```
'''
Soren DeOrlow
IDSN 599, Spring 2021
Deorlow@usc.edu
Final Project Part 2
'''
```

PART 1

Domain: What is the type/nature of the problem that you are thinking to solve

Through the process of searching for data sources, there has been an abundance of data surrounding covid-19 along with interesting projects attempting to draw new insights from this vast data. I seek to become aware of the work that is happening in this space and through prototyping, gain new insight on data patterns around covid-19.

Dataset: Where did you get your dataset? How big is it – how many rows and attributes? Why/how does it contain the data you need to solve the problem you are thinking of solving?

I have found numerous datasets on Kaggle with fascinating data on covid 19, including lung x-ray imaging and covid vaccine related tweets. The cleanest dataset that I found maps vaccine data globally in relation to total population as well as covid fatalities. It contains 21k rows of data on global patterns. I also discovered that Roche has launched a project called the uncover Covid-19 challenge focused on data exploration and research. This project has launched 18 sub tasks inviting data research contributions to provide new perspective and answers to questions such as, "Predicting illness severity in a particular patient or demographic." I have chosen two similar datasets that I am working with to gain greater insight into the patterns of covid-19 from a public health standpoint and how various populations around the world have responded to vaccines and other measures.

Problem Type: Are you creating a predictor, a classifier, or something else? Why do you think this is the way to go to solve the problem?

I plan to use classification to gain insight into how different parts of the world are approaching vaccination and the velocity of vaccination over time. I also am looking at the relationships of GDP, vaccination and covid morbidity. I'm less concerned with the relationship between covid morbidity, but I will not rule out any insights that might be found. Ultimately, I would like to see changes over time.

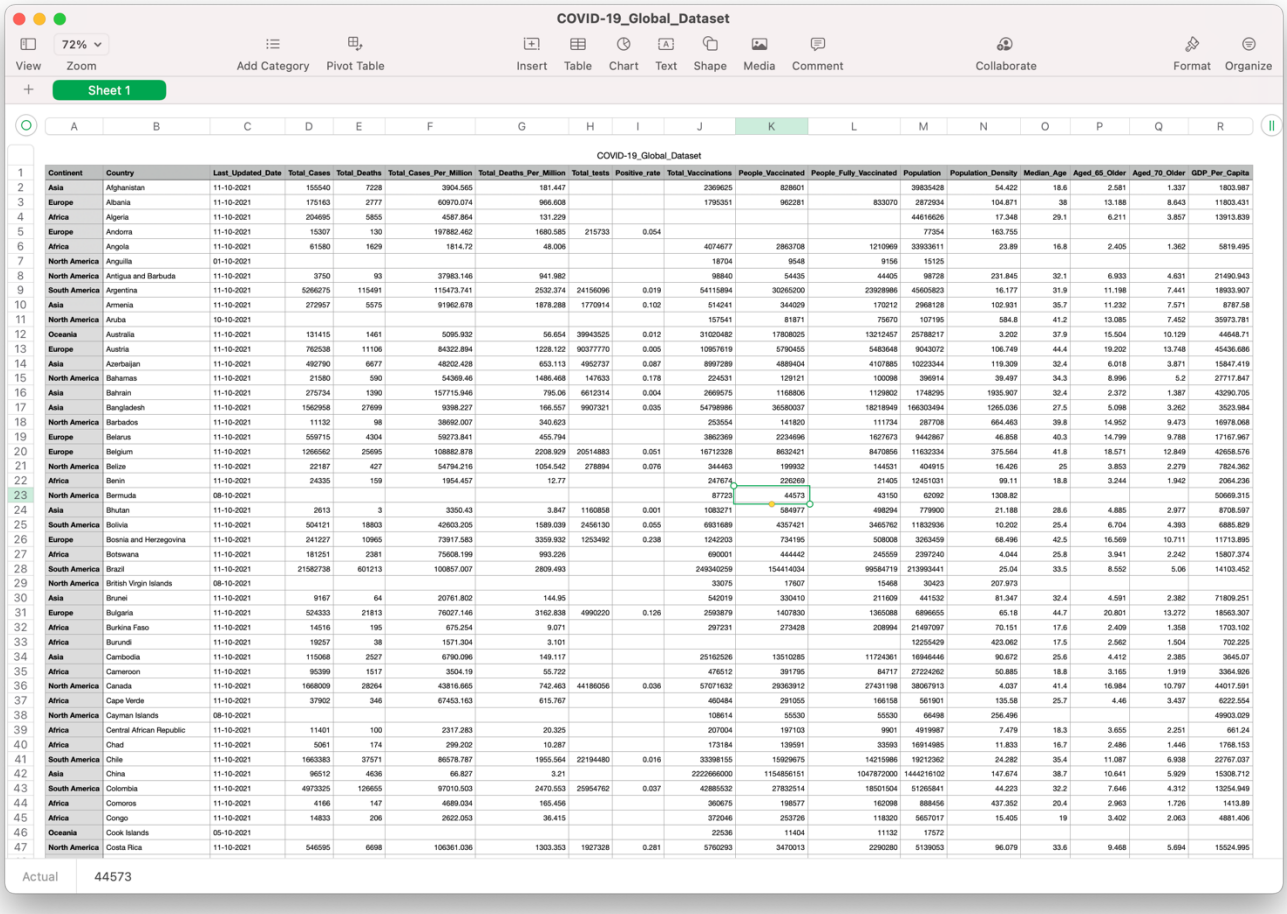
Attributes: What are the attributes of the dataset? Which are numeric and which are text? Do you have any missing values for attributes?

