Life Expectancy: A Data-Driven Exploration and Prediction

Milestone: Data Exploration and Visualization

Group 13

Sai Varun Kumar Namburi

Prajwal Srinivas

namburi.sai@northeastern.edu

srinivas.pra@northeastern.edu

Percentage of Effort Contributed by Student1: 50%

Percentage of Effort Contributed by Student2: 50%

Signature of Student 1: Sai Varun

Signature of Student 2: Prajwal

Date of Submission: 5th Mar 2023

Data Source: This dataset is taken from kaggle.com
https://www.kaggle.com/code/philbowman212/life-expectancy-exploratory-data-analysis/data

Data Description:

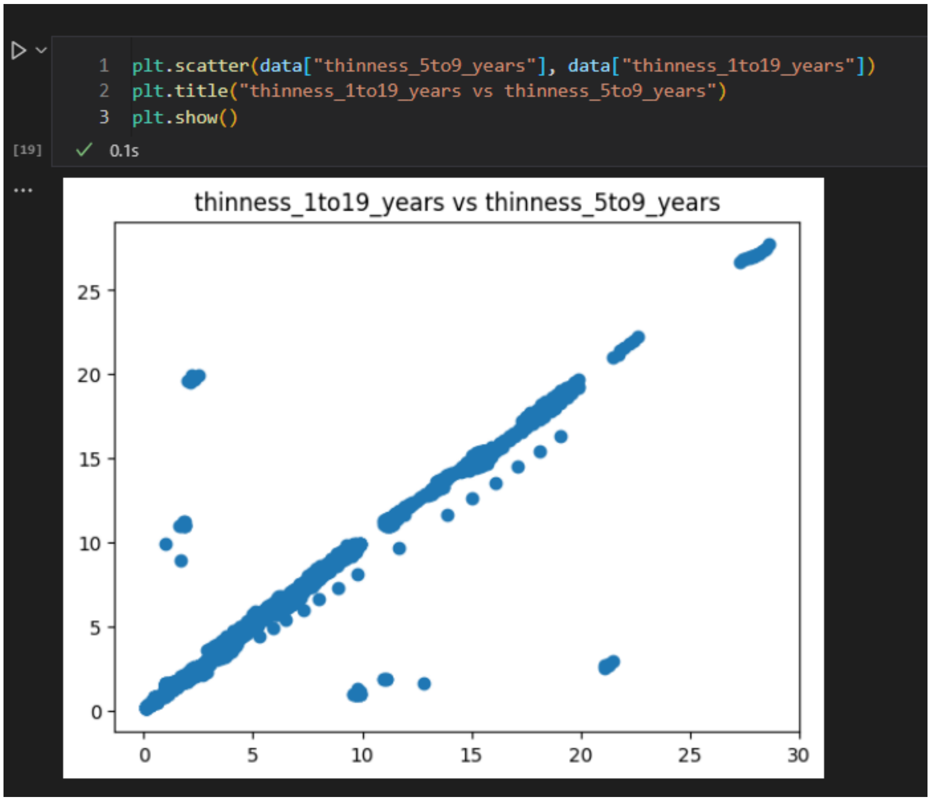
The World Health Organization's Global Health Observatory (GHO) maintains records of the health status and related factors of all countries. The data concerning life expectancy and health factors for 193 countries were obtained from the WHO's Global Health Observatory data repository. It was noted that over the past 15 years, there has been significant progress in the health sector, leading to a significant improvement in human mortality rates, particularly in developing nations compared to the last 30 years. In this project, data from the years 2000 to 2015 for 193 countries were selected for further analysis. In this dataset, we have 22 columns as below

(Country, Year, Status, Life expectancy, Adult Mortality, infant deaths, Alcohol, percentage expenditure, Hepatitis B, Measles, BMI, under-five deaths, Polio, Total expenditure, Diphtheria, HIV/AIDS, GDP, Population, thinness 1-19 years, thinness 5-9 years, Income composition of resources, Schooling)

Data Exploration:

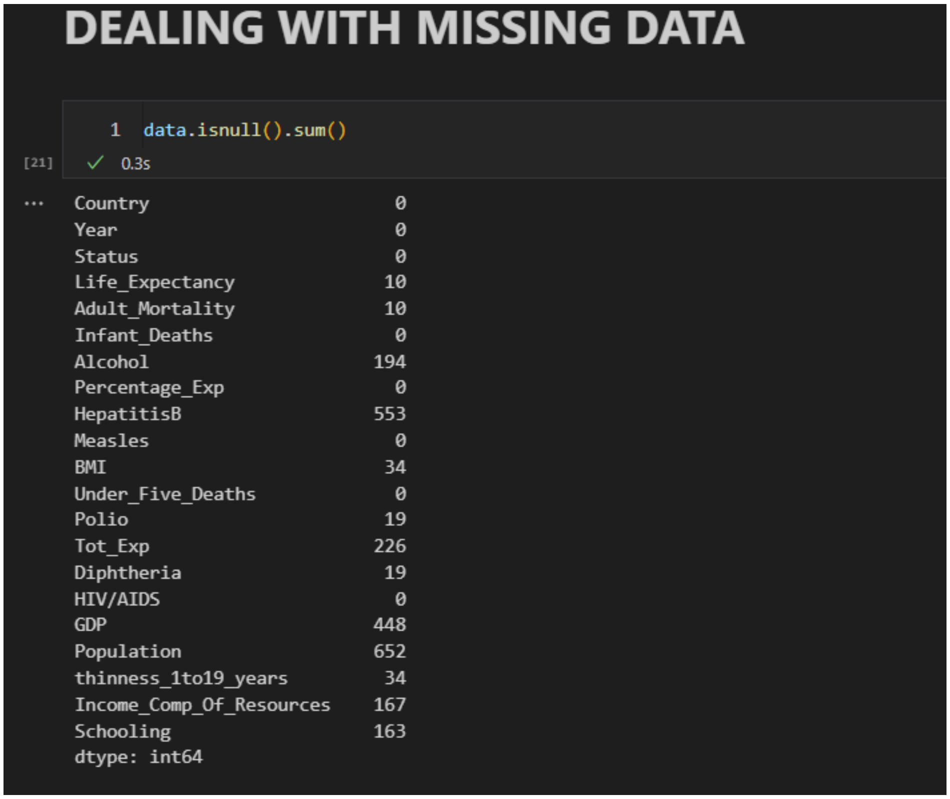
Observation

- Life Expectancy has a negative correlation with adult mortality
- Life Expectancy has a strong correlation with Schooling and Income composition of resources
- The strong correlation between thinness_1to19_years and thinness_5to9_years
- There is a non-negligible correlation between Life Expectancy and BMI and body diseases



```
plt.scatter(data["thinness_5to9_years"], data["thinness_1to19_years"])
plt.title("thinness_1to19_years vs thinness_5to9_years")
plt.show()
```

Dealing with missing data:

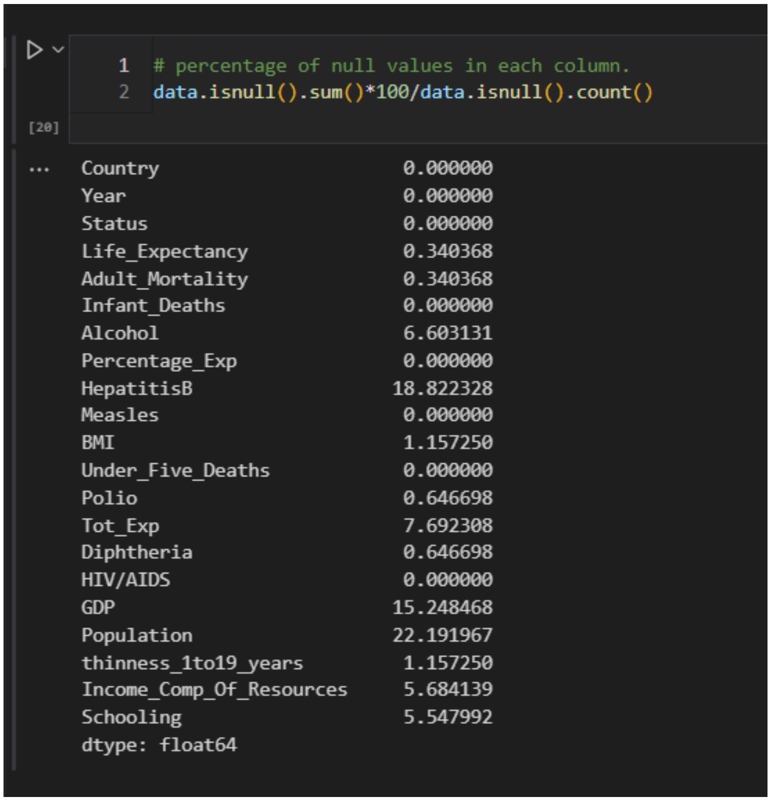


DEALING WITH MISSING DATA

```python
data.isnull().sum()
```

```
Country                      0
Year                         0
Status                       0
Life_Expectancy             10
Adult_Mortality             10
Infant_Deaths                0
Alcohol                    194
Percentage_Exp               0
HepatitisB                