3MTT FINAL PROJECT

TOPIC: CORONA VIRUS (COVID19)

BY: MARYAM HUSSAIN

FELLOW ID:FE/23/39684605

COURSE: DATA ANALYSIS AND VISUALIZATION.

DATA CLEANING AND PROCESSING OF CORONA VIRUS CASES.

SOURCE OF DATASET:The dataset used in this project was gotten from an online source (kaggle.com dataset)

1.IMPORTING LIBRARIES

In [12]:
```
#importing library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from plotly.subplots import make_subplots
from datetime import datetime
```

In [13]:
```
covid_df=pd.read_csv('/covid19_eco.csv')
```

DISPLAYING THE FIRST FEW ROWS OF MY DATAFRAME

In [15]:
```
covid_df.head(10)
```

Out [15]:

	ID	DATE	CONFIRMED_CASE	NEW_CASE	TOTAL_CASE	DEATH_CASE	GLOBAL_CONFIRMED_CASE	PERCENTAGE_GCC	G
0	COV-NG001	27/02/2020	0	1	1	0	82294	0.001215	1
1	COV-NG002	28/02/2020	1	0	1	0	83652	0.001195	1
2	COV-NG003	29/02/2020	1	0	1	0	84403	0.001185	1
3	COV-NG004	01/03/2020	1	0	1	0	87137	0.001148	1
4	COV-NG005	02/03/2020	1	0	1	0	88948	0.001124	1
5	COV-NG006	03/03/2020	1	0	1	0	90869	0.001100	1
6	COV-NG007	04/03/2020	1	0	1	0	93091	0.001074	2
7	COV-NG008	05/03/2020	1	0	1	0	95324	0.001049	2
8	COV-NG009	06/03/2020	1	0	1	0	98192	0.001018	2
9	COV-NG010	07/03/2020	1	0	1	0	101927	0.000981	3

GETTING SOME INFORMATION ABOUT MY DATASET

In [16]:
```
covid_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 113 entries, 0 to 112
Data columns (total 20 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   ID                     113 non-null    object
 1   DATE                   113 non-null    object
 2   CONFIRMED_CASE         113 non-null    int64
 3   NEW_CASE               113 non-null    int64
 4   TOTAL_CASE             113 non-null    int64
 5   DEATH_CASE             113 non-null    int64
 6   GLOBAL_CONFIRMED_CASE  113 non-null    int64
 7   PERCENTAGE_GCC         113 non-null    float64
 8   GLOBAL_NEW_CASES       113 non-null    object
 9   PERCENTAGE_GNC         113 non-null    float64
```

```
 10  GLOBAL_DEATH_CASES      113 non-null    int64
 11  PERCENTAGE_GDC          113 non-null    float64
 12  WEATHER                 113 non-null    int64
 13  BUYING_RATE             113 non-null    float64
 14  CENTRAL_RATE            113 non-null    float64
 15  SELLING_RATE            113 non-null    float64
 16  SHARE_INDEX             113 non-null    object
 17  OIL_PRICE               113 non-null    float64
 18  GROSS_RESERVES          113 non-null    int64
 19  LIQUID_RESERVES         113 non-null    int64
dtypes: float64(7), int64(9), object(4)
memory usage: 17.8+ KB
```

BASIC STATISTICS ABOUT MY DATASET

In [17]:
```
covid_df.describe()
```

Out [17]:

	CONFIRMED_CASE	NEW_CASE	TOTAL_CASE	DEATH_CASE	GLOBAL_CONFIRMED_CASE	PERCENTAGE_GCC	PERCENTAGE_GN
count	113.000000	113.000000	113.000000	113.000000	1.130000e+02	113.000000	113.000000
mean	3857.460177	163.769912	4021.230088	4.203540	2.954492e+06	0.075050	0.162695
std	5070.394409	185.477521	5240.724514	5.423427	2.511797e+06	0.073562	0.160561
min	0.000000	0.000000	1.000000	0.000000	8.229400e+04	0.000981	0.000000
25%	46.000000	6.000000	65.000000	0.000000	4.626840e+05	0.013748	0.015066
50%	874.000000	91.000000	982.000000	2.000000	2.544792e+06	0.038589	0.107185
75%	6696.000000	265.000000	7035.000000	7.000000	4.893186e+06	0.143771	0.276733
max	17762.000000	745.000000	18507.000000	31.000000	8.242999e+06	0.224518	0.608715

DELETING SOME UNWANTED COLUMNS

In [18]:
```
covid_df.drop(['GLOBAL_CONFIRMED_CASE','PERCENTAGE_GCC','GLOBAL_NEW_CASES','PERCENTAGE_GNC','GLOBAL_I
```

In [19]:
```
covid_df.head()
```

Out [19]:

	ID	DATE	CONFIRMED_CASE	NEW_CASE	TOTAL_CASE	DEATH_CASE
0	COV-NG001	27/02/2020	0	1	1	0
1	COV-NG002	28/02/2020	1	0	1	0
2	COV-NG003	29/02/2020	1	0	1	0
3	COV-NG004	01/03/2020	1	0	1	0
4	COV-NG005	02/03/2020	1	0	1	0

