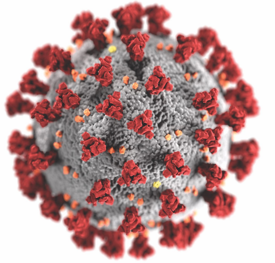
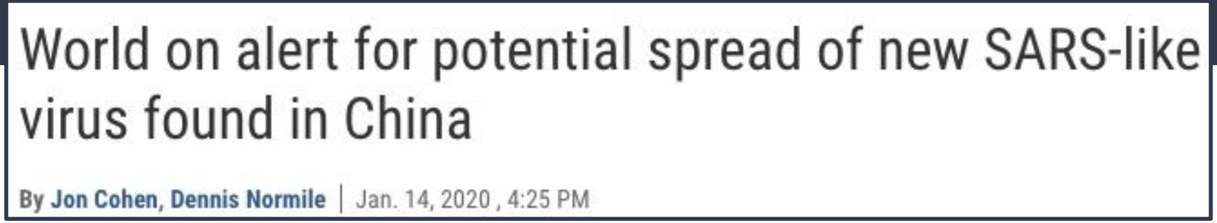
Project One:

Exploring Attributable Factors of COVID-19 Deaths in the U.S.

Group Members: Kholiswa Tsotetsi, Azzy Caceres, Michael Striffler

Background





World on alert for potential spread of new SARS-like virus found in China

By Jon Cohen, Dennis Normile | Jan. 14, 2020 , 4:25 PM

First identified in Wuhan, China late 2019, the novel coronavirus disease ,which is caused by SARS-COV-2 , was declared a global pandemic by WHO early March.

Currently, there are 30. 5 million cases worldwide, with a global death toll of 946,000. The United States represents 6.7 million of these cases with a death toll of close to 200,000

Data: NIAID/John Hopkins

Questions to Answer

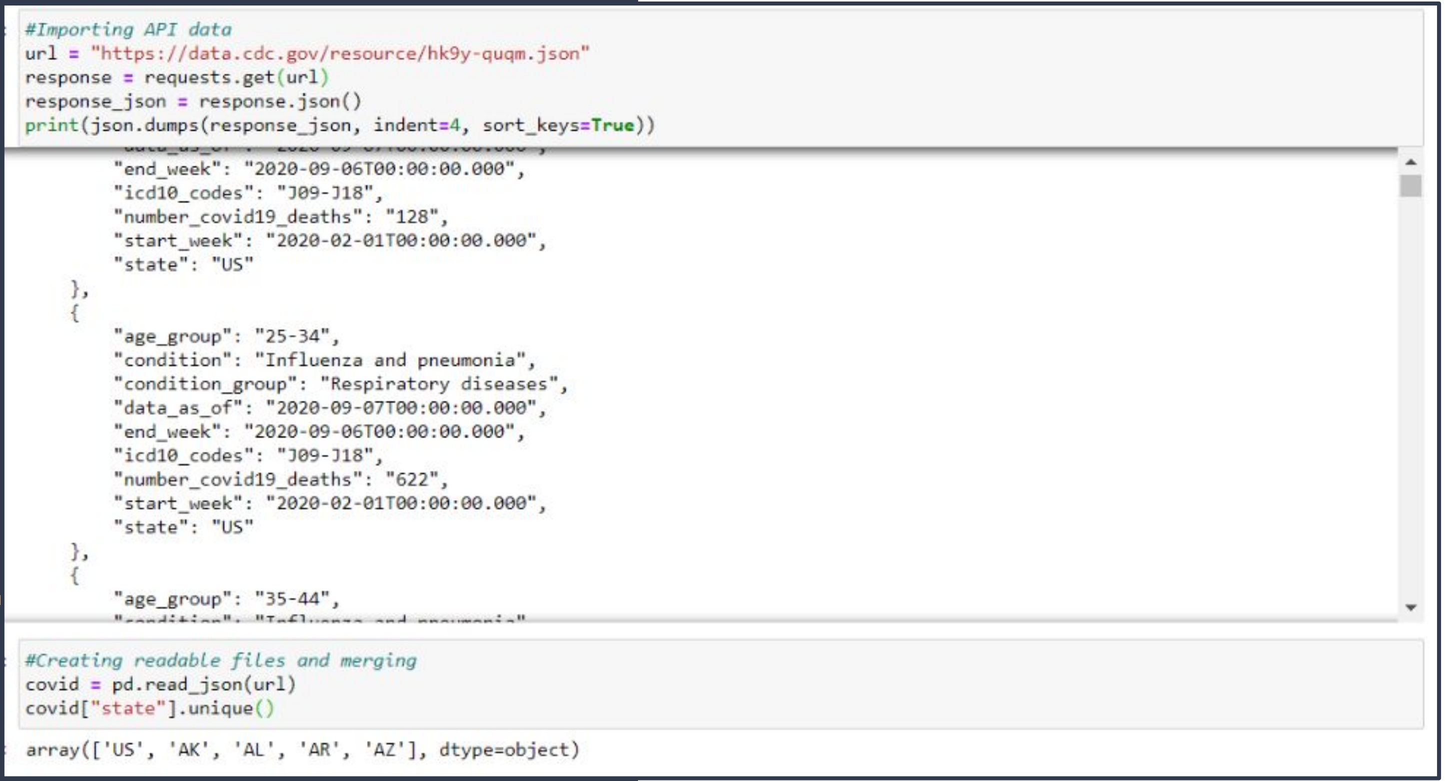
1) What is the spatial distribution of Covid-19 deaths in the United States?

2) Do pre-existing conditions increase the risk of death caused by Covid-19? If so, which pre-existing conditions have resulted in the highest mortality in reported Covid-19 related deaths?

3) Does age correlate with the number of reported Covid-19 deaths?

4) Overall, what is the mortality rate of Covid-19 in the United States?

Importing API file



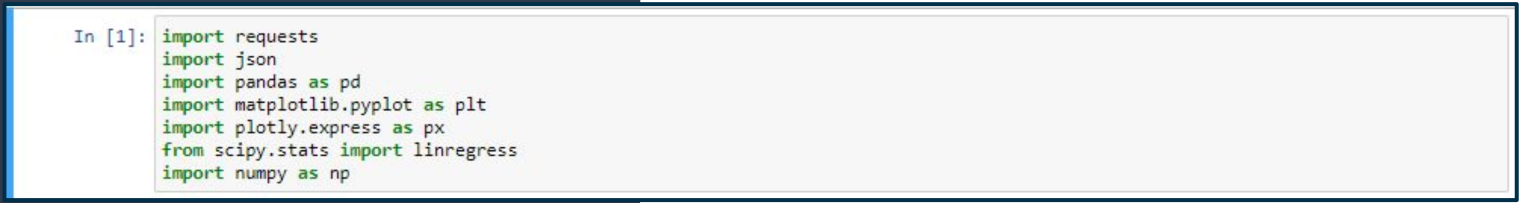
```
#Importing API data
url = "https://data.cdc.gov/resource/hk9y-quqm.json"
response = requests.get(url)
response_json = response.json()
print(json.dumps(response_json, indent=4, sort_keys=True))
```

```
        "data_as_of": "2020-09-07T00:00:00.000",
        "end_week": "2020-09-06T00:00:00.000",
        "icd10_codes": "J09-J18",
        "number_covid19_deaths": "128",
        "start_week": "2020-02-01T00:00:00.000",
        "state": "US"
    },
    {
        "age_group": "25-34",
        "condition": "Influenza and pneumonia",
        "condition_group": "Respiratory diseases",
        "data_as_of": "2020-09-07T00:00:00.000",
        "end_week": "2020-09-06T00:00:00.000",
        "icd10_codes": "J09-J18",
        "number_covid19_deaths": "622",
        "start_week": "2020-02-01T00:00:00.000",
        "state": "US"
    },
    {
        "age_group": "35-44",
        "condition": "Influenza and pneumonia",
```

```
#Creating readable files and merging
covid = pd.read_json(url)
covid["state"].unique()
```

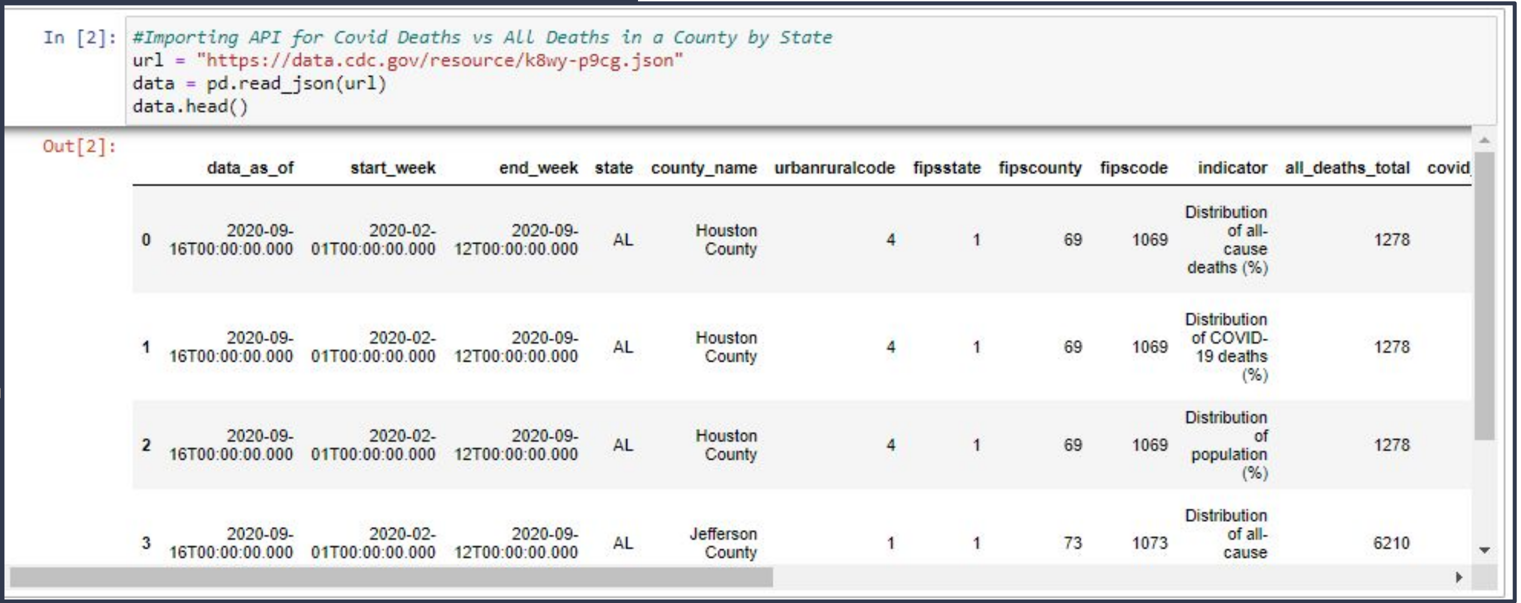
```
array(['US', 'AK', 'AL', 'AR', 'AZ'], dtype=object)
```

Adding the Dependencies



```
In [1]: import requests
        import json
        import pandas as pd
        import matplotlib.pyplot as plt
        import plotly.express as px
        from scipy.stats import linregress
        import numpy as np
```

