Project 1: The Effects of the Number of Confirmed Case, Elderly Population, Air Pollution Exposure, Adult Education Rate and Health Spending on the Mortality Rate.

This section of the project 2 contains revised version of project 1. The project 1 in revised in three ways,

1) fixing typos,

2) updating the data to a more recent data,

3) adding US data into the data set,

4) deleting "Urban population" data,

5) changing the response variable to Mortality rate and setting the initial response variable, confirmed case, as a predictor variable,

6) editing graphical figures,

7) redefining "Adult Education" variable.

This edit was performed due to the increasing number of covid confirmed cases over the world. Many reports have stated that covid-19 is very contagious but not too strong to kill a healthy patient. In addition, unlikely to the reports from the begining of the 1st wave, covid-19 is more contagious among elders, recent findings and trends show that the it is contagious among all ages. Therefore, it is more reasonable to look at the mortality rate than the number of confirmed cases.

Additionally, the data is updated to a more recent data since the second and the third waves have begun globally, so the number of confirmed cases and deaths have significantly increased, therefore the updated data is able to give more accurate and up to date results.

US data was added as the response variable has changed. When the confirmed case was the response variable, because the US's number of confirmed case was extraordinarily big, it behaved like an outlier, so was excluded. Because the confirmed case is still a variable we consider, its threat as a possible outlier still presents, but because it has useful data for other variables, it is included in the revised version.

Urban population data is excluded because it contained too many missing data. As a result of deleting the variable, the we were able to attain 7 more countries' data.

The unit of confirmed case is changed. The number is multiplied by 0.001 to work with a smaller number.

The graphical figures are edited to be more pleasant visually and the code itself is changed. One function is defined to generate a histogram so the code got shorter.

Finally, the adult education variable is redefined to the percentage of the population who has completed post-secondary education over the total population from the the percentage of the population who has not completed post-secondary education over the total population.

Introduction

The COVID - 19 Data is given by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University and all the other data are from OECD statistics. The world has been suffered from the pandemic for more than 3/4 of the year and resulted in a tremendous number of casualities, however, how much a country suffer from covid-19 is quite different. Therefore, in this project, we will look closely at factors, such as the number of confirmed case, the elderly population, air pollution exposure, adult education rate and health spending of a nation and examine if these factors, predictor variables, in other words, have any impact on the mortality rate, the response variable.

The elderly population data shows the share of the total elderly population (aged 60+) expressed as a ratio of the total population. The air pollution exposure data refer to population exposure to more than 10 micrograms/m3 and are expressed as annual averages. The education rate data shows the percentage of the population who has completed post-secondary education over the total population and the health spending data measures the final consumption of health care goods and services from both privata and public sectors and is presented in USD per capita.

Data

We will first clean and merge data for the oecd data. Each data files have different countries, so we will only use the common data and merge them as one dataframe.

In [1]:
```
import os
import random
import pandas as pd
from matplotlib import pyplot as plt
```

In [2]:
```
def clean_oecd(dir, rename):
    data = pd.read_csv(dir)
    drop = ["INDICATOR", "SUBJECT", 'MEASURE', 'FREQUENCY', 'TIME', 'Flag Codes']
    data.drop(drop, axis = 1, inplace = True)
    data = data.set_index("LOCATION")
    data.rename(columns = {"Value":rename}, inplace = True)
    return data

#clean elderly population data
eld = "C:₩₩Users₩₩lsy76₩₩Desktop₩₩20-21₩₩fall₩₩ECO225₩₩Project 1₩₩2018_elderly_popult
air = ("C:₩₩Users₩₩lsy76₩₩Desktop₩₩20-21₩₩fall₩₩ECO225₩₩Project 1₩₩2017_air_pollution_
edu = ("C:₩₩Users₩₩lsy76₩₩Desktop₩₩20-21₩₩fall₩₩ECO225₩₩Project 1₩₩2019_adult_educatio
health = ("C:₩₩Users₩₩lsy76₩₩Desktop₩₩20-21₩₩fall₩₩ECO225₩₩Project 1₩₩2019_health_spea

eld = clean_oecd(eld, "Elderly Population")
air = clean_oecd(air, "Air Pollution Exposure")
edu = clean_oecd(edu, "Adult Education")
hea = clean_oecd(health, "Health Spending")

oecd = pd.concat([eld, air], axis = 1)
oecd = pd.concat([oecd, edu], axis = 1)
oecd = pd.concat([oecd, hea], axis = 1)
oecd = oecd.dropna()
```

```
#oecd = oecd.drop(["USA"])

def change(lst1, lst2, df):
    for i in range(0, len(lst1)):
        df = df.rename(index={lst1[i]:lst2[i]})
    return df

oecd_name = ['AUS', 'AUT', 'BEL', 'CAN', 'CZE', 'DNK', 'FIN', 'FRA', 'DEU', 'GRC',
       'HUN', 'ISL', 'IRL', 'ITA', 'KOR', 'LUX', 'MEX', 'NLD','NZL', 'NOR', 'POL', 'PR
       'SVK', 'ESP', 'SWE', 'CHE', 'GBR', "USA", "COL", 'EST', 'ISR', 'SVN', "LVA", "L

covid_name = ['Australia', "Austria", "Belgium", "Canada", "Czechia", "Denmark", "Fin
              "Greece", "Hungary", "Iceland","Ireland", "Italy", "Korea, South", "Luxe
              "New Zealand", "Norway", "Poland", "Portugal", "Slovakia", "Spain", "Swe
             "Colombia", "Estonia", "Israel",  "Slovenia", "Latvia", "Lithuania"]

oecd = change(oecd_name, covid_name, oecd)
oecd
```

Out[2]:

	Elderly Population	Air Pollution Exposure	Adult Education	Health Spending
Australia	15.663	8.52437	17.132658	5187.419
Austria	18.759	12.68774	14.435863	5851.055
Belgium	18.835	13.09935	21.313864	5427.957
Canada	17.157	6.45931	8.142036	5418.383
Czechia	19.414	16.21481	6.247575	3426.039
Denmark	19.461	10.35431	18.410515	5567.899
Finland	21.613	5.90603	9.736092	4578.420
France	19.837	11.95962	19.561680	5375.699
Germany	21.466	12.08939	13.277470	6645.757
Greece	21.894	16.47663	26.043503	2383.628
Hungary	19.143	16.08462	15.026365	2222.426
Iceland	14.136	6.78283	21.476316	4811.424
Ireland	13.864	8.27177	16.342726	5275.543
Italy	22.675	16.49710	37.837608	3649.206
Korea, South	14.294	25.13889	11.270410	3384.158
Luxembourg	14.354	10.17904	24.854612	5558.322
Mexico	7.246	21.23615	60.211422	1153.581
Netherlands	19.015	12.08600	20.440771	5765.098
New Zealand	15.288	5.99181	19.302570	4203.989
Norway	17.087	7.04580	17.480188	6646.713
Poland	17.230	20.94327	7.367686	2292.146
Portugal	21.672	8.12088	47.824013	3378.627
Slovakia	15.782	17.93355	8.789214	2353.645
Spain	19.290	9.91419	38.685894	3616.459
Sweden	19.861	6.13443	16.366867	5782.292

	Elderly Population	Air Pollution Exposure	Adult Education	Health Spending
Switzerland	18.371	10.44407	10.993076	7732.412
United Kingdom	18.312	10.44482	19.939764	4653.058
US	16.026	7.36365	9.189871	11071.715
Colombia	8.060	16.91747	43.289780	1212.604
Estonia	19.676	6.84199	9.943496	2578.793
Israel	11.654	20.81130	12.920499	2932.461
Slovenia	19.670	16.25411	11.202892	3224.020
Latvia	20.181	14.10963	11.564806	1972.572
Lithuania	19.706	11.90224	6.735801	2638.133

Now, we will open the most recent covid-19 data. Also, we will drop unnecessary data entries, as well as countries. We will only save the data from countries that we have the OECD data for.

In [3]:
```
#clean csse covid data
dir = "C:₩₩Users₩₩lsy76₩₩Desktop₩₩20-21₩₩fall₩₩ECO225₩₩Project 1₩₩csse_covid_19_data₩₩
covid = pd.read_csv(dir)

drop = ["FIPS", "Admin2", "Province_State", "Last_Update",  "Combined_Key",
        "Incident_Rate", "Case_Fatality_Ratio", "Recovered", "Active", "Lat", "Long_"]
covid.drop(drop, axis = 1, inplace = True)
covid.rename(columns = {"Country_Region":"LOCATION"}, inplace = True)
covid.rename(columns = {"Confirmed":"Confirmed Case"}, inplace = True)
covid
```

Out[3]:

	LOCATION	Confirmed Case	Deaths
0	Afghanistan	46717	1797
1	Albania	39014	822
2	Algeria	84152	2447
3	Andorra	6790	76
4	Angola	15251	350
...	...	...	...
3971	West Bank and Gaza	88004	747
3972	Western Sahara	10	1
3973	Yemen	2197	619
3974	Zambia	17665	357
3975	Zimbabwe	10129	277

3976 rows × 3 columns

Now, we will 1) merge the OECD data with the COVID-19 data. 2) add a new column, mortality rate. 3) drop the column, deaths. 4) rearrange the columns.

In [4]:
```
def extract(loc, df):

    d = df[df["LOCATION"] == loc].groupby("LOCATION").sum()
```

```
    return d

d = extract("Australia", covid)


for i in covid_name:
    if i == "Australia":
        pass
    else:
        a = extract(i, covid)
        d= pd.concat([d, a])

df = pd.concat([d, oecd], axis = 1)
mortality = (df["Deaths"]/df["Confirmed Case"])*100
confirmed = df["Confirmed Case"] * 0.001
df["Confirmed Case"] = confirmed
df["Mortality Rate"] = mortality
df.drop(["Deaths"], axis = 1, inplace=True)
df = df[["Mortality Rate", "Confirmed Case", "Elderly Population", "Air Pollution Expo
         "Adult Education", "Health Spending"]]
df["Adult Education"] = 100 - df["Adult Education"]

df
```

Out[4]:

LOCATION	Mortality Rate	Confirmed Case	Elderly Population	Air Pollution Exposure	Adult Education	Health Spending
Australia	3.251800	27.923	15.663	8.52437	82.867342	5187.419
Austria	1.164668	285.489	18.759	12.68774	85.564137	5851.055
Belgium	2.898075	579.212	18.835	13.09935	78.686136	5427.957
Canada	3.159524	387.052	17.157	6.45931	91.857964	5418.383
Czechia	1.590807	528.474	19.414	16.21481	93.752425	3426.039
Denmark	1.025828	82.470	19.461	10.35431	81.589485	5567.899
Finland	1.567041	25.462	21.613	5.90603	90.263908	4578.420
France	2.321364	2275.429	19.837	11.95962	80.438320	5375.699
Germany	1.569137	1094.678	21.466	12.08939	86.722530	6645.757
Greece	2.342049	107.470	21.894	16.47663	73.956497	2383.628
Hungary	2.251293	221.073	19.143	16.08462	84.973635	2222.426
Iceland	0.498799	5.413	14.136	6.78283	78.523684	4811.424
Ireland	2.842111	72.798	13.864	8.27177	83.657274	5275.543
Italy	3.477140	1620.901	22.675	16.49710	62.162392	3649.206
Korea, South	1.495891	35.163	14.294	25.13889	88.729590	3384.158
Luxembourg	0.939395	35.129	14.354	10.17904	75.145388	5558.322
Mexico	9.512528	1122.362	7.246	21.23615	39.788578	1153.581
Netherlands	1.775319	536.129	19.015	12.08600	79.559229	5765.098

LOCATION	Mortality Rate	Confirmed Case	Elderly Population	Air Pollution Exposure	Adult Education	Health Spending
New Zealand	1.213592	2.060	15.288	5.99181	80.697430	4203.989
Norway	0.912793	36.591	17.087	7.04580	82.519812	6646.713
Poland	1.760034	999.924	17.230	20.94327	92.632314	2292.146
Portugal	1.523321	300.462	21.672	8.12088	52.175987	3378.627
Slovakia	0.809830	107.183	15.782	17.93355	91.210786	2353.645
Spain	2.747512	1656.444	19.290	9.91419	61.314106	3616.459
Sweden	2.607015	260.758	19.861	6.13443	83.633133	5782.292
Switzerland	1.493015	330.874	18.371	10.44407	89.006924	7732.412
United Kingdom	3.590755	1647.230	18.312	10.44482	80.060236	4653.058
US	1.972422	13721.304	16.026	7.36365	90.810129	11071.715
Colombia	2.787909	1324.792	8.060	16.91747	56.710220	1212.604
Estonia	0.968232	12.497	19.676	6.84199	90.056504	2578.793
Israel	0.850864	338.127	11.654	20.81130	87.079501	2932.461
Slovenia	1.931678	77.135	19.670	16.25411	88.797108	3224.020
Latvia	1.187313	17.687	20.181	14.10963	88.435194	1972.572
Lithuania	0.830201	62.515	19.706	11.90224	93.264199	2638.133

In [5]:
```
df.describe()
```

Out[5]:

	Mortality Rate	Confirmed Case	Elderly Population	Air Pollution Exposure	Adult Education	Health Spending
count	34.000000	34.000000	34.000000	34.000000	34.000000	34.000000
mean	2.084390	880.535588	17.549765	12.388858	80.783591	4352.107441
std	1.568783	2348.453659	3.634771	5.114503	12.624792	2038.564702
min	0.498799	2.060000	7.246000	5.906030	39.788578	1153.581000
25%	1.170329	43.072000	15.692750	8.158603	78.904409	2711.715000
50%	1.675420	273.123500	18.797000	11.930930	83.645204	4391.204500
75%	2.712388	894.746000	19.698500	16.244285	88.954470	5525.730750
max	9.512528	13721.304000	22.675000	25.138890	93.752425	11071.715000

This table shows that among 34 OECD countries, the average rate of the mortality rate is 2.4% and the median is 1.79%.

In [6]:
```
df.corr()
```

Out[6]:

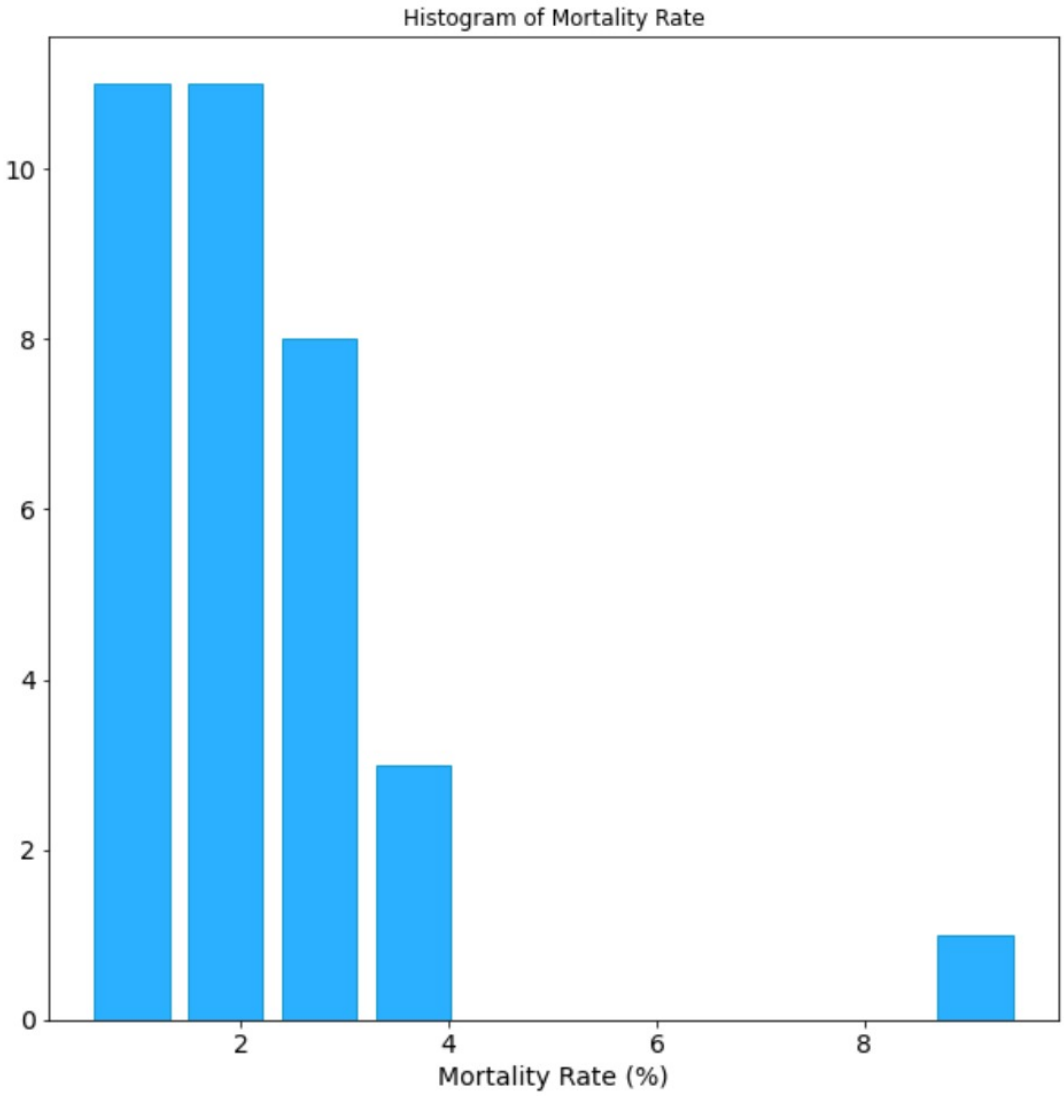
	Mortality Rate	Confirmed Case	Elderly Population	Air Pollution Exposure	Adult Education	Health Spending

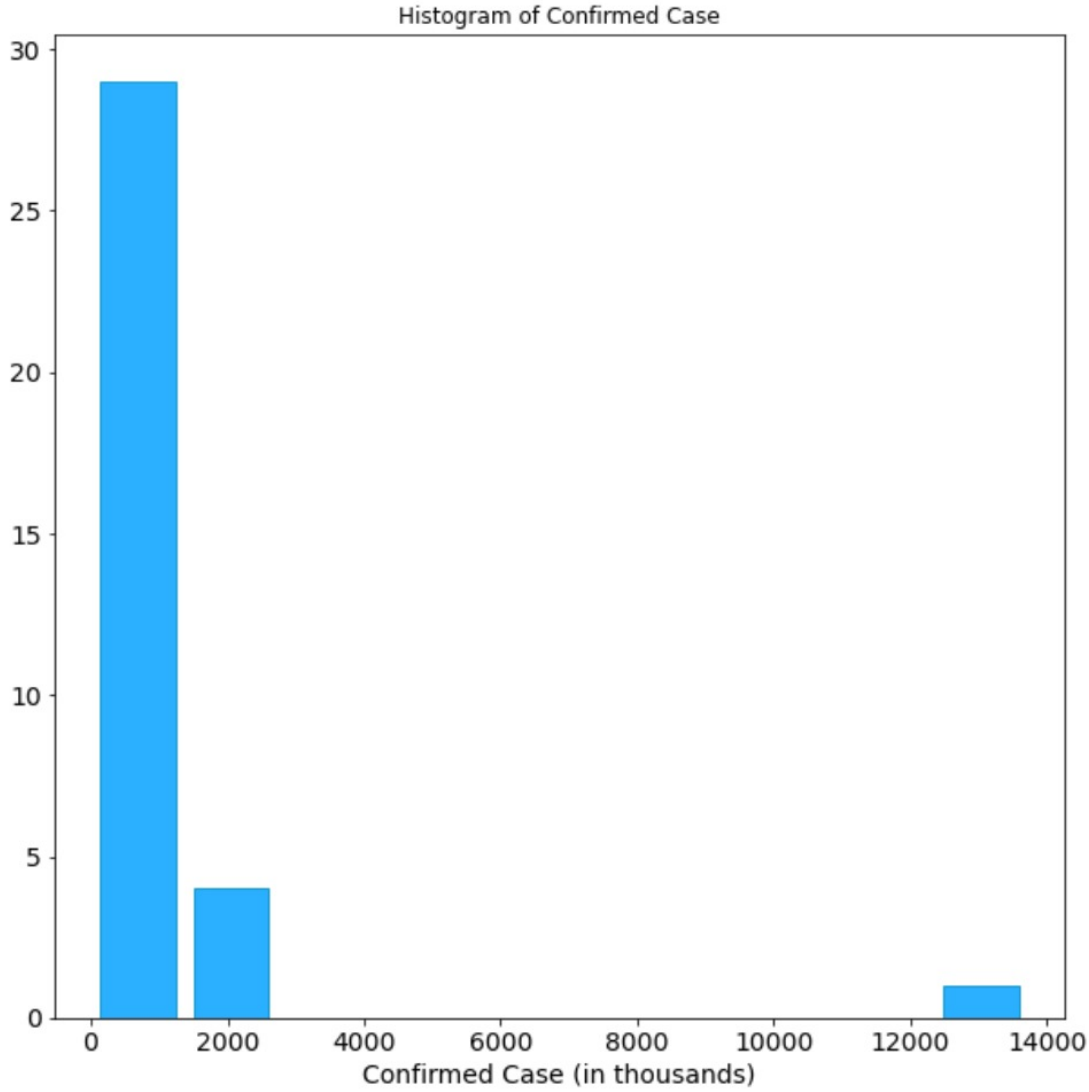
	Mortality Rate	Confirmed Case	Elderly Population	Air Pollution Exposure	Adult Education	Health Spending
Mortality Rate	1.000000	0.102445	-0.376780	0.237223	-0.628824	-0.208491
Confirmed Case	0.102445	1.000000	-0.070733	-0.115027	0.025460	0.550410
Elderly Population	-0.376780	-0.070733	1.000000	-0.295892	0.310982	0.180489
Air Pollution Exposure	0.237223	-0.115027	-0.295892	1.000000	-0.120592	-0.563072
Adult Education	-0.628824	0.025460	0.310982	-0.120592	1.000000	0.304335
Health Spending	-0.208491	0.550410	0.180489	-0.563072	0.304335	1.000000

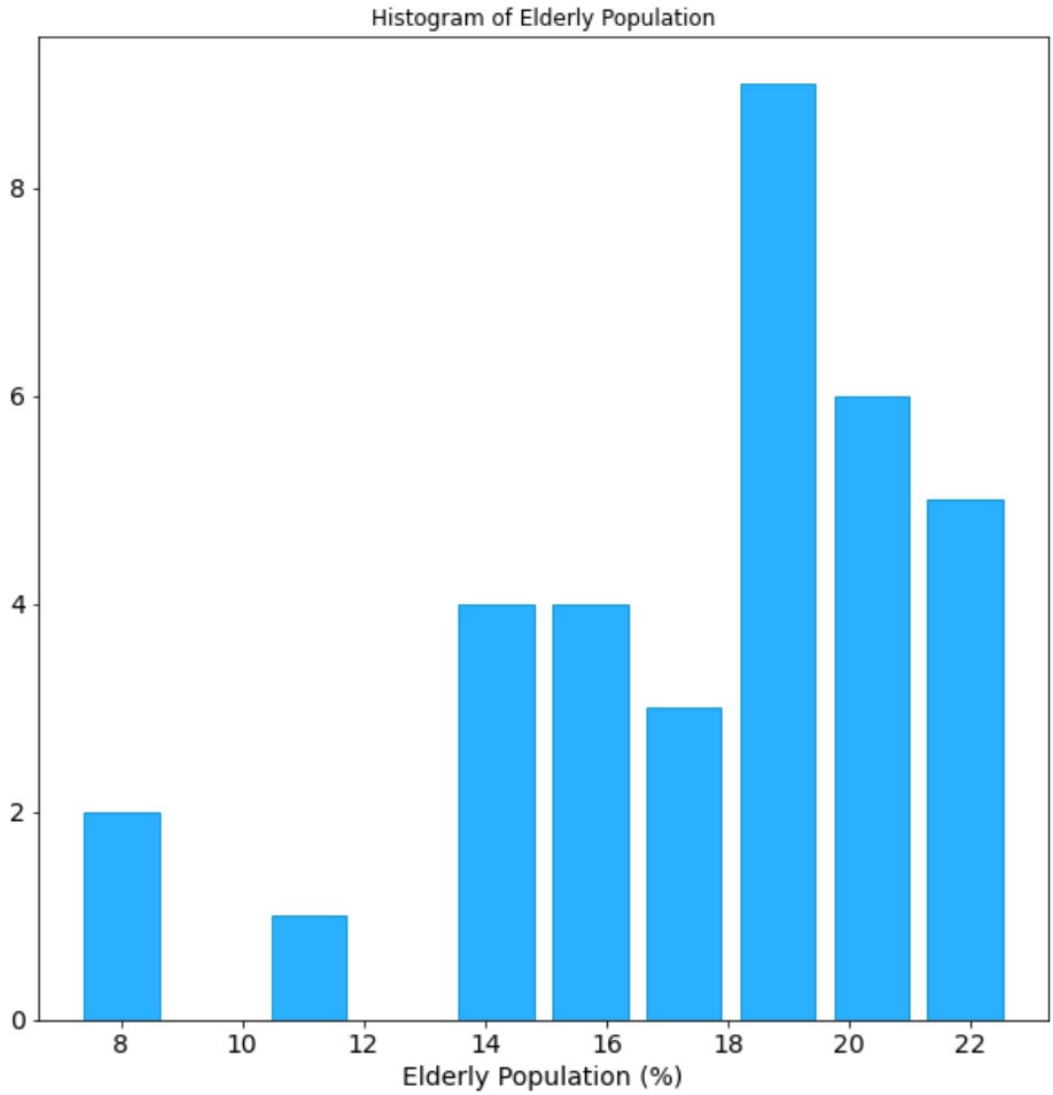
This table show the correlation between the mortality variable and all other covariates and the correlation between each covariates. It shows that as the number of confirmed case, or air pollution exposure rate increases, the mortality rate increases the mortality rate increases. In contrast, as the rate of elderly population, aldult education or the amount of health spending increased, the mortality rate decreases.

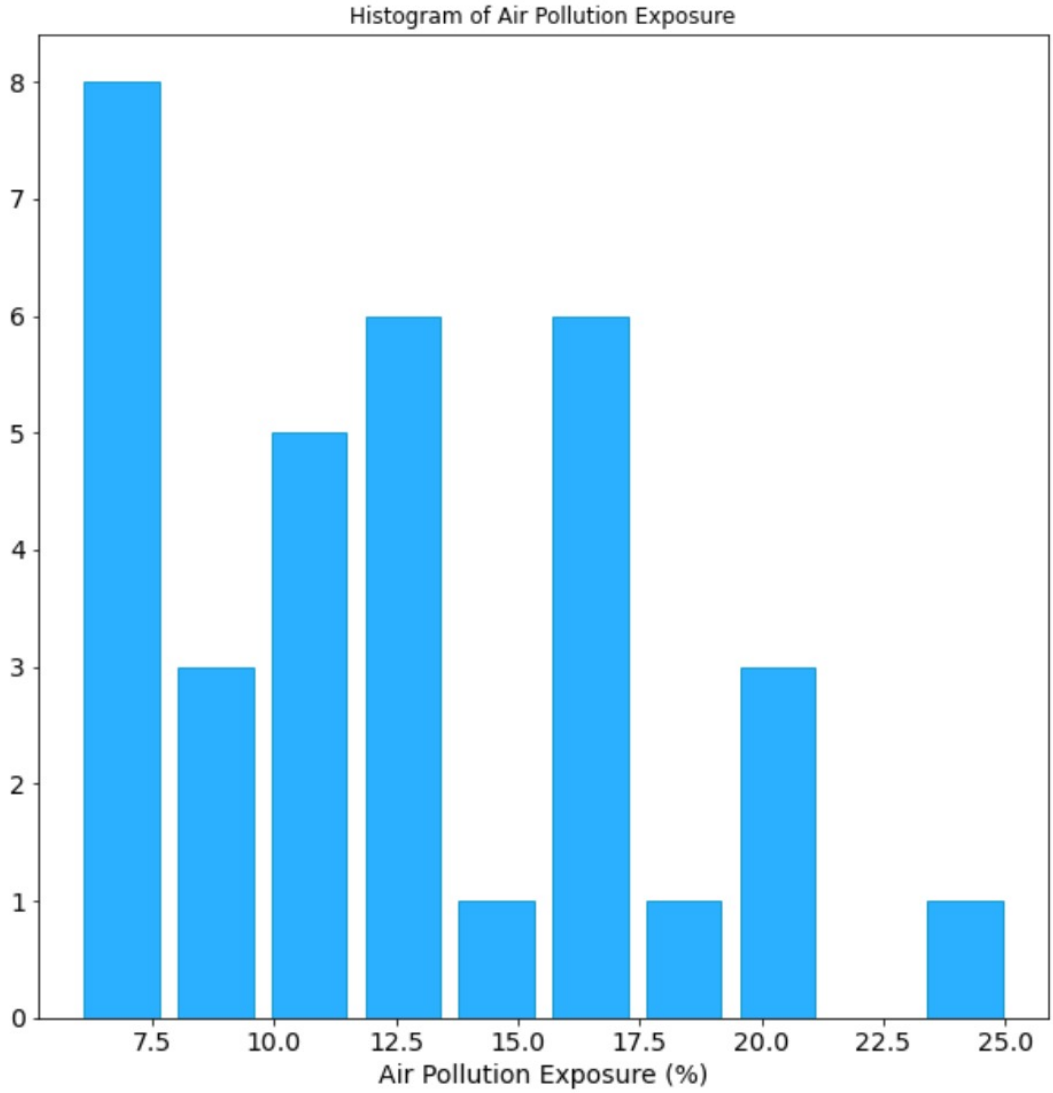
Graphical Figures

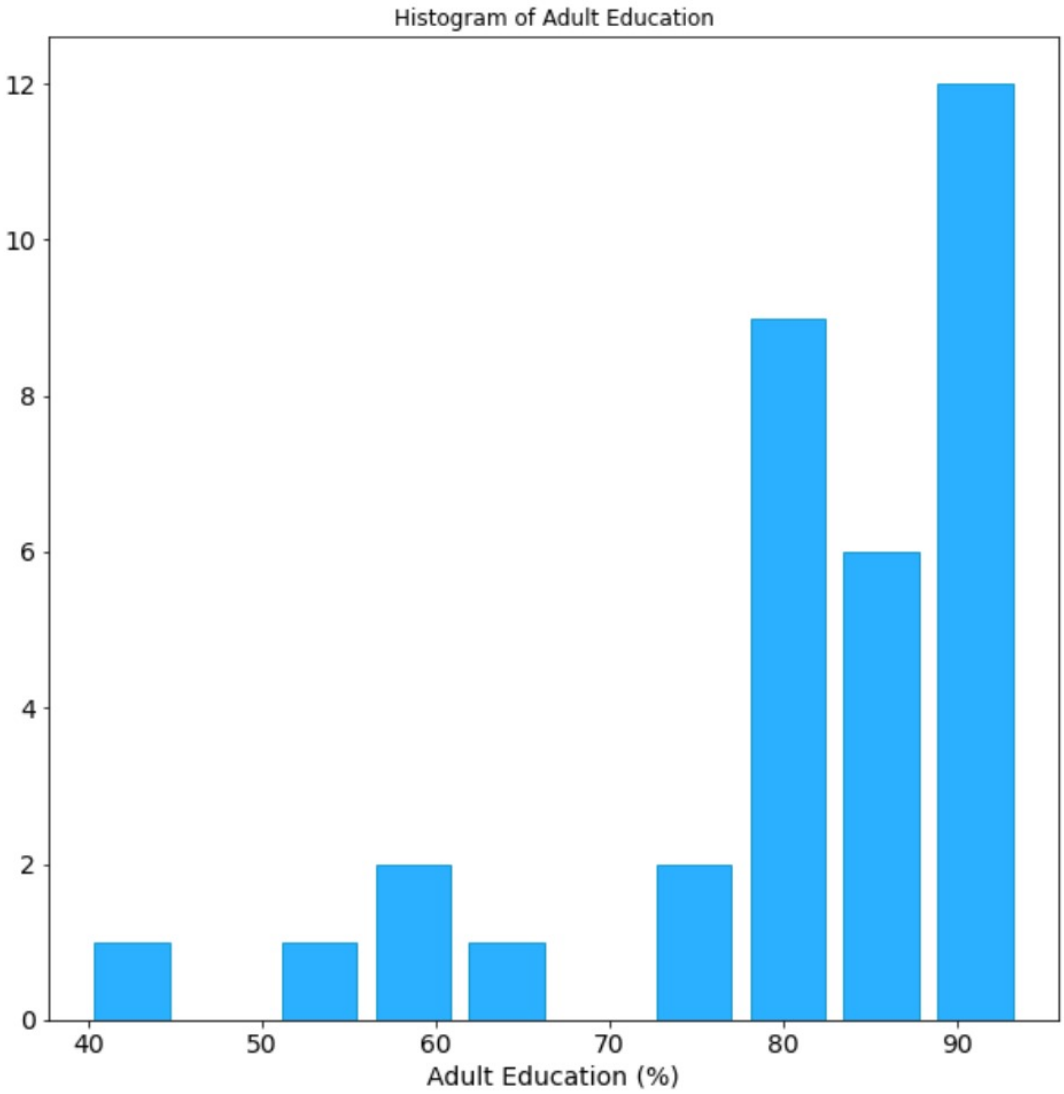
In [7]:
```
lst = ["Mortality Rate", "Confirmed Case", "Elderly Population", "Air Pollution Exposu
       "Adult Education", "Health Spending"]

def get_histogram(data, name):

    fig, ax = plt.subplots(figsize = (10,10))

    if name == "Confirmed Case":
        ax= plt.hist(data[name], rwidth = 0.8, facecolor = '#2ab0ff', edgecolor='#169
        plt.xlabel(name + " (in thousands)", fontsize = 14)

    elif name == "Health Spending":

        ax = plt.hist(data[name], rwidth = 0.8, facecolor = '#2ab0ff', edgecolor='#16
        plt.xlabel(name + " (USD per Capita)", fontsize = 14)

    else:
        ax = plt.hist(data[name], rwidth = 0.8, facecolor = '#2ab0ff', edgecolor='#16
        plt.xlabel(name + " (%)", fontsize = 14)

    plt.xticks(fontsize = 14)
    plt.yticks(fontsize = 14)

    plt.title("Histogram of " + name)
    #plt.style.use('seaborn-whitegrid')


    plt.show()

for i in lst:
    get_histogram(df, i)
```

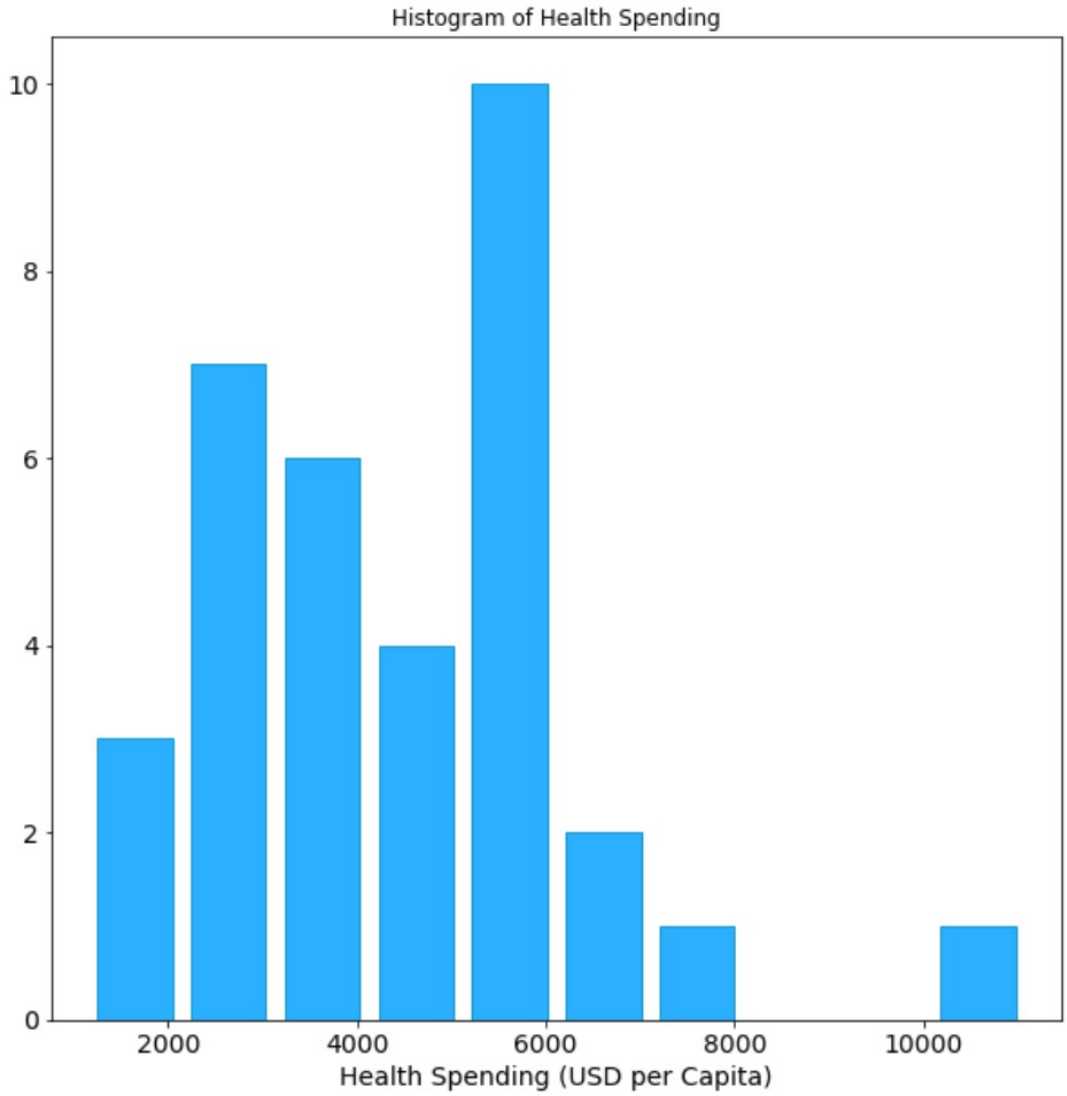

Histogram of Mortality Rate
10
8
6
4
2
0
2
4
6
8
Mortality Rate (%)


Histogram of Confirmed Case
30
25
20
15
10
5
0
0
2000
4000
6000
8000
10000
12000
14000
Confirmed Case (in thousands)

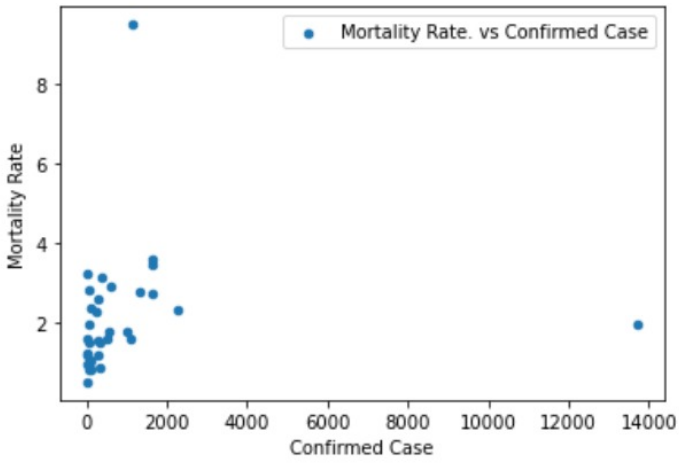

Histogram of Elderly Population
0
2
4
6
8
8
10
12
14
16
18
20
22
Elderly Population (%)

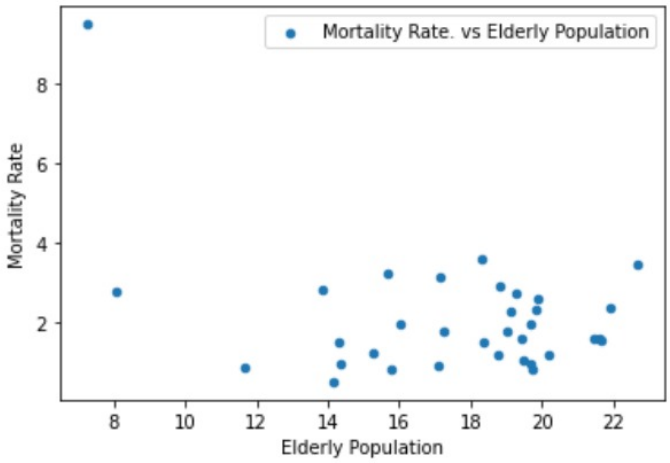

Histogram of Air Pollution Exposure
8
7
6
5
4
3
2
1
0
7.5
10.0
12.5
15.0
17.5
20.0
22.5
25.0
Air Pollution Exposure (%)

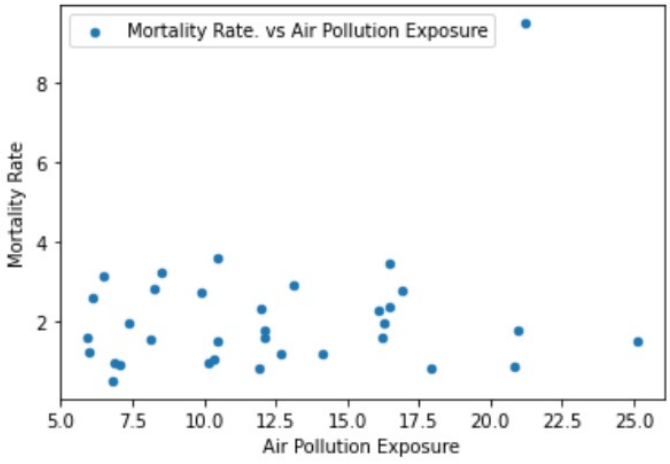

Histogram of Adult Education
12
10
8
6
4
2
0
40
50
60
70
80
90
Adult Education (%)

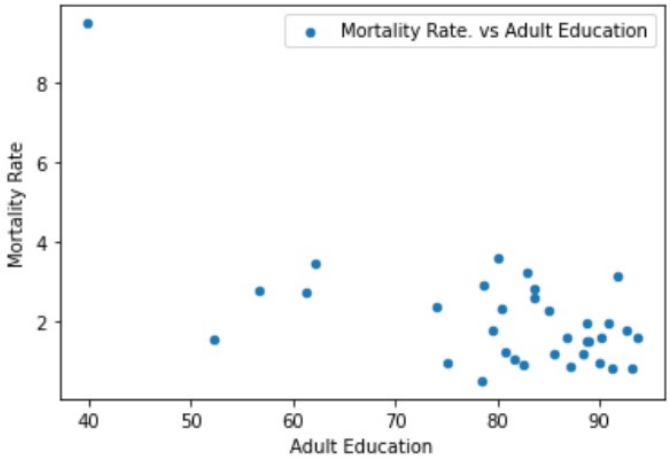


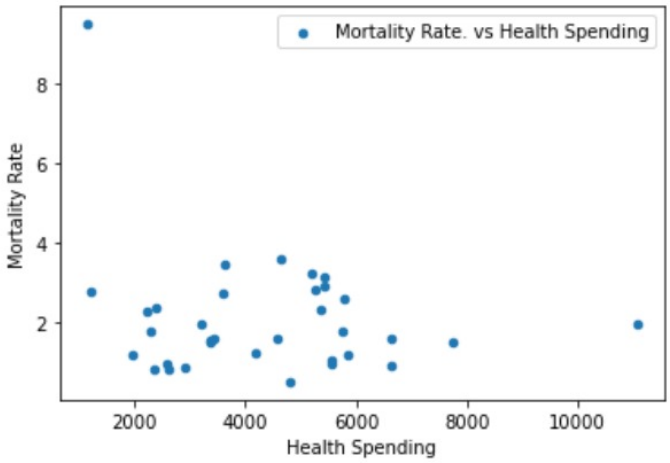
In [8]:
```
lst = ["Confirmed Case", "Elderly Population", "Air Pollution Exposure",
       "Adult Education", "Health Spending"]
for i in lst:
    df.plot.scatter(i, "Mortality Rate" , label = "Mortality Rate. vs " + i )
```




Mortality Rate. vs Elderly Population
Mortality Rate
Elderly Population
8
6
4
2
8
10
12
14
16
18
20
22


Mortality Rate. vs Air Pollution Exposure
Mortality Rate
Air Pollution Exposure
8
6
4
2
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5
25.0


Mortality Rate. vs Adult Education
Mortality Rate
Adult Education
8
6
4
2
40
50
60
70
80
90



Project 2: The Effects of the Number of Confirmed Case, Elderly Population, Air Pollution Exposure, Adult Education Rate and Health Spending on the Mortality Rate.

Introduction

The COVID - 19 Data is given by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University and all the other data are from OECD statistics. The world has been suffered from the pandemic for more than 3/4 of the year and resulted in a tremendous number of casualities, however, how much a country suffer from covid-19 is quite different. Therefore, in this project, we will look closely at factors, such as the number of confirmed case, the elderly population, air pollution exposure, adult education rate and health spending of a nation and examine if these factors, predictor variables, in other words, have any impact on the mortality rate, the response variable.

The elderly population data shows the share of the total elderly population (aged 60+) expressed as a ratio of the total population. The air pollution exposure data refer to population exposure to more than 10 micrograms/m3 and are expressed as annual averages. The education rate data shows the percentage of the population who has completed post-secondary education over the total population and the health spending data measures the final consumption of health care goods and services from both privata and public sectors and is presented in USD per capita.

In [9]:
```
import matplotlib.colors as mplc
import matplotlib.patches as patches
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import statsmodels.formula.api as sm #for linear regression: sm.ols
import geopandas as gpd
import seaborn as sns
import geoplot as gplt

from pandas_datareader import DataReader
```

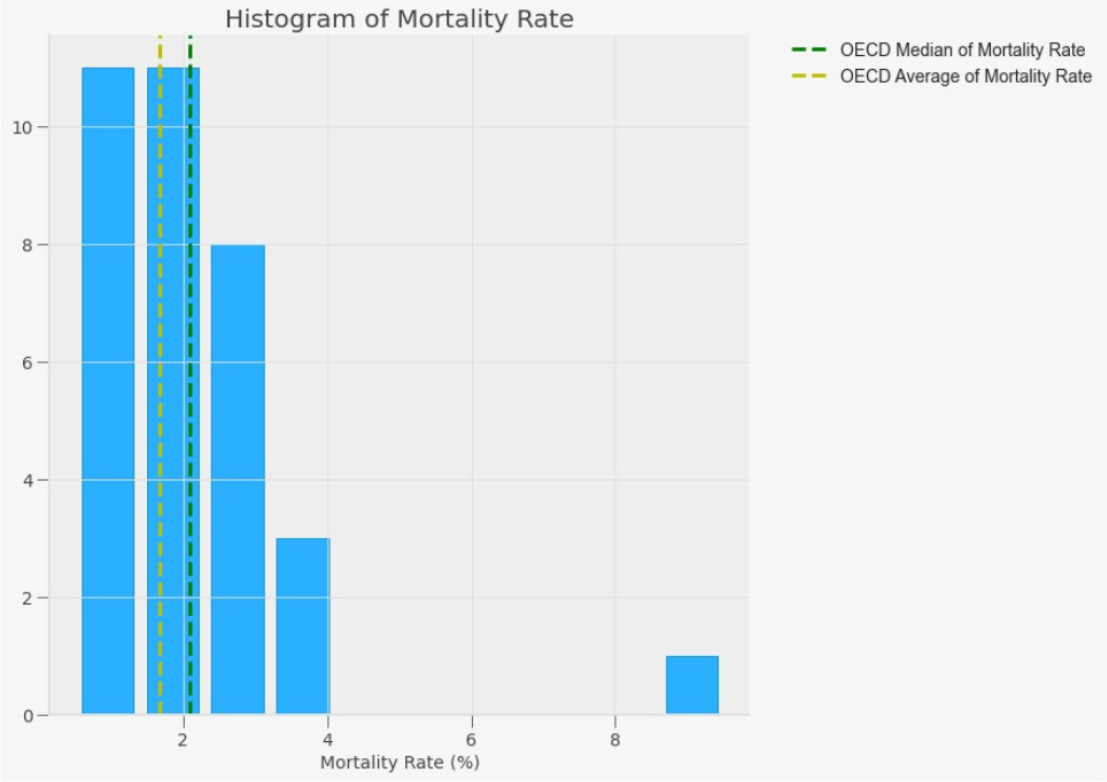
```
%matplotlib inline
# activate plot theme
import qeds
qeds.themes.mpl_style();

from shapely.geometry import Point
```

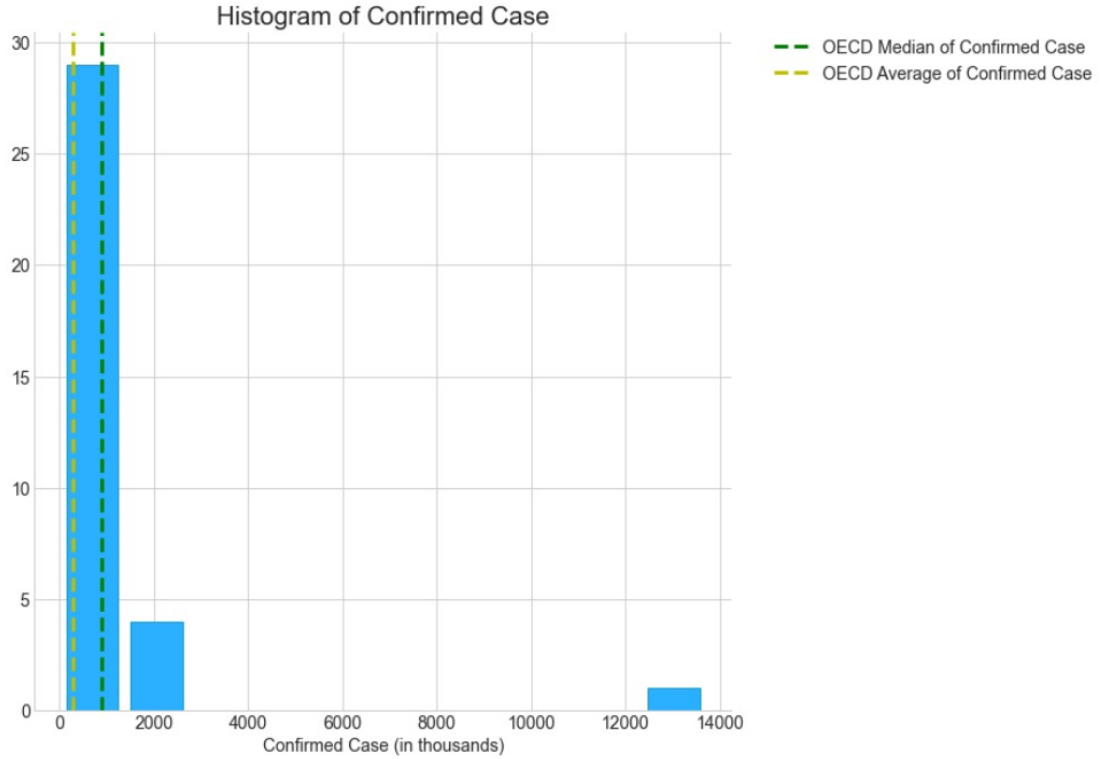
Part 2

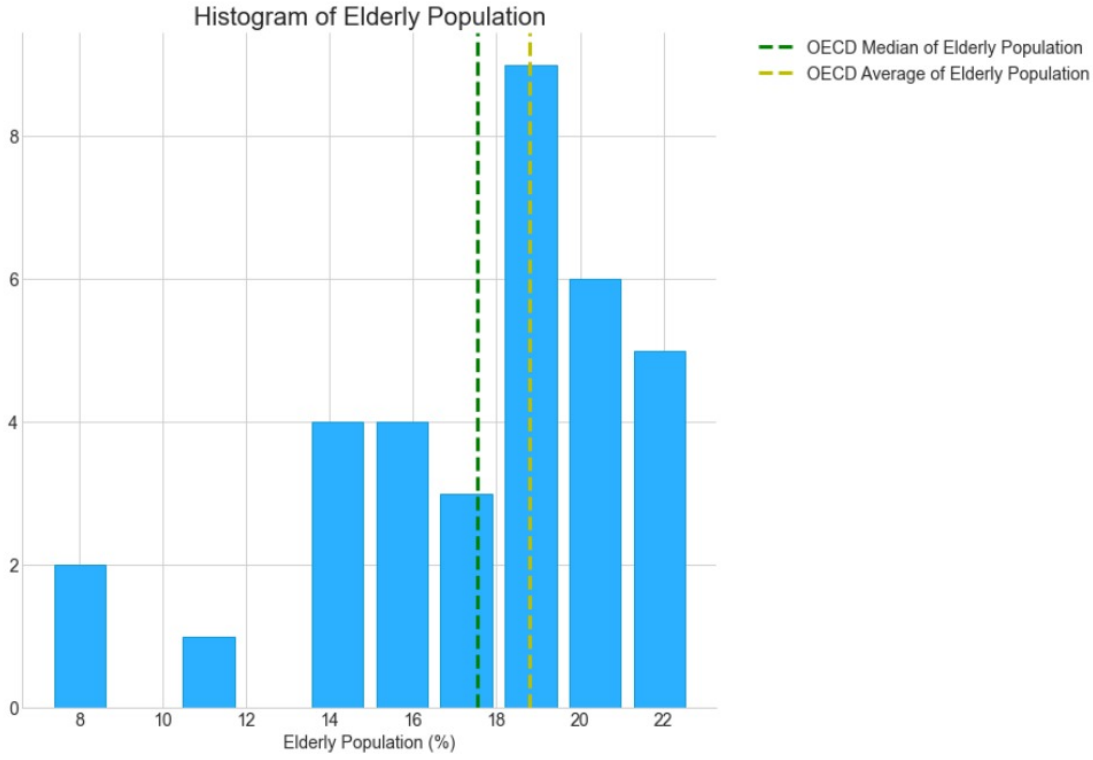
Graphical Figures

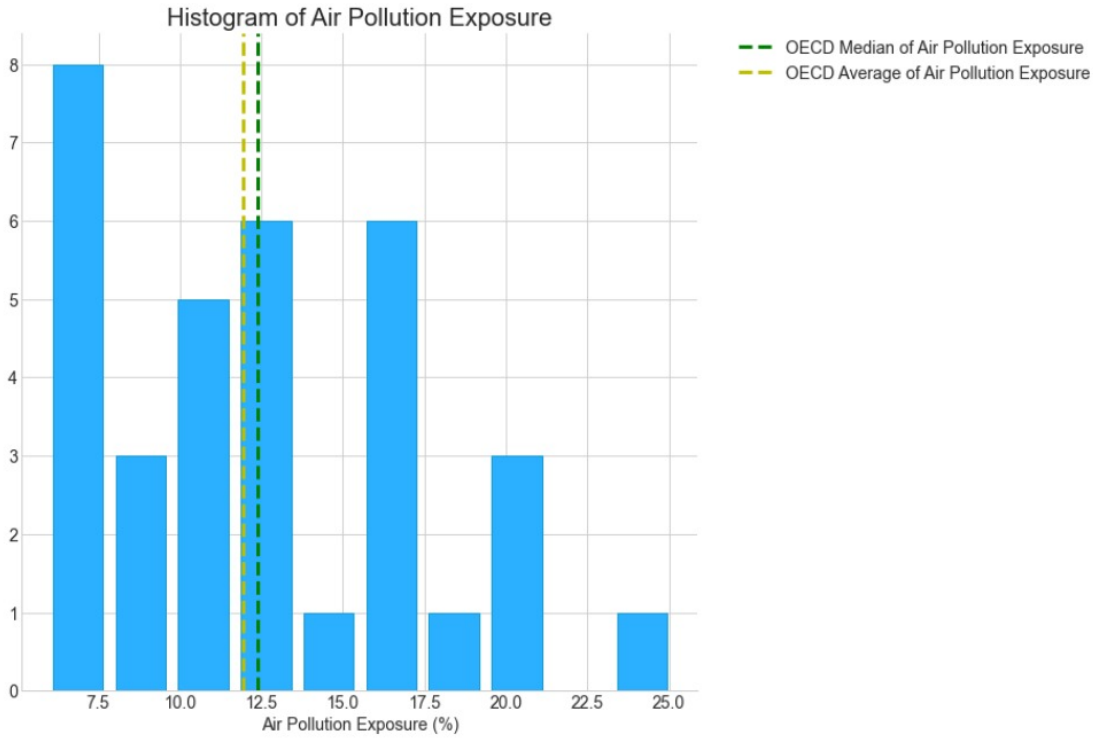
Histogram

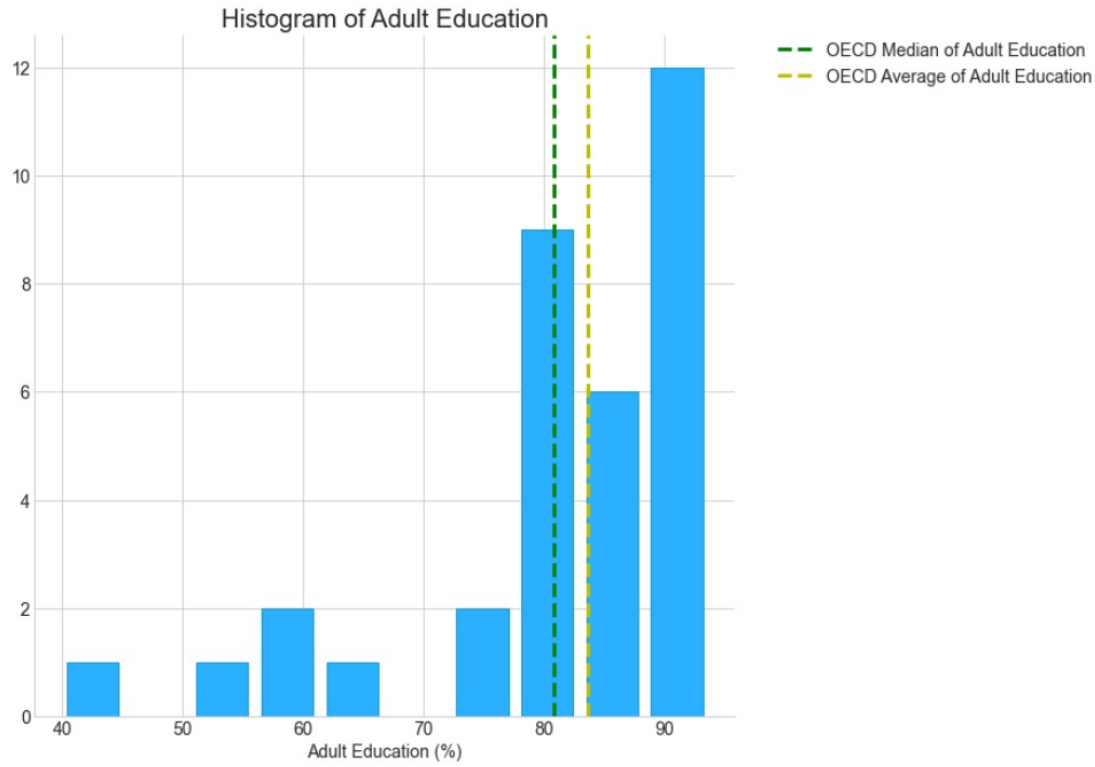
In [10]:
```
lst = ["Mortality Rate", "Confirmed Case", "Elderly Population", "Air Pollution Exposu
       "Adult Education", "Health Spending"]

def get_histogram(data, name):

    fig, ax = plt.subplots(figsize = (10,10))

    if name == "Confirmed Case":
        plt.hist(data[name], rwidth = 0.8, facecolor = '#2ab0ff', edgecolor='#169acf'
        plt.xlabel(name + " (in thousands)", fontsize = 14)

    elif name == "Health Spending":

        plt.hist(data[name], rwidth = 0.8, facecolor = '#2ab0ff', edgecolor='#169acf'
        plt.xlabel(name + " (USD per Capita)", fontsize = 14)

    else:

        plt.hist(data[name], rwidth = 0.8, facecolor = '#2ab0ff', edgecolor='#169acf'
        plt.xlabel(name + " (%)", fontsize = 14)

    plt.xticks(fontsize = 14)
    plt.yticks(fontsize = 14)

    plt.title("Histogram of " + name)
    plt.style.use('seaborn-whitegrid')

    p = plt.axvline(data[name].mean(), color='g', linestyle='dashed', linewidth=3)
    m = plt.axvline(data[name].median(), color='y', linestyle='dashed', linewidth=3)
    ax.legend((p,m), ("OECD Median of " + name, "OECD Average of " + name), bbox_to_a


    plt.show()

for i in lst:
    get_histogram(df, i)
```

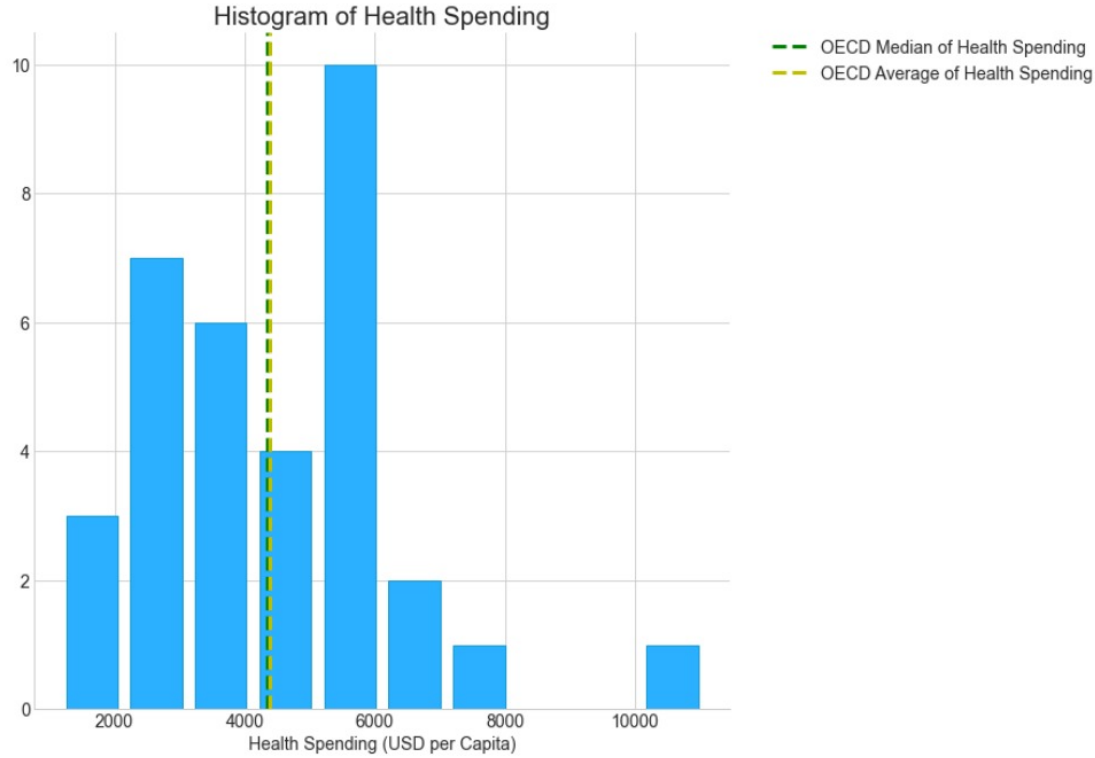

Histogram of Mortality Rate
OECD Median of Mortality Rate
OECD Average of Mortality Rate
0
2
4
6
8
10
2
4
6
8
Mortality Rate (%)


Histogram of Confirmed Case
OECD Median of Confirmed Case
OECD Average of Confirmed Case
0
5
10
15
20
25
30
0
2000
4000
6000
8000
10000
12000
14000
Confirmed Case (in thousands)


Histogram of Elderly Population
OECD Median of Elderly Population
OECD Average of Elderly Population
0
2
4
6
8
8
10
12
14
16
18
20
22
Elderly Population (%)


Histogram of Air Pollution Exposure
OECD Median of Air Pollution Exposure
OECD Average of Air Pollution Exposure
0
1
2
3
4
5
6
7
8
7.5
10.0
12.5
15.0
17.5
20.0
22.5
25.0
Air Pollution Exposure (%)





Above histograms is revised version of the histograms from the project 1, and show frequency distributions of mortality rate, number of confirmed case, elderly population, air pollution exposure rate, adult education rate, and the amount of health spending. All of these histograms are skewed, either positively or negatively, showing either right or left tail is longer, and the mean and median of the data are not likely to be the same. To display the mean and median of the data to help the interpretation, the green and yellow dashed lines are included in the histogram.

Histogram of Mortality Rate

From the histogram, we can find that the skewness of the distribution of mortality rate is positive, since it is skewed right. The green and yellow dashed lines show that the average mortality rate is around 1-2% and the median rate is around 2-3%. The interesting thing that the histogram shows is that even though most of countries share similar mortality rate, there is an extreme case, nearly 10%. Despite the similar rates that the majority of the countries share, since rates vary by countries, one can question why they vary and so, this histogram represents the purpose of the project, "what influences mortality rate?".

Histogram of Confirmed Case

This histogram is also skewed right. However, the distribution looks somewhat ugly, because there is a huge break between the bars. In fact, it is cause by the data from the US, because they have an extraordinary number of confirmed case. Due to the abnormal data, there exist a huge deviation and so, the average number of confirmed case is not very useful, so the median can better represent the data. the median number of confirmed case is around 1,000,000.

Histogram of Elderly Population

The shape of the histogram is negative, i.e, it is skewed left. The mean of rate of elderly population among OECD countries is around 18&-19% and the median is around 17%-18%.

Histogram of Air Pollution Exposure

The shape of the histogram is positive and the median and mean value of the percentage of air pollution exposure of citizens from OECD countries are similar as they are both around 12.5%.

Histogram of Adult Education

The shape is left skewed and the mean and median values of the rate of adult population who has completed post-secondary education are similar as they are both in between 80%-85%.

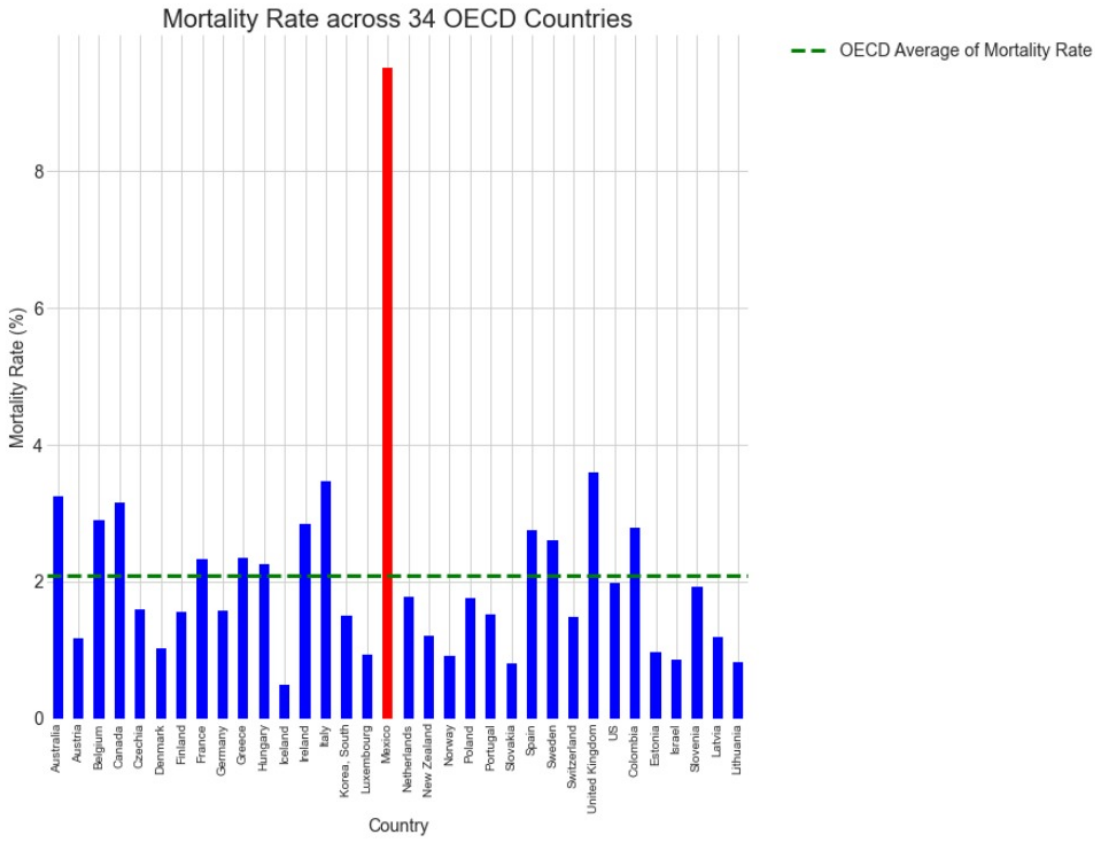
Histogram of Health Spending

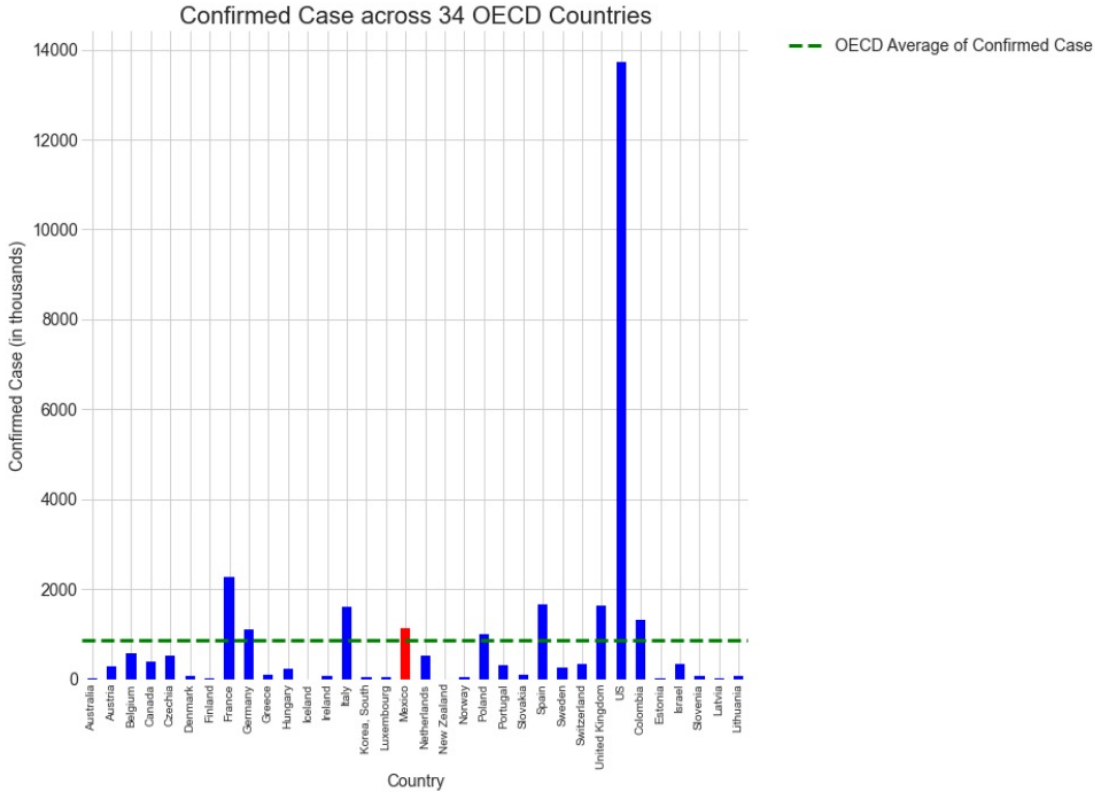
The shape is positively skewed and the median and mean value of the annula health spending of among OECD countries are very similar around 4000-4500 USD per capital.

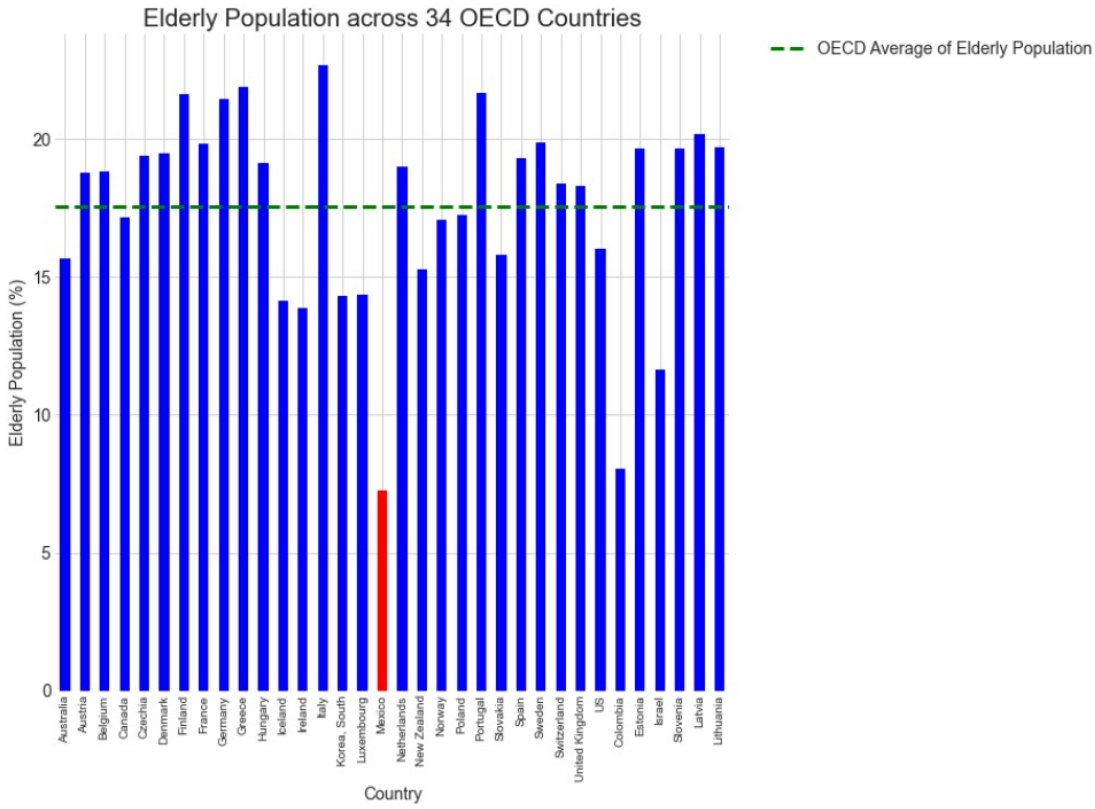
Bar Chart

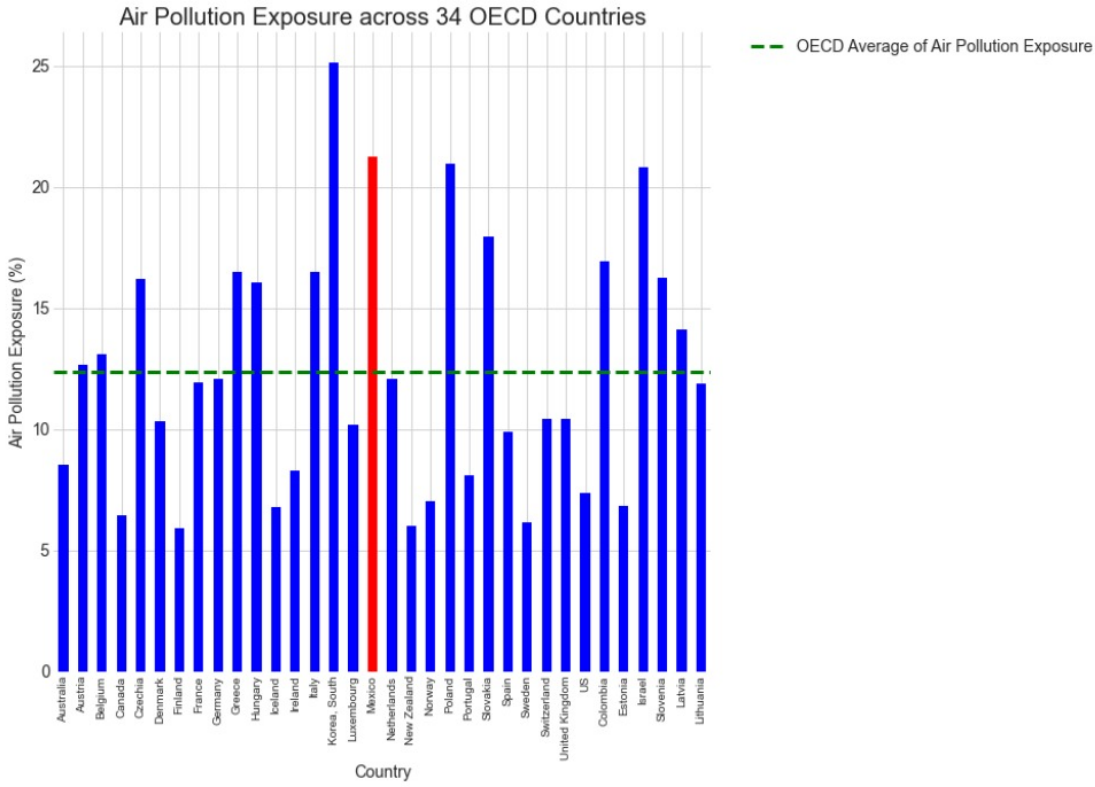
In [11]:
```
lst = ["Mortality Rate", "Confirmed Case", "Elderly Population", "Air Pollution Exposu
       "Adult Education", "Health Spending"]

def get_bar_chart(data, name):
    d = data[name]

    fig, ax = plt.subplots(figsize = (10,10))

    for side in ["right", "top", "left", "bottom"]:
        ax.spines[side].set_visible(False)

    lst = []

    for i in data.iterrows():

        if i[0] == "Mexico":
```

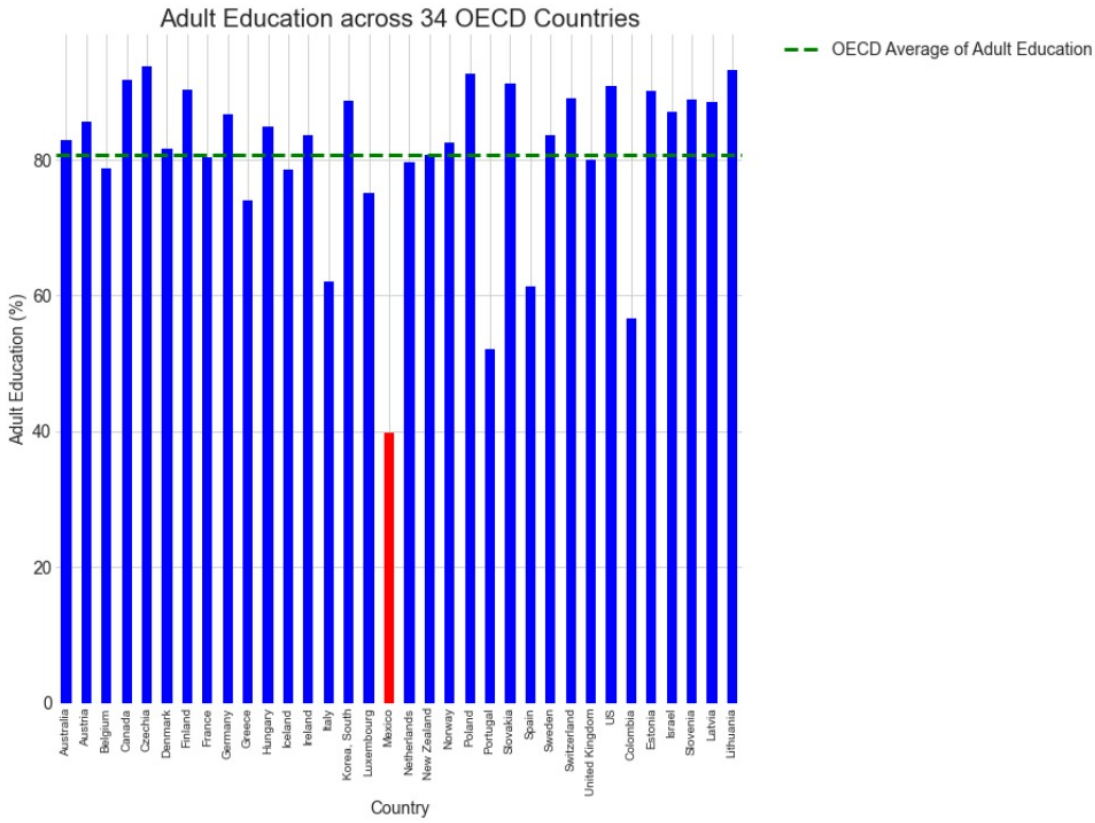
```
            lst.append("r")

        else:
            lst.append("b")

    d.plot(kind="bar", ax=ax, color=lst)

    ax.spines['right'].set_visible(False)
    ax.spines['top'].set_visible(False)
    ax.set_title(name + " across 34 OECD Countries")
    ax.set_xlabel("Country", fontsize = 14)

    plt.tick_params(axis='x', labelsize=10 )

    ax.set_facecolor('white')
    fig.set_facecolor('white')
    #ax.grid(False)


    if name == "Confirmed Case":
        ax.set_ylabel(name + " (in thousands)", fontsize = 14)


    elif name == "Health Spending":
        ax.set_ylabel(name + " (USD per Capita)", fontsize = 14)

    else:
        ax.set_ylabel(name + " (%)", fontsize = 14)

    p = plt.axhline(data[name].mean(), color='g', linestyle='dashed', linewidth=3)
    ax.legend([p], ["OECD Average of " + name], bbox_to_anchor=(1.05, 1), loc=2, bord




for i in lst:
    get_bar_chart(df, i)
```

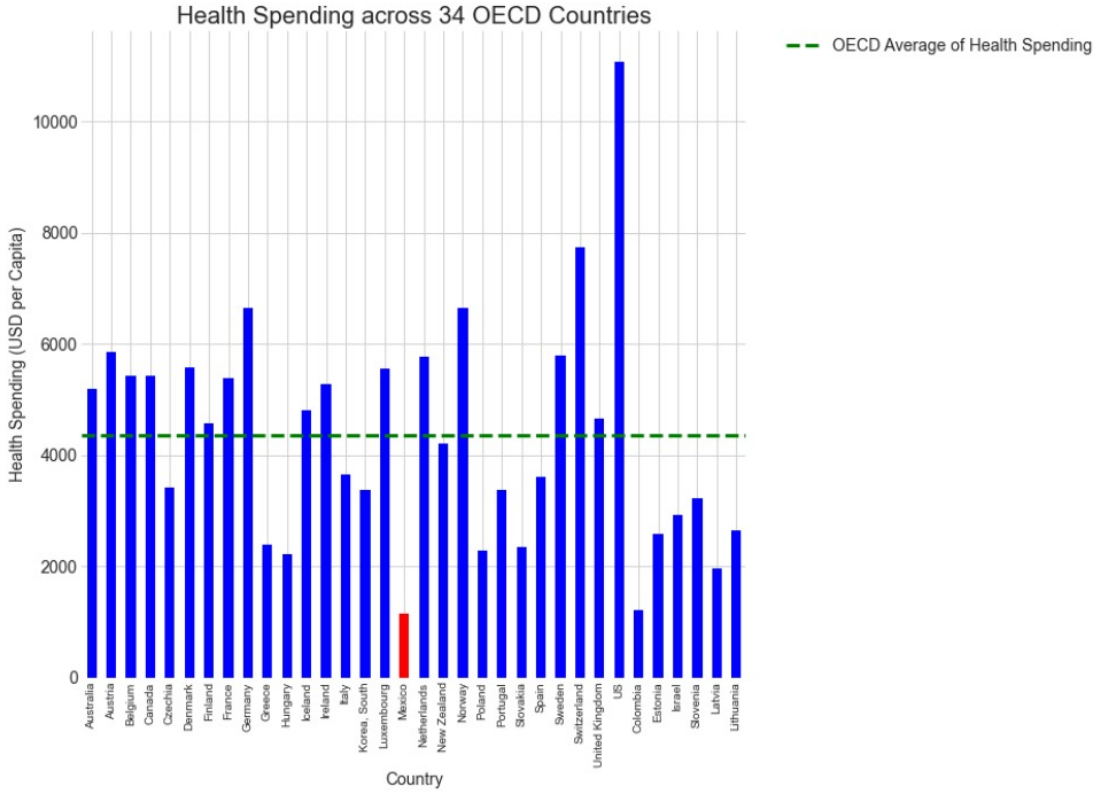

Mortality Rate across 34 OECD Countries
OECD Average of Mortality Rate
Mortality Rate (%)
0
2
4
6
8
Australia
Austria
Belgium
Canada
Czechia
Denmark
Finland
France
Germany
Greece
Hungary
Iceland
Ireland
Italy
Korea, South
Luxembourg
Mexico
Netherlands
New Zealand
Norway
Poland
Portugal
Slovakia
Spain
Sweden
Switzerland
United Kingdom
US
Colombia
Estonia
Israel
Slovenia
Latvia
Lithuania
Country


Confirmed Case across 34 OECD Countries
OECD Average of Confirmed Case
Confirmed Case (in thousands)
0
2000
4000
6000
8000
10000
12000
14000
Australia
Austria
Belgium
Canada
Czechia
Denmark
Finland
France
Germany
Greece
Hungary
Iceland
Ireland
Italy
Korea, South
Luxembourg
Mexico
Netherlands
New Zealand
Norway
Poland
Portugal
Slovakia
Spain
Sweden
Switzerland
United Kingdom
US
Colombia
Estonia
Israel
Slovenia
Latvia
Lithuania
Country


Elderly Population across 34 OECD Countries
OECD Average of Elderly Population
Elderly Population (%)
0
5
10
15
20
Australia
Austria
Belgium
Canada
Czechia
Denmark
Finland
France
Germany
Greece
Hungary
Iceland
Ireland
Italy
Korea, South
Luxembourg
Mexico
Netherlands
New Zealand
Norway
Poland
Portugal
Slovakia
Spain
Sweden
Switzerland
United Kingdom
US
Colombia
Estonia
Israel
Slovenia
Latvia
Lithuania
Country


Air Pollution Exposure across 34 OECD Countries
OECD Average of Air Pollution Exposure
Air Pollution Exposure (%)
0
5
10
15
20
25
Australia
Austria
Belgium
Canada
Czechia
Denmark
Finland
France
Germany
Greece
Hungary
Iceland
Ireland
Italy
Korea, South
Luxembourg
Mexico
Netherlands
New Zealand
Norway
Poland
Portugal
Slovakia
Spain
Sweden
Switzerland
United Kingdom
US
Colombia
Estonia
Israel
Slovenia
Latvia
Lithuania
Country

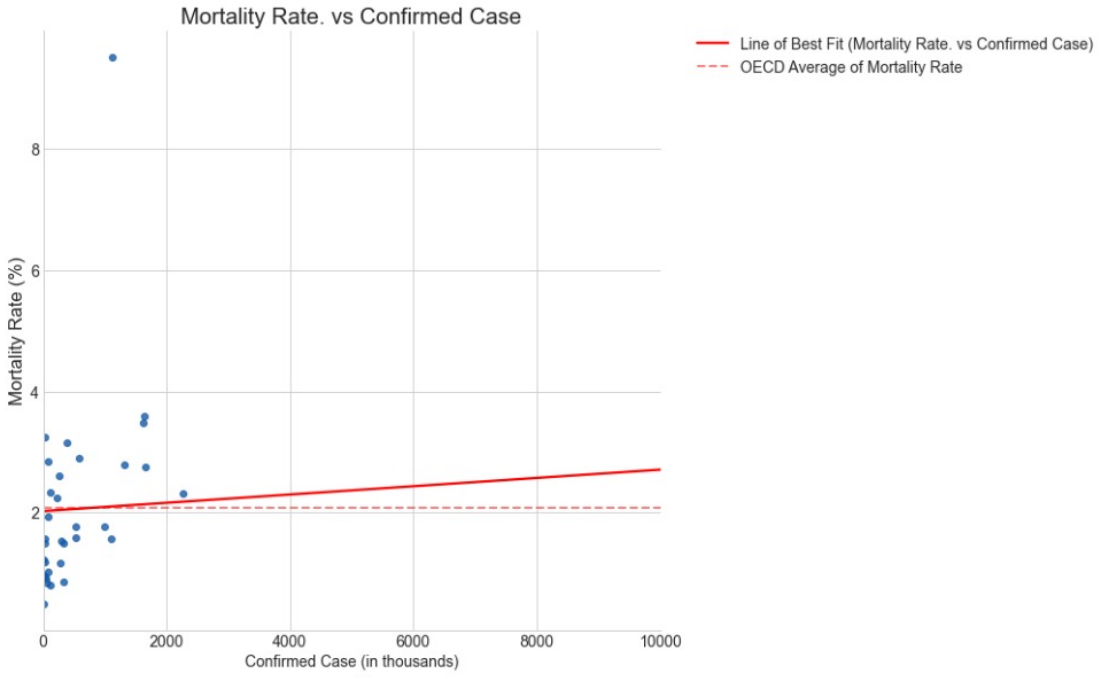


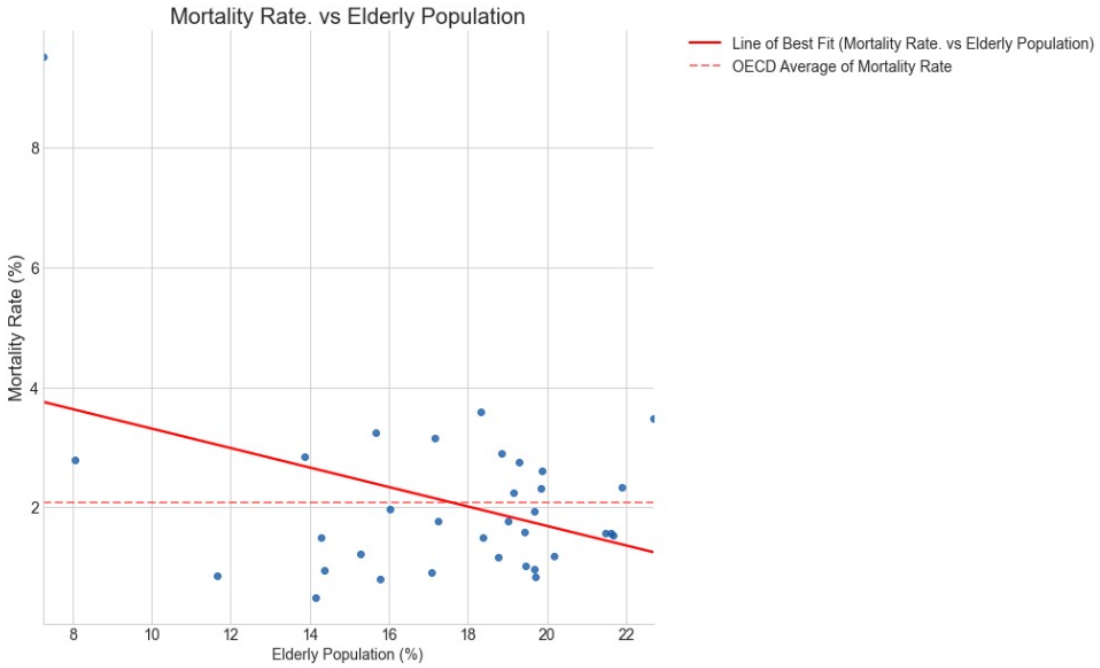


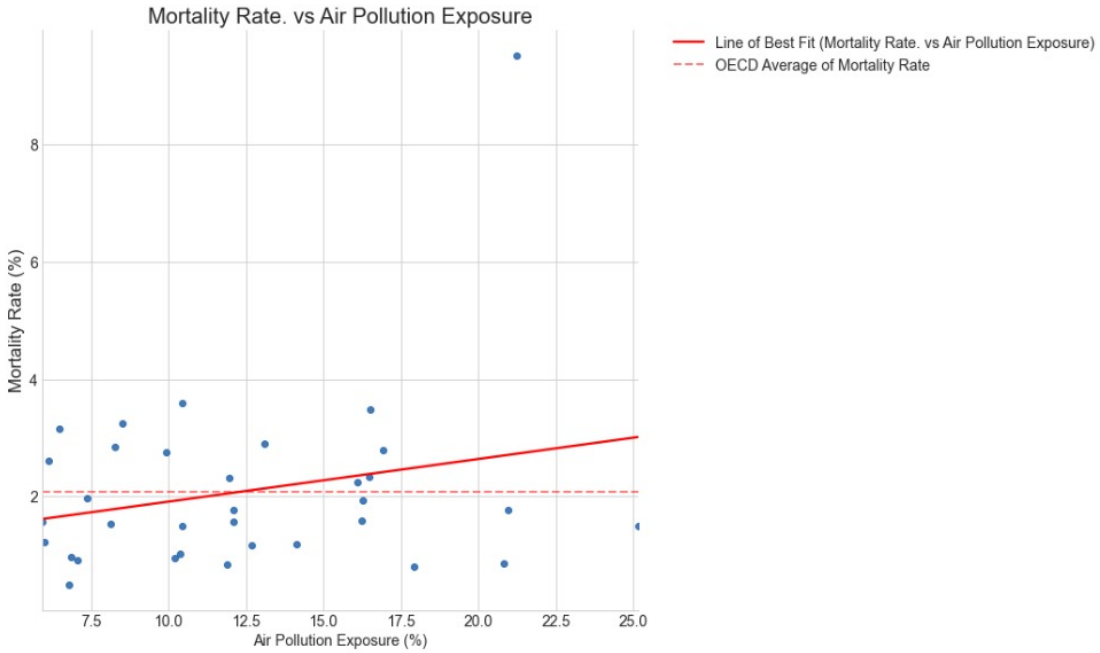
The box plot is included to show each countries' data more efficiently. The data from Mexico is coloured in red, because Mexico has the greatest mortality rate and so looking at Mexico's data can give a insight how these factors, predictor variables, influences the mortality rate. Also the average data is displayed as a green dashed line. This line can effectlvely show where each countries' data are located compare to the average data.

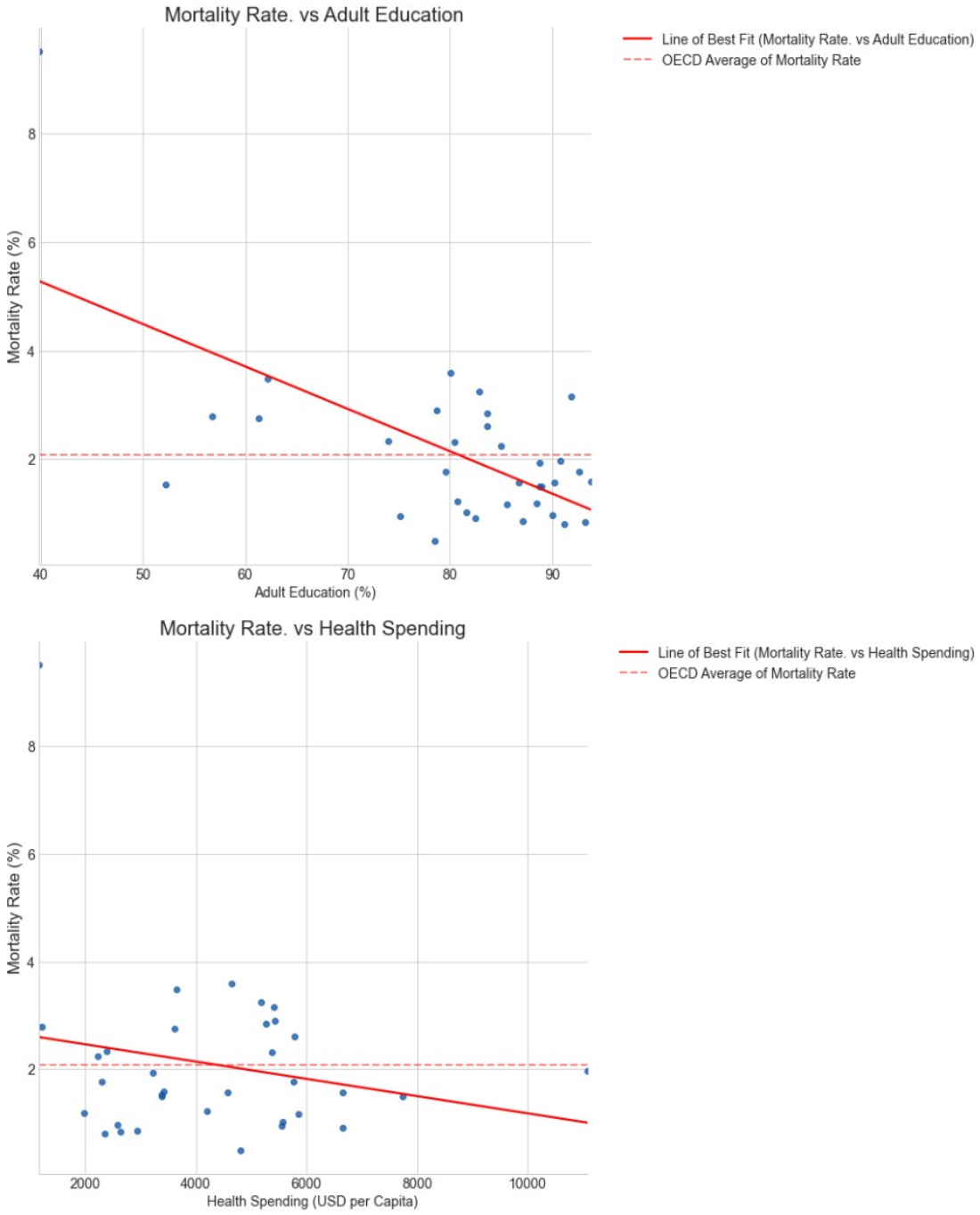
Scatter Plot

In [12]:
```
lst = ["Confirmed Case", "Elderly Population", "Air Pollution Exposure",
       "Adult Education", "Health Spending"]

def get_line_plot(data, name):

    fig, ax = plt.subplots(figsize = (10,10))

    ax = sns.regplot(x=name, y='Mortality Rate', ci=None,
                     line_kws={'color': 'red', 'label':"Line of Best Fit (Mortality Ra

    if name == "Confirmed Case":
        ax.set_xlabel(name + " (in thousands)", fontsize = 14)
        ax.set_xlim(0, 10000)

    elif name == "Health Spending":

        ax.set_xlabel(name + " (USD per Capita)", fontsize = 14)


    else:

        ax.set_xlabel(name + " (%)", fontsize = 14)

    p = plt.axhline(data["Mortality Rate"].mean(), color='r', linestyle='dashed', lin
                    label = "OECD Average of Mortality Rate",)

    ax.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0)
    ax.set_title("Mortality Rate. vs "+name)
    ax.set_ylabel("Mortality Rate (%)")
    ax.set_facecolor('white')
    fig.set_facecolor('white')


for i in lst:
    get_line_plot(df, i)
```




Mortality Rate. vs Elderly Population
Line of Best Fit (Mortality Rate. vs Elderly Population)
OECD Average of Mortality Rate
Mortality Rate (%)
Elderly Population (%)
8
6
4
2
8
10
12
14
16
18
20
22


Mortality Rate. vs Air Pollution Exposure
Line of Best Fit (Mortality Rate. vs Air Pollution Exposure)
OECD Average of Mortality Rate
Mortality Rate (%)
Air Pollution Exposure (%)
8
6
4
2
7.5
10.0
12.5
15.0
17.5
20.0
22.5
25.0



These scatter plots are imporved version of the scatter plots from the project 1 with some curves added to the inital scatter plot. The red straight line is the regression line of the data, line of best fit, and the blurry red dashed line is shows the average mortality rate. They help to understand the correaltion between the response variable, mortality rate, and predictor variables.

Mortality Rate .vs Confirmed Case

Single points on the scatter is, for the same reason as explained while interpreting the histogram of confirmed case, is quite concentrated on the left and there is one point, the US data, on the far right. The regression line shows that there is a positive correlation between mortality rate and the number of confirmed case.

Mortality Rate .vs Elderly Population

Single points are scattered all around but we can find more points on the right than the left, and more points at the bottom than the top. Along with single points, the regression line shows a negative correlation between the mortality rate and the rate of elderly population, i.e., as the rate of elderly population decreases the mortality rate increases. This result is quite questionable since covid-19 is known to be more risky to the elders than young people.

Mortality Rate .vs Air Pollution Exposure

This scatter plot shows a positive correlation between the mortality rate and the rate of air pollution exposure.

Mortality Rate .vs Adult Education

We can see that most of the points are located at the bottom left. The regression line is pretty steep and thus, it shows a negative correlation between the mortality rate and adult education rate. More specifically, as the rate of the adult population who has completed post-secondary education increases, the mortality rate decreases.

Mortality Rate .vs Health Spending

The scatter plot shows a negative correlation between mortality rate and health spending, as the amount of health spending increases, the mortality rate decreases. We can detect that as there are more points on the bottom left and a negatively sloped regression iine.

In [13]:
```
world = gpd.read_file(gpd.datasets.get_path("naturalearth_lowres"))
world = world[(world.name!="Antarctica")]
world =  world.replace({"name": "South Korea"}, {"name": 'Korea, South'})
world =  world.replace({"name": "United States of America"}, {"name": 'US'})
```

In [14]:
```
data = pd.merge(world, df, left_on="name", right_on="LOCATION", how="left")
drop= ["pop_est", "continent", "iso_a3", "gdp_md_est"]
data.drop(drop, axis = 1, inplace = True)
```

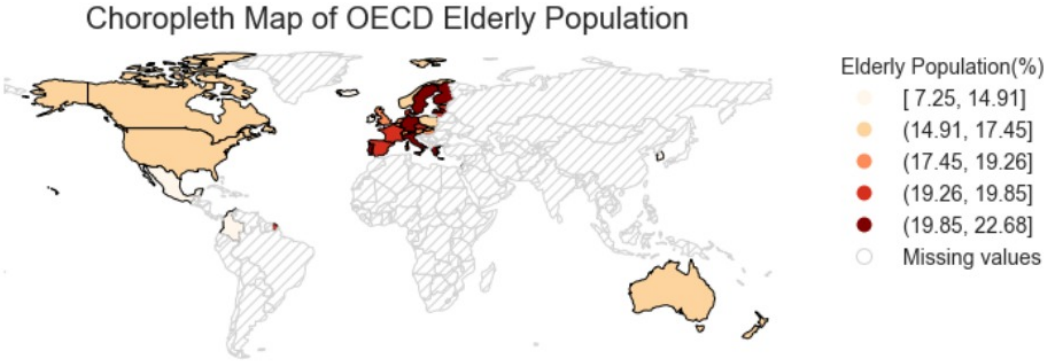
Part 3

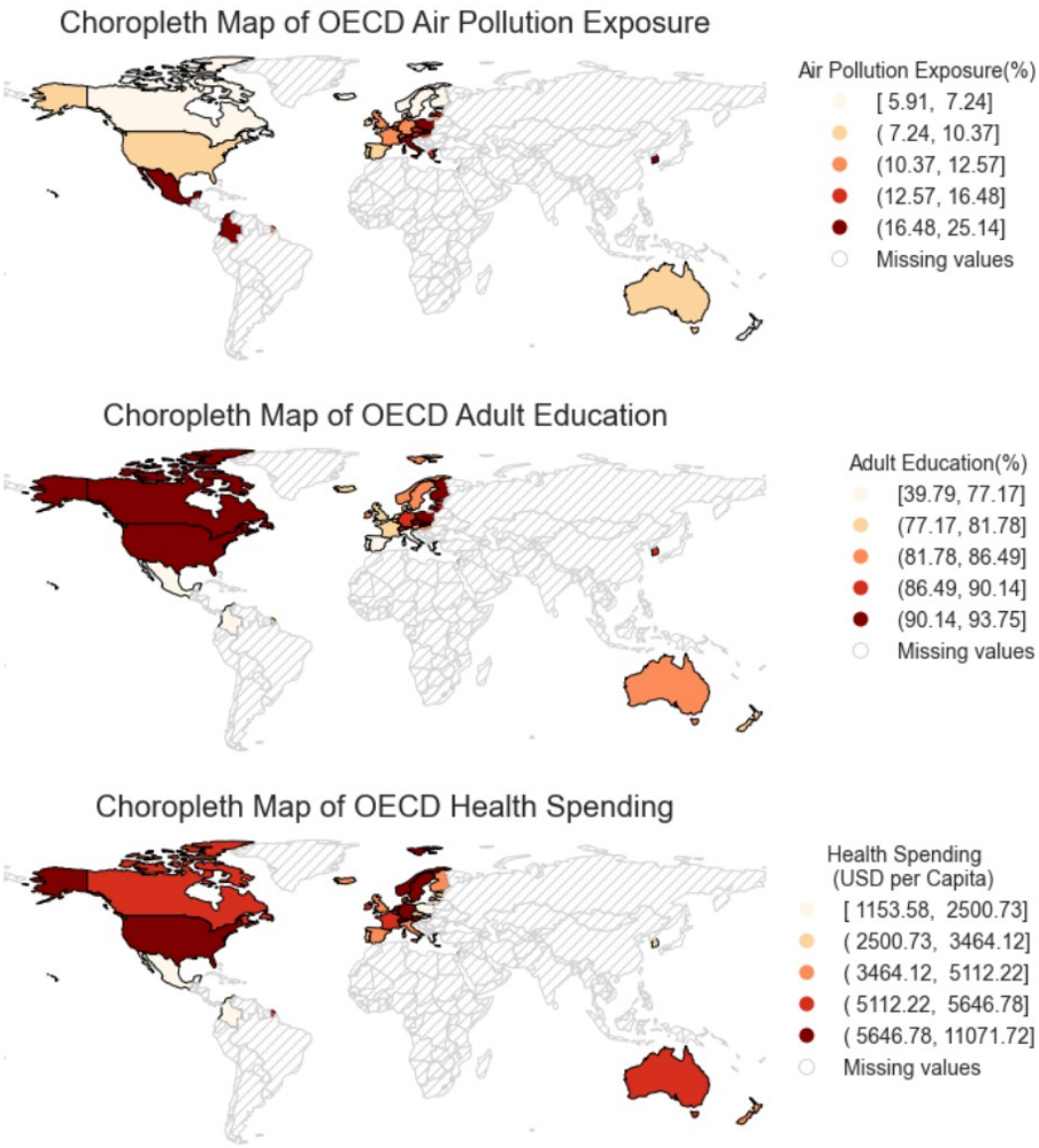
Maps

Cholopleth Map

In [15]:
```
def get_map(data, name):

    fig, gax = plt.subplots(figsize = (10,8))


    if name == "Confirmed Case":
            data.plot(ax=gax, edgecolor='black', column=name, legend=True,
                      cmap='OrRd', missing_kwds = {"color":"white", "edgecolor": "ligh
                                                   "label":"Missing values"}, scheme=
                      legend_kwds={'bbox_to_anchor': (1.3, 1), "loc":'upper right', "f
                     )
    elif name == "Health Spending":
        data.plot(ax=gax, edgecolor='black', column=name, legend=True,
                      cmap='OrRd', missing_kwds = {"color":"white", "edgecolor": "ligh
                                                   "label":"Missing values"}, scheme=
                      legend_kwds={'bbox_to_anchor': (1.3, 1), "loc":'upper right', "f
                     )
```

```
    else:
        data.plot(ax=gax, edgecolor='black', column=name, legend=True,
                  cmap='OrRd', missing_kwds = {"color":"white", "edgecolor": "lig
                                               "label":"Missing values"}, scheme=
                  legend_kwds={'bbox_to_anchor': (1.3, 1), "loc":'upper right', "
                 )

    gax.set_axis_off()
    gax.set_title("Choropleth Map of OECD " + name)

    plt.show()




lst = ["Mortality Rate", "Confirmed Case", "Elderly Population", "Air Pollution Exposu
       "Adult Education", "Health Spending"]

for i in lst:
    get_map(data, i)
```





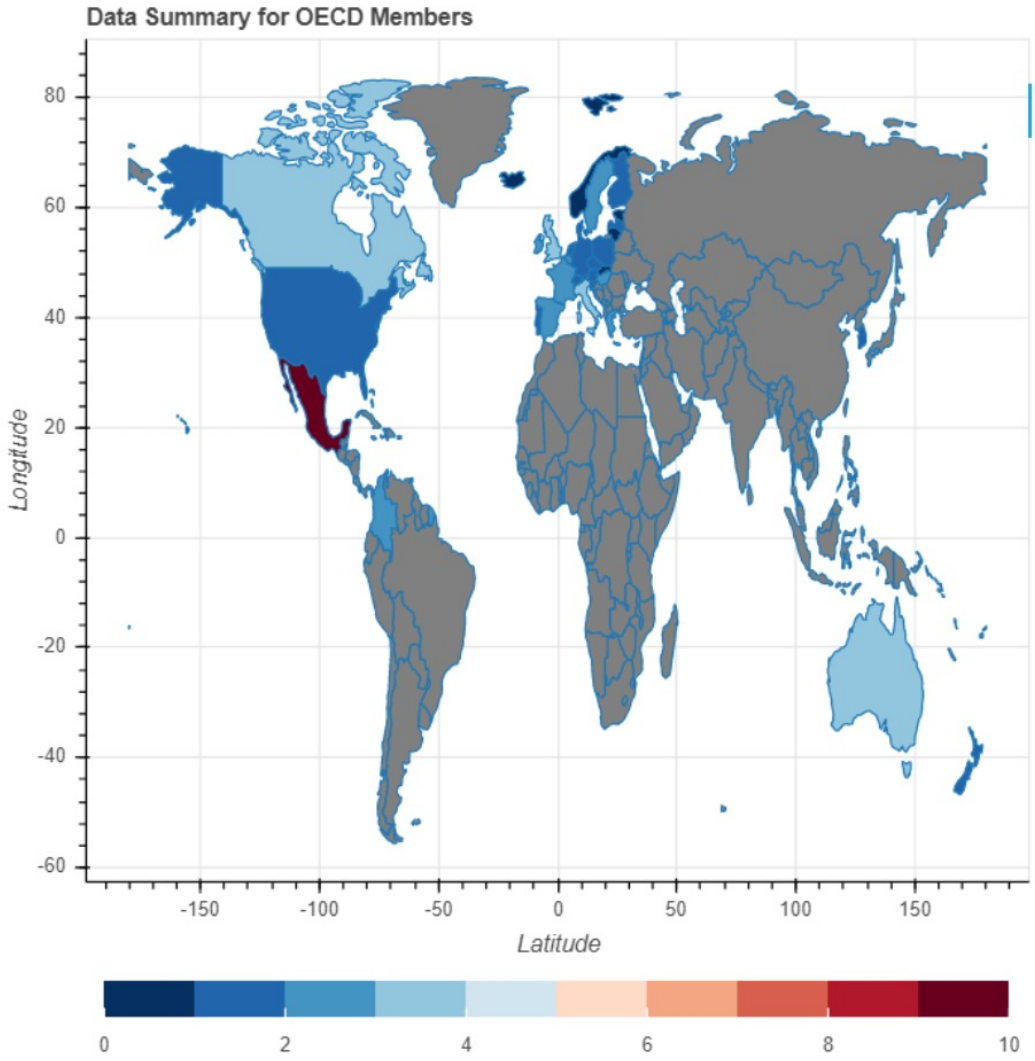
These are choropleth maps are shaded in different colours in proportion to the mortality rate, number of confirmed case, elderly population, air pollution exposure, adult education rate and the amount of health spending of OECE countries. Each data is divided into 5 ranges of equal probability as each legends shows. The areas shaded in white and filled with grey slashes are non-OECD countries or OECD countries that miss some data. The dark red colour indicates that the value is relatively the highest or the biggest and the value gets lowers or smaller as the colour gets lighter. These maps efficiently visualizes the statistical values of each variables.

In [16]:
```
from bokeh.io import output_notebook
from bokeh.plotting import figure, ColumnDataSource
from bokeh.io import output_notebook, show, output_file
from bokeh.plotting import figure
from bokeh.models import GeoJSONDataSource, LinearColorMapper, ColorBar, HoverTool
from bokeh.palettes import brewer
output_notebook()
import json

#Convert data to geojson for bokeh
data = data.fillna("No data")
data_geo=GeoJSONDataSource(geojson=data.to_json())
```

BokehJS 2.2.1 successfully loaded.

Interaction Map

In [17]:
```
color_mapper = LinearColorMapper(palette = brewer['RdBu'][10], low = 0, high = 10)
color_bar = ColorBar(color_mapper=color_mapper, label_standoff=8,width = 500, height
                     border_line_color=None,location = (0,0), orientation = 'horizont


hover = HoverTool(tooltips = [ ("Country",'@name'),('Mortality Rate (%)', '@{Mortality
                               ('Confirmed Case','@{Confirmed Case}'),
                               ('Elderly Population (%)','@{Elderly Population}'), ('A
                              ('Adult Education (%)','@{Adult Education}'), ('Health Sp

p = figure(title="Data Summary for OECD Members", tools=[hover], x_axis_label = "Lati

p.patches("xs","ys",source=data_geo,
          fill_color = {'field' :'Mortality Rate', 'transform' : color_mapper})

p.add_layout(color_bar, 'below')


show(p)
```



The interaction map effectively shows the data of each countries and the different colours show the mortality rate of each countries. The areas shaded in grey are non-OECD countries or OECD countries that contains some missed data.

for the HTML version, visit:
https://drive.google.com/file/d/1WU91WkruSKRxOV40LMPJIGmQBdaIp0Sb/view?usp=sharing

Conclusion

The project investigates the influence of the number of confirmed case, elderly population, air pollution exposure, adult education rate and health spending on the mortality rate. With the graphical figures, maps and the description of the data, we have found correlation between variables - as the number of confirmed case, or air pollution exposure rate increases, the mortality rate increases the mortality rate increases. In contrast, as the rate of elderly population, aldult education or the amount of health spending increased, the mortality rate decreases. Because the vaccinations and medications to prevent and treat COVID-19 is not yet completely developed, the pandamic is expected to continue. Therefore, the findings from the project are subject to change in the future, therefore, it is importatnt to constantly update the most recent data for a more accurate and representative interpretation.

Project 3: The Effects of Various Socio-Economic Factors on the Mortality Rate and Incident Rate of COVID-19.

Introduction

The purpose of the project is to investigate effects of various socioeconomic factors on the mortality rate and the incident rate of COVID-19. The socioeconomic factors include elderly population, adult education attainment rate, health spending, gross domestic product and passenger traffic by air. These socioeconomic data are gained from OECD Statistics and World Bank. The mortality rate and the indicent rate of COVID-19 are gained from the Center for System Science nd Engineering (CSSE) at the Johns Hopkins University. The socioeconomic factors are used as predictor variables and the mortality rate and the incident rate are set as response variables.

The elderly population data shows the share of the total elderly population (aged 60+) expressed as a ratio of the total population. The adult education attainment rate data shows the percentage of the population who has completed post-secondary education over the total population and the health spending data measures the final consumption of health care goods and services from both privata and public sectors and is presented in USD per capita. The GDP data is based on GDP (PPP) and is in US dollars per capita. Lastly the passenger traffic by air data shows the number of passengers transported by airlines that are registered in the respective country.

The project focuses on the data from 43 countries across the world. Most of the countries are the members of OECD, but some of them are not, and the countries come from both emerging markets and developed markets.

Part 2

Newly Added Data Using Web-Scraping

Two of the socioeconomic factors are newly added as predictor variables to the project. The new predictor variables are the GDP per capita and the passenger traffic by air. The reason why these two data are added is that the GDP can show a country's economic status numerically, so we can find if there is a relationship between a country's wealth and the response variables. Also, the passenger traffic by air can represent an openess of a country.

Additionally, they are added using web-scraping because these data are not cleanly and nicely structured. Therefore, by using web scraping, we could get a clean and structed data efficently.

Challenges

The biggest challenge is that for the sake of the accuracy of the project, it is crucial to updata the COVID-19 data daily, as governments around the world updates the daily confirmend cases and deaths.

Cleaning and loading dataset.

In [41]:
```
def clean(dir, rename):
    d = pd.read_csv(dir)
    d = d.groupby("LOCATION").tail(1)
    drop = ['INDICATOR', 'SUBJECT', 'MEASURE', 'FREQUENCY', 'TIME', 'Flag Codes']
    d.drop(drop, axis = 1, inplace = True)
    d.rename(columns = {"Value":rename}, inplace = True)
    return d

eld = "C:₩₩Users₩₩lsy76₩₩Desktop₩₩20-21₩₩fall₩₩ECO225₩₩Project 1₩₩data₩₩eld.csv"
edu = ("C:₩₩Users₩₩lsy76₩₩Desktop₩₩20-21₩₩fall₩₩ECO225₩₩Project 1₩₩data₩₩edu.csv")
health = ("C:₩₩Users₩₩lsy76₩₩Desktop₩₩20-21₩₩fall₩₩ECO225₩₩Project 1₩₩data₩₩health.csv
pop = ("C:₩₩Users₩₩lsy76₩₩Desktop₩₩20-21₩₩fall₩₩ECO225₩₩Project 1₩₩pop.csv")
eld = clean(eld, "Elderly Population")
edu = clean(edu, "Adult Education")
health = clean(health, "Health Spending")
pop = clean(pop, "pop")

d = eld.merge(edu, left_on="LOCATION", right_on="LOCATION", how="outer")
d = d.merge(health, left_on="LOCATION", right_on="LOCATION", how="outer")
d = d.merge(pop, left_on="LOCATION", right_on="LOCATION", how="inner")
d = d.dropna()
d = d.set_index("LOCATION")

d["pop"] = d['pop']*1000000

lst1 = ['AUS', 'AUT', 'BEL', 'CAN', 'CZE', 'DNK', 'FIN', 'FRA', 'DEU', 'GRC',
       'HUN', 'ISL', 'IRL', 'ITA', "JPN", 'KOR', 'LUX', 'MEX', 'NLD',"NZL", 'NOR', 'PO
       'SVK', 'ESP', 'SWE', 'CHE', 'TUR', 'GBR', "USA", "BRA", "CHL", "COL", 'EST', 'I
             "CHN", "CRI", "IND", "IDN", "ZAF", "LTU"]

lst2 = ['Australia', "Austria", "Belgium", "Canada", "Czechia", "Denmark", "Finland",
              "Greece", "Hungary", "Iceland","Ireland", "Italy", "Japan", "Korea, Sout
              "New Zealand", "Norway", "Poland", "Portugal", "Slovakia", "Spain", "Swe
             "Brazil", "Chile", "Colombia", "Estonia", "Israel",  "Russia", "Slovenia'
              "Indonesia", "South Africa", "Lithuania"]

for i in range(0, len(lst1)):
    d = d.rename(index={lst1[i]:lst2[i]})
```

```
dir = "C:₩₩Users₩₩lsy76₩₩Desktop₩₩20-21₩₩fall₩₩ECO225₩₩Project 1₩₩csse_covid_19_data₩₩

covid = pd.read_csv(dir)

drop = ["FIPS", "Admin2", "Province_State", "Last_Update",  "Combined_Key", "Incident_
covid.drop(drop, axis = 1, inplace = True)
covid.rename(columns = {"Country_Region":"LOCATION"}, inplace = True)

covid = covid.groupby("LOCATION").sum()
d = d.merge(covid, left_index=True, right_index=True, how="inner")

d['Mortality Rate'] = d["Deaths"]/d["Confirmed"] * 100
d["Incident Rate"] = d["Confirmed"]/d["pop"]*100
drop = ["Confirmed", "Deaths", "pop"]
d.drop(drop, axis = 1, inplace = True)
d = d[['Mortality Rate', 'Incident Rate', 'Elderly Population', 'Adult Education', 'He

df = d
```

Web-Scraping

We start by requesting for the web contents for wikipedia's "List of countries by GDP (PPP) per capita" page with the URL - https://en.wikipedia.org/wiki/List_of_countries_by_GDP_(PPP)_per_capita/, and the "List of countries by airline passengers" page with the URL - https://en.wikipedia.org/wiki/List_of_countries_by_airline_passengers/.

In [42]:
```
import requests
from bs4 import BeautifulSoup

#GDP data
web_url_gdp = "https://en.wikipedia.org/wiki/List_of_countries_by_GDP_(PPP)_per_capita
response_gdp = requests.get(web_url_gdp)
print("GDP data: ", response_gdp)

#Passenger Traffic by Air data

web_url_pta = 'https://en.wikipedia.org/wiki/List_of_countries_by_airline_passengers'
response_pta = requests.get(web_url_pta)
print("Passenger Traffic by Air data: ", response_pta)
```

```
GDP data:  <Response [200]>
Passenger Traffic by Air data:  <Response [200]>
```

We have obtained a successful responses. Now, we will proceed to get nicely structured soup objects passing the response contents to a BeautifulSoup() method.

In [43]:
```
#GDP data
soup_object_gdp = BeautifulSoup(response_gdp.content)

#Passenger Traffic by Air data
soup_object_pta = BeautifulSoup(response_pta.content)
```

Now identify the type of HTML tag which contains all the data along with any id names or class names associated to that HTML tag.

In [44]:
```
#GDP data
gdp = soup_object_gdp.find_all("table", "wikitable sortable")
gdp_values = gdp[0].find_all('tr')

#Passenger Traffic by Air data
```

```
pta = soup_object_pta.find_all("table", "wikitable sortable")
pta_values = pta[0].find_all('tr')

def show(text, values):
    print(text)
    print(values[0])
    print('--')
    print(values[1])
    print('--')
    print(values[2])

show("GDP data", gdp_values)
print("_____________________________________________")
show("Passenger Traffic by Air data", pta_values)
```

```
GDP data
<tr>
<th data-sort-type="number">Rank
</th>
<th>Country/Territory
</th>
<th>Int$
</th></tr>
--
<tr>
<td>1</td>
<td align="left"><span class="flagicon"><img alt="" class="thumbborder" data-file-heig
ht="600" data-file-width="1000" decoding="async" height="14" src="//upload.wikimedia.o
rg/wikipedia/commons/thumb/d/da/Flag_of_Luxembourg.svg/23px-Flag_of_Luxembourg.svg.pn
g" srcset="//upload.wikimedia.org/wikipedia/commons/thumb/d/da/Flag_of_Luxembourg.svg/
35px-Flag_of_Luxembourg.svg.png 1.5x, //upload.wikimedia.org/wikipedia/commons/thumb/
d/da/Flag_of_Luxembourg.svg/46px-Flag_of_Luxembourg.svg.png 2x" width="23"/> </span><a
href="/wiki/Luxembourg" title="Luxembourg">Luxembourg</a></td>
<td>112,875
</td></tr>
--
<tr>
<td>2</td>
<td align="left"><span class="flagicon"><img alt="" class="thumbborder" data-file-heig
ht="2880" data-file-width="4320" decoding="async" height="15" src="//upload.wikimedia.
org/wikipedia/commons/thumb/4/48/Flag_of_Singapore.svg/23px-Flag_of_Singapore.svg.png"
srcset="//upload.wikimedia.org/wikipedia/commons/thumb/4/48/Flag_of_Singapore.svg/35px
-Flag_of_Singapore.svg.png 1.5x, //upload.wikimedia.org/wikipedia/commons/thumb/4/48/F
lag_of_Singapore.svg/45px-Flag_of_Singapore.svg.png 2x" width="23"/> </span><a href="/
wiki/Singapore" title="Singapore">Singapore</a></td>
<td>95,603
</td></tr>
_____________________________________________
Passenger Traffic by Air data
<tr>
<th>Rank
</th>
<th>Country
</th>
<th>Airline passengers carried
</th>
<th>Year
</th></tr>
--
<tr>
<td>1
</td>
<td><span class="flagicon"><img alt="" class="thumbborder" data-file-height="650" data
-file-width="1235" decoding="async" height="12" src="//upload.wikimedia.org/wikipedia/
en/thumb/a/a4/Flag_of_the_United_States.svg/23px-Flag_of_the_United_States.svg.png" sr
cset="//upload.wikimedia.org/wikipedia/en/thumb/a/a4/Flag_of_the_United_States.svg/35p
x-Flag_of_the_United_States.svg.png 1.5x, //upload.wikimedia.org/wikipedia/en/thumb/a/
a4/Flag_of_the_United_States.svg/46px-Flag_of_the_United_States.svg.png 2x" width="2
3"/> </span><a href="/wiki/United_States" title="United States">United States</a>
```

```
</td>
<td>849,403,000
</td>
<td>2017
</td></tr>
--
<tr>
<td>2
</td>
<td><span class="flagicon"><img alt="" class="thumbborder" data-file-height="600" data
-file-width="900" decoding="async" height="15" src="//upload.wikimedia.org/wikipedia/c
ommons/thumb/f/fa/Flag_of_the_People%27s_Republic_of_China.svg/23px-Flag_of_the_Peopl
e%27s_Republic_of_China.svg.png" srcset="//upload.wikimedia.org/wikipedia/commons/thum
b/f/fa/Flag_of_the_People%27s_Republic_of_China.svg/35px-Flag_of_the_People%27s_Republ
ic_of_China.svg.png 1.5x, //upload.wikimedia.org/wikipedia/commons/thumb/f/fa/Flag_of_
the_People%27s_Republic_of_China.svg/45px-Flag_of_the_People%27s_Republic_of_China.sv
g.png 2x" width="23"/> </span><a href="/wiki/China" title="China">China</a>
</td>
<td>551,234,510
</td>
<td>2017
</td></tr>
```

GDP Data

The first element of the list contains the column names 'Rank, Country/Territory, and Int$'. The next elements of the list contain soup objects which contain the country's gdp data. This data can be extracted in a loop since the structure for all the list elements is the same.

An empty dataframe gdp_df is created with the column names LOCATION and GDP. The index is initiated to zero and a for loop is designed to go through all the elements of the list in order and extract the country name and the gdp per capita from the list element which are enclosed in the HTML tag.

A find_all() will return a list of td tags. The .text attribute can be used to just pick the text part from the tag and because there are some spaces on the right and left of the extracted texts, strip() methods are used. Because some of the country names are different from the ones from the original dataframe we use, the names are changed. The comma from gdp values are removed, and because the data types for the gdp values are strings, not intgers or floats, they are casted to integers. Finally, values are then put into the dataframe and the index value is incremented.

In [45]:
```
df_gdp= pd.DataFrame(columns = ['LOCATION', 'GDP']) # Create an empty dataframe

ix = 0 # Initialise index to zero

for row in gdp_values[1:]:

    values = row.find_all('td')

    i = values[1].text.rstrip().lstrip()

    if i == "United States":
        LOCATION = "US"
    elif i == "South Korea":
        LOCATION = "Korea, South"
    elif i == "Czech Republic":
        LOCATION = "Czechia"
    else:
        LOCATION = i

    j = values[2].text.rstrip().lstrip().replace(",", "")
```

```
    GDP = float(j)*0.001
    df_gdp.loc[ix] = [LOCATION, GDP] # Store it in the dataframe as a row

    ix += 1

df_gdp = df_gdp.set_index("LOCATION")
df_gdp
```

Out[45]:

LOCATION	GDP
Luxembourg	112.875
Singapore	95.603
Qatar	91.897
Ireland	89.383
Switzerland	68.340
...	...
Malawi	0.995
Democratic Republic of the Congo	0.978
Central African Republic	0.972
South Sudan	0.884
Burundi	0.783

191 rows × 1 columns

Passenger Traffic by Air Data

The first element of the list contains the column names 'Rank, Country, Airline passengers carried and Year'. The next elements of the list contain soup objects which contain the country's gdp data. This data can be extracted in a loop since the structure for all the list elements is the same.

An empty dataframe pta_df is created with the column names LOCATION and Passenger Traffic by Air. The index is initiated to zero and a for loop is designed to go through all the elements of the list in order and extract the country name and the number of passenger carried by air from the list element which are enclosed in the HTML tag.

A find_all() will return a list of td tags. The .text attribute can be used to just pick the text part from the tag and because there are some spaces on the right and left of the extracted texts, strip() methods are used. Because some of the country names are different from the main dataframe we use, the names are changed. The comma from the passenger carried values are removed, and because the data types for the passenger carried values are strings, not intgers or floats, they are casted to integers. Finally, values are then put into the dataframe and the index value is incremented.

In [46]:
```
df_pta= pd.DataFrame(columns = ['LOCATION', 'Passenger Traffic by Air']) # Create an

ix = 0 # Initialise index to zero

for row in pta_values[1:-1]:
```

```
    values = row.find_all('td')

    i = values[1].text.rstrip().lstrip()

    if i == "United States":
        LOCATION = "US"
    elif i == "South Korea":
        LOCATION = "Korea, South"
    elif i == "Czech Republic":
        LOCATION = "Czechia"
    else:
        LOCATION = i


    j = values[2].text.rstrip().lstrip().replace(",", "")
    Passenger_Carried = float(j) * 0.000001
    df_pta.loc[ix] = [LOCATION, Passenger_Carried] # Store it in the dataframe as a r
    ix += 1

df_pta = df_pta.set_index("LOCATION")
df_pta
```

Out[46]:

LOCATION	Passenger Traffic by Air
US	849.40300
China	551.23451
Ireland	153.53755
United Kingdom	151.86728
India	139.75242
...	...
Dominican Republic	0.01190
Bosnia and Herzegovina	0.00707
Somalia	0.00449
Benin	0.00090
Monaco	0.00032

165 rows × 1 columns

In [47]:
```
scraped = pd.merge(df_gdp, df_pta, left_index=True, right_index=True, how="outer")
```

Merge

In [48]:
```
data = pd.merge(df, scraped, left_index=True, right_index=True, how="left")
data = data.dropna()

data
```

Out[48]:

LOCATION	Mortality Rate	Incident Rate	Elderly Population	Adult Education	Health Spending	GDP	Passenger Traffic by Air

LOCATION	Mortality Rate	Incident Rate	Elderly Population	Adult Education	Health Spending	GDP	Passenger Traffic by Air
Australia	3.251800	0.111724	15.663	47.129978	9.334	50.845	74.257330
Austria	1.164668	3.230351	18.759	33.773777	10.408	55.406	16.171640
Belgium	2.898075	5.079141	18.835	40.670090	10.349	50.114	13.676840
Canada	3.159524	1.044425	17.157	59.374844	10.790	47.569	91.404000
Czechia	1.590807	4.973204	19.414	24.212393	7.763	40.293	5.450670
Denmark	1.025828	1.424363	19.461	40.353836	10.031	57.781	6.428700
Finland	1.567041	0.461642	21.613	45.934380	9.086	49.334	14.831370
France	2.321364	3.399121	19.837	37.904724	11.186	45.454	68.316470
Germany	1.569137	1.320254	21.466	29.900126	11.654	53.571	116.847030
Greece	2.342049	1.001969	21.894	31.890486	7.791	29.045	13.861830
Hungary	2.251293	2.263330	19.143	25.981535	6.360	32.434	26.066290
Iceland	0.498799	1.534636	14.136	45.038918	8.788	54.482	7.242610
Ireland	2.842111	1.498822	13.864	47.306587	6.841	89.383	153.537550
Italy	3.477140	2.682643	22.675	19.632418	8.661	40.066	27.836420
Japan	1.396911	0.119402	28.137	52.677990	11.055	41.637	123.898000
Korea, South	1.495891	0.068099	14.294	50.031040	8.044	44.292	84.045160
Luxembourg	0.939395	5.778271	14.354	51.642971	5.397	112.875	1.901010
Mexico	9.512528	0.895541	7.246	18.259947	5.477	18.804	58.536880
Netherlands	1.775319	3.111309	19.015	40.385136	9.965	57.101	42.770630
New Zealand	1.213592	0.042166	15.288	39.065784	9.319	41.072	16.271520
Norway	0.912793	0.688847	17.087	44.129421	10.489	64.856	54.099013
Poland	1.760034	2.603078	17.230	32.014767	6.159	33.739	7.376930
Portugal	1.523321	2.921696	21.672	26.284128	9.563	33.131	15.946360
Spain	2.747512	3.544482	19.290	38.596992	8.999	38.143	71.908520
Sweden	2.607015	2.562678	19.861	43.970543	10.882	52.477	11.623900
Switzerland	1.493015	3.886587	18.371	44.412537	12.142	68.340	26.737540
Turkey	2.083243	0.821742	8.649	21.951159	4.376	28.294	107.917330
United Kingdom	3.590755	2.479441	18.312	47.189125	10.254	44.288	151.867280
US	1.972422	4.193970	16.026	48.340187	16.960	63.051	849.403000
Brazil	2.721509	3.063282	9.222	18.429998	9.423	14.563	96.395710
Chile	2.790921	2.948387	11.547	25.168179	9.102	23.455	17.641090
Colombia	2.787909	2.658397	8.060	23.819326	7.260	14.137	32.506880
Estonia	0.968232	0.945327	19.676	41.368298	6.786	37.033	0.013100

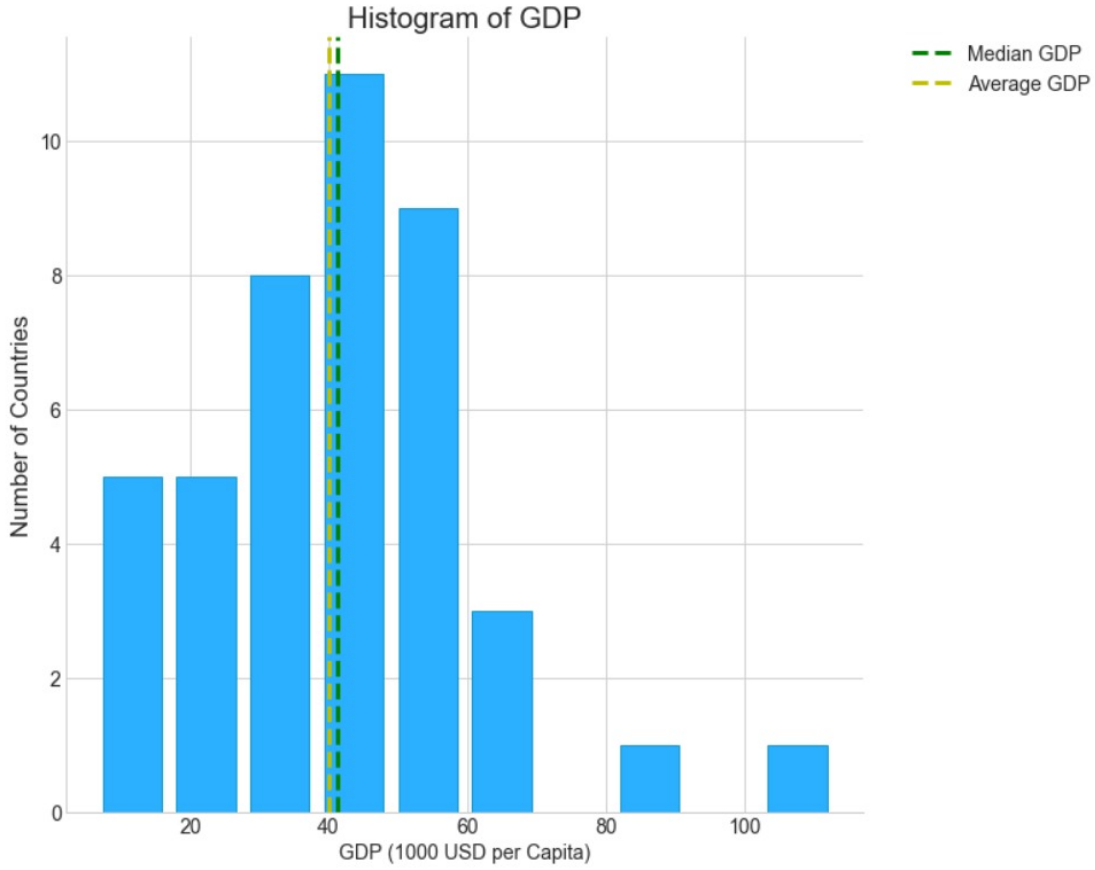
LOCATION	Mortality Rate	Incident Rate	Elderly Population	Adult Education	Health Spending	GDP	Passenger Traffic by Air
Israel	0.850864	3.810765	11.654	50.246292	7.452	39.126	7.065040
Russia	1.739745	1.593221	14.746	56.725521	5.316	27.394	89.373640
Slovenia	1.931678	3.726238	19.670	33.283279	8.255	38.506	1.087080
Latvia	1.187313	0.917771	20.181	35.707733	6.255	30.579	3.441020
China	5.100384	0.006572	11.194	9.681174	5.047	17.206	551.234510
Costa Rica	1.234911	2.801534	8.180	25.066828	7.510	19.309	1.822880
India	1.454336	0.698851	6.176	10.595189	3.603	6.284	139.752420
Indonesia	3.140034	0.205262	5.812	11.857569	2.988	12.345	110.252910
Lithuania	0.830201	2.231450	19.706	43.149418	6.803	38.605	1.069270
South Africa	2.731797	1.372526	5.587	7.197612	8.113	11.911	20.821040

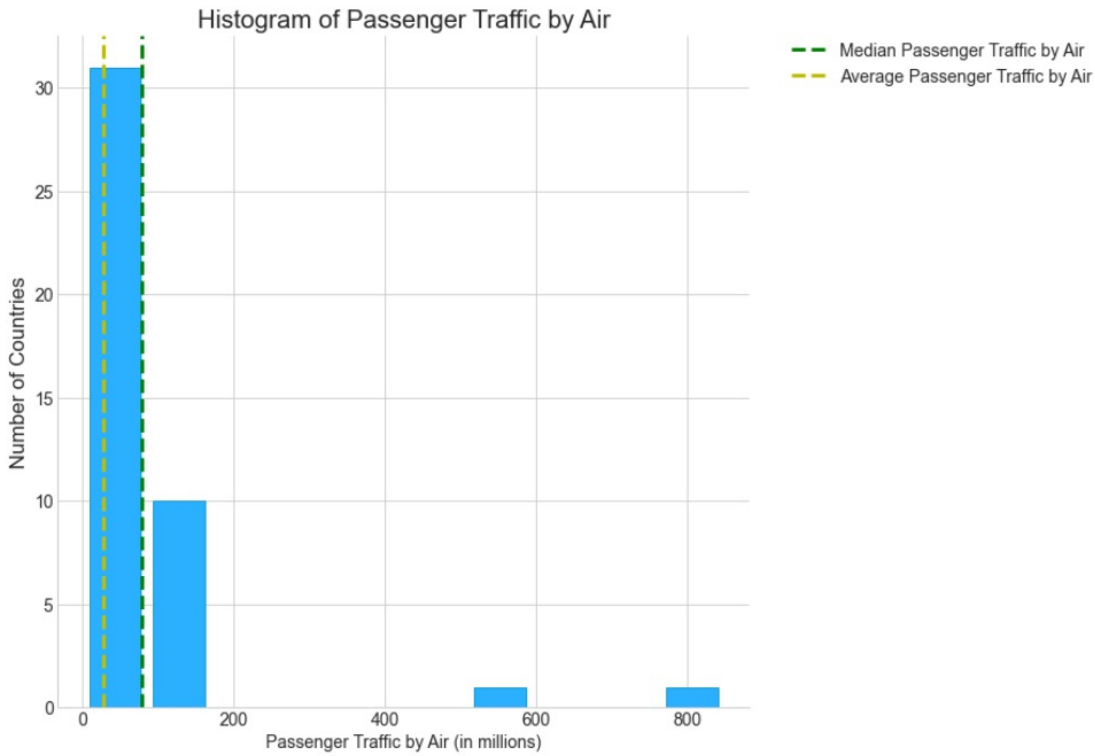
Visualization of the New Data

Histogram

In [49]:
```
def get_histogram(data, name):

    fig, ax = plt.subplots(figsize = (10,10))

    if name == "GDP":

        plt.hist(data[name], rwidth = 0.8, facecolor = '#2ab0ff', edgecolor='#169acf'
        plt.xlabel(name + " (1000 USD per Capita)", fontsize = 14)

    elif name == "Passenger Traffic by Air":

        plt.hist(data[name], rwidth = 0.8, facecolor = '#2ab0ff', edgecolor='#169acf'
        plt.xlabel(name + " (in millions)", fontsize = 14)

    else:

        plt.hist(data[name], rwidth = 0.8, facecolor = '#2ab0ff', edgecolor='#169acf'
        plt.xlabel(name + " (%)", fontsize = 14)

    plt.xticks(fontsize = 14)
    plt.yticks(fontsize = 14)

    plt.title("Histogram of " + name)
    plt.ylabel("Number of Countries")
    plt.style.use('seaborn-whitegrid')

    p = plt.axvline(data[name].mean(), color='g', linestyle='dashed', linewidth=3)
    m = plt.axvline(data[name].median(), color='y', linestyle='dashed', linewidth=3)
    ax.legend((p,m), ("Median " + name, "Average " + name), bbox_to_anchor=(1.05, 1),


    plt.show()

lst = ["GDP", "Passenger Traffic by Air"]
```

```
for i in lst:
    get_histogram(data, i)
```





Interpretation

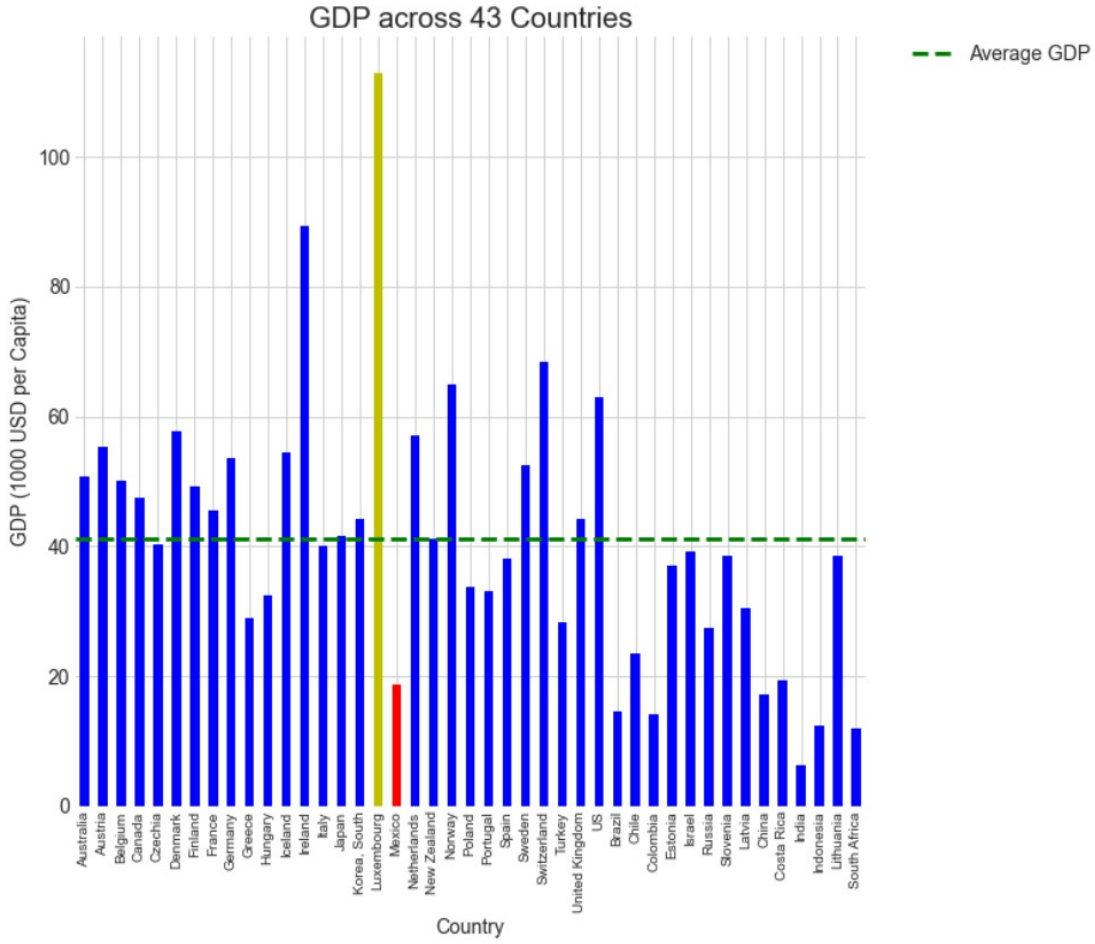
The histogram of GDP per capita data and the histogram of Passenger Traffic by Air data show an approximate distribution the GDP per capita and the number of passengers carried by flights

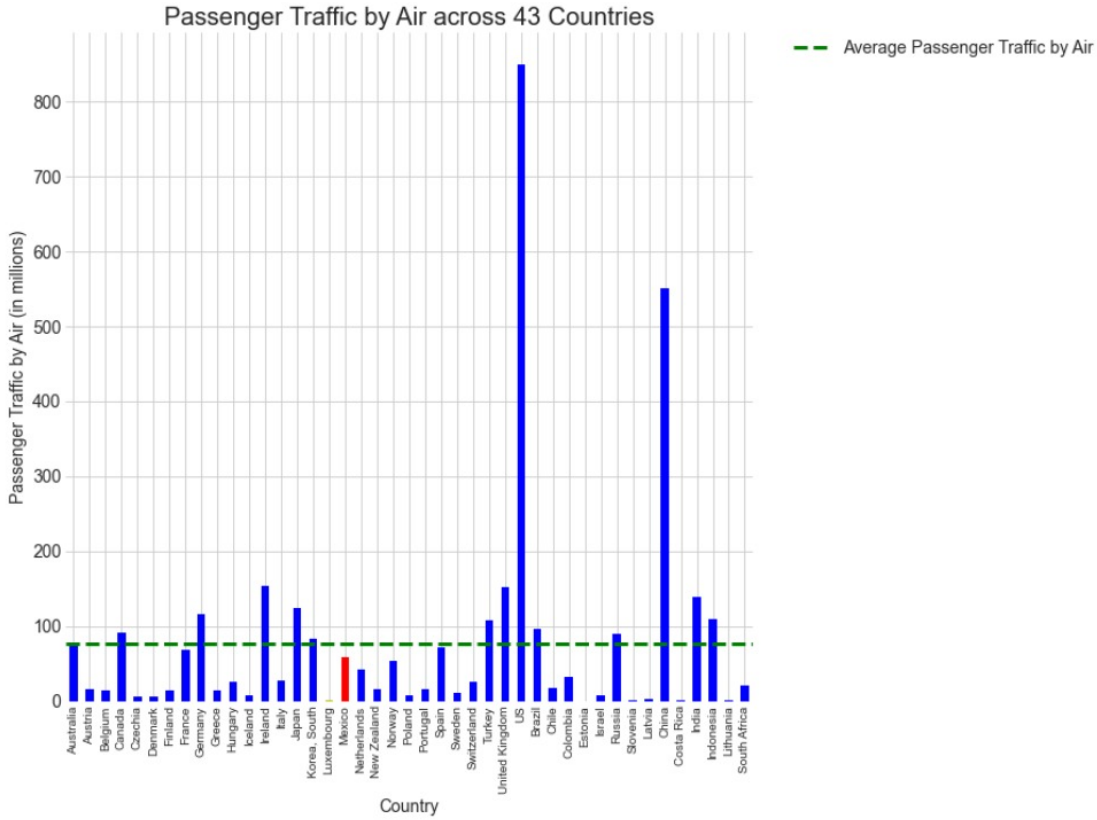
among 43 selected countries around the world. The histogram of GDP looks quite similar to a similar to the typical bell shape of a normal distribution, but still slightly skewed to the right. The mean and median GDP per capita across the countries are around 40,000 USD to 45,000 USD. In contrast, the Passenger Traffic by Air histogram shows an extremly right skewed distribution. The mean number of passengers carried by flights per year is around 20 to 30 million, while the median value is about 70 to 30 million passengers. Additionally, we can find some extraordinary data point at 800 million passengers.

Bar Chart

In [50]:
```
def get_bar_chart(data, name):

    d = data[name]

    fig, ax = plt.subplots(figsize = (10,10))

    for side in ["right", "top", "left", "bottom"]:
        ax.spines[side].set_visible(False)

    lst = []

    for i in data.iterrows():

        if i[0] == "Mexico":
            lst.append("r")

        elif i[0] == "Luxembourg":
            lst.append("y")

        else:
            lst.append("b")

    d.plot(kind="bar", ax=ax, color=lst)

    ax.spines['right'].set_visible(False)
    ax.spines['top'].set_visible(False)
    ax.set_title(name + " across 43 Countries")
    ax.set_xlabel("Country", fontsize = 14)

    plt.tick_params(axis='x', labelsize=10 )

    ax.set_facecolor('white')
    fig.set_facecolor('white')



    if name == "GDP":
        ax.set_ylabel(name + " (1000 USD per Capita)", fontsize = 14)

    elif name == "Passenger Traffic by Air":

        ax.set_ylabel(name + " (in millions)", fontsize = 14)

    else:
        ax.set_ylabel(name + " (%)", fontsize = 14)

    p = plt.axhline(data[name].mean(), color='g', linestyle='dashed', linewidth=3)
    ax.legend([p], ["Average " + name], bbox_to_anchor=(1.05, 1), loc=2, borderaxespa


for i in lst:
```

```
get_bar_chart(data, i)
```





Interpretation

The bars coloured in olive green and red indicates the country with the largest incident rate and mortality rate of covid-19 respectively. According to the bar chart of GDP data, Luxembourg has the highest GDP, as well as the largest incident rate. It is an interesting data because the deviation between the GDP of Luxembourg and the average GDP is quite huge. Therefore, it could be a potential outlier that demonstrates an inaccurate relationship between the GDP and the incident rate. Similarly, Mexico, which has the largest mortality rate has GDP that is relatively low. The devition between the GDP of Mexico and the average GDP is also pretty big, so we can get a clue that there may exist outliers.

Furthermore, the Passenger Air Traffic data visualized as a bar chart also shows some interesting trends. Luxembourg that has the largest GDP and incident rate has the least number of passengers carried by flights. Also, we see an extremly big data from the US and China. So, we need to keep these data in mind when investigating the relationship.

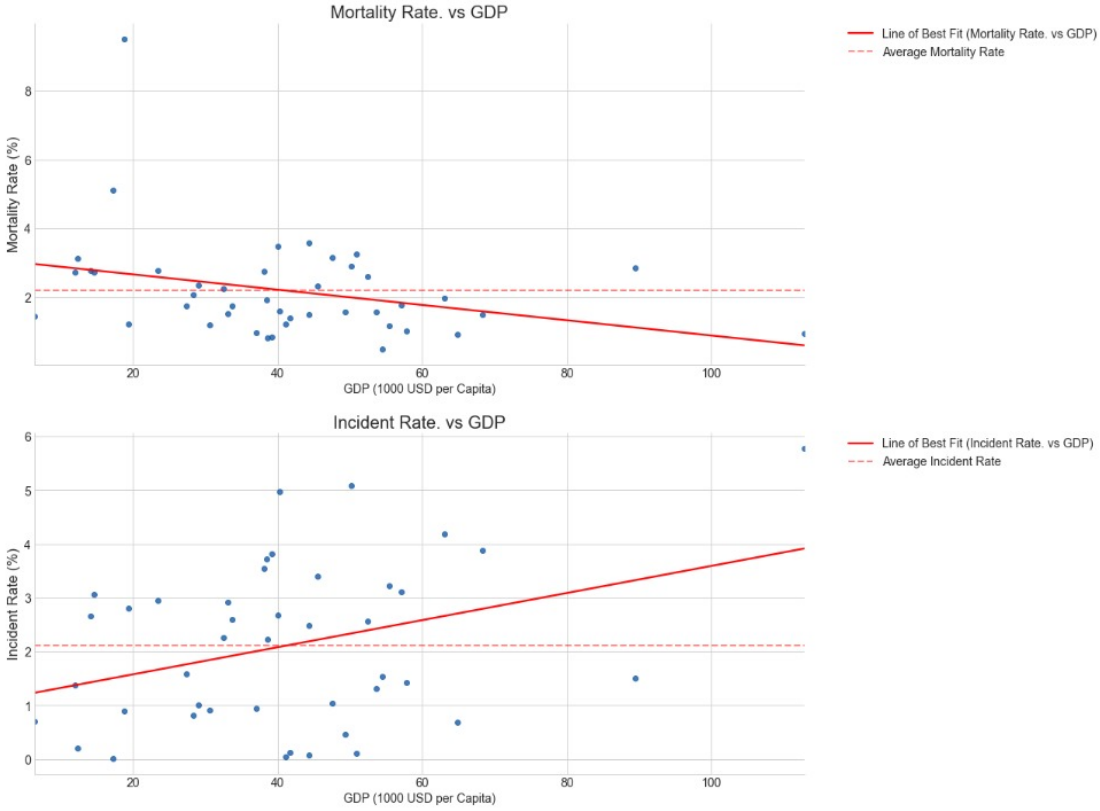
Scatter Plot

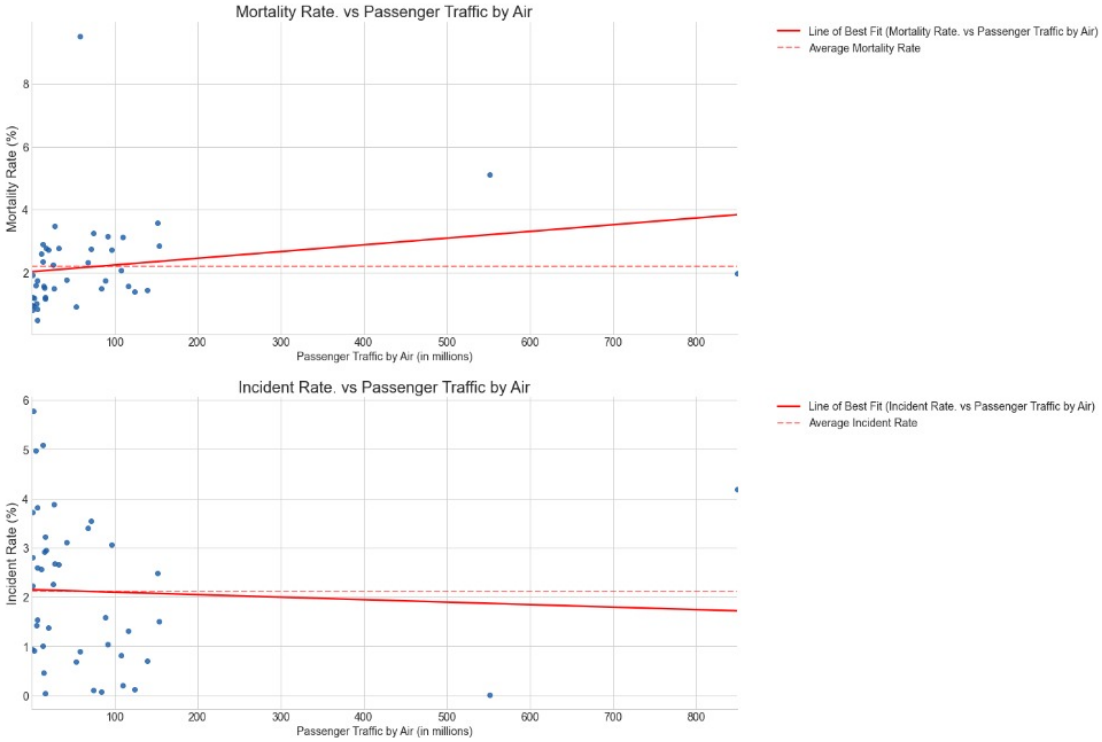
In [51]:
```
def get_line_plot(data, name):

    fig, (ax1, ax2)  = plt.subplots(nrows = 2)
    fig.set_size_inches(16, 16)

    sns.regplot(x=name, y='Mortality Rate', ci=None,
                line_kws={'color': 'red', 'label':"Line of Best Fit (Mortality Rate. v
    ax1.axhline(data["Mortality Rate"].mean(), color='r', linestyle='dashed', linewid
                    label = "Average Mortality Rate",)

    sns.regplot(x=name, y='Incident Rate', ci=None,
                     line_kws={'color': 'red', 'label':"Line of Best Fit (Incident Rat
    ax2.axhline(data["Incident Rate"].mean(), color='r', linestyle='dashed', linewidt
                    label = "Average Incident Rate",)

    if name == "GDP":
        ax1.set_xlabel(name + " (1000 USD per Capita)", fontsize = 14)
        ax2.set_xlabel(name + " (1000 USD per Capita)", fontsize = 14)

    elif name == "Passenger Traffic by Air":

        ax1.set_xlabel(name + " (in millions)", fontsize = 14)
        ax2.set_xlabel(name + " (in millions)", fontsize = 14)


    else:

        ax1.set_xlabel(name + " (%)", fontsize = 14)
        ax2.set_xlabel(name + " (%)", fontsize = 14)


    ax1.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0)
    ax1.set_title("Mortality Rate. vs "+name)
    ax1.set_ylabel("Mortality Rate (%)")
    ax1.set_facecolor('white')


    ax2.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0)
    ax2.set_title("Incident Rate. vs "+name)
    ax2.set_ylabel("Incident Rate (%)")
    ax2.set_facecolor('white')
    fig.set_facecolor('white')
```

```
for i in lst:
    get_line_plot(data, i)

data.corr()
```

Out[51]:

	Mortality Rate	Incident Rate	Elderly Population	Adult Education	Health Spending	GDP	Passenger Traffic by Air
Mortality Rate	1.000000	-0.157312	-0.321151	-0.369160	-0.170369	-0.308314	0.216003
Incident Rate	-0.157312	1.000000	0.132550	0.101377	0.287759	0.344203	-0.050485
Elderly Population	-0.321151	0.132550	1.000000	0.503240	0.494349	0.428781	-0.131963
Adult Education	-0.369160	0.101377	0.503240	1.000000	0.434959	0.675630	-0.024456
Health Spending	-0.170369	0.287759	0.494349	0.434959	1.000000	0.408025	0.295907
GDP	-0.308314	0.344203	0.428781	0.675630	0.408025	1.000000	0.024050
Passenger Traffic by Air	0.216003	-0.050485	-0.131963	-0.024456	0.295907	0.024050	1.000000





Interpretation

GDP

The scatter plots display values for the mortality rate and GDP per capita, and the incident rate and GDP per capita. Additionally the line of best fit shows the linear relationship between the variables. The scatter plot of the mortality rate and the GDP per capita shows a negative relationship, i.e., the mortality rate decreases as a country's GDP increases. According to the correlation table in the output, the mortality rate decreases by 0.31% as the GDP per capita increases by 1000 USD. Also, the incident rate and the GDP per capita has a positive relationship. The incident rate increases by 0.34% as the GDP per capita increases by 1000 USD.

Passenger Traffic by Air

The scatter plot shows the relationship between the mortality rate and the number of passengers carried by flights per year, and the incident rate and the number of passengers carried by flights per year. According to the diagrams, there exists a positive relationship between the mortality rate and the number of passenger carried by flights.The a country's mortality rate increases by 0.22% as the number of passenger carried by flights increases by 1 million passengers. As well, the incident rate and the number of passenger carried have a negative relationship. A country's incident rate decreases by 0.05% as 1 million more passengers are carried by flights.

Yet, for both data, the linear relationship illustrated by the scatter plot and the correlation table cannot be simply accepted due to the possible outliers and the hypothesis testings have to be conducted if those numerial values representing the relationships have statistical significances.

Conclusion

The project is conducted to find a meaningful relationship between socioeconomic factors and the incident rate or the mortality rate of COVID-19. The socioeconomic factors the project

specifically focuses on are the ratio of elderly population to the total population, the ratio of the adult population that completed post-secondary education, the amount of total health spending per capita, GDP per capita, and the number of passengers carried by flight per year of a country. The data are obtained from 43 different countries across the world, where most of them are members of OECD and they include both emerging markets and developed markets.

The reason why various socioeconomic factors are considered is that the corona virus shows a tendency that it is highly contagious, while not too fatal. However, still, it has caused numerous deaths and economic loss in a tremendous scale. Additionally, experts warn the world that a pandemic, like this one, will surely happen in the future and will appear more frequently. Therefore, it is worth to look at socioeconomic characteristics of countries and compare how these characteristics have affected their societies. Furthermore, by finding the relationships, it is expected to reform or develop some old and new policies to minimize the future loss when such pandemic hits the world again.

So far, we have observed that the mortality rate and socioeconomic factors, except the number of passengers carried by a flight, have a negative relationship, and opposite results for the relationship between the incident rate and socioeconomic factors, except the number of passengers carried by flights. However, it the relationships may not represent the data properly and may not be statistically significant. But, just by the findings so far, we can conclude that to minimize the mortality rate of infectious diseases, it is essential to invest in developing policies to improve public education, health spending, and GDP. For the incident rate, it is not clear if doing the opposite of solution given to miniize the mortality rate would be effective. However, the findings show that there is a tendency of a higher incident rate in a developed market. Therefore, it gives a clue that developed markets must come up with some case specific policies to minimize the loss from a high incident rate.