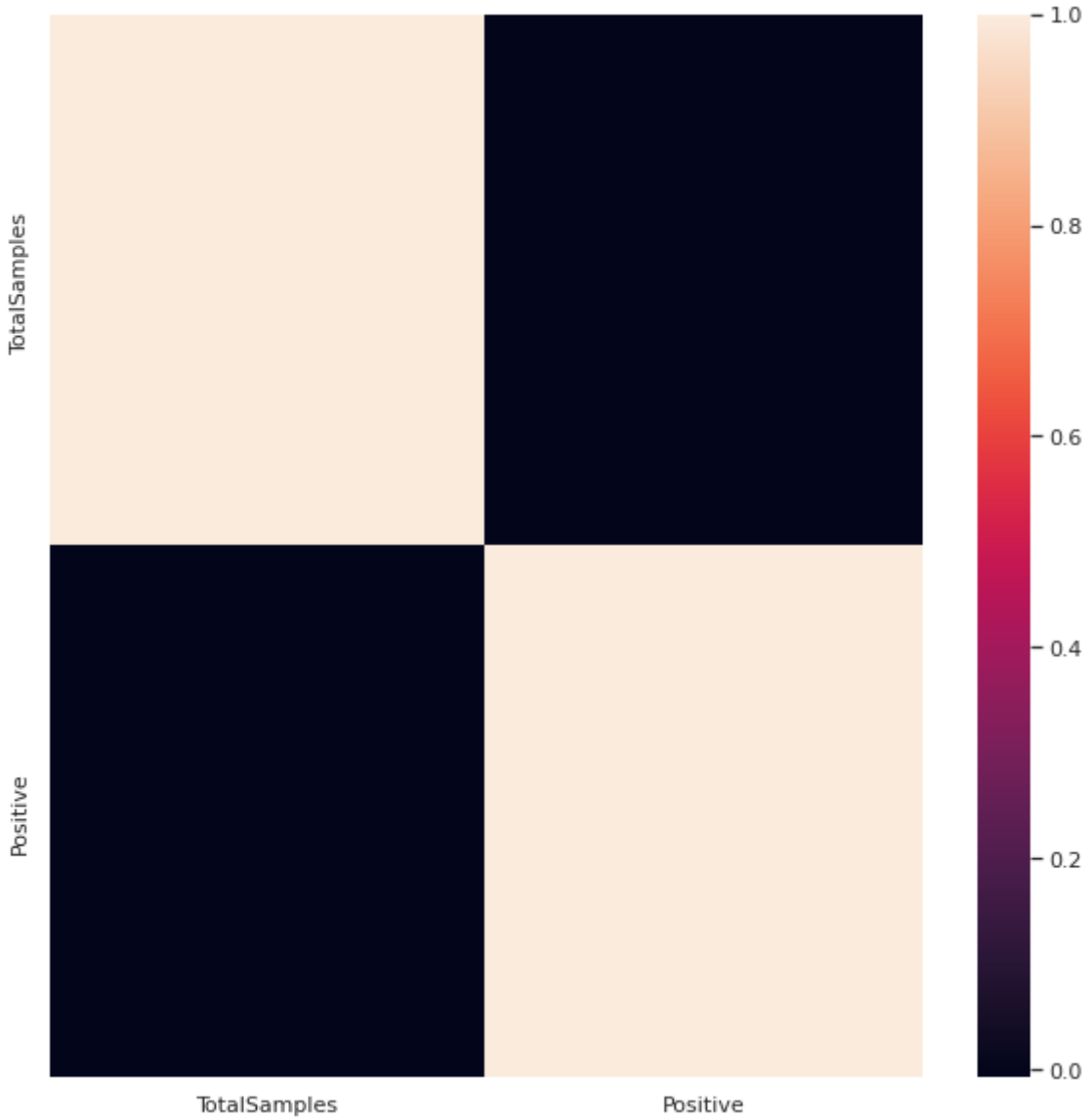
```
fig, ax = plt.subplots()
fig.set_figwidth(15)
fig.set_figheight(6)
ax.plot(df_Gujarat["Date"],df_Gujarat["Positive"])
```

```
[<matplotlib.lines.Line2D at 0x7f97f8f32cd0>]
```



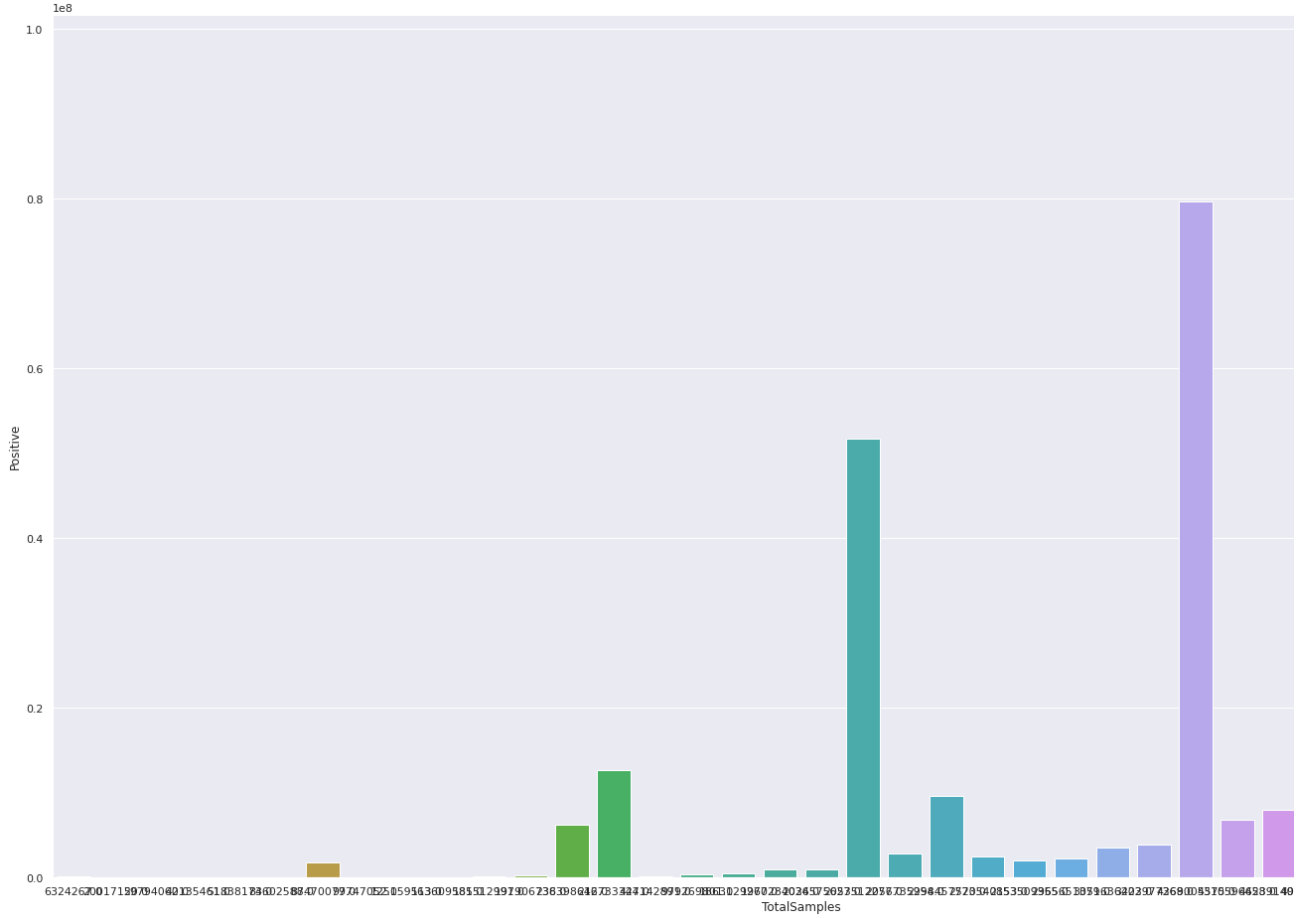

```
sns.set(rc={'figure.figsize':(10,10)})
sns.heatmap(data=covid.corr())
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f97f5f9cc90>
```




```
sns.set(rc={'figure.figsize':(27,16)})
sns.barplot(x='TotalSamples',y='Positive',data=covid_new)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f97f5d31d50>
```



Data Segmentation

here I use k-means clustering algorithm to cluster the data

```
covid1=(covid_new['TotalSamples'],covid_new['Positive'])
```

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=2, init='k-means++', random_state=0).fit(covid1)
```

```
kmeans.labels_
```

```
array([0, 1], dtype=int32)
```

```
kmeans.inertia_
```

```
0.0
```

```
kmeans.n_iter_
```

```
1
```

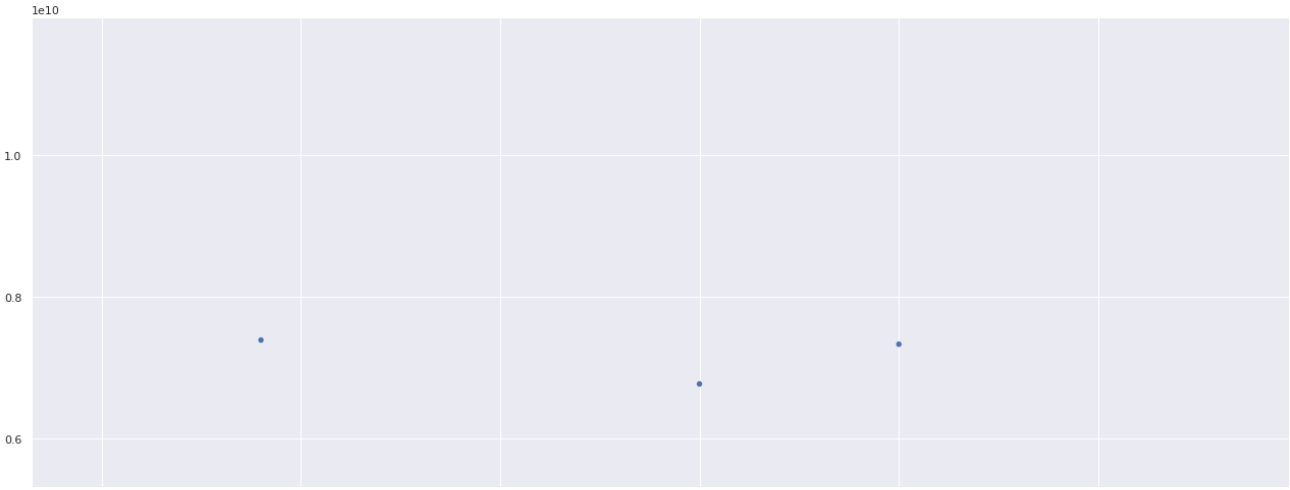
```
kmeans.cluster_centers_
```

```
array([[8.74700770e+07, 4.96777335e+09, 1.63609581e+08, 2.85350936e+09,
        7.39279636e+09, 9.97470520e+07, 1.86312928e+09, 6.32426700e+06,
        4.31059646e+09, 1.97906736e+08, 4.62391409e+09, 2.05673595e+09,
        4.27142891e+08, 1.96028403e+09, 2.05351228e+09, 6.77324794e+09,
        4.26900558e+09, 4.21354610e+07, 2.00171500e+07, 2.29845777e+09,
        7.33457382e+09, 1.85512991e+08, 1.25159513e+08, 8.46025880e+07,
        5.18381730e+07, 2.96565138e+09, 2.38398612e+08, 2.02657563e+09,
        2.52054015e+09, 2.97940600e+07, 6.71118866e+09, 3.42297737e+09,
        2.46733344e+08, 1.13881815e+10, 9.79269801e+08, 3.05163620e+09],
       [1.76359100e+06, 3.85926000e+06, 5.12450000e+04, 2.06599100e+06,
        1.85934500e+06, 5.91950000e+04, 4.67857000e+05, 1.69010000e+05,
        6.84817300e+06, 2.66181000e+05, 8.00951700e+06, 2.83015300e+06,
        1.19494000e+05, 9.77615000e+05, 5.16964950e+07, 4.70119700e+06,
        7.97231750e+07, 8.90270000e+04, 0.00000000e+00, 9.58408000e+06,
        9.69015830e+07, 1.01501000e+05, 3.39040000e+04, 1.97850000e+04,
        9.06820000e+04, 2.21445800e+06, 6.28732300e+06, 9.60287000e+05,
        2.44507600e+06, 1.76440000e+04, 1.27726040e+07, 3.85537300e+06,
        1.26307660e+07, 2.74397100e+06, 3.50257000e+05, 3.48743100e+06]])
```

```
from collections import Counter
Counter(kmeans.labels_)
```

```
Counter({0: 1, 1: 1})
```

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.scatterplot(data=covid1)
plt.show()
```




```
sns.scatterplot(data=covid1)
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1],
            marker="X", c="r", s=80, label="centroids")
plt.legend()
plt.show()
```

1e10

Silhoutte score for clustering

```
covid1=str(covid1)
covid1['TotalSamples']=int(covid1['TotalSamples'])
#covid1 = covid1[['TotalSamples']].astype({'TotalSamples': float})
#covid1=int(float(covid1))
```

```
from sklearn import datasets
from sklearn.cluster import KMeans

X1 = covid1['Totalsamples']
#X=X.astype(int)
y = covid1.Positive

km = KMeans(n_clusters=2, random_state=42)

km.fit_predict(X)

score = silhouette_score(X, km.labels_, metric='euclidean')

print('Silhouetter Score: %.3f' % score)


from yellowbrick.cluster import SilhouetteVisualizer

fig, ax = plt.subplots(2, 2, figsize=(15,8))
for i in [2, 3, 4, 5]:
    '''
    Create KMeans instance for different number of clusters
    '''
    km = KMeans(n_clusters=i, init='k-means++', n_init=10, max_iter=100, random_state=42)
    q, mod = divmod(i, 2)
    '''
    Create SilhouetteVisualizer instance with KMeans instance
    Fit the visualizer
    '''
    visualizer = SilhouetteVisualizer(km, colors='yellowbrick', ax=ax[q-1][mod])
    visualizer.fit(covid1.Positive)
```