CREATING A PIVOT TABLE

In [20]:
```
dated=pd.pivot_table(covid_df,values=['CONFIRMED_CASE','NEW_CASE','DEATH_CASE'],index='DATE',aggfun
```

In [21]:
```
dated['Mortality Rate']= dated['DEATH_CASE']*100/dated['CONFIRMED_CASE']
```

In [22]:
```
dated=dated.sort_values(by='CONFIRMED_CASE',ascending=False)
```

In [23]:
```
dated.style.background_gradient(cmap='cubehelix')
```

Out [23]:

DATE	CONFIRMED_CASE	DEATH_CASE	NEW_CASE	Mortality Rate
18/06/2020	17762	6	745	0.033780
17/06/2020	17175	14	587	0.081514
16/06/2020	16685	31	490	0.185796
15/06/2020	16112	4	573	0.024826
14/06/2020	15709	13	403	0.082755
13/06/2020	15208	8	501	0.052604
12/06/2020	14581	12	627	0.082299
11/06/2020	13900	5	681	0.035971
10/06/2020	13491	17	409	0.126010
09/06/2020	12828	4	663	0.031182
08/06/2020	12513	7	315	0.055942
07/06/2020	12253	12	260	0.097935

DATE	CONFIRMED_CASE	DEATH_CASE	NEW_CASE	Mortality Rate
06/06/2020	11864	9	389	0.075860
05/06/2020	11536	10	328	0.086685
04/06/2020	11186	8	350	0.071518
03/06/2020	10838	1	348	0.009227
02/06/2020	10597	15	241	0.141549
01/06/2020	10181	12	416	0.117867
31/05/2020	9874	14	307	0.141787
30/05/2020	9321	12	553	0.128742
29/05/2020	8934	2	387	0.022386
28/05/2020	8752	5	182	0.057130
27/05/2020	8363	5	389	0.059787
26/05/2020	8087	16	276	0.197848
25/05/2020	7858	7	229	0.089081
24/05/2020	7545	5	313	0.066269
23/05/2020	7280	0	265	0.000000
22/05/2020	7035	10	245	0.142146
21/05/2020	6696	9	339	0.134409
20/05/2020	6412	10	284	0.155958
19/05/2020	6186	1	226	0.016166
18/05/2020	5970	9	216	0.150754
17/05/2020	5632	6	338	0.106534
16/05/2020	5456	5	176	0.091642
15/05/2020	5168	4	288	0.077399
14/05/2020	4975	3	193	0.060302
13/05/2020	4791	6	184	0.125235
12/05/2020	4645	6	146	0.129171
11/05/2020	4403	10	242	0.227118
10/05/2020	4155	17	248	0.409146
09/05/2020	3916	11	239	0.280899
08/05/2020	3530	10	386	0.283286
07/05/2020	3149	4	381	0.127024
06/05/2020	2954	5	195	0.169262
05/05/2020	2806	5	148	0.178190
04/05/2020	2561	6	245	0.234283
03/05/2020	2391	2	170	0.083647
02/05/2020	2171	17	220	0.783049
01/05/2020	1933	10	238	0.517331
30/04/2020	1729	7	204	0.404858
29/04/2020	1533	7	196	0.456621
28/04/2020	1338	4	195	0.298954
27/04/2020	1274	0	64	0.000000
26/04/2020	1183	5	91	0.422654
25/04/2020	1096	3	87	0.273723
24/04/2020	982	1	114	0.101833
23/04/2020	874	3	108	0.343249
22/04/2020	783	3	91	0.383142
21/04/2020	666	3	117	0.450450
20/04/2020	628	1	38	0.159236
19/04/2020	542	2	86	0.369004
18/04/2020	493	2	48	0.405680
17/04/2020	442	4	51	0.904977
16/04/2020	407	1	35	0.245700
15/04/2020	373	1	34	0.268097