As I was exploring various datasets, I was careful to select datasets that were robust and consisted primarily of numeric data, with limited missing attributes. The datasets that I have chosen, list countries and how they are doing in the fight against covid-19. Below is a sample of both dataset including a complete view of the column headers.

Dataset 1

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	New_deaths	population	ratio
0	Afghanistan	AFG	5/11/21	504502	448878	55624	12	40146987	1.118086396
1	Afghanistan	AFG	5/20/21	547901	470341	77560	10	40146987	1.171547444
2	Afghanistan	AFG	5/24/21	573277	476367	96910	10	40146987	1.186557288
3	Afghanistan	AFG	5/26/21	590454	479372	111082	19	40146987	1.194042283
4	Afghanistan	AFG	5/27/21	593313	479574	113739	14	40146987	1.194545434
5	Afghanistan	AFG	5/30/21	600152	480226	119926	20	40146987	1.196169466

Dataset 2

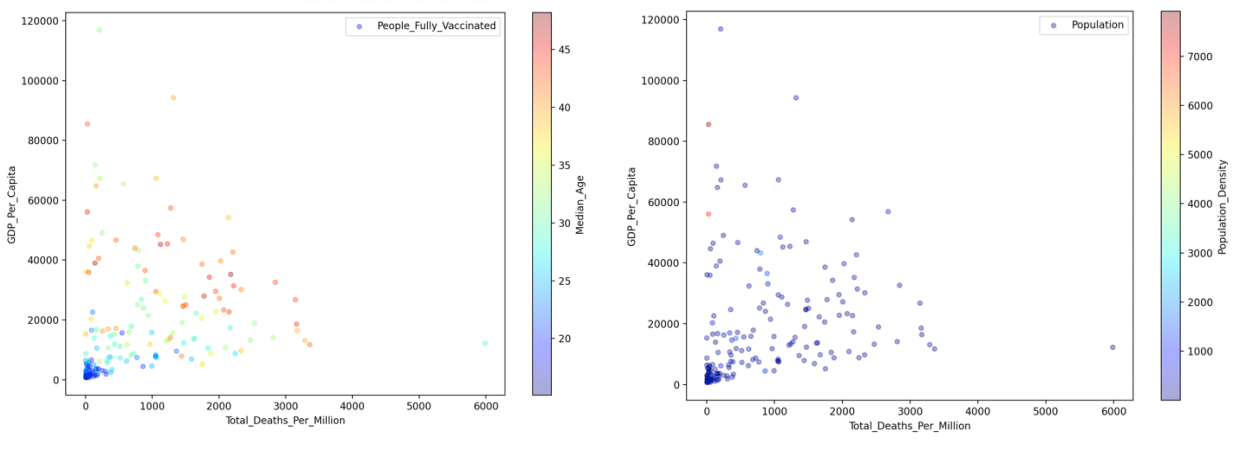


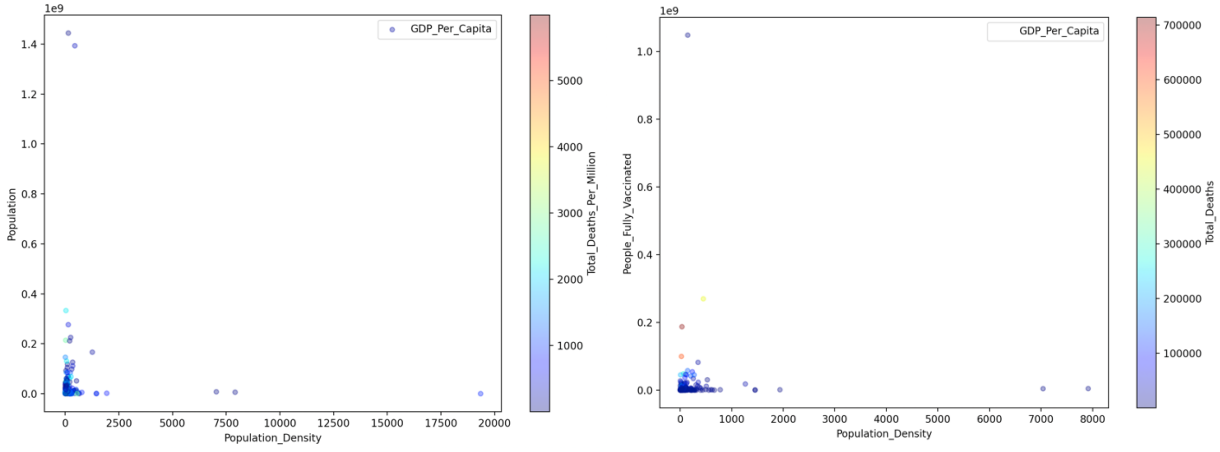
COVID-19_Global_Dataset

	Continent	Country	Last_Updated_Date	Total_Cases	Total_Deaths	Total_Cases_Per_Million	Total_Deaths_Per_Million	Total_tests	Positive_rate	Total_Vaccinations	People_Vaccinated	People_Fully_Vaccinated	Population	Population_Density	Median_Age	Aged_65_Older	Aged_70_Older	GDP_Per_Capita
2	Asia	Afghanistan	11-10-2021	155540	7228	3904.565	181.447			2369625	828601		39835428	54.422	18.6	2.581	1.337	1803.987
3	Europe	Albania	11-10-2021	175163	2777	60970.074	966.608			1795351	962281	833070	2872934	104.871	38	13.188	8.643	11803.431
4	Africa	Algeria	11-10-2021	204695	5855	4587.864	131.229						44616626	17.348	29.1	6.211	3.857	13913.839
5	Europe	Andorra	11-10-2021	15307	130	197882.462	1680.585	215733	0.054				77354	163.755				
6	Africa	Angola	11-10-2021	61580	1629	1814.72	48.006			4074677	2863708	1210969	33933611	23.89	16.8	2.405	1.362	5819.495
7	North America	Anguilla	01-10-2021							18704	9548	9156	15125					
8	North America	Antigua and Barbuda	11-10-2021	3750	93	37983.146	941.982			98840	54435	44405	98728	231.845	32.1	6.933	4.631	21490.943
9	South America	Argentina	11-10-2021	5266275	115491	115473.741	2532.374	24156096	0.019	54115894	30265200	23928986	45605823	16.177	31.9	11.198	7.441	18933.907
10	Asia	Armenia	11-10-2021	272957	5575	91962.678	1878.288	1770914	0.102	514241	344029	170212	2968128	102.931	35.7	11.232	7.571	8787.58
11	North America	Aruba	10-10-2021							157541	81871	75670	107195	584.8	41.2	13.085	7.452	35973.781
12	Oceania	Australia	11-10-2021	131415	1461	5095.932	56.654	39943525	0.012	31020482	17808025	13212457	25788217	3.202	37.9	15.504	10.129	44648.71
13	Europe	Austria	11-10-2021	762538	11106	84322.894	1228.122	90377770	0.005	10957619	5790455	5483648	9043072	106.749	44.4	19.202	13.748	45436.686
14	Asia	Azerbaijan	11-10-2021	492790	6677	48202.428	653.113	4952737	0.087	8997289	4889404	4107885	10223344	119.309	32.4	6.018	3.871	15847.419
15	North America	Bahamas	11-10-2021	21580	590	54369.46	1486.468	147633	0.178	224531	129121	100098	396914	39.497	34.3	8.996	5.2	27717.847
16	Asia	Bahrain	11-10-2021	275734	1390	157715.946	795.06	6612314	0.004	2669575	1168806	1129802	1748295	1935.907	32.4	2.372	1.387	43290.705
17	Asia	Bangladesh	11-10-2021	1562958	27699	9398.227	166.557	9907321	0.035	54798986	36580037	18218949	166303494	1265.036	27.5	5.098	3.262	3523.984
18	North America	Barbados	11-10-2021	11132	98	38692.007	340.623			253554	141820	111734	287708	664.463	39.8	14.952	9.473	16978.068
19	Europe	Belarus	11-10-2021	559715	4304	59273.841	455.794			3862369	2234696	1627673	9442867	46.858	40.3	14.799	9.788	17167.967
20	Europe	Belgium	11-10-2021	1266562	25695	108882.878	2208.929	20514883	0.051	16712328	8632421	8470856	11632334	375.564	41.8	18.571	12.849	42658.576
21	North America	Belize	11-10-2021	22187	427	54794.216	1054.542	278894	0.076	344463	199932	144531	404915	16.426	25	3.853	2.279	7824.362
22	Africa	Benin	11-10-2021	24335	159	1954.457	12.77			247674	226269	21405	12451031	99.11	18.8	3.244	1.942	2064.236
23	North America	Bermuda	08-10-2021							87723	44573	43150	62092	1308.82				50669.315
24	Asia	Bhutan	11-10-2021	2613	3	3350.43	3.847	1160858	0.001	1083271	584977	498294	779900	21.188	28.6	4.885	2.977	8708.597
25	South America	Bolivia	11-10-2021	504121	18803	42603.205	1589.039	2456130	0.055	6931689	4357421	3465762	11832936	10.202	25.4	6.704	4.393	6885.829
26	Europe	Bosnia and Herzegovina	11-10-2021	241227	10965	73917.583	3359.932	1253492	0.238	1242203	734195	508008	3263459	68.496	42.5	16.569	10.711	11713.895
27	Africa	Botswana	11-10-2021	181251	2381	75608.199	993.226			690001	444442	245559	2397240	4.044	25.8	3.941	2.242	15807.374
28	South America	Brazil	11-10-2021	21582738	601213	100857.007	2809.493			249340259	154414034	99584719	213993441	25.04	33.5	8.552	5.06	14103.452
