Data Preparation and Management Questions

Covid 19 is an ongoing threat to our current world as we know it. In order to truly understand the impact of Covid 19, we ask the questions of how many cases of Covid 19 spread on a given date, how many deaths has it caused and lastly, and lastly what countries has Covid 19 impacted the most and by what percent. In order to conduct a thorough exploratory analysis we must for prep the data to see if the data set. I do this in lines 6-8 in the appendix python code. Here, I made sure that every variable had the same number of values and found that there were missing values for geoID, countryTerritory, popData2019 and continentExp. I then created a new dataframe to exclude NA values and made sure the data fell in the 2019-2020 range and displayed the table to show that the values of every variable were 25767. Now that the data prep has created a viable table, I moved onto exploration to create insights.

Data Exploration

For my exploration, I chose to analyze the impact of Covid on the country Afghanistan. I wanted to specifically explore the number of total cases per day (per millions) and the number of deaths per day (per millions) for a date range of April-June. I did this in lines 47-59 in the appendix output code. I analyzed the number of cases and deaths in the values per millions due to the volume of the numbers. I wanted the graphs to have readable trend values so by dividing the values per million, I was able to produce a readable that shows that cases and deaths increased over the month of June.

Data Scaling and Comparisons

Utilizing min max scalers, I was able to show the number of cases on a scaled basis as well as the number of deaths on a scaled basis as well. The purpose of the min max scaler is to transform the features of death and cases by scaling them to a given range. Instead of now using a zero mean value, we now have unit variance scaling which is more accurate for modeling. This is done in lines 66-70 in the python appendix code.

Insights and Analysis

After data prepping, exploration and analysis, I finally got to develop insights for analyzing the impact of Covid on Afghanistan. The first insight I made was on line 22 where I found that Afghanistan accounts for 2.23% of deaths for total deaths worldwide. This is significant because we can now understand how the number of cases also compares worldwide. When diving further,I also found that while there was the largest spike in case reportings on 5/17/2020 (around 1000 cases) , the number of deaths spiked as well (around 30) but was not the largest spike of deaths which occurred on 6/10/2020 (40 deaths). This is shown from lines 54-56. This showed that while the number of cases are somewhat correlated to the deaths, they are not correlated on the same date range. This could be due to the 14 day period where cases could be asymptotic or not, meaning that deaths could be delayed from the time a case is reported. I further looked into the relationship of deaths vs cases in line 62 where I created a bivariate distribution of cases vs deaths. We see that there is not a large scattering pattern with a large cluster of values near the 0-10 range for deaths and 0-200 range for reportings. The pattern looks positively linear meaning that there is a definite correlation between cases and deaths, however we must account for the delay in deaths due to the 14 day period. Lastly, in lines 61-63, I created a univariate plot to show the distribution of cases and deaths in Afghanistan. Here I found that both deaths cases spiked initially per day with around 8000 cases per day and 120 deaths, but death values plummeted to around 20 deaths while cases dropped and created a normal distribution with a peak of around 3000 cases per day. Overall, this exploratory data analysis helped me analyze the rate of covid cases, deaths and spread in Afghanistan and the world.

Python Appendix

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt import
IPython
from IPython.display import display
import sys import
sklearn
```

In [2]:

```
from sklearn.preprocessing import MinMaxScaler from
sklearn.preprocessing import minmax_scale from
sklearn.preprocessing import MaxAbsScaler from
sklearn.preprocessing import StandardScaler from
sklearn.preprocessing import RobustScaler from
sklearn.preprocessing import Normalizer
from sklearn.preprocessing import QuantileTransformer
```

In [4]:

```
corona_df = pd.read_csv("covid.csv")
```

In [5]:

```
corona_df.head()
```

Out[5]:

	dateRep	day	month	year	cases	deaths	countriesAndTerritories	geoId	countryterritoryC
0	26/06/2020	26	6	2020	460	36	Afghanistan	AF	
1	25/06/2020	25	6	2020	234	21	Afghanistan	AF	
2	24/06/2020	24	6	2020	338	20	Afghanistan	AF	
3	23/06/2020	23	6	2020	310	17	Afghanistan	AF	
4	22/06/2020	22	6	2020	409	12	Afghanistan	AF	

In [6]:

```
corona_df.count()
```

Out[6]:

```
dateRep                    25935
day                        25935
month                      25935
year                       25935
cases                      25935
deaths                     25935
countriesAndTerritories    25935
geoId                      25831
countryterritoryCode       25871
popData2019                25871
continentExp               25935
dtype: int64
```

In [ ]:

In [7]:

```
new_corona=corona_df.dropna()
new_corona.count()
```

Out[7]:

```
dateRep                    25767
day                        25767
month                      25767
year                       25767
cases                      25767
deaths                     25767
countriesAndTerritories    25767
geoId                      25767
countryterritoryCode       25767
popData2019                25767
continentExp               25767
dtype: int64
```

In [ ]:

In [8]:

```
new_corona[(new_corona.day >= 1) & (new_corona.day <= 31) & (new_corona.month >= 1) & (new_corona.month <= 12)
& (new_corona.year >= 2019) & (new_corona.year <= 2020) & (new_corona.cases >= 0) & (new_corona.deaths >= 0)]
```

Out[8]:

	dateRep	day	month	year	cases	deaths	countriesAndTerritories	geoId	countryterrit
0	26/06/2020	26	6	2020	460	36	Afghanistan	AF	
1	25/06/2020	25	6	2020	234	21	Afghanistan	AF	
2	24/06/2020	24	6	2020	338	20	Afghanistan	AF	
3	23/06/2020	23	6	2020	310	17	Afghanistan	AF	
4	22/06/2020	22	6	2020	409	12	Afghanistan	AF	
5	21/06/2020	21	6	2020	546	21	Afghanistan	AF	
6	20/06/2020	20	6	2020	346	2	Afghanistan	AF	
7	19/06/2020	19	6	2020	658	42	Afghanistan	AF	
8	18/06/2020	18	6	2020	564	13	Afghanistan	AF	
9	17/06/2020	17	6	2020	783	13	Afghanistan	AF	
10	16/06/2020	16	6	2020	761	7	Afghanistan	AF	
11	15/06/2020	15	6	2020	664	20	Afghanistan	AF	
12	14/06/2020	14	6	2020	556	5	Afghanistan	AF	
13	13/06/2020	13	6	2020	656	20	Afghanistan	AF	
14	12/06/2020	12	6	2020	747	21	Afghanistan	AF	
15	11/06/2020	11	6	2020	684	21	Afghanistan	AF	
16	10/06/2020	10	6	2020	542	15	Afghanistan	AF	
17	09/06/2020	9	6	2020	575	12	Afghanistan	AF	
18	08/06/2020	8	6	2020	791	30	Afghanistan	AF	
19	07/06/2020	7	6	2020	582	18	Afghanistan	AF	
20	06/06/2020	6	6	2020	915	9	Afghanistan	AF	
21	05/06/2020	5	6	2020	787	6	Afghanistan	AF	
22	04/06/2020	4	6	2020	758	24	Afghanistan	AF	
23	03/06/2020	3	6	2020	759	5	Afghanistan	AF	
24	02/06/2020	2	6	2020	545	8	Afghanistan	AF	
25	01/06/2020	1	6	2020	680	8	Afghanistan	AF	
26	31/05/2020	31	5	2020	866	3	Afghanistan	AF	
27	30/05/2020	30	5	2020	623	11	Afghanistan	AF	
28	29/05/2020	29	5	2020	580	8	Afghanistan	AF	
29	28/05/2020	28	5	2020	625	7	Afghanistan	AF	
...	...	...	...	...	...	...	...	...	
25905	19/04/2020	19	4	2020	1	0	Zimbabwe	ZW	
25906	18/04/2020	18	4	2020	0	0	Zimbabwe	ZW	

	dateRep	day	month	year	cases	deaths	countriesAndTerritories	geoId	countryterrit
25907	17/04/2020	17	4	2020	1	0	Zimbabwe	ZW	
25908	16/04/2020	16	4	2020	6	0	Zimbabwe	ZW	
25909	15/04/2020	15	4	2020	0	0	Zimbabwe	ZW	
25910	14/04/2020	14	4	2020	3	0	Zimbabwe	ZW	
25911	13/04/2020	13	4	2020	0	0	Zimbabwe	ZW	
25912	12/04/2020	12	4	2020	3	0	Zimbabwe	ZW	
25913	11/04/2020	11	4	2020	0	0	Zimbabwe	ZW	
25914	10/04/2020	10	4	2020	0	1	Zimbabwe	ZW	
25915	09/04/2020	9	4	2020	1	1	Zimbabwe	ZW	
25916	08/04/2020	8	4	2020	1	0	Zimbabwe	ZW	
25917	07/04/2020	7	4	2020	0	0	Zimbabwe	ZW	
25918	06/04/2020	6	4	2020	0	0	Zimbabwe	ZW	
25919	05/04/2020	5	4	2020	0	0	Zimbabwe	ZW	
25920	04/04/2020	4	4	2020	1	0	Zimbabwe	ZW	
25921	03/04/2020	3	4	2020	0	0	Zimbabwe	ZW	
25922	02/04/2020	2	4	2020	0	0	Zimbabwe	ZW	
25923	01/04/2020	1	4	2020	1	0	Zimbabwe	ZW	
25924	31/03/2020	31	3	2020	0	0	Zimbabwe	ZW	
25925	30/03/2020	30	3	2020	0	0	Zimbabwe	ZW	
25926	29/03/2020	29	3	2020	2	0	Zimbabwe	ZW	
25927	28/03/2020	28	3	2020	2	0	Zimbabwe	ZW	
25928	27/03/2020	27	3	2020	0	0	Zimbabwe	ZW	
25929	26/03/2020	26	3	2020	1	0	Zimbabwe	ZW	
25930	25/03/2020	25	3	2020	0	0	Zimbabwe	ZW	
25931	24/03/2020	24	3	2020	0	1	Zimbabwe	ZW	
25932	23/03/2020	23	3	2020	0	0	Zimbabwe	ZW	
25933	22/03/2020	22	3	2020	1	0	Zimbabwe	ZW	
25934	21/03/2020	21	3	2020	1	0	Zimbabwe	ZW	

25752 rows × 11 columns

```
In [ ]:
```

In [ ]:

```

```

In [18]:

```
new_corona['pop_in_millions'] = (new_corona['popData2019']/1000000) new_corona['deaths_per_mil'] =
((new_corona['deaths'])/(new_corona['pop_in_million s']))
new_corona['cases_per_mil'] = ((new_corona['cases'])/(new_corona['pop_in_millions'
]))
new_corona.head()
```

```
/Users/avadhani/anaconda3/lib/python3.7/site-packages/ipykernel_launche
r.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-d
ocs/stable/indexing.html#indexing-view-versus-copy
  """Entry point for launching an IPython kernel.
/Users/avadhani/anaconda3/lib/python3.7/site-packages/ipykernel_launche
r.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-d
ocs/stable/indexing.html#indexing-view-versus-copy

/Users/avadhani/anaconda3/lib/python3.7/site-packages/ipykernel_launche
r.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-d
ocs/stable/indexing.html#indexing-view-versus-copy
  This is separate from the ipykernel package so we can avoid doing imp
orts until
```

Out[18]:

	dateRep	day	month	year	cases	deaths	countriesAndTerritories	geoId	countryterritoryC
0	26/06/2020	26	6	2020	460	36	Afghanistan	AF	
1	25/06/2020	25	6	2020	234	21	Afghanistan	AF	
2	24/06/2020	24	6	2020	338	20	Afghanistan	AF	
3	23/06/2020	23	6	2020	310	17	Afghanistan	AF	
4	22/06/2020	22	6	2020	409	12	Afghanistan	AF	

In [19]:

In [21]:

```
100*((new_corona.groupby(new_corona.countriesAndTerritories)['deaths'].sum())/(new_corona.groupby(new_corona.countriesAndTerritories)['cases'].sum()))
```

```
Out[21]:

countriesAndTerritories
Afghanistan                            2.236951
Albania                                2.235401
Algeria                                7.055042
Andorra                                6.081871
Angola                                 5.076142
Anguilla                               0.000000
Antigua_and_Barbuda                    4.615385
Argentina                              2.143239
Armenia                                1.763668
Aruba                                  2.970297
Australia                              1.376025
Austria                                4.004360
Azerbaijan                             1.211958
Bahamas                               10.576923
Bahrain                                0.294838
Bangladesh                             1.280350
Barbados                               7.216495
Belarus                                0.607797
Belgium                               15.942433
Belize                                 8.695652
Benin                                  1.376598
Bermuda                                6.164384
Bhutan                                 0.000000
Bolivia                                3.203172
Bonaire, Saint Eustatius and Saba      0.000000
Bosnia_and_Herzegovina                 4.610116
Botswana                               1.123596
Brazil                                 4.476050
British_Virgin_Islands                12.500000
Brunei_Darussalam                      2.127660
                                        ...
Sri_Lanka                              0.547264
Sudan                                  6.188780
Suriname                               2.680965
Sweden                                 8.185945
Switzerland                            5.362897
Syria                                  2.892562
Taiwan                                 1.565996
Tajikistan                             0.913723
Thailand                               1.836605
Timor_Leste                            0.000000
Togo                                   2.380952
Trinidad_and_Tobago                    6.504065
Tunisia                                4.302926
Turkey                                 2.612951
Turks_and_Caicos_islands               6.666667
Uganda                                 0.000000
Ukraine                                2.666967
United_Arab_Emirates                   0.661469
United_Kingdom                        14.036626
United_Republic_of_Tanzania            4.125737
United_States_Virgin_Islands           7.407407
United_States_of_America               5.136254
```

```
   Uruguay                                2.866593
```

```
Uzbekistan                      0.276702
Venezuela                       0.854701
Vietnam                         0.000000
Western_Sahara                  4.000000
Yemen                          26.765799
Zambia                          1.202405
Zimbabwe                        1.088929
Length: 208, dtype: float64
```

In [ ]:

```

```

In [46]:

```
Afghanistan= new_corona.loc[corona_df['geoId'] == 'AF']
```

In [47]:

```
Afghanistan.describe()
```

Out[47]:

	day	month	year	cases	deaths	popData2019	pop_in_millio
count	169.000000	169.000000	169.000000	169.000000	169.000000	169.0	1.690000e+
mean	15.899408	3.520710	2019.994083	178.550296	3.994083	38041757.0	3.804176e+
std	8.753669	1.851926	0.076923	266.013223	7.338335	0.0	2.850617e-
min	1.000000	1.000000	2019.000000	0.000000	0.000000	38041757.0	3.804176e+
25%	8.000000	2.000000	2020.000000	0.000000	0.000000	38041757.0	3.804176e+
50%	16.000000	4.000000	2020.000000	26.000000	0.000000	38041757.0	3.804176e+
75%	23.000000	5.000000	2020.000000	280.000000	5.000000	38041757.0	3.804176e+
max	31.000000	12.000000	2020.000000	1063.000000	42.000000	38041757.0	3.804176e+

In [48]:

```
Afghanistan = Afghanistan.set_index('dateRep')
```

In [ ]:

```

```

In [50]:

```
Afghanistan.index
```

Out[50]:

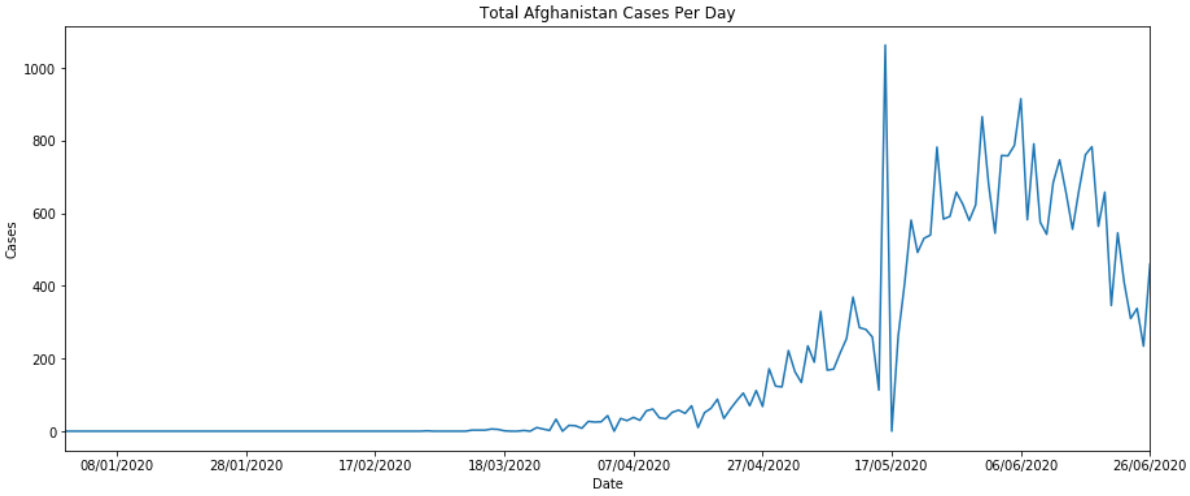
```
Index(['26/06/2020', '25/06/2020', '24/06/2020', '23/06/2020', '22/06/2
020',
       '21/06/2020', '20/06/2020', '19/06/2020', '18/06/2020', '17/06/2
020',
       ...
       '09/01/2020', '08/01/2020', '07/01/2020', '06/01/2020', '05/01/2
020',
       '04/01/2020', '03/01/2020', '02/01/2020', '01/01/2020', '31/12/2
019'],
      dtype='object', name='dateRep', length=169)
```

In [ ]:

```

```

In [54]:

```
y = Afghanistan['cases']
y.plot(figsize=(15, 6))
plt.gca().invert_xaxis()
plt.title('Total Afghanistan Cases Per Day') plt.xlabel('Date')
plt.ylabel('Cases') plt.show()
```

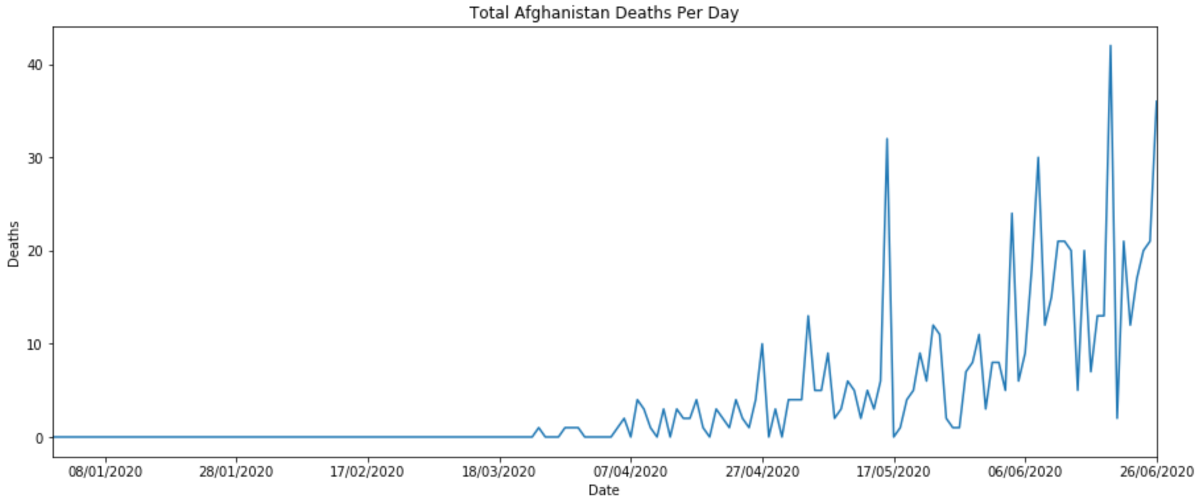


In [55]:

```

```

In [56]:

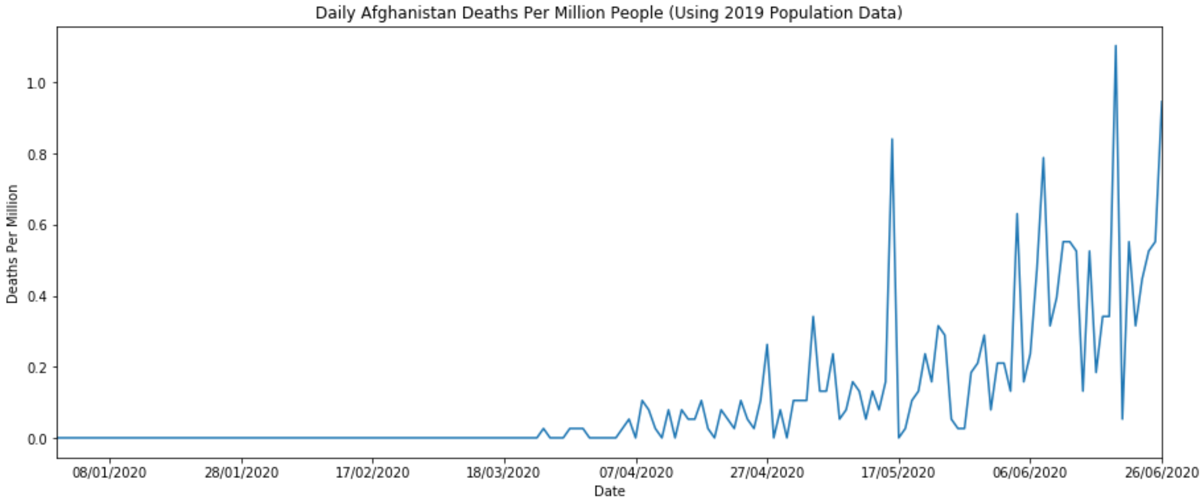
```
z = Afghanistan['deaths']
z.plot(figsize=(15,6))
plt.gca().invert_xaxis()
plt.title('Total Afghanistan Deaths Per Day') plt.xlabel('Date')
plt.ylabel('Deaths') plt.show()
```



In [57]:

In [59]:

```
b = Afghanistan['deaths_per_mil']
b.plot(figsize=(15,6)) plt.gca().invert_xaxis()
plt.title('Daily Afghanistan Deaths Per Million People (Using 2019 Population Dat a)')
plt.xlabel('Date') plt.ylabel('Deaths Per Million')
plt.show()
```

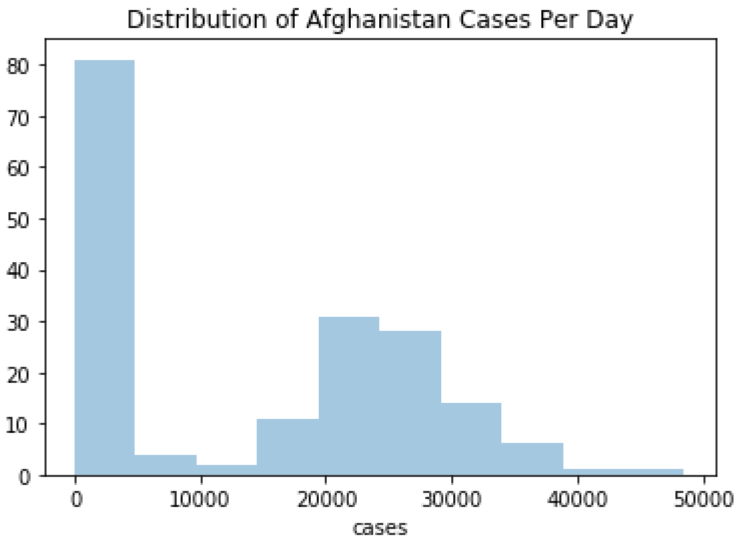


In [ ]:

```

```

In [61]:

```
Afghanistan_cases=USA['cases']
sns.distplot(Afghanistan_cases, bins=10, kde=False).set_title("Distribution of Afgh anistan Cases Per Day")
plt.show()
```



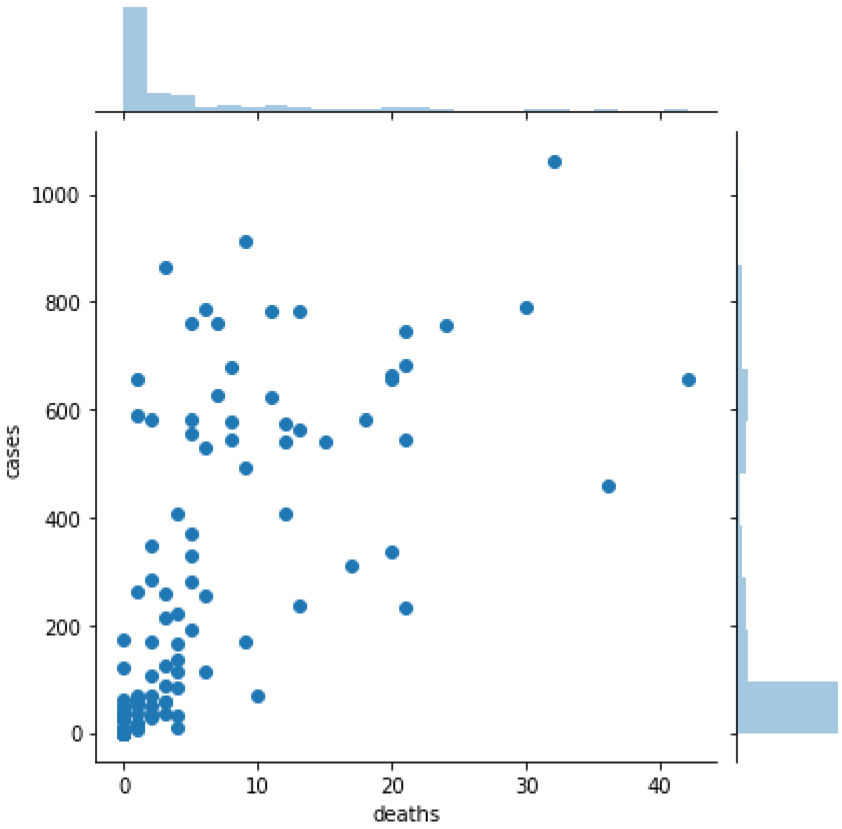
In [ ]:

In [62]:

```
sns.jointplot(data=Afghanistan,x='deaths', y='cases')
```

Out[62]:

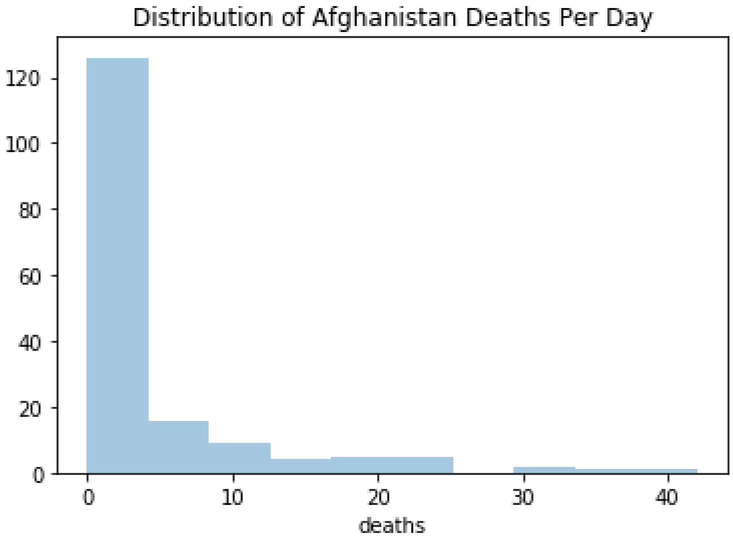
```
<seaborn.axisgrid.JointGrid at 0x1a2cc2ce80>
```



In [ ]:

In [63]:

```
Afghanistan_deaths=Afghanistan['deaths']
sns.distplot(Afghanistan_deaths, bins=10,kde=False).set_title("Distribution of Afgh anistan Deaths Per Day")
plt.show()
```



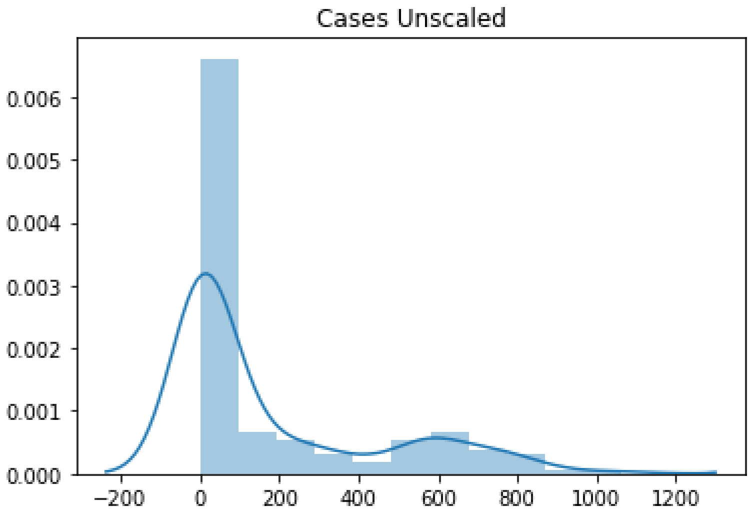
In [ ]:

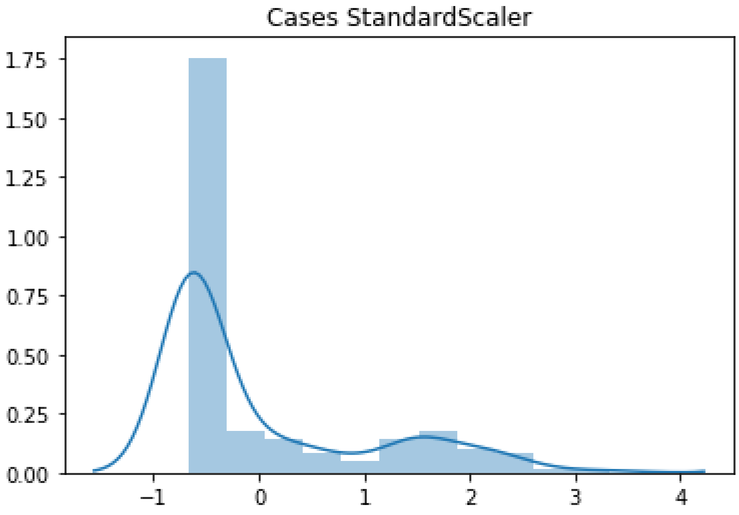
```
```

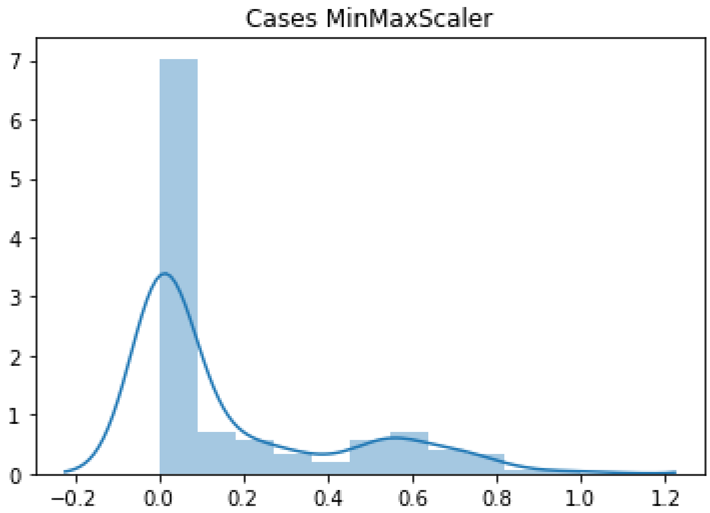
In [66]:

```
scale_cases = pd.DataFrame(Afghanistan['cases'])
scale_cases.describe()
sc = scale_cases

unscaled_fig, ax = plt.subplots()
sns.distplot(sc).set_title('Cases Unscaled')
unscaled_fig.savefig('Transformation-Unscaled' + '.pdf',
    bbox_inches = 'tight', dpi=None, facecolor='w', edgecolor='b',
    orientation='portrait', papertype=None, format=None,
    transparent=True, pad_inches=0.25, frameon=None)

standard_fig, ax = plt.subplots()
sns.distplot(StandardScaler().fit_transform(np.array(sc).reshape(-1,1))).set_title(
'Cases StandardScaler')
standard_fig.savefig('Transformation-StandardScaler' + '.pdf',
    bbox_inches = 'tight', dpi=None, facecolor='w', edgecolor='b',
    orientation='portrait', papertype=None, format=None,
    transparent=True, pad_inches=0.25, frameon=None)

minmax_fig, ax = plt.subplots()
sns.distplot(MinMaxScaler().fit_transform(np.array(sc).reshape(-1,1))).set_title('C
ases MinMaxScaler')
minmax_fig.savefig('Transformation-MinMaxScaler' + '.pdf',
    bbox_inches = 'tight', dpi=None, facecolor='w', edgecolor='b',
    orientation='portrait', papertype=None, format=None,
    transparent=True, pad_inches=0.25, frameon=None)
```

```
/Users/avadhani/anaconda3/lib/python3.7/site-packages/ipykernel_launche
r.py:10: MatplotlibDeprecationWarning:
The frameon kwarg was deprecated in Matplotlib 3.1 and will be removed
in 3.3. Use facecolor instead.
  # Remove the CWD from sys.path while we load stuff.
/Users/avadhani/anaconda3/lib/python3.7/site-packages/ipykernel_launche
r.py:17: MatplotlibDeprecationWarning:
The frameon kwarg was deprecated in Matplotlib 3.1 and will be removed
in 3.3. Use facecolor instead.
/Users/avadhani/anaconda3/lib/python3.7/site-packages/ipykernel_launche
r.py:24: MatplotlibDeprecationWarning:
The frameon kwarg was deprecated in Matplotlib 3.1 and will be removed
in 3.3. Use facecolor instead.
```





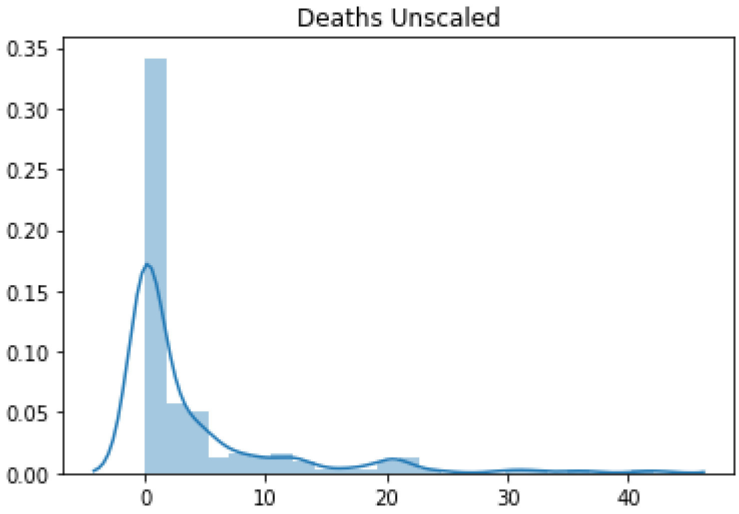


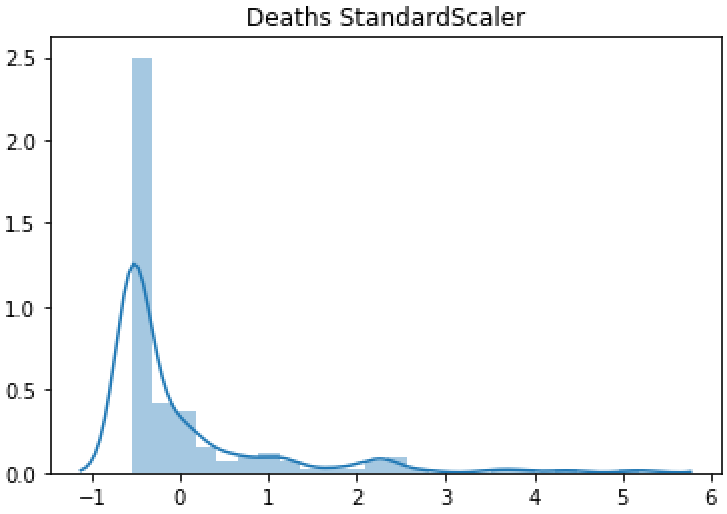
```
In [ ]:
```

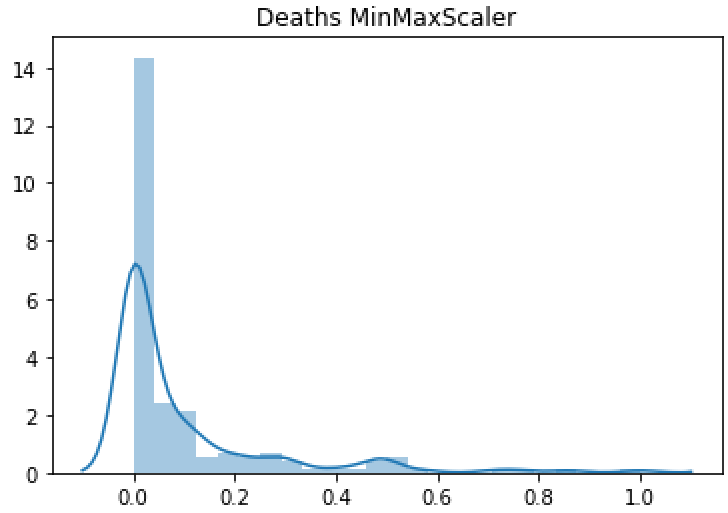
In [70]:

```
scale_deaths = pd.DataFrame(Afghanistan['deaths'])
sd = scale_deaths

unscaled_fig, ax = plt.subplots()
sns.distplot(sd).set_title('Deaths Unscaled')
unscaled_fig.savefig('Transformation-Unscaled' + '.pdf',
    bbox_inches = 'tight', dpi=None, facecolor='w', edgecolor='b',
    orientation='portrait', papertype=None, format=None,
    transparent=True, pad_inches=0.25, frameon=None)

standard_fig, ax = plt.subplots()
sns.distplot(StandardScaler().fit_transform(np.array(sd).reshape(-1,1))).set_title('Deaths StandardScaler')
standard_fig.savefig('Transformation-StandardScaler' + '.pdf',
    bbox_inches = 'tight', dpi=None, facecolor='w', edgecolor='b',
    orientation='portrait', papertype=None, format=None,
    transparent=True, pad_inches=0.25, frameon=None)

minmax_fig, ax = plt.subplots()
sns.distplot(MinMaxScaler().fit_transform(np.array(sd).reshape(-1,1))).set_title('Deaths MinMaxScaler')
minmax_fig.savefig('Transformation-MinMaxScaler' + '.pdf',
    bbox_inches = 'tight', dpi=None, facecolor='w', edgecolor='b',
    orientation='portrait', papertype=None, format=None)
```

```
/Users/avadhani/anaconda3/lib/python3.7/site-packages/ipykernel_launche
r.py:9: MatplotlibDeprecationWarning:
The frameon kwarg was deprecated in Matplotlib 3.1 and will be removed
in 3.3. Use facecolor instead.
  if __name__ == '__main__':
/Users/avadhani/anaconda3/lib/python3.7/site-packages/ipykernel_launche
r.py:16: MatplotlibDeprecationWarning:
The frameon kwarg was deprecated in Matplotlib 3.1 and will be removed
in 3.3. Use facecolor instead.
  app.launch_new_instance()
```






Deaths MinMaxScaler
14
12
10
8
6
4
2
0
0.0
0.2
0.4
0.6
0.8
1.0