DATE	CONFIRMED_CASE	DEATH_CASE	NEW_CASE	Mortality Rate
14/04/2020	343	1	30	0.291545
13/04/2020	323	0	20	0.000000
12/04/2020	318	0	5	0.000000
11/04/2020	305	3	13	0.983607
10/04/2020	288	0	17	0.000000
09/04/2020	276	1	12	0.362319
08/04/2020	254	0	22	0.000000
07/04/2020	238	1	16	0.420168
06/04/2020	232	0	6	0.000000
05/04/2020	214	0	18	0.000000
04/04/2020	209	0	5	0.000000
03/04/2020	184	0	25	0.000000
02/04/2020	174	0	10	0.000000
01/04/2020	139	0	35	0.000000
31/03/2020	131	0	8	0.000000
30/03/2020	111	1	20	0.900901
29/03/2020	97	0	14	0.000000
28/03/2020	70	0	27	0.000000
27/03/2020	65	0	5	0.000000
26/03/2020	46	0	19	0.000000
25/03/2020	36	0	10	0.000000
24/03/2020	30	1	6	3.333333
23/03/2020	20	0	10	0.000000
22/03/2020	15	0	5	0.000000
21/03/2020	12	0	3	0.000000
20/03/2020	12	0	0	0.000000
19/03/2020	8	0	4	0.000000
18/03/2020	3	0	5	0.000000
17/03/2020	3	0	0	0.000000
15/03/2020	2	0	0	0.000000
16/03/2020	2	0	1	0.000000
14/03/2020	2	0	0	0.000000
13/03/2020	2	0	0	0.000000
12/03/2020	2	0	0	0.000000
11/03/2020	2	0	0	0.000000
10/03/2020	2	0	0	0.000000
09/03/2020	2	0	0	0.000000
06/03/2020	1	0	0	0.000000
02/03/2020	1	0	0	0.000000
03/03/2020	1	0	0	0.000000
04/03/2020	1	0	0	0.000000
05/03/2020	1	0	0	0.000000
29/02/2020	1	0	0	0.000000
07/03/2020	1	0	0	0.000000
08/03/2020	1	0	1	0.000000
28/02/2020	1	0	0	0.000000
01/03/2020	1	0	0	0.000000
27/02/2020	0	0	1	nan

PLOTING A CHART

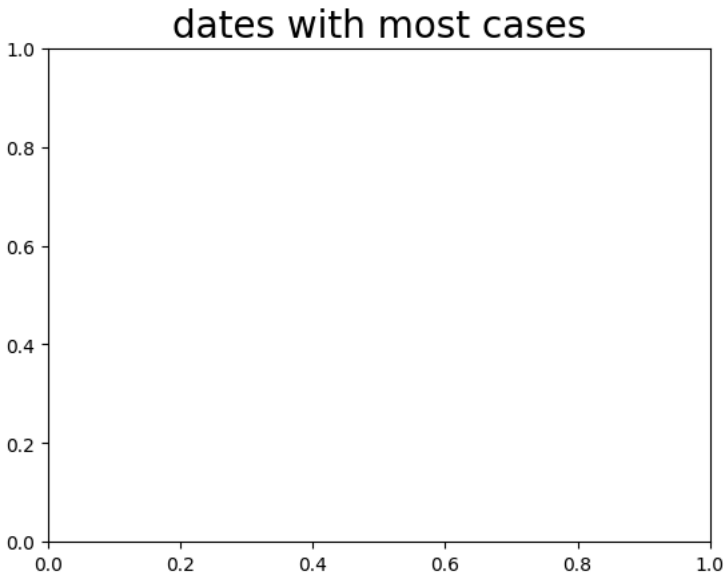
In [24]:
```
#DATES WITH MOST CASES
dates_with_most_cases= covid_df.groupby(by='DEATH_CASE').max()[['CONFIRMED_CASE','DATE']].sort_value
```

In [36]:
```
fig=plt.figure(figsize=(30,9))
```

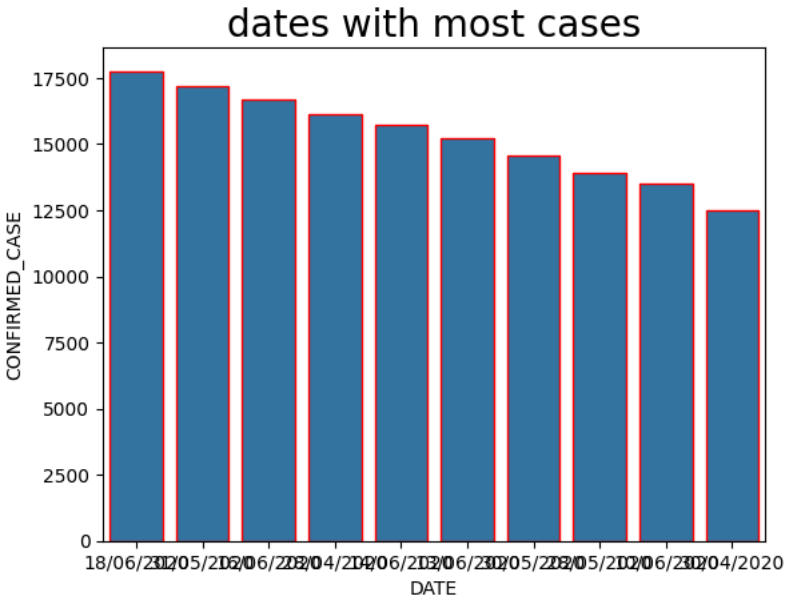
```
<Figure size 3000x900 with 0 Axes>
```

In [37]:
```
plt.title('dates with most cases',size =20)
```

Out [37]:
```
Text(0.5, 1.0, 'dates with most cases')
```



In [38]:
```
plt.title('dates with most cases',size=20)
ax=sns.barplot(data=dates_with_most_cases.iloc[:10],y='CONFIRMED_CASE',x='DATE',linewidth =1,edgecol
```



In [ ]:

In [ ]: