Gurjus Singh

MSDS 422 – Practical Machine Learning

September 20th, 2020

COVID EDA ASSIGNMENT 1

Data preparation & Data Exploration

In order to find out what is in the data, it is important to first explore the data before preparing it for visualization. In order to understand the columns, I first viewed the head of the data. I saw several data types mainly :

date which is an object data type

day, month, year, cases, deaths which are all integers

countriesandTerritories, geold, countryterritoryCode and continentExp which are all objects

populationData and Cumulative number which are both floating numbers

The most important columns in the dataset were cases, deaths, population, and cumulative_number per 100,000 as they were important in understanding each county's COVID-19 performance and what policies they had in place for COVID prevention. Cumulative_number is the cases added up for 14 days divided by country's population multiplied by 100,000.

After viewing the data frame using .head() in the COVID dataset, I decided to use the describe method as this is used to find out statistics. I noticed that some of the data can be statistically analyzed, but cases, deaths are the most important and only ones to do statistical operations on.

In the data frame, I noticed there were some negative cases, deaths, and cumulative numbers so I decided to turn them into positive cases as they could have been mistakenly entered as negatives.

After data cleaning, I retook a look at the dataset, and saw that the numbers were how I wanted them to be, which is to make them all positive. From having an initial look at the data after data cleaning, I noticed the max number of deaths was a whopping 4,928 cases in one day while the maximum number of deaths was 97,824! Average number of deaths was at 21, and average number of cases per day was around 690.

Data visualization

After data prep and look at initial statistics I decided to see the distribution of cases, deaths using a histogram for the whole world. After examining the histogram, I saw that the majority of cases are between 0-20,000 while the majority of deaths was between 0-1000. The data for both histograms is right skewed. Next I wanted to put together a bar plot of top five by cases and deaths. I hypothesized that the United States would be in the top five, and I correctly guessed it as it was in the top five. But the other I didn't know about, for example in number of cases the other four top countries were India, Brazil, Russia and Peru, while in deaths the other four top countries were India, Brazil, Mexico and UK. Although this does not show a great comparison, it shows what countries contribute to the Worldwide COVID cases the most. Next I wanted to see if I could find out in what month did the most cases occur, so I decide to make pie chart. The most cases occurred in August 2020 while the most deaths occurred in April 2020.

Data Scaling and Comparisons

Since both variables are highly skewed, I think we could have scaled both deaths and cases. What I noticed afterwards is that the histogram was still mostly skewed, so minmax and standard scaling did not have much effect on normalizing the data.

Since the population is not the same for most countries, we need to transform the data to compare how good each country is doing. In the data set, they already accounted by transforming

to 100,000 cases, so I decided to feature create my own variable by finding out the cumulative 14 days for 100,000 deaths. After doing this feature creation I found that USA had almost 4 people out of 100,000 dying, while both Canada and China reported 0.5 people dying on average using a 14-day rolling average in deaths. This shows that China and Canada were quick to respond to the Covid-19 outbreak. I also compared China to Canada and found that Canada was higher while China's COVID deaths were flat. This suggested that the early lockdown measures as noticed in January through March in China seemed to work or they are under reporting their numbers. It is unclear which one it is, but with CDC recommending a lockdown for the U.S. it is likely the lockdown that helped.

Insights from analysis

This was a really interesting Assignment surrounding COVID-19 and I found it insightful comparing countries. I did not know that China's cases were close to 0 according to the scaling part of the assignment and I thought their cases were going up. I also thought since I had not heard much about Canada's statistics on COVID that they had none either, but they are also having some deaths like the U.S. , but not as high. It was also astonishing that Peru and Mexico made the list as well.

I did see India was in my Top Five Bar Plot, which confirms their news recently that their numbers have been rising. I also did not know Brazil was in the top five in cases as I had not heard about them either. Going further I hope to see how the U.S. is getting 400k deaths by January using their prediction models. I think we will learn what type of model to use for COVID during this class. I would think it would involve multiple linear regression in my opinion.

Overall, this was a great assignment to do, and I can now say I learned about COVID and applied my Data Science skills to assess a real-world problem. The management and analytic questions surrounding the COVID-19 problem is what are the best ways to prevent the virus, what actions are country's taking and which countries are doing the best to prevent the virus? With this assignment I have felt I could only answer the third question. The other two would have to be answered by researching the country's and their policies on the Google.

Appendix

Data preparation & Data Exploration

#import matplotlib since that is what I used for the visualizations, import pandas for dataframe, import scalers to normalize data; I then loaded the CSV file from the link provided in the assignment.

```
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
```

```
In [160]: #Import Statements
          import pandas as pd
          import matplotlib.pyplot as plt

In [161]: #Open CSV file
          CovidDF = pd.read_csv('/Users/gurjy/Downloads/download')
```

#Initial look at head; I already saw a feature creation variable; decided to make a cumulative number for deaths since they already had cases.

```
[162]: #see head of dataframe
       CovidDF.head()
```

Out[162]:

dateRep	day	month	year	cases	deaths	countriesAndTerritories	geoId	countryterritoryCode	popData2019	continentExp	Cumulative_number_for_14_days_of_C 19_cases_per_1
09/2020	17	9	2020	17	0	Afghanistan	AF	AFG	38041757.0	Asia	1.6
09/2020	16	9	2020	40	10	Afghanistan	AF	AFG	38041757.0	Asia	1.7
09/2020	15	9	2020	99	6	Afghanistan	AF	AFG	38041757.0	Asia	1.6
09/2020	14	9	2020	75	0	Afghanistan	AF	AFG	38041757.0	Asia	1.4
09/2020	13	9	2020	35	0	Afghanistan	AF	AFG	38041757.0	Asia	1.3

#look at the types of data; get an understanding what each one is

```
In [163]: #evaluate types
          CovidDF.dtypes

Out[163]: dateRep                                                  object
          day                                                       int64
          month                                                     int64
          year                                                      int64
          cases                                                     int64
          deaths                                                    int64
          countriesAndTerritories                                  object
          geoId                                                    object
          countryterritoryCode                                     object
          popData2019                                             float64
          continentExp                                             object
          Cumulative_number_for_14_days_of_COVID-19_cases_per_100000    float64
          dtype: object
```

#look at data set; I see minimums in the negatives

```
In [164]: #see statistics in data set
          CovidDF.describe()
```

Out[164]:

	day	month	year	cases	deaths	popData2019	Cumulative_number_for_14_days_of_COVID-19_cases_per_100000
count	43300.000000	43300.000000	43300.000000	43300.000000	43300.000000	4.323600e+04	40519.000000
mean	15.619423	5.586351	2019.998453	690.588614	21.738822	4.293025e+07	32.678808
std	8.814361	2.191485	0.039306	4303.681856	126.497018	1.580210e+08	75.480389
min	1.000000	1.000000	2019.000000	-8261.000000	-1918.000000	8.150000e+02	-147.419587
25%	8.000000	4.000000	2020.000000	0.000000	0.000000	1.355982e+06	0.361227
50%	15.000000	6.000000	2020.000000	9.000000	0.000000	8.519373e+06	4.521082
75%	23.000000	7.000000	2020.000000	147.000000	3.000000	2.916192e+07	26.216640
max	31.000000	12.000000	2020.000000	97894.000000	4928.000000	1.433784e+09	1058.225943

#Made all negative numbers positive using abs function

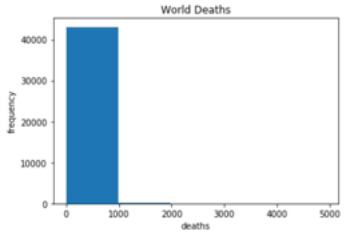
```
165]: #convert all numbers in cases, deaths, cumulative_number to positives
      CovidDF['cases'] = CovidDF['cases'] .abs()
      CovidDF['deaths'] = CovidDF['deaths'].abs()
      CovidDF['Cumulative_number_for_14_days_of_COVID-19_cases_per_100000'] = CovidDF['Cumulative_number_for_14_days_of_COVII
```

```
= CovidDF['Cumulative_number_for_14_days_of_COVID-19_cases_per_100000'].abs()
```

Data visualization

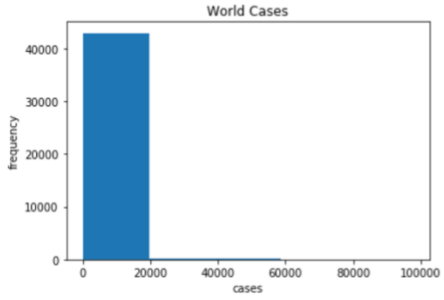
#Highly skewed data sets of cases and deaths world wide

```
In [167]: plt.hist(CovidDF['deaths'], bins = 5)
          plt.xlabel('deaths')
          plt.ylabel('frequency')
          plt.title('World Deaths')

Out[167]: Text(0.5, 1.0, 'World Deaths')
```



In [168]:
```
plt.hist(CovidDF['cases'], bins = 5)
plt.xlabel('cases')
plt.ylabel('frequency')
plt.title('World Cases')
```

Out[168]: Text(0.5, 1.0, 'World Cases')



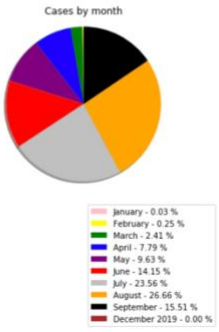
#Dataframe by month

In [169]:
```
CovidDF.groupby(['month']).sum()
```

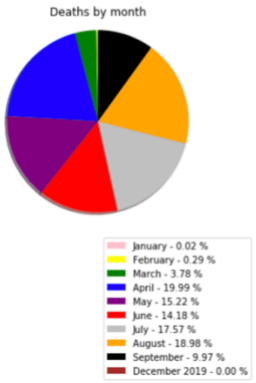
Out[169]:

month	day	year	cases	deaths	popData2019	Cumulative_number_for_14_days_of_COVID-19_cases_per_100000
1	33232	4195540	9797	213	1.798195e+11	9.494658
2	29145	3924860	75412	2708	1.682183e+11	150.000095
3	76501	7766900	722398	35753	1.993784e+11	38167.319418
4	95353	12372500	2332148	189137	2.293312e+11	167948.344128
5	103558	13057280	2883815	144030	2.377151e+11	138220.379320
6	97167	12663380	4234818	134161	2.300480e+11	179096.776684
7	103664	13087580	7053372	166234	2.377457e+11	248553.435609
8	103664	13087580	7980201	179537	2.377457e+11	343420.712947
9	31960	7175040	4643221	94338	1.303297e+11	212409.139239
12	2077	135273	27	0	5.800630e+09	0.000000

In [219]:
```
x =  ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'December 2019']
sizes = CovidDF.groupby(['month']).sum()['cases']
percent = 100.*sizes/sizes.sum()
colors = ['pink', 'yellow', 'green', 'blue', 'purple', 'red', 'silver', 'orange', 'black', 'brown']
explode = (0, 0, 0, 0, 0, 0,0,0,0,0)  # explode 1st slice
labels = ['{0} - {1:1.2f} %'.format(i,j) for i,j in zip(x, percent)]
# Plot
plt.title("Cases by month")
patches, texts = plt.pie(sizes, colors=colors, shadow=True, startangle=90)
plt.legend(patches, labels, loc="lower left", bbox_to_anchor=(0.5,-0.80))
plt.axis('equal')
plt.show()
```



```
In [220]: x =  ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'December 2019']
sizes = CovidDF.groupby(['month']).sum()['deaths']
percent = 100.*sizes/sizes.sum()
colors = ['pink', 'yellow', 'green', 'blue', 'purple', 'red', 'silver', 'orange', 'black', 'brown']
explode = (0, 0, 0, 0, 0, 0,0,0,0,0)  # explode 1st slice
labels = ['{0} - {1:1.2f} %'.format(i,j) for i,j in zip(x, percent)]
# Plot
plt.title("Deaths by month")
patches, texts = plt.pie(sizes, colors=colors, shadow=True, startangle=90)
plt.legend(patches, labels, loc="lower left", bbox_to_anchor=(0.5,-0.80))
plt.axis('equal')
plt.show()
```



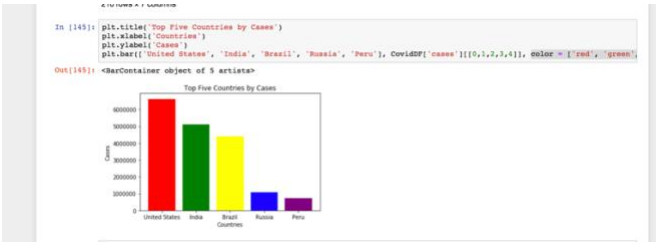
#Rank Countries by cases

```
In [142]: CovidDF = CovidDF.groupby(['countriesAndTerritories']).sum().sort_values(['cases'], ascending = False)
CovidDF
```

Out[142]:

countriesAndTerritories	day	month	year	cases	deaths	popData2019	Cumulative_number_for_14_days_of_COVID-19_cases_per_100000
United_States_of_America	4029	1267	529239	6630891	196802	8.621501e+10	27199.331614
India	4020	1264	527219	5118253	83198	3.566350e+11	4631.584427
Brazil	4029	1267	529239	4419083	134106	5.529497e+10	28058.085742
Russia	4029	1267	529239	1079519	18917	3.821853e+10	10017.866317
Peru	3038	1145	391880	744400	31051	6.307030e+09	30469.822299
...	...	...	...	...	...	...	...
Greenland	2877	1109	367640	14	0	1.031212e+07	211.789622
Montserrat	2856	1102	363600	13	1	8.983800e+05	2825.085153
Falkland_Islands_(Malvinas)	2565	1061	337340	13	0	5.631240e+05	3529.062871
Holy_See	2954	1124	377740	12	0	1.524050e+05	18527.607362
Anguilla	2716	1088	353500	3	0	2.602600e+06	67.240452

210 rows × 7 columns



#Rank Countries by Deaths

United States India Brazil Russia Peru
Countries

```
In [146]: CovidDF = CovidDF.groupby(['countriesAndTerritories']).sum().sort_values(['deaths'], ascending = False)
          CovidDF
```

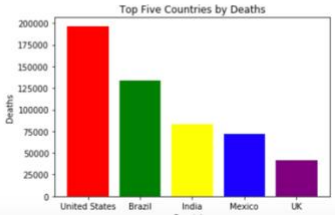
Out[146]:

countriesAndTerritories	day	month	year	cases	deaths	popData2019	Cumulative_number_for_14_days_of_COVID-19_cases_per_100000
United_States_of_America	4029	1267	529239	6630891	196802	8.621501e+10	27199.331614
Brazil	4029	1267	529239	4419083	134106	5.529497e+10	28058.085742
India	4020	1264	527219	5118253	83198	3.566350e+11	4631.584427
Mexico	3975	1243	513079	680931	71978	3.240418e+10	7135.728201
United_Kingdom	4029	1267	529239	378219	41684	1.746154e+10	7515.737516
...	...	...	...	...	...	...	...
Greenland	2877	1109	367640	14	0	1.031212e+07	211.789622
Grenada	2814	1100	361580	24	0	2.004836e+07	223.210300
Bhutan	2965	1127	379760	246	0	1.434617e+08	432.450000
Saint_Lucia	2962	1124	377740	27	0	3.418266e+07	188.188955
Saint_Kitts_and_Nevis	2742	1091	355520	17	0	9.298784e+06	287.693531

210 rows × 7 columns

```
In [147]: plt.title('Top Five Countries by Deaths')
          plt.xlabel('Countries')
          plt.ylabel('Deaths')
          plt.bar(['United States', 'Brazil', 'India', 'Mexico', 'UK'], CovidDF['deaths'][[0,1,2,3,4]], color = ['red', 'green',
```

Out[147]: <BarContainer object of 5 artists>

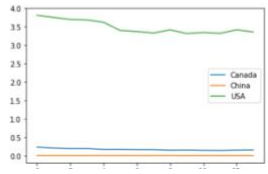


#Comparing Using Feature Transformation

Accomplished this by using .rolling on dataframe and comparing three countries China, Canada, and USA

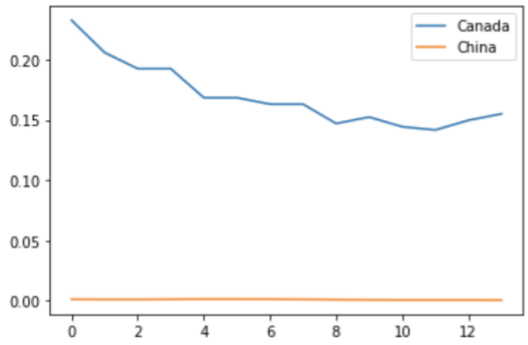
```
In [400]: CovidDF = CovidDF.dropna()

          CovidDFusa = CovidDF[CovidDF['countriesAndTerritories'] == 'United_States_of_America']
          CovidDFCanada = CovidDF[CovidDF['countriesAndTerritories'] == 'Canada']
          CovidDFChina = CovidDF[CovidDF['countriesAndTerritories'] == 'China']
          rollingaverageCANADA = CovidDFCanada['deaths'].rolling(14).sum()/CovidDFCanada['popData2019'][7236]*100000
          fourteencumulativedaydeathsCANADA = [rollingaverageCANADA[x] for x in range(7249, 7263)]

          rollingaverageUSA = CovidDFusa['deaths'].rolling(14).sum()/CovidDFusa['popData2019'][41374]*100000
          fourteencumulativedaydeathsUSA = [rollingaverageUSA[x] for x in range(41387, 41401)]

          rollingaverageCHINA= CovidDFChina['deaths'].rolling(14).sum()/CovidDFChina['popData2019'][8490]*100000
          fourteencumulativedaydeathsCHINA = [rollingaverageCHINA[x] for x in range(8503, 8517)]


          plt.plot(range(0, 14), fourteencumulativedaydeathsCANADA[::-1])

          plt.plot(range(0, 14), fourteencumulativedaydeathsCHINA[::-1])

          plt.plot(range(0, 14), fourteencumulativedaydeathsUSA[::-1])

          plt.legend(['Canada', 'China', 'USA'])
```

Out[400]: <matplotlib.legend.Legend at 0x126a99390>



Compared China and Canada ; Canada has higher number of deaths in September

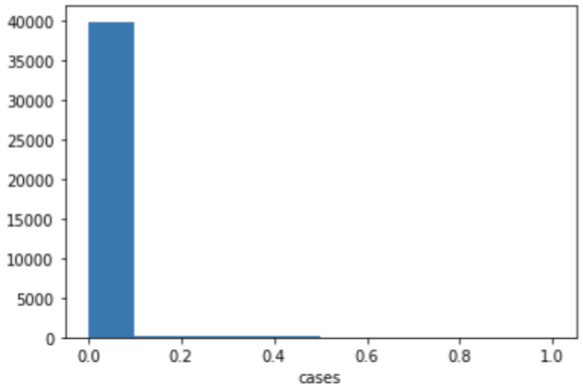
```
In [402]: plt.plot(range(0, 14), fourteencumulativedaydeathsCANADA[::-1])
          plt.plot(range(0, 14), fourteencumulativedaydeathsCHINA[::-1])
          plt.legend(['Canada', 'China'])
Out[402]: <matplotlib.legend.Legend at 0x1270a84d0>
```



Data Scaling and Comparisons

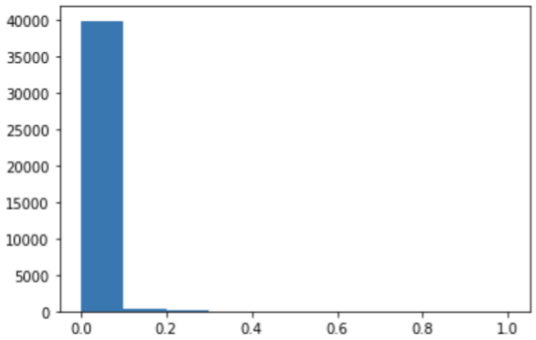
#Scaled the data using both minmax and standard scaling, but did not see much of a difference. The Data was till right skewed afterwards.

```
In [98]: df['cases'] = MinMaxScaler().fit_transform(df[['cases']])
         plt.hist(df['cases'])
         plt.xlabel('cases')
Out[98]: Text(0.5, 0, 'cases')
```



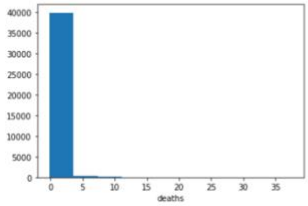
In [97]:
```
df = pd.DataFrame(CovidDF['deaths'])
df['cases'] = CovidDF['cases']
df['deaths'] = MinMaxScaler().fit_transform(df[['deaths']])
plt.hist(df['deaths'])
```

Out[97]:
```
(array([3.9802e+04, 3.3300e+02, 1.6400e+02, 2.5000e+01, 1.3000e+01,
        2.0000e+00, 1.0000e+00, 4.0000e+00, 0.0000e+00, 1.0000e+00]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <a list of 10 Patch objects>)
```



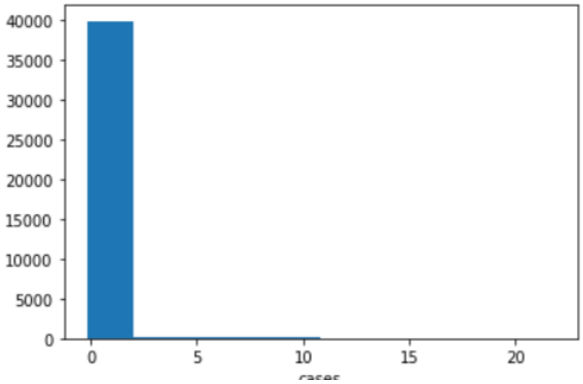
In [102]:
```
testdf['deaths'] = StandardScaler().fit_transform(testdf[['deaths']])
plt.hist(testdf['deaths'])
plt.xlabel('deaths')
```

Out[102]:
```
Text(0.5, 0, 'deaths')
```



In [101]:
```
testdf = pd.DataFrame(CovidDF['deaths'])
testdf['cases'] = CovidDF['cases']
testdf['cases'] = StandardScaler().fit_transform(testdf[['cases']])
plt.hist(testdf['cases'])
plt.xlabel('cases')
```

Out[101]:
```
Text(0.5, 0, 'cases')
```



References

https://www.ecdc.europa.eu/en/publications-data/download-todays-data-geographic-distribution-covid-19-cases-worldwide

https://www.npr.org/sections/coronavirus-live-updates/2020/09/17/913475045/india-is-on-track-to-surpass-u-s-as-the-country-worst-affected-by-covid-19

https://www.mathworks.com/help/matlab/ref/legend.html#:~:text=Plot%20two%20lines%20and%20add,arguments%20to%20the%20legend%20function.&text=If%20you%20add%20or%20delete,name%2Dvalue%20pair%20during%20creation.

https://stackoverflow.com/questions/23577505/how-to-avoid-overlapping-of-labels-autopct-in-a-matplotlib-pie-chart

https://matplotlib.org/tutorials/introductory/pyplot.html

https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html#sphx-glr-auto-examples-preprocessing-plot-all-scaling-py