Importing API

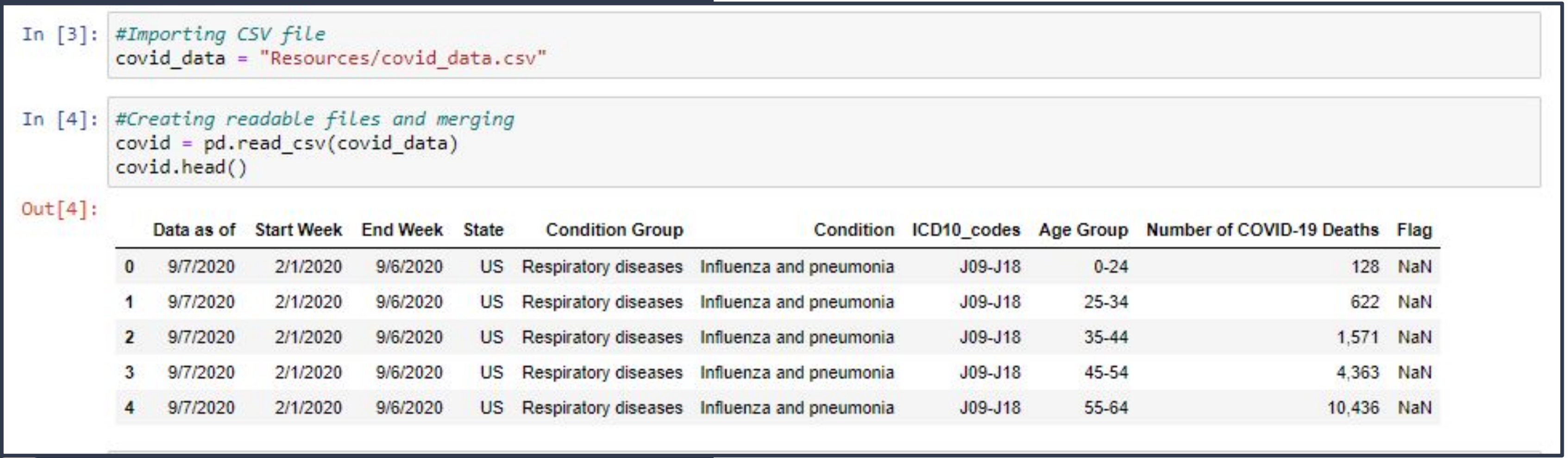


```
In [2]: #Importing API for Covid Deaths vs All Deaths in a County by State
        url = "https://data.cdc.gov/resource/k8wy-p9cg.json"
        data = pd.read_json(url)
        data.head()
```

Out[2]:

	data_as_of	start_week	end_week	state	county_name	urbanruralcode	fipsstate	fipscounty	fipscode	indicator	all_deaths_total	covid
0	2020-09-16T00:00:00.000	2020-02-01T00:00:00.000	2020-09-12T00:00:00.000	AL	Houston County	4	1	69	1069	Distribution of all-cause deaths (%)	1278	
1	2020-09-16T00:00:00.000	2020-02-01T00:00:00.000	2020-09-12T00:00:00.000	AL	Houston County	4	1	69	1069	Distribution of COVID-19 deaths (%)	1278	
2	2020-09-16T00:00:00.000	2020-02-01T00:00:00.000	2020-09-12T00:00:00.000	AL	Houston County	4	1	69	1069	Distribution of population (%)	1278	
3	2020-09-16T00:00:00.000	2020-02-01T00:00:00.000	2020-09-12T00:00:00.000	AL	Jefferson County	1	1	73	1073	Distribution of all-cause	6210	

Importing CSV file



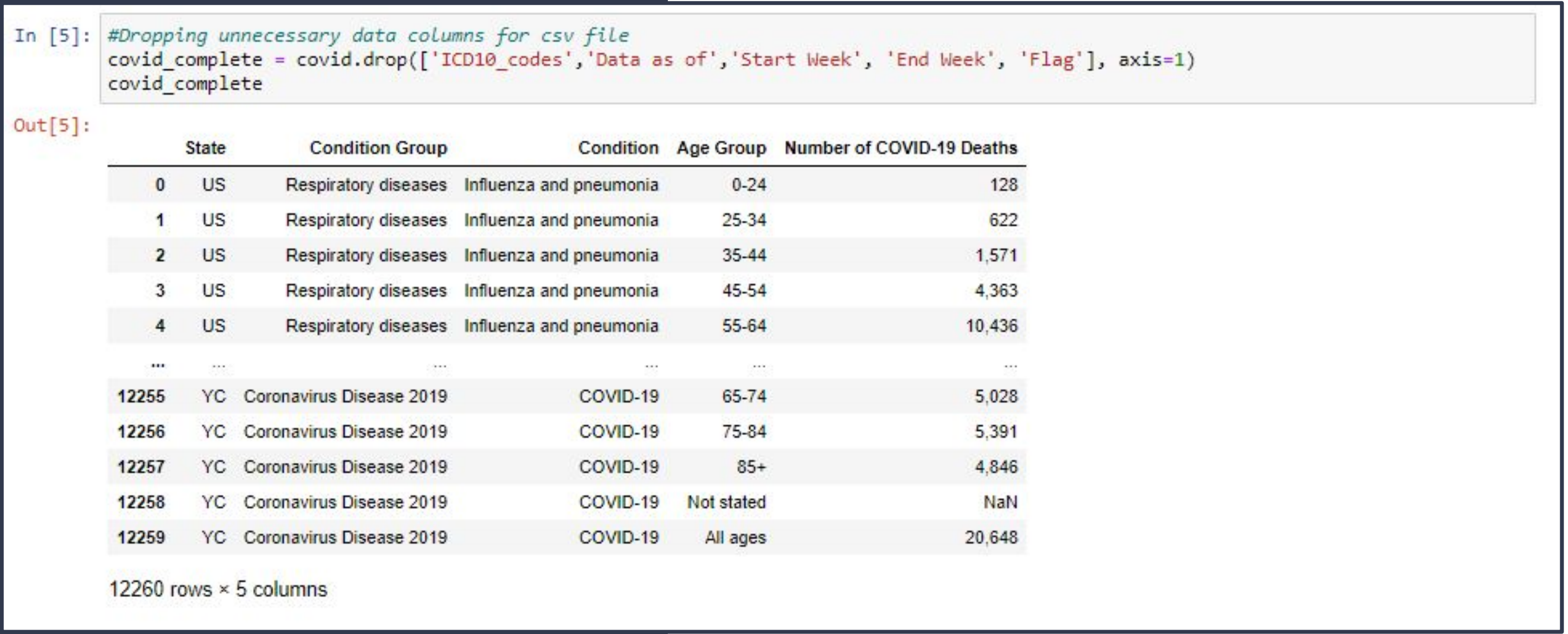
```
In [3]: #Importing CSV file
        covid_data = "Resources/covid_data.csv"
```

```
In [4]: #Creating readable files and merging
        covid = pd.read_csv(covid_data)
        covid.head()
```

Out[4]:

	Data as of	Start Week	End Week	State	Condition Group	Condition	ICD10_codes	Age Group	Number of COVID-19 Deaths	Flag
0	9/7/2020	2/1/2020	9/6/2020	US	Respiratory diseases	Influenza and pneumonia	J09-J18	0-24	128	NaN
1	9/7/2020	2/1/2020	9/6/2020	US	Respiratory diseases	Influenza and pneumonia	J09-J18	25-34	622	NaN
2	9/7/2020	2/1/2020	9/6/2020	US	Respiratory diseases	Influenza and pneumonia	J09-J18	35-44	1,571	NaN
3	9/7/2020	2/1/2020	9/6/2020	US	Respiratory diseases	Influenza and pneumonia	J09-J18	45-54	4,363	NaN
4	9/7/2020	2/1/2020	9/6/2020	US	Respiratory diseases	Influenza and pneumonia	J09-J18	55-64	10,436	NaN

Dropping the Columns



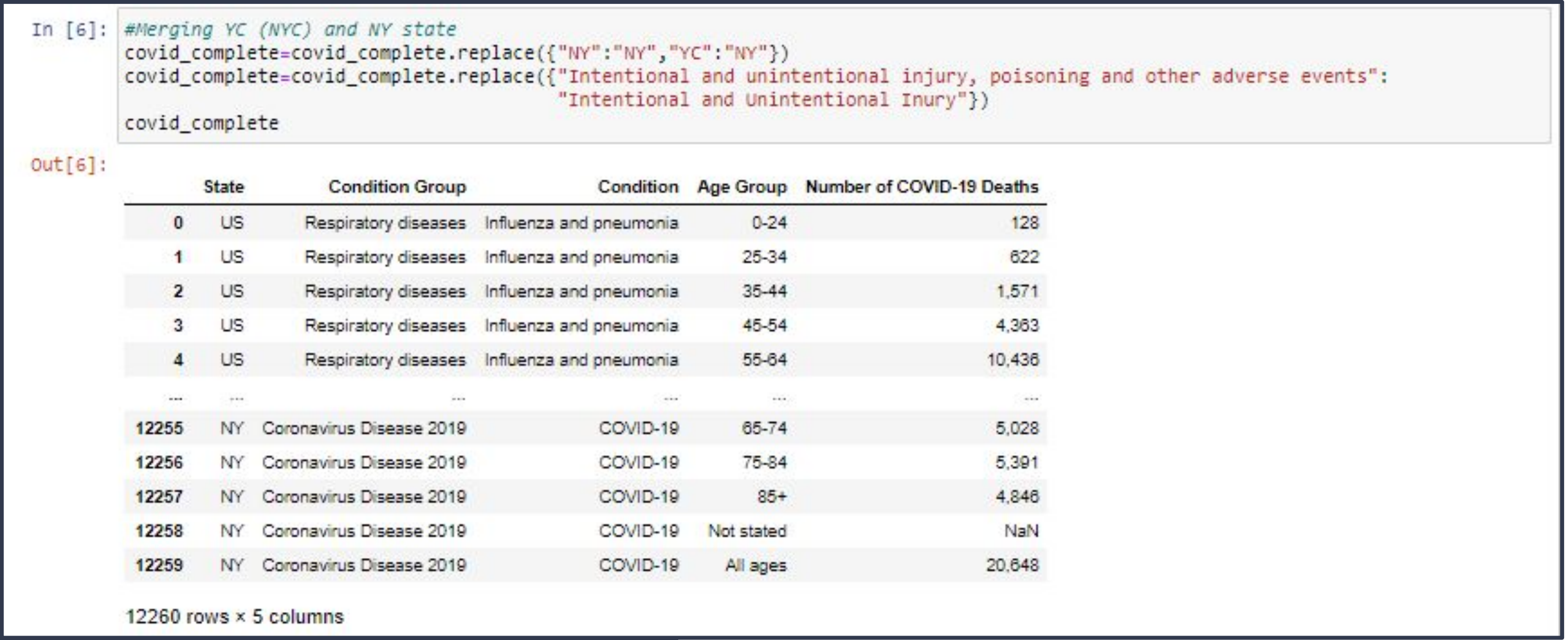
```
In [5]: #Dropping unnecessary data columns for csv file
        covid_complete = covid.drop(['ICD10_codes','Data as of','Start Week', 'End Week', 'Flag'], axis=1)
        covid_complete
```

Out[5]:

	State	Condition Group	Condition	Age Group	Number of COVID-19 Deaths
0	US	Respiratory diseases	Influenza and pneumonia	0-24	128
1	US	Respiratory diseases	Influenza and pneumonia	25-34	622
2	US	Respiratory diseases	Influenza and pneumonia	35-44	1,571
3	US	Respiratory diseases	Influenza and pneumonia	45-54	4,363
4	US	Respiratory diseases	Influenza and pneumonia	55-64	10,436
...	...	...	...	...	...
12255	YC	Coronavirus Disease 2019	COVID-19	65-74	5,028
12256	YC	Coronavirus Disease 2019	COVID-19	75-84	5,391
12257	YC	Coronavirus Disease 2019	COVID-19	85+	4,846
12258	YC	Coronavirus Disease 2019	COVID-19	Not stated	NaN
12259	YC	Coronavirus Disease 2019	COVID-19	All ages	20,648

12260 rows × 5 columns

Merged YC to NY



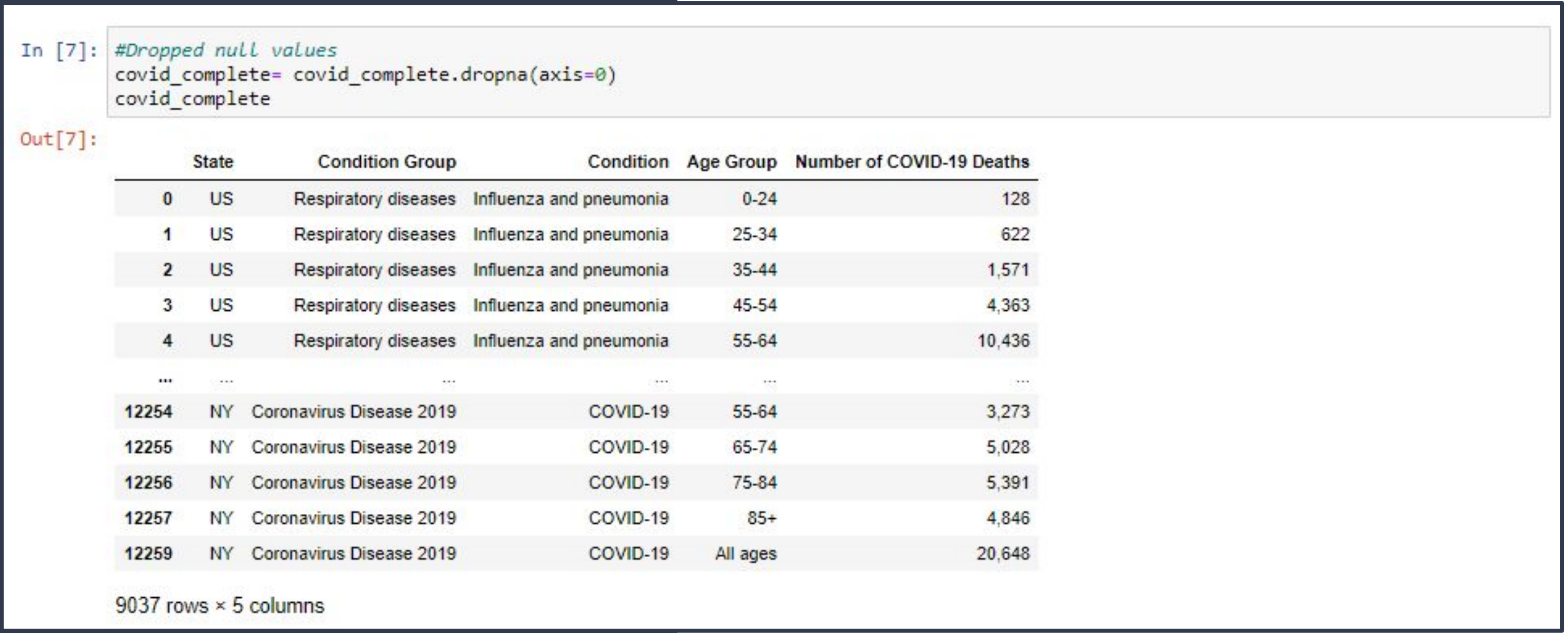
```
In [6]: #Merging YC (NYC) and NY state
        covid_complete=covid_complete.replace({"NY":"NY","YC":"NY"})
        covid_complete=covid_complete.replace({"Intentional and unintentional injury, poisoning and other adverse events":
                                               "Intentional and Unintentional Inury"})
        covid_complete
```

Out[6]:

	State	Condition Group	Condition	Age Group	Number of COVID-19 Deaths
0	US	Respiratory diseases	Influenza and pneumonia	0-24	128
1	US	Respiratory diseases	Influenza and pneumonia	25-34	622
2	US	Respiratory diseases	Influenza and pneumonia	35-44	1,571
3	US	Respiratory diseases	Influenza and pneumonia	45-54	4,363
4	US	Respiratory diseases	Influenza and pneumonia	55-64	10,436
...	...	...	...	...	...
12255	NY	Coronavirus Disease 2019	COVID-19	65-74	5,028
12256	NY	Coronavirus Disease 2019	COVID-19	75-84	5,391
12257	NY	Coronavirus Disease 2019	COVID-19	85+	4,846
12258	NY	Coronavirus Disease 2019	COVID-19	Not stated	NaN
12259	NY	Coronavirus Disease 2019	COVID-19	All ages	20,648

12260 rows × 5 columns

Dropped Null Values



In [7]:

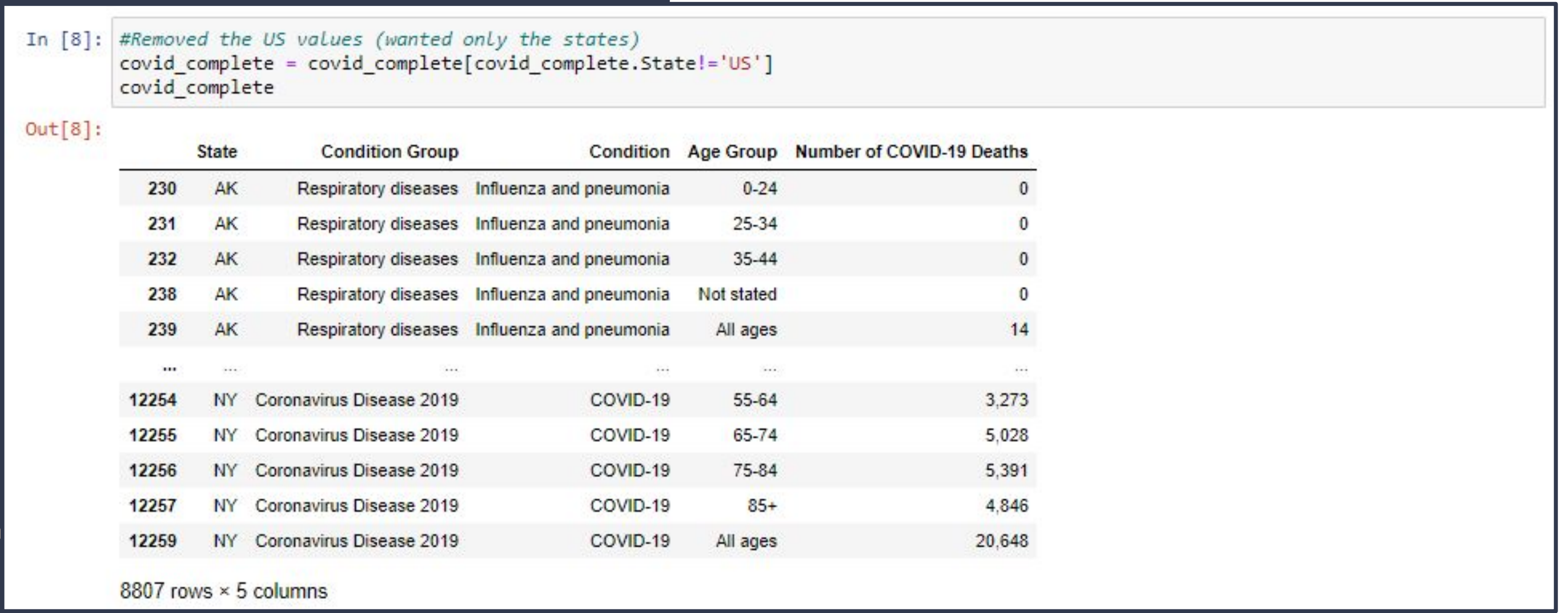
```
#Dropped null values
covid_complete= covid_complete.dropna(axis=0)
covid_complete
```

Out[7]:

	State	Condition Group	Condition	Age Group	Number of COVID-19 Deaths
0	US	Respiratory diseases	Influenza and pneumonia	0-24	128
1	US	Respiratory diseases	Influenza and pneumonia	25-34	622
2	US	Respiratory diseases	Influenza and pneumonia	35-44	1,571
3	US	Respiratory diseases	Influenza and pneumonia	45-54	4,363
4	US	Respiratory diseases	Influenza and pneumonia	55-64	10,436
...	...	...	...	...	...
12254	NY	Coronavirus Disease 2019	COVID-19	55-64	3,273
12255	NY	Coronavirus Disease 2019	COVID-19	65-74	5,028
12256	NY	Coronavirus Disease 2019	COVID-19	75-84	5,391
12257	NY	Coronavirus Disease 2019	COVID-19	85+	4,846
12259	NY	Coronavirus Disease 2019	COVID-19	All ages	20,648

9037 rows × 5 columns

Removed US from Dataframe



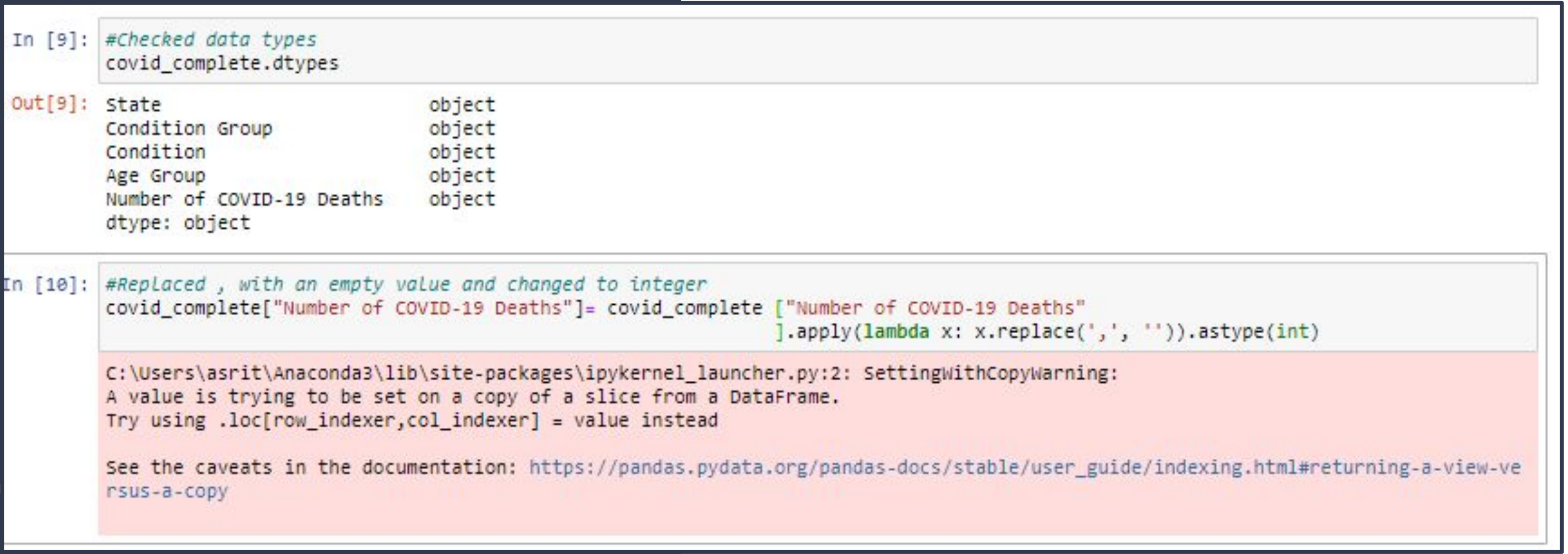
```
In [8]: #Removed the US values (wanted only the states)
        covid_complete = covid_complete[covid_complete.State!='US']
        covid_complete
```

Out[8]:

	State	Condition Group	Condition	Age Group	Number of COVID-19 Deaths
230	AK	Respiratory diseases	Influenza and pneumonia	0-24	0
231	AK	Respiratory diseases	Influenza and pneumonia	25-34	0
232	AK	Respiratory diseases	Influenza and pneumonia	35-44	0
238	AK	Respiratory diseases	Influenza and pneumonia	Not stated	0
239	AK	Respiratory diseases	Influenza and pneumonia	All ages	14
...	...	...	...	...	...
12254	NY	Coronavirus Disease 2019	COVID-19	55-64	3,273
12255	NY	Coronavirus Disease 2019	COVID-19	65-74	5,028
12256	NY	Coronavirus Disease 2019	COVID-19	75-84	5,391
12257	NY	Coronavirus Disease 2019	COVID-19	85+	4,846
12259	NY	Coronavirus Disease 2019	COVID-19	All ages	20,648

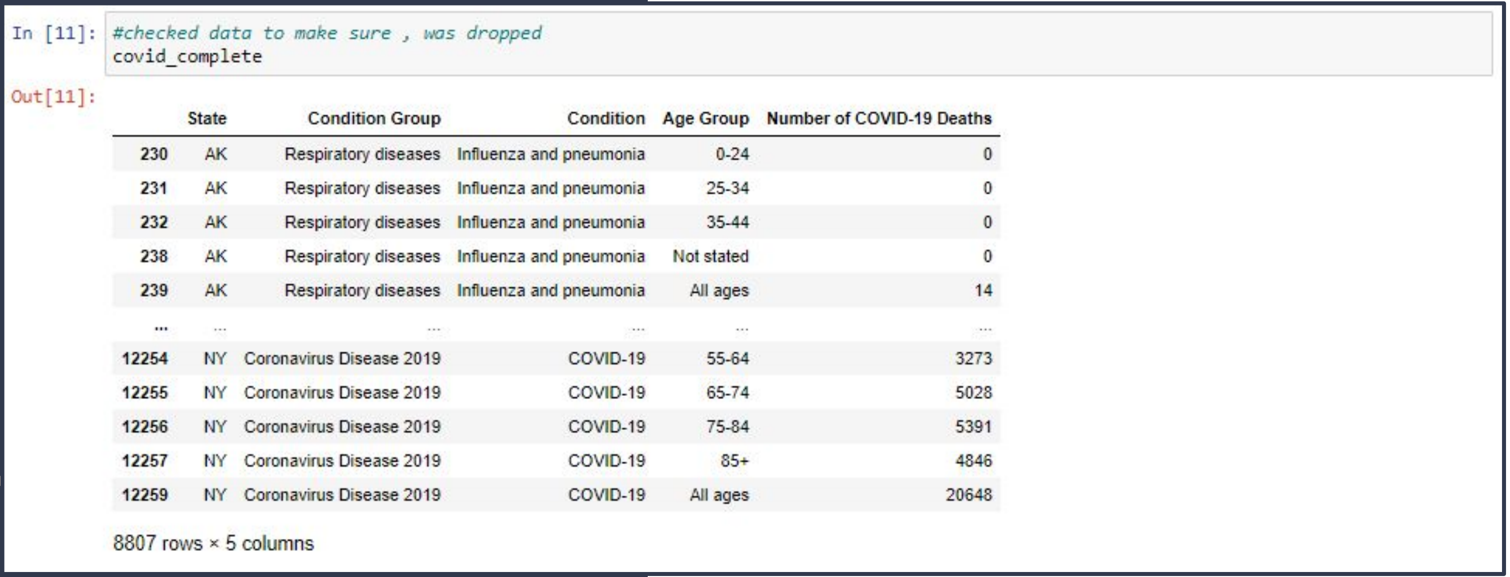
8807 rows × 5 columns

Changing Data type for Number of Deaths



```
In [9]:  #Checked data types
         covid_complete.dtypes

Out[9]:  State                         object
         Condition Group               object
         Condition                     object
         Age Group                     object
         Number of COVID-19 Deaths     object
         dtype: object

In [10]: #Replaced , with an empty value and changed to integer
         covid_complete["Number of COVID-19 Deaths"]= covid_complete ["Number of COVID-19 Deaths"
                                                               ].apply(lambda x: x.replace(',', '')).astype(int)

         C:\Users\asrit\Anaconda3\lib\site-packages\ipykernel_launcher.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: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ve
         rsus-a-copy
```

Checked Deaths Column Removal of Comma



In [11]:

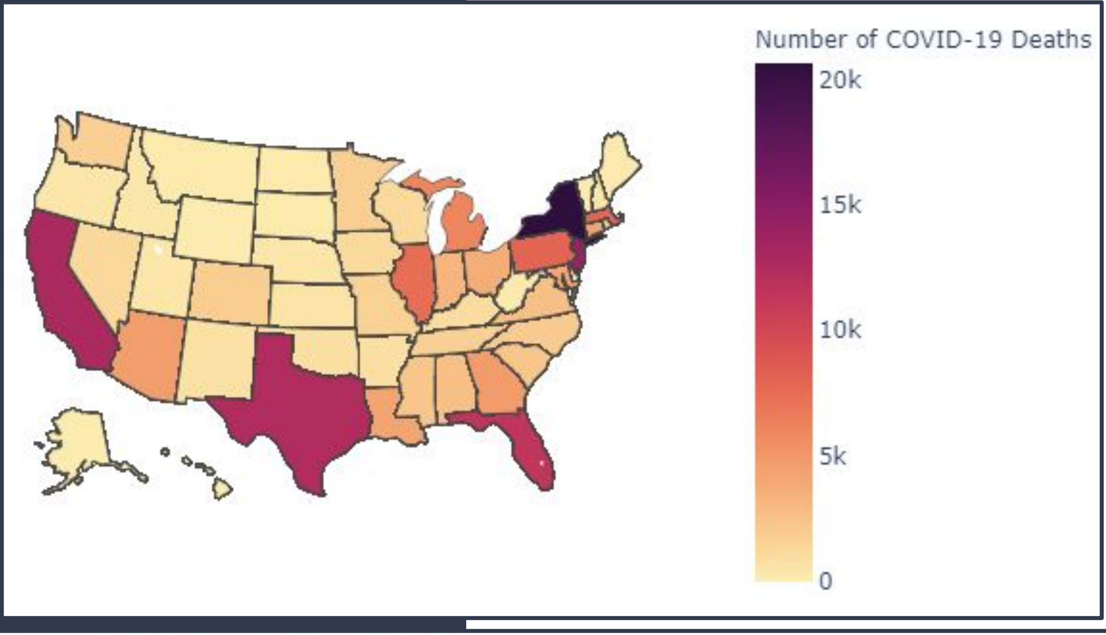
```
#checked data to make sure , was dropped
covid_complete
```

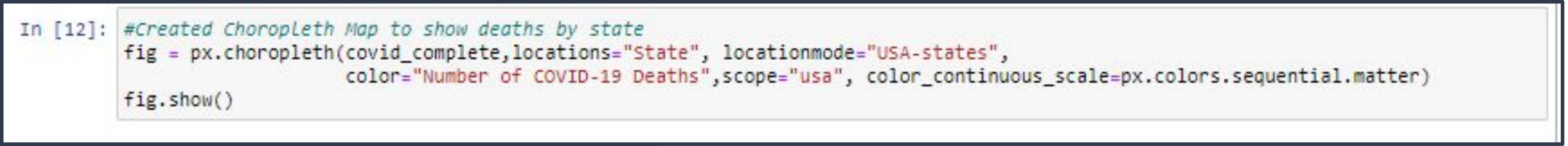
Out[11]:

	State	Condition Group	Condition	Age Group	Number of COVID-19 Deaths
230	AK	Respiratory diseases	Influenza and pneumonia	0-24	0
231	AK	Respiratory diseases	Influenza and pneumonia	25-34	0
232	AK	Respiratory diseases	Influenza and pneumonia	35-44	0
238	AK	Respiratory diseases	Influenza and pneumonia	Not stated	0
239	AK	Respiratory diseases	Influenza and pneumonia	All ages	14
...	...	...	...	...	...
12254	NY	Coronavirus Disease 2019	COVID-19	55-64	3273
12255	NY	Coronavirus Disease 2019	COVID-19	65-74	5028
12256	NY	Coronavirus Disease 2019	COVID-19	75-84	5391
12257	NY	Coronavirus Disease 2019	COVID-19	85+	4846
12259	NY	Coronavirus Disease 2019	COVID-19	All ages	20648

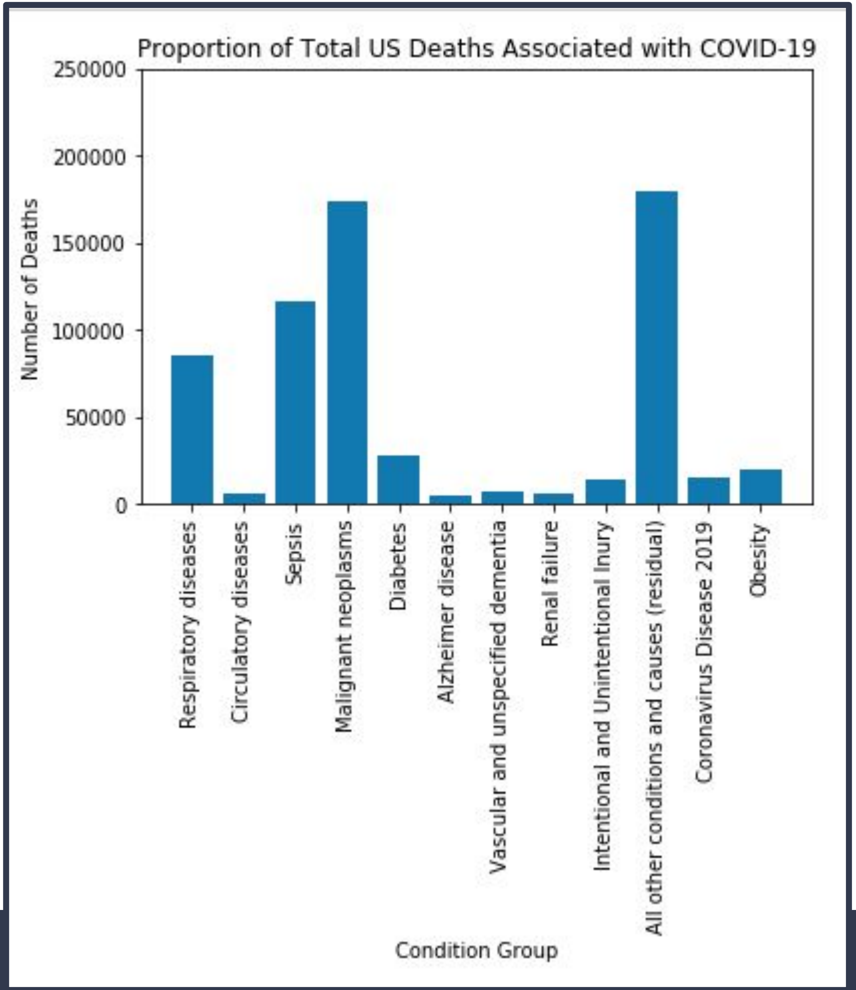
8807 rows × 5 columns

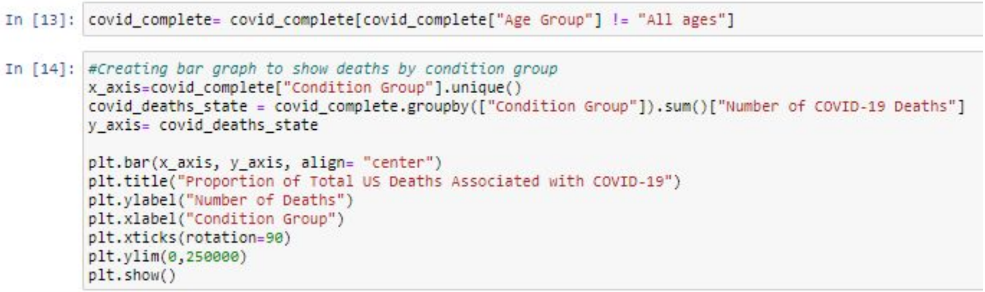
Spatial Distribution of Covid-19 Fatalities





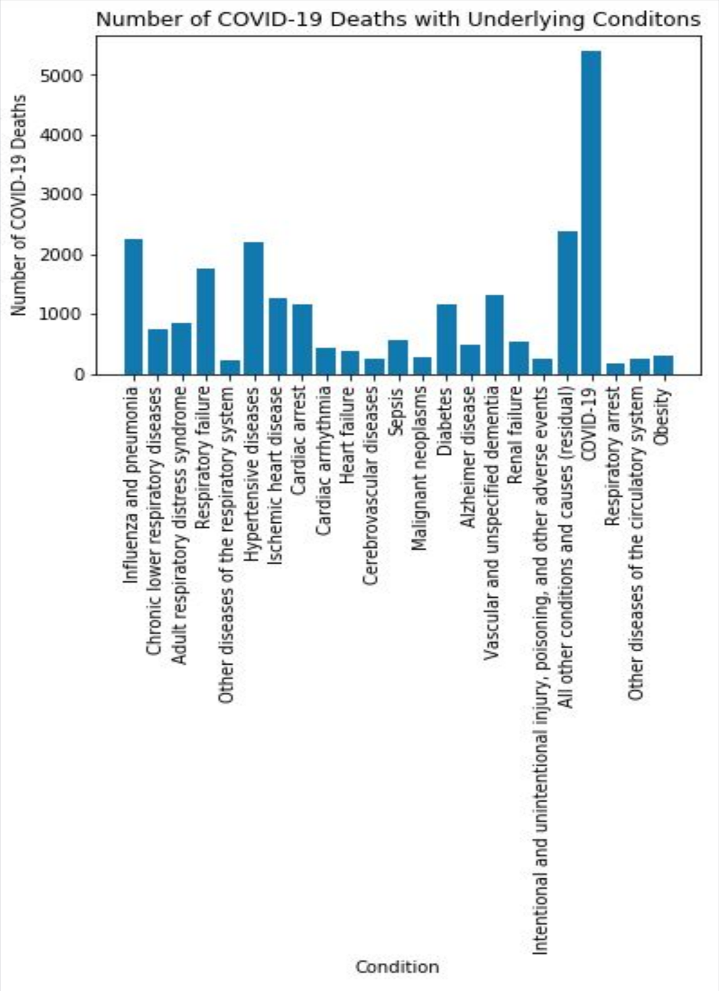
```
In [12]: #Created Choropleth Map to show deaths by state
         fig = px.choropleth(covid_complete,locations="State", locationmode="USA-states",
                             color="Number of COVID-19 Deaths",scope="usa", color_continuous_scale=px.colors.sequential.matter)
         fig.show()
```

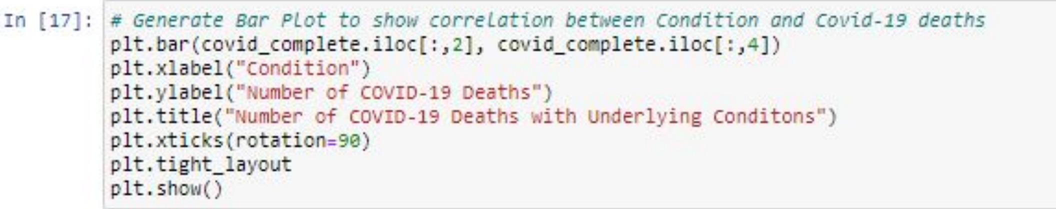




```
In [13]: covid_complete= covid_complete[covid_complete["Age Group"] != "All ages"]
```

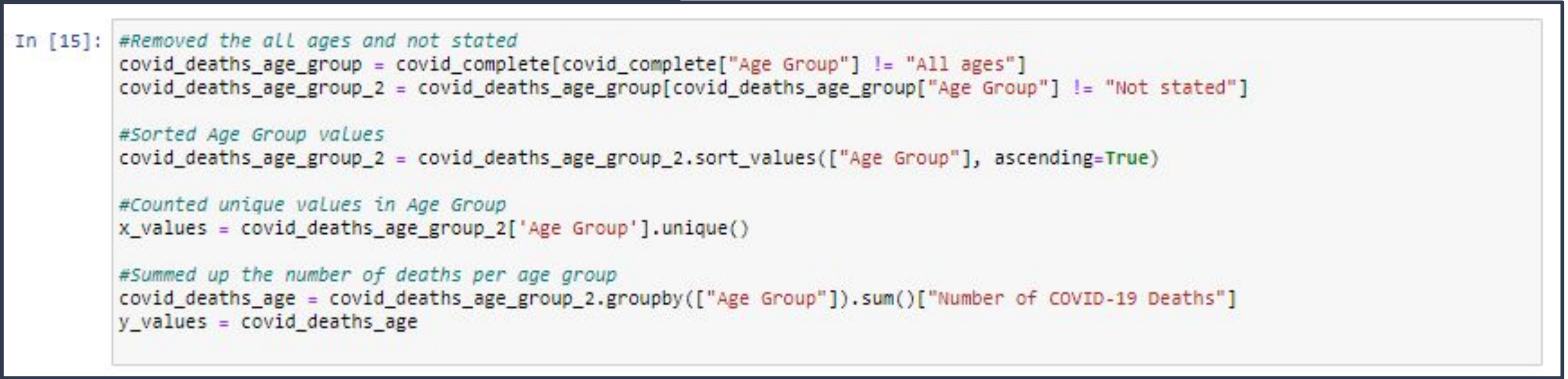
```
In [14]: #Creating bar graph to show deaths by condition group
         x_axis=covid_complete["Condition Group"].unique()
         covid_deaths_state = covid_complete.groupby(["Condition Group"]).sum()["Number of COVID-19 Deaths"]
         y_axis= covid_deaths_state

         plt.bar(x_axis, y_axis, align= "center")
         plt.title("Proportion of Total US Deaths Associated with COVID-19")
         plt.ylabel("Number of Deaths")
         plt.xlabel("Condition Group")
         plt.xticks(rotation=90)
         plt.ylim(0,250000)
         plt.show()
```





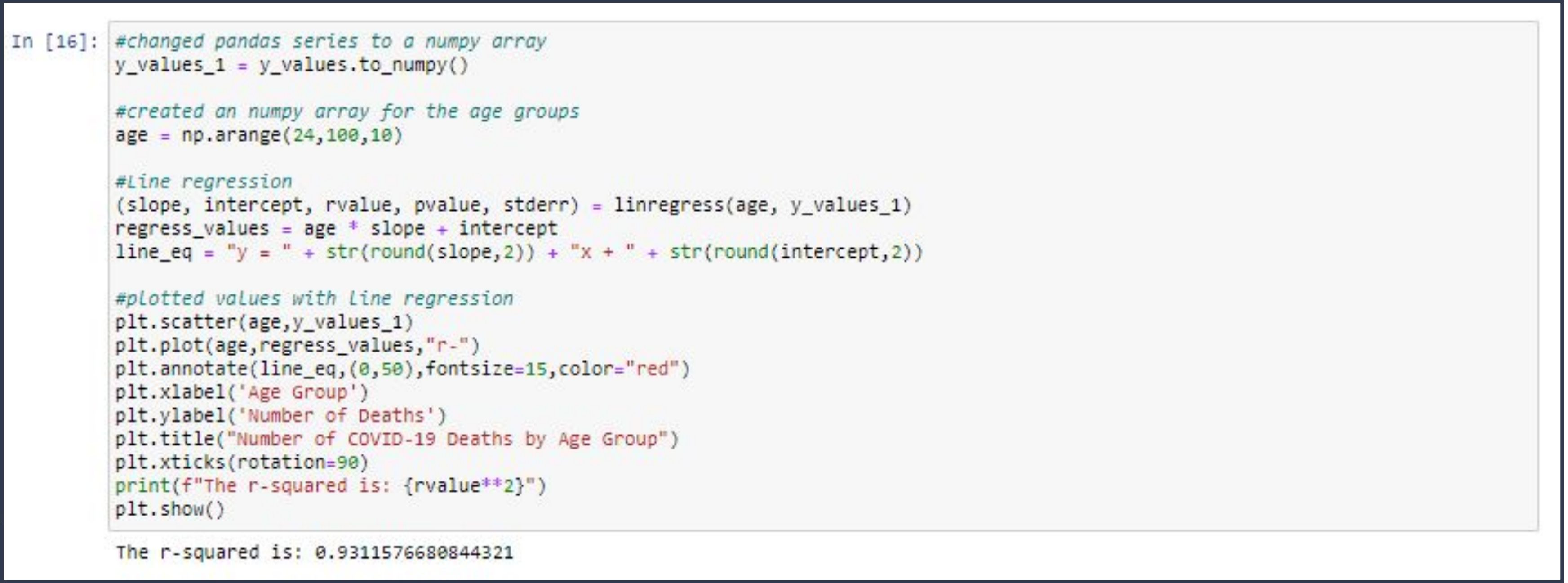
```
In [17]: # Generate Bar Plot to show correlation between Condition and Covid-19 deaths
         plt.bar(covid_complete.iloc[:,2], covid_complete.iloc[:,4])
         plt.xlabel("Condition")
         plt.ylabel("Number of COVID-19 Deaths")
         plt.title("Number of COVID-19 Deaths with Underlying Conditons")
         plt.xticks(rotation=90)
         plt.tight_layout
         plt.show()
```

Created Variables for Scatter Plot



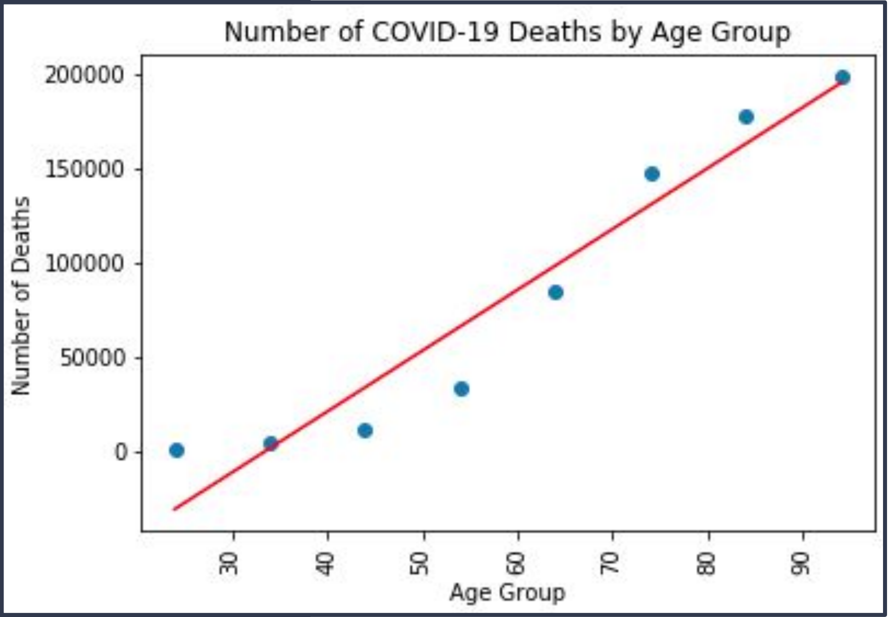
```
In [15]: #Removed the all ages and not stated
         covid_deaths_age_group = covid_complete[covid_complete["Age Group"] != "All ages"]
         covid_deaths_age_group_2 = covid_deaths_age_group[covid_deaths_age_group["Age Group"] != "Not stated"]

         #Sorted Age Group values
         covid_deaths_age_group_2 = covid_deaths_age_group_2.sort_values(["Age Group"], ascending=True)

         #Counted unique values in Age Group
         x_values = covid_deaths_age_group_2['Age Group'].unique()

         #Summed up the number of deaths per age group
         covid_deaths_age = covid_deaths_age_group_2.groupby(["Age Group"]).sum()["Number of COVID-19 Deaths"]
         y_values = covid_deaths_age
```

Changed Pandas Series to Numpy Array Plotted Graph with Regression Line

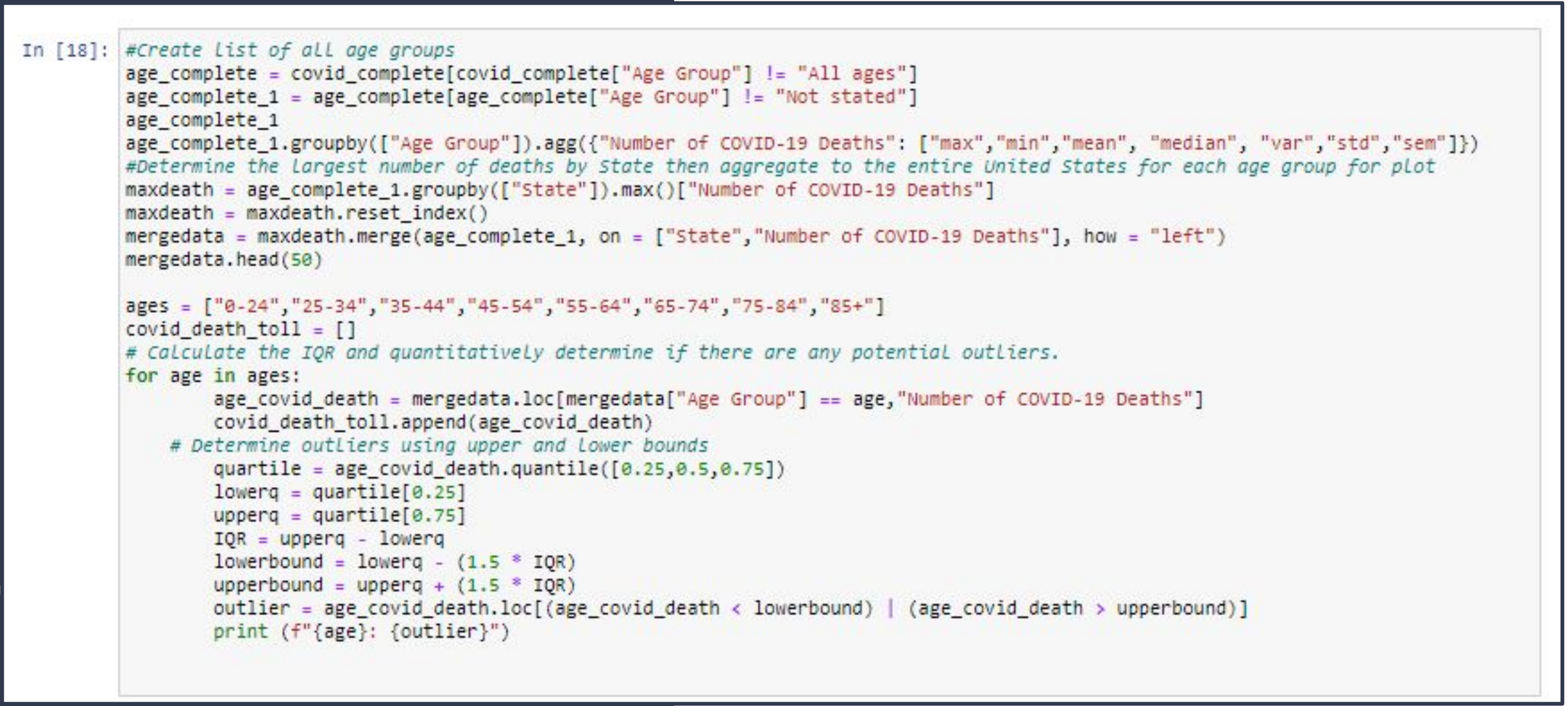


```
In [16]: #changed pandas series to a numpy array
         y_values_1 = y_values.to_numpy()

         #created an numpy array for the age groups
         age = np.arange(24,100,10)

         #Line regression
         (slope, intercept, rvalue, pvalue, stderr) = linregress(age, y_values_1)
         regress_values = age * slope + intercept
         line_eq = "y = " + str(round(slope,2)) + "x + " + str(round(intercept,2))

         #plotted values with line regression
         plt.scatter(age,y_values_1)
         plt.plot(age,regress_values,"r-")
         plt.annotate(line_eq,(0,50),fontsize=15,color="red")
         plt.xlabel('Age Group')
         plt.ylabel('Number of Deaths')
         plt.title("Number of COVID-19 Deaths by Age Group")
         plt.xticks(rotation=90)
         print(f"The r-squared is: {rvalue**2}")
         plt.show()

         The r-squared is: 0.9311576680844321
```

Correlation Between Age Group and Number of Covid-19 Deaths

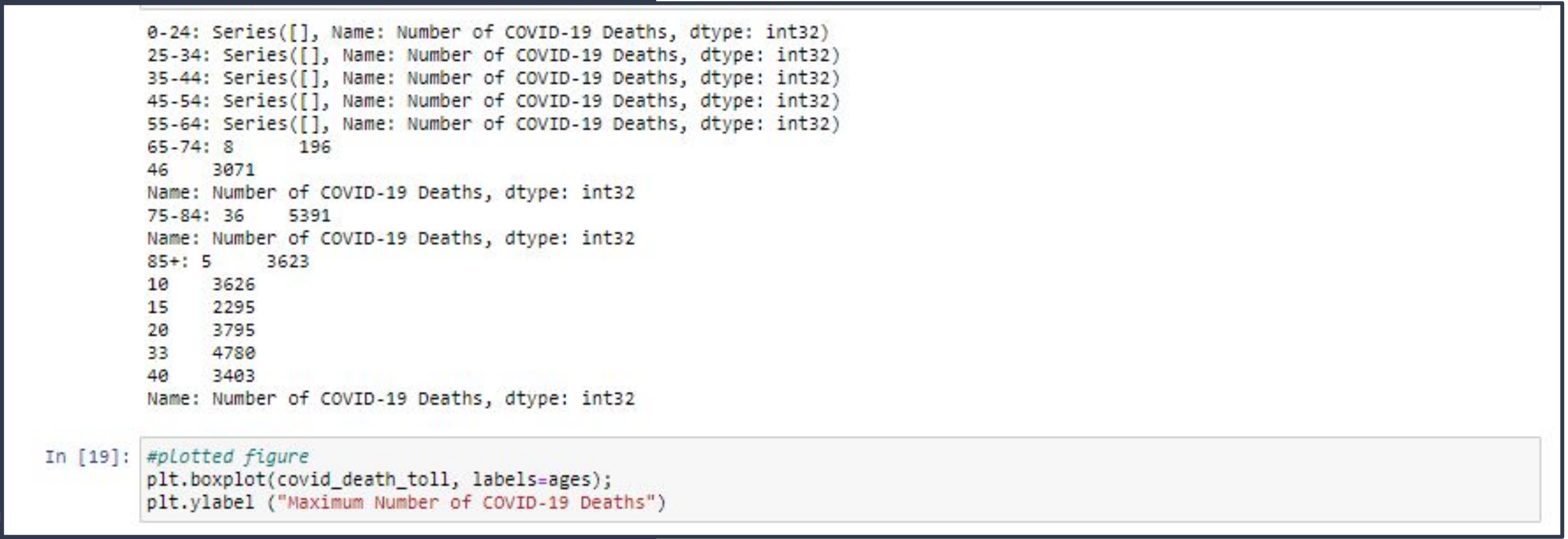


Set Variables for BoxPlot



```
In [18]: #Create list of all age groups
age_complete = covid_complete[covid_complete["Age Group"] != "All ages"]
age_complete_1 = age_complete[age_complete["Age Group"] != "Not stated"]
age_complete_1
age_complete_1.groupby(["Age Group"]).agg({"Number of COVID-19 Deaths": ["max","min","mean", "median", "var","std","sem"]})
#Determine the largest number of deaths by State then aggregate to the entire United States for each age group for plot
maxdeath = age_complete_1.groupby(["State"]).max()["Number of COVID-19 Deaths"]
maxdeath = maxdeath.reset_index()
mergedata = maxdeath.merge(age_complete_1, on = ["State","Number of COVID-19 Deaths"], how = "left")
mergedata.head(50)

ages = ["0-24","25-34","35-44","45-54","55-64","65-74","75-84","85+"]
covid_death_toll = []
# Calculate the IQR and quantitatively determine if there are any potential outliers.
for age in ages:
        age_covid_death = mergedata.loc[mergedata["Age Group"] == age,"Number of COVID-19 Deaths"]
        covid_death_toll.append(age_covid_death)
    # Determine outliers using upper and lower bounds
        quartile = age_covid_death.quantile([0.25,0.5,0.75])
        lowerq = quartile[0.25]
        upperq = quartile[0.75]
        IQR = upperq - lowerq
        lowerbound = lowerq - (1.5 * IQR)
        upperbound = upperq + (1.5 * IQR)
        outlier = age_covid_death.loc[(age_covid_death < lowerbound) | (age_covid_death > upperbound)]
        print (f"{age}: {outlier}")
```

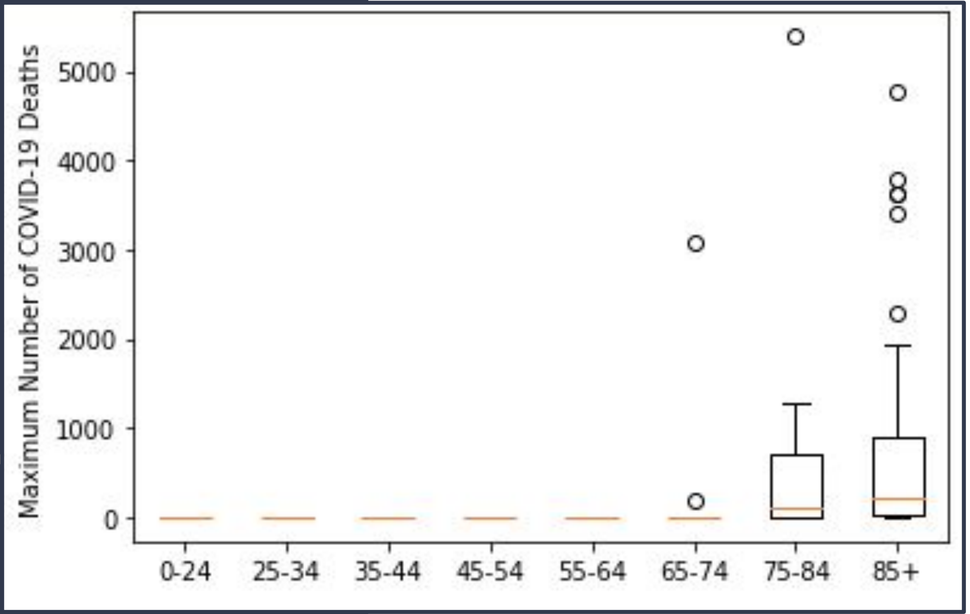
Output of Prior Code



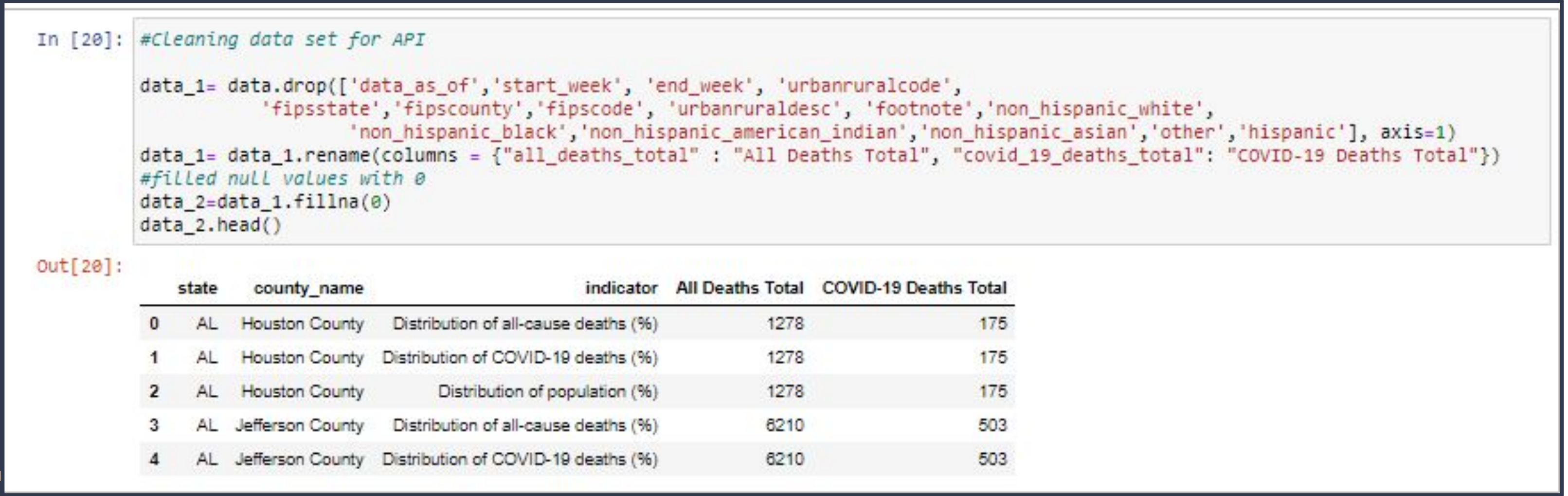
```
0-24: Series([], Name: Number of COVID-19 Deaths, dtype: int32)
25-34: Series([], Name: Number of COVID-19 Deaths, dtype: int32)
35-44: Series([], Name: Number of COVID-19 Deaths, dtype: int32)
45-54: Series([], Name: Number of COVID-19 Deaths, dtype: int32)
55-64: Series([], Name: Number of COVID-19 Deaths, dtype: int32)
65-74: 8      196
46     3071
Name: Number of COVID-19 Deaths, dtype: int32
75-84: 36    5391
Name: Number of COVID-19 Deaths, dtype: int32
85+: 5     3623
10    3626
15    2295
20    3795
33    4780
40    3403
Name: Number of COVID-19 Deaths, dtype: int32
```

```
In [19]: #plotted figure
         plt.boxplot(covid_death_toll, labels=ages);
         plt.ylabel ("Maximum Number of COVID-19 Deaths")
```

Maximum Number of Covid-19 Deaths By Age Group



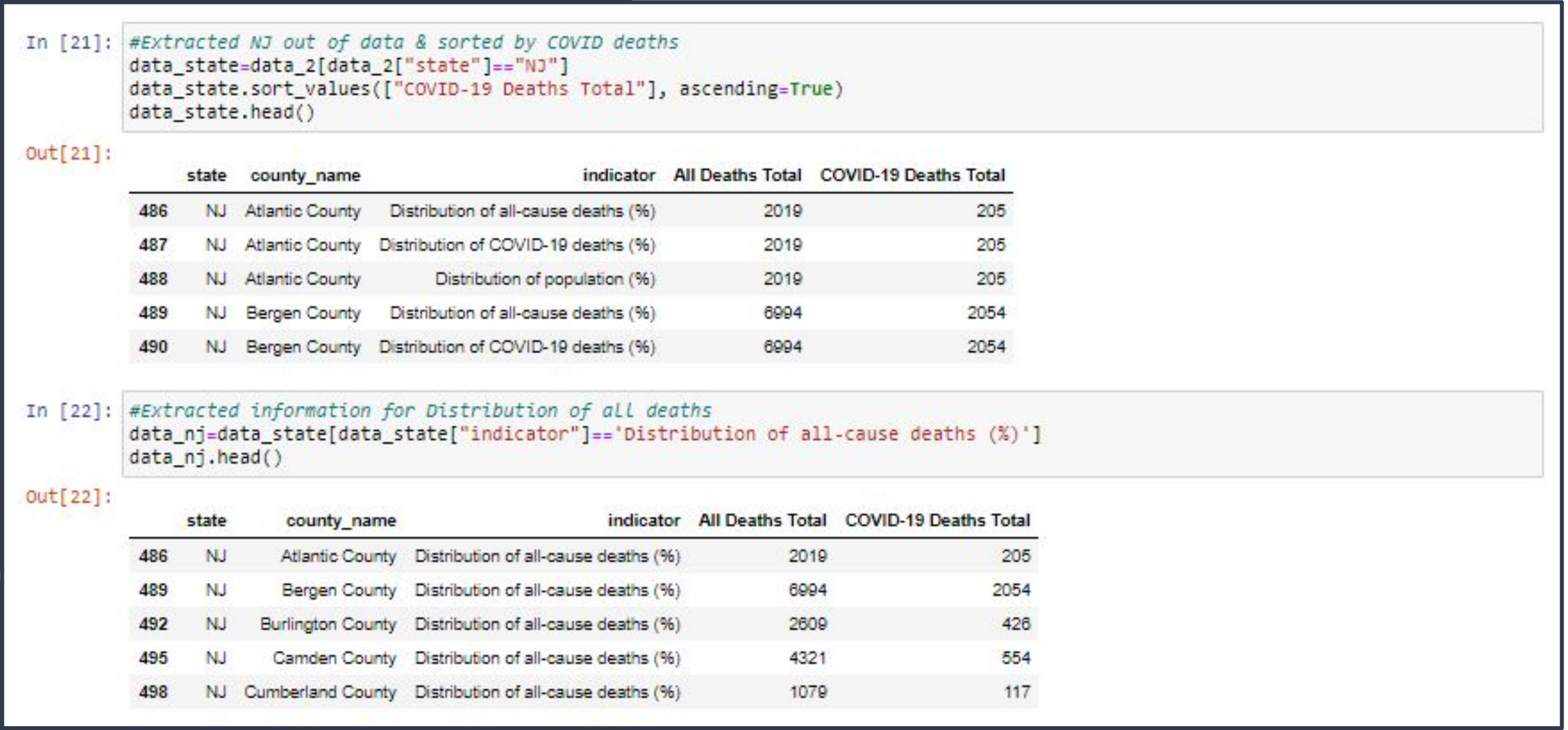
Cleaning the API Data



```
In [20]: #Cleaning data set for API

         data_1= data.drop(['data_as_of','start_week', 'end_week', 'urbanruralcode',
                     'fipsstate','fipscounty','fipscode', 'urbanruraldesc', 'footnote','non_hispanic_white',
                             'non_hispanic_black','non_hispanic_american_indian','non_hispanic_asian','other','hispanic'], axis=1)
         data_1= data_1.rename(columns = {"all_deaths_total" : "All Deaths Total", "covid_19_deaths_total": "COVID-19 Deaths Total"})
         #filled null values with 0
         data_2=data_1.fillna(0)
         data_2.head()
```

```
Out[20]:
      state     county_name                          indicator  All Deaths Total  COVID-19 Deaths Total
   0     AL  Houston County  Distribution of all-cause deaths (%)              1278                    175
   1     AL  Houston County   Distribution of COVID-19 deaths (%)              1278                    175
   2     AL  Houston County        Distribution of population (%)              1278                    175
   3     AL Jefferson County Distribution of all-cause deaths (%)              6210                    503
   4     AL Jefferson County  Distribution of COVID-19 deaths (%)              6210                    503
```

Extracted and Sorted Data



```
In [21]: #Extracted NJ out of data & sorted by COVID deaths
         data_state=data_2[data_2["state"]=="NJ"]
         data_state.sort_values(["COVID-19 Deaths Total"], ascending=True)
         data_state.head()
```

Out[21]:

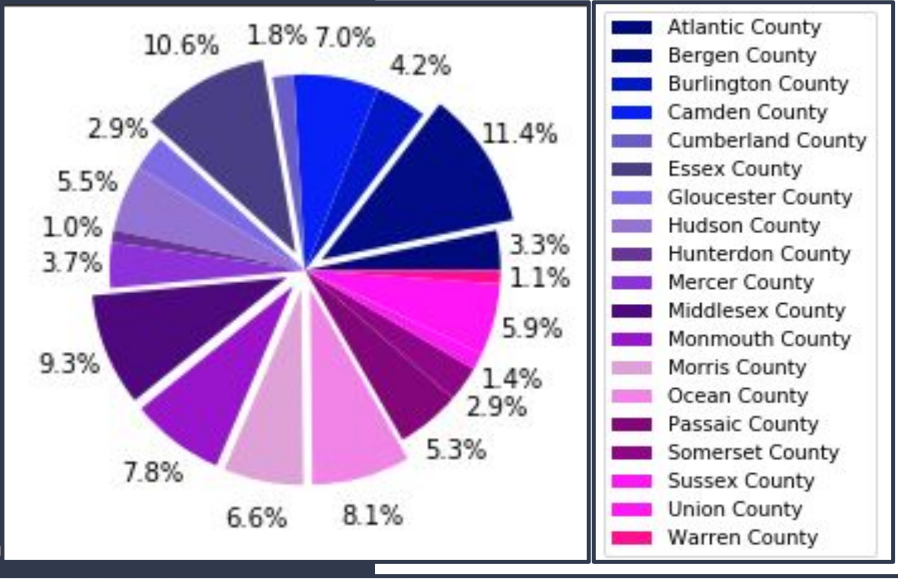
	state	county_name	indicator	All Deaths Total	COVID-19 Deaths Total
486	NJ	Atlantic County	Distribution of all-cause deaths (%)	2019	205
487	NJ	Atlantic County	Distribution of COVID-19 deaths (%)	2019	205
488	NJ	Atlantic County	Distribution of population (%)	2019	205
489	NJ	Bergen County	Distribution of all-cause deaths (%)	6994	2054
490	NJ	Bergen County	Distribution of COVID-19 deaths (%)	6994	2054

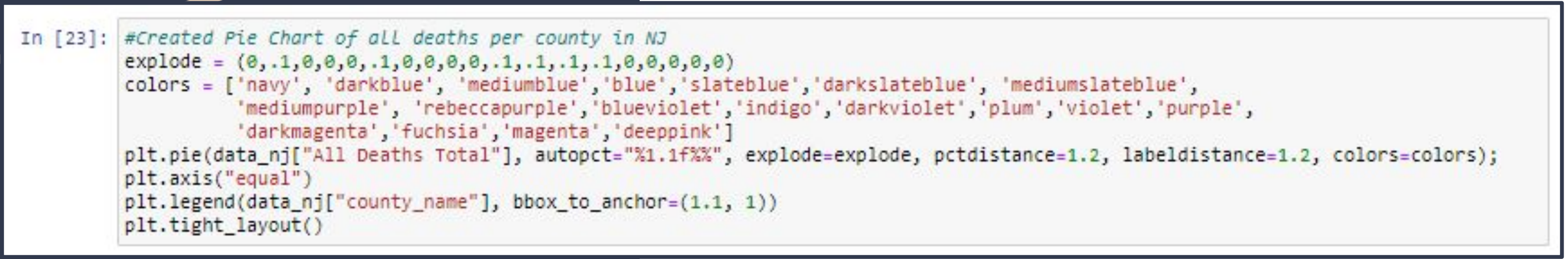
```
In [22]: #Extracted information for Distribution of all deaths
         data_nj=data_state[data_state["indicator"]=='Distribution of all-cause deaths (%)']
         data_nj.head()
```

Out[22]:

	state	county_name	indicator	All Deaths Total	COVID-19 Deaths Total
486	NJ	Atlantic County	Distribution of all-cause deaths (%)	2019	205
489	NJ	Bergen County	Distribution of all-cause deaths (%)	6994	2054
492	NJ	Burlington County	Distribution of all-cause deaths (%)	2609	426
495	NJ	Camden County	Distribution of all-cause deaths (%)	4321	554
498	NJ	Cumberland County	Distribution of all-cause deaths (%)	1079	117

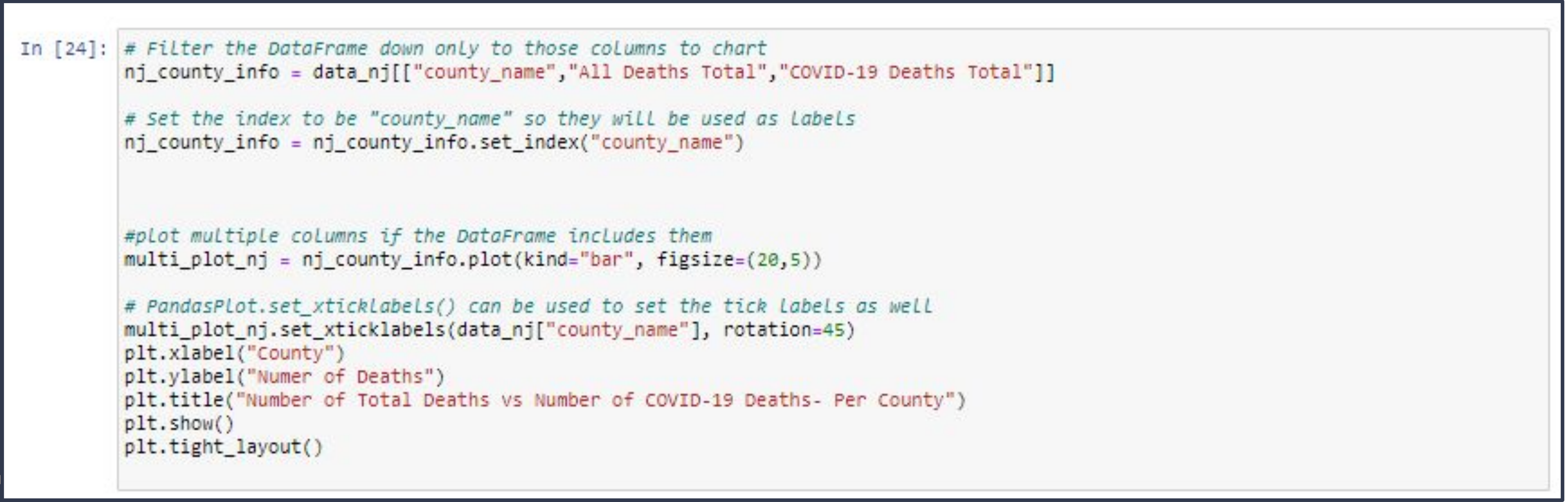
Mortality Rate of All Deaths by County in New Jersey





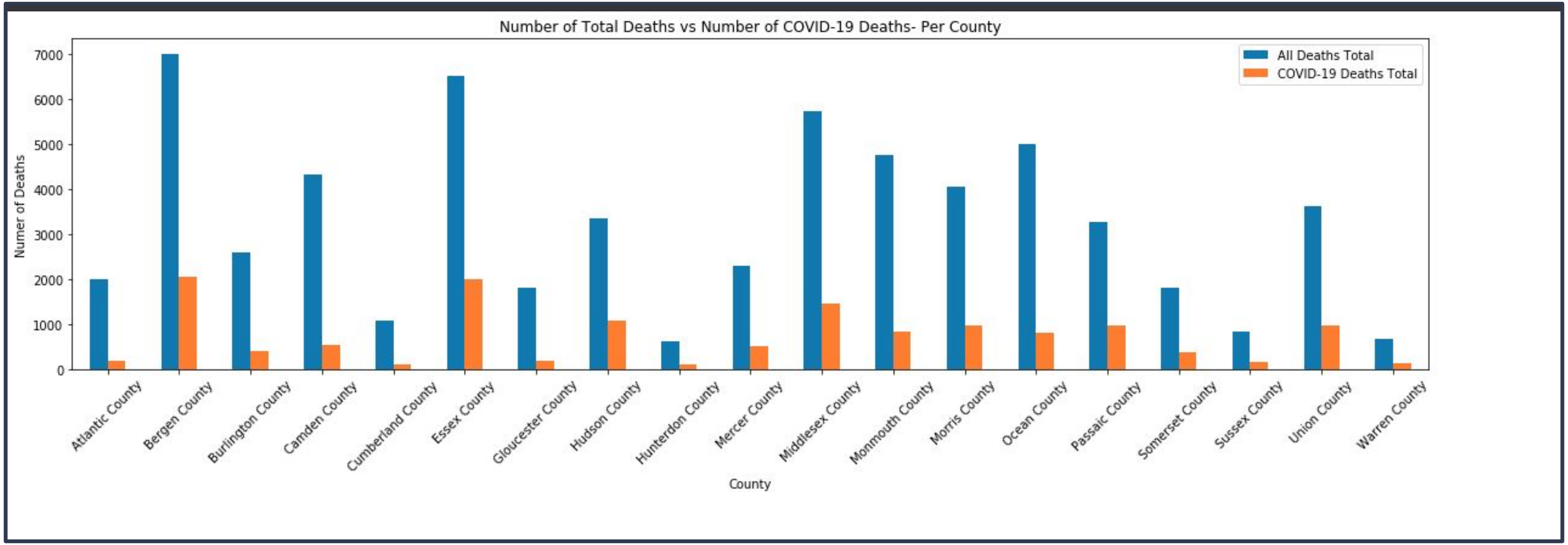
```
In [23]: #Created Pie Chart of all deaths per county in NJ
         explode = (0,.1,0,0,0,.1,0,0,0,0,.1,.1,.1,.1,0,0,0,0,0)
         colors = ['navy', 'darkblue', 'mediumblue','blue','slateblue','darkslateblue', 'mediumslateblue',
                   'mediumpurple', 'rebeccapurple','blueviolet','indigo','darkviolet','plum','violet','purple',
                   'darkmagenta','fuchsia','magenta','deeppink']
         plt.pie(data_nj["All Deaths Total"], autopct="%1.1f%%", explode=explode, pctdistance=1.2, labeldistance=1.2, colors=colors);
         plt.axis("equal")
         plt.legend(data_nj["county_name"], bbox_to_anchor=(1.1, 1))
         plt.tight_layout()
```

Creating Bar Graph Comparing COVID-19 Deaths to All Deaths Per County



In [24]:

```
# Filter the DataFrame down only to those columns to chart
nj_county_info = data_nj[["county_name","All Deaths Total","COVID-19 Deaths Total"]]

# Set the index to be "county_name" so they will be used as labels
nj_county_info = nj_county_info.set_index("county_name")



#plot multiple columns if the DataFrame includes them
multi_plot_nj = nj_county_info.plot(kind="bar", figsize=(20,5))

# PandasPlot.set_xticklabels() can be used to set the tick labels as well
multi_plot_nj.set_xticklabels(data_nj["county_name"], rotation=45)
plt.xlabel("County")
plt.ylabel("Numer of Deaths")
plt.title("Number of Total Deaths vs Number of COVID-19 Deaths- Per County")
plt.show()
plt.tight_layout()
```


Number of Total Deaths vs Number of COVID-19 Deaths- Per County
All Deaths Total
COVID-19 Deaths Total
Numer of Deaths
0
1000
2000
3000
4000
5000
6000
7000
Atlantic County
Bergen County
Burlington County
Camden County
Cumberland County
Essex County
Gloucester County
Hudson County
Hunterdon County
Mercer County
Middlesex County
Monmouth County
Morris County
Ocean County
Passaic County
Somerset County
Sussex County
Union County
Warren County
County

Thank You!

(& Stay Safe!)