29	North America	British Virgin Islands	08-10-2021							33075	17607	15468	30423	207.973				
30	Asia	Brunei	11-10-2021	9167	64	20761.802	144.95			542019	330410	211609	441532	81.347	32.4	4.591	2.382	71809.251
31	Europe	Bulgaria	11-10-2021	524333	21813	76027.146	3162.838	4990220	0.126	2593879	1407830	1365088	6896655	65.18	44.7	20.801	13.272	18563.307
32	Africa	Burkina Faso	11-10-2021	14516	195	675.254	9.071			297231	273428	208994	21497097	70.151	17.6	2.409	1.358	1703.102
33	Africa	Burundi	11-10-2021	19257	38	1571.304	3.101						12255429	423.062	17.5	2.562	1.504	702.225
34	Asia	Cambodia	11-10-2021	115068	2527	6790.096	149.117			25162526	13510285	11724361	16946446	90.672	25.6	4.412	2.385	3645.07
35	Africa	Cameroon	11-10-2021	95399	1517	3504.19	55.722			476512	391795	84717	27224262	50.885	18.8	3.165	1.919	3364.926
36	North America	Canada	11-10-2021	1668009	28264	43816.665	742.463	44186056	0.036	57071632	29363912	27431198	38067913	4.037	41.4	16.984	10.797	44017.591
37	Africa	Cape Verde	11-10-2021	37902	346	67453.163	615.767			460484	291055	166158	561901	135.58	25.7	4.46	3.437	6222.554
38	North America	Cayman Islands	08-10-2021							108614	55530	55530	66498	256.496				49903.029
39	Africa	Central African Republic	11-10-2021	11401	100	2317.283	20.325			207004	197103	9901	4919987	7.479	18.3	3.655	2.251	661.24
40	Africa	Chad	11-10-2021	5061	174	299.202	10.287			173184	139591	33593	16914985	11.833	16.7	2.486	1.446	1768.153
41	South America	Chile	11-10-2021	1663383	37571	86578.787	1955.564	22194480	0.016	33398155	15929675	14215986	19212362	24.282	35.4	11.087	6.938	22767.037
42	Asia	China	11-10-2021	96512	4636	66.827	3.21			2222666000	1154856151	1047872000	1444216102	147.674	38.7	10.641	5.929	15308.712
43	South America	Colombia	11-10-2021	4973325	126655	97010.503	2470.553	25954762	0.037	42885532	27832514	18501504	51265841	44.223	32.2	7.646	4.312	13254.949
44	Africa	Comoros	11-10-2021	4166	147	4689.034	165.456			360675	198577	162098	888456	437.352	20.4	2.963	1.726	1413.89
45	Africa	Congo	11-10-2021	14833	206	2622.053	36.415			372046	253726	118320	5657017	15.405	19	3.402	2.063	4881.406
46	Oceania	Cook Islands	05-10-2021							22536	11404	11132	17572					
47	North America	Costa Rica	11-10-2021	546595	6698	106361.036	1303.353	1927328	0.281	5760293	3470013	2290280	5139053	96.079	33.6	9.468	5.694	15524.995

Actual 44573

Initial explorations into data patterns



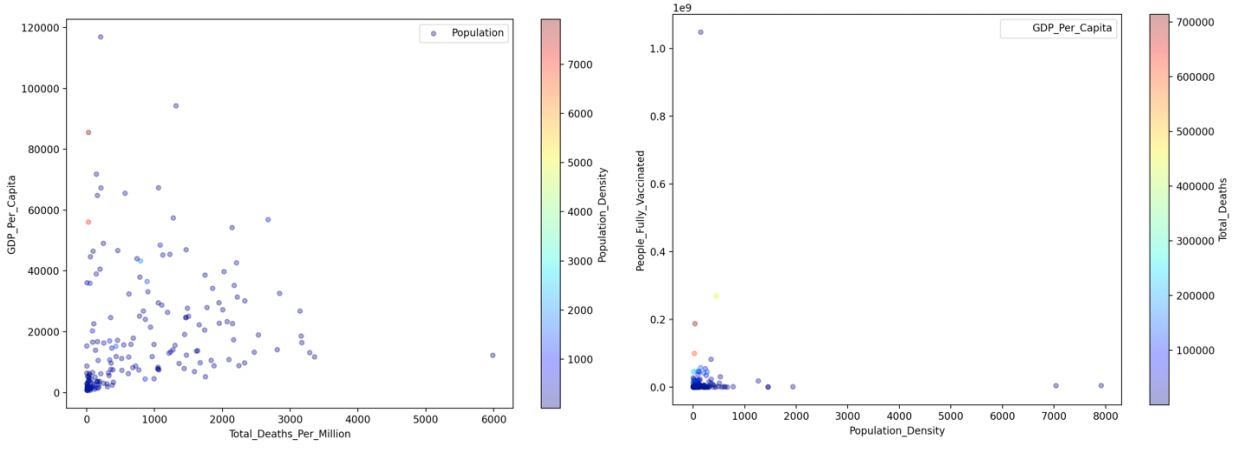


PART 2

ML Algorithm: *You are to choose three ML algorithms and describe why you have selected those algorithms. Justify your choices of ML algorithms.*

I will be using pd.get_dummies (one hot encoder) to categorize countries
I will be using cartopy.io.shapereader in order to populate a global map with my dataset.
I will be using matplotlib.pyplot to evaluate and cluster my dataset to reveal relevant patterns.

Correlations: *Show some numbers, or screen shots of plots to show which pairs of your data attribute correlations are most likely to be of value in your analysis. I suggest also showing plots as it really shows the quality of the correlation, instead of just numeric values.*



Within the first dataset there is a correlation between GDP Per Capita and People Fully Vaccinated.

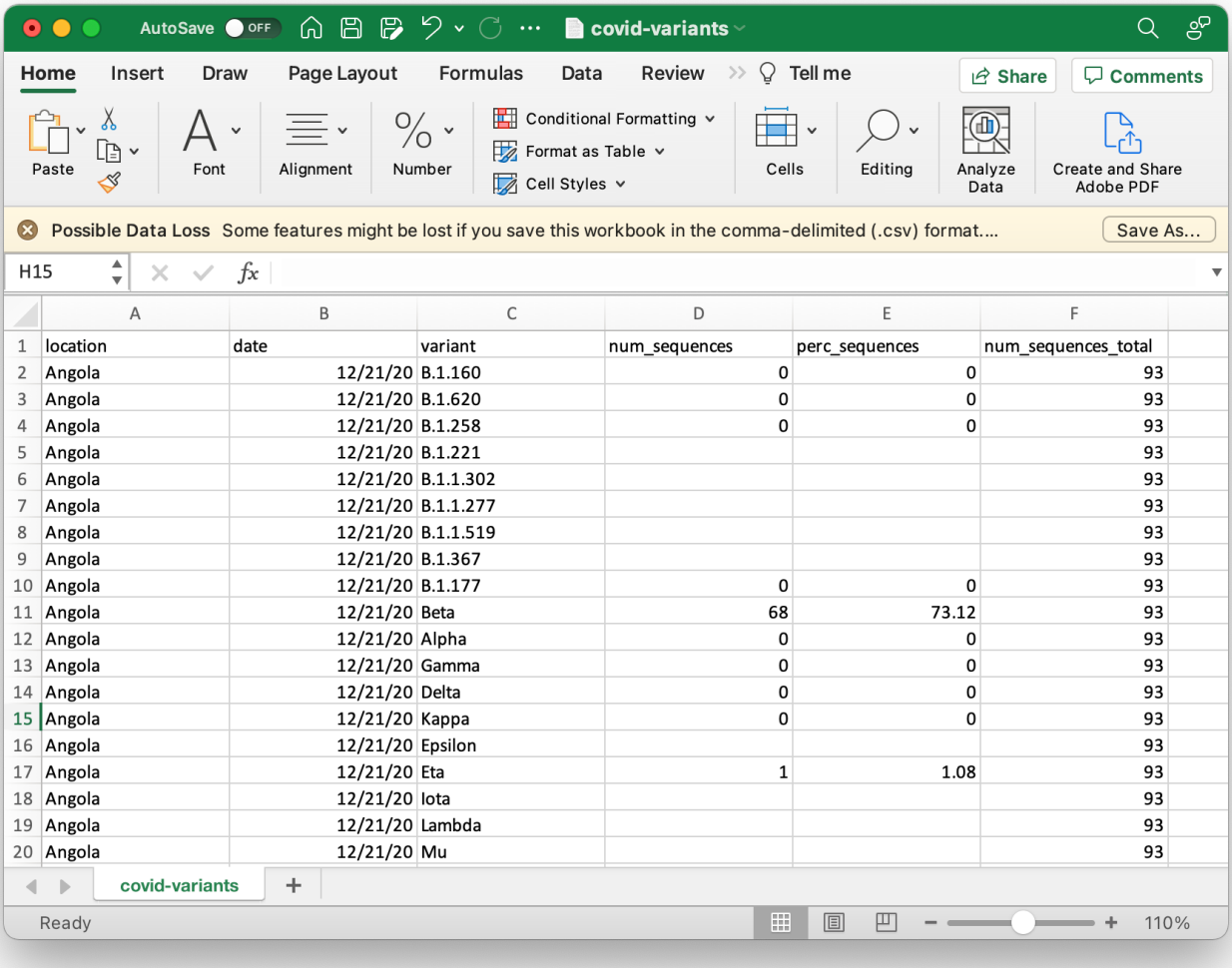
Transformers: *Discuss each of your transformers, whether custom or not. Explain why you chose this transformer and what are each of them doing. I suggest using at least the Pipeline class. You may also want to use the ColumnTransformer class if you have categorical (text) data.*

I will be using pd.get_dummies (one hot encoder) to transform nominal categorical data into categorized countries. This transformation will allow us to involve countries as a data source.

For my second dataset, I will create a custom transformer that will convert the categorical names of the different types of variants into a data source.

General notes.

The two datasets that I have selected show great promise. I am excited about exploring the migration of variants across the globe and the conditions that have enabled a variant to proliferate.

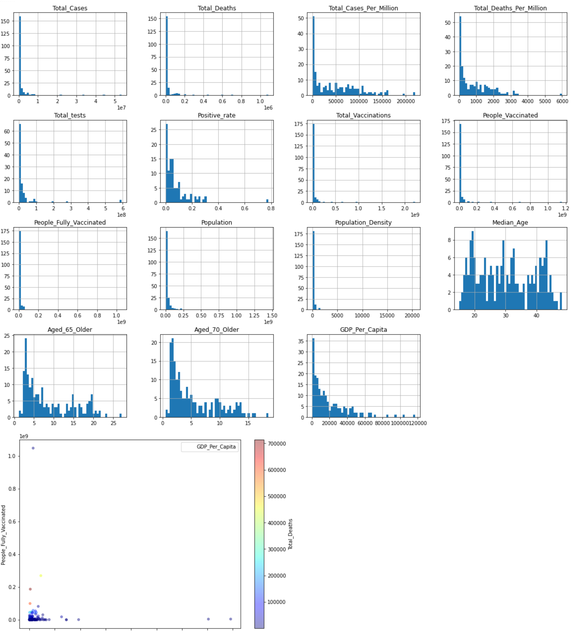


	A	B	C	D	E	F
1	location	date	variant	num_sequences	perc_sequences	num_sequences_total
2	Angola	12/21/20	B.1.160	0	0	93
3	Angola	12/21/20	B.1.620	0	0	93
4	Angola	12/21/20	B.1.258	0	0	93
5	Angola	12/21/20	B.1.221			93
6	Angola	12/21/20	B.1.1.302			93
7	Angola	12/21/20	B.1.1.277			93
8	Angola	12/21/20	B.1.1.519			93
9	Angola	12/21/20	B.1.367			93
10	Angola	12/21/20	B.1.177	0	0	93
11	Angola	12/21/20	Beta	68	73.12	93
12	Angola	12/21/20	Alpha	0	0	93
13	Angola	12/21/20	Gamma	0	0	93
14	Angola	12/21/20	Delta	0	0	93
15	Angola	12/21/20	Kappa	0	0	93
16	Angola	12/21/20	Epsilon			93
17	Angola	12/21/20	Eta	1	1.08	93
18	Angola	12/21/20	Iota			93
19	Angola	12/21/20	Lambda			93
20	Angola	12/21/20	Mu			93

In [12]:

```
import pandas as pd
import os
import matplotlib.pyplot as plt



COVID19_PATH = os.path.join("COVID-19_Global_Dataset.csv")

def load_covid19_data(covid19_path = COVID19_PATH):
    return pd.read_csv(csv_path)


covid19 = pd.read_csv(COVID19_PATH)
covid19.info()  #The info method is usefult to get a quick description of the data
print(covid19.info())
#print(covid19["people_vaccinated"].value_counts()) #Shows what categories exist and how many districts belong to each category
#print(covid19.describe()) #This method shows a summary of the numerical attributes
covid19.hist( bins=50, figsize=(20,15)) #shows the number of instances (vertical axis) that have a given value range
plt.show()#Plots a histogram for each numerical attribute
#covid19.plot(kind="scatter", x="GDP_Per_Capita", y="Total_Deaths", alpha=0.1)
#covid19.plot(kind="scatter", x="Total_Cases_Per_Million", y="GDP_Per_Capita", alpha=0.1)
covid19.plot(kind="scatter", x="Population_Density", y="People_Fully_Vaccinated", alpha=0.4, label="GDP_Per_Capita", figsize=(10,7),c="Total_Deaths",cmap=plt.get_cmap("jet"),colorbar=True)
#print(covid19.median().values)
plt.legend()
plt.show() #This is required in PyCharm - not shown in book
```

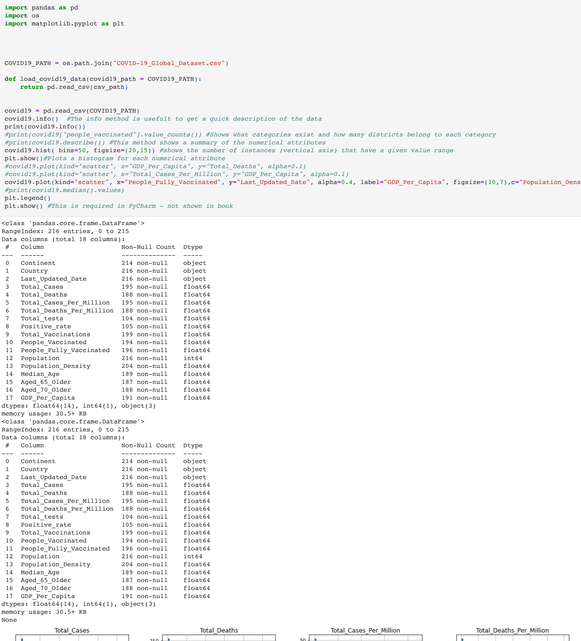
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 216 entries, 0 to 215
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Continent                   214 non-null    object
 1   Country                     216 non-null    object
 2   Last_Updated_Date           216 non-null    object
 3   Total_Cases                 195 non-null    float64
 4   Total_Deaths                188 non-null    float64
 5   Total_Cases_Per_Million     195 non-null    float64
 6   Total_Deaths_Per_Million    188 non-null    float64
 7   Total_tests                 104 non-null    float64
 8   Positive_rate               105 non-null    float64
 9   Total_Vaccinations          199 non-null    float64
 10  People_Vaccinated           194 non-null    float64
 11  People_Fully_Vaccinated     196 non-null    float64
 12  Population                  216 non-null    int64
 13  Population_Density          204 non-null    float64
 14  Median_Age                  189 non-null    float64
 15  Aged_65_Older               187 non-null    float64
 16  Aged_70_Older               188 non-null    float64
 17  GDP_Per_Capita              191 non-null    float64
dtypes: float64(14), int64(1), object(3)
memory usage: 30.5+ KB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 216 entries, 0 to 215
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Continent                   214 non-null    object
 1   Country                     216 non-null    object
 2   Last_Updated_Date           216 non-null    object
 3   Total_Cases                 195 non-null    float64
 4   Total_Deaths                188 non-null    float64
 5   Total_Cases_Per_Million     195 non-null    float64
 6   Total_Deaths_Per_Million    188 non-null    float64
 7   Total_tests                 104 non-null    float64
 8   Positive_rate               105 non-null    float64
 9   Total_Vaccinations          199 non-null    float64
 10  People_Vaccinated           194 non-null    float64
 11  People_Fully_Vaccinated     196 non-null    float64
 12  Population                  216 non-null    int64
 13  Population_Density          204 non-null    float64
 14  Median_Age                  189 non-null    float64
 15  Aged_65_Older               187 non-null    float64
 16  Aged_70_Older               188 non-null    float64
 17  GDP_Per_Capita              191 non-null    float64
dtypes: float64(14), int64(1), object(3)
memory usage: 30.5+ KB
None
```



In [11]:

```
import pandas as pd
import os
import matplotlib.pyplot as plt



COVID19_PATH = os.path.join("COVID-19_Global_Dataset.csv")

def load_covid19_data(covid19_path = COVID19_PATH):
    return pd.read_csv(csv_path)


covid19 = pd.read_csv(COVID19_PATH)
covid19.info()  #The info method is usefult to get a quick description of the data
print(covid19.info())
#print(covid19["people_vaccinated"].value_counts()) #Shows what categories exist and how many districts belong to each category
#print(covid19.describe()) #This method shows a summary of the numerical attributes
covid19.hist( bins=50, figsize=(20,15)) #shows the number of instances (vertical axis) that have a given value range
plt.show()#Plots a histogram for each numerical attribute
#covid19.plot(kind="scatter", x="GDP_Per_Capita", y="Total_Deaths", alpha=0.1)
#covid19.plot(kind="scatter", x="Total_Cases_Per_Million", y="GDP_Per_Capita", alpha=0.1)
covid19.plot(kind="scatter", x="People_Fully_Vaccinated", y="Last_Updated_Date", alpha=0.4, label="GDP_Per_Capita", figsize=(10,7),c="Population_Density",cmap=plt.get_cmap("jet"),colorbar=True)
#print(covid19.median().values)
plt.legend()
plt.show() #This is required in PyCharm - not shown in book
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 216 entries, 0 to 215
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Continent                   214 non-null    object
 1   Country                     216 non-null    object
 2   Last_Updated_Date           216 non-null    object
 3   Total_Cases                 195 non-null    float64
 4   Total_Deaths                188 non-null    float64
 5   Total_Cases_Per_Million     195 non-null    float64
 6   Total_Deaths_Per_Million    188 non-null    float64
 7   Total_tests                 104 non-null    float64
 8   Positive_rate               105 non-null    float64
 9   Total_Vaccinations          199 non-null    float64
 10  People_Vaccinated           194 non-null    float64
 11  People_Fully_Vaccinated     196 non-null    float64
 12  Population                  216 non-null    int64
 13  Population_Density          204 non-null    float64
 14  Median_Age                  189 non-null    float64
 15  Aged_65_Older               187 non-null    float64
 16  Aged_70_Older               188 non-null    float64
 17  GDP_Per_Capita              191 non-null    float64
dtypes: float64(14), int64(1), object(3)
memory usage: 30.5+ KB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 216 entries, 0 to 215
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Continent                   214 non-null    object
 1   Country                     216 non-null    object
 2   Last_Updated_Date           216 non-null    object
 3   Total_Cases                 195 non-null    float64
 4   Total_Deaths                188 non-null    float64
 5   Total_Cases_Per_Million     195 non-null    float64
 6   Total_Deaths_Per_Million    188 non-null    float64
 7   Total_tests                 104 non-null    float64
 8   Positive_rate               105 non-null    float64
 9   Total_Vaccinations          199 non-null    float64
 10  People_Vaccinated           194 non-null    float64
 11  People_Fully_Vaccinated     196 non-null    float64
 12  Population                  216 non-null    int64
 13  Population_Density          204 non-null    float64
 14  Median_Age                  189 non-null    float64
 15  Aged_65_Older               187 non-null    float64
 16  Aged_70_Older               188 non-null    float64
 17  GDP_Per_Capita              191 non-null    float64
dtypes: float64(14), int64(1), object(3)
memory usage: 30.5+ KB
None
```



In [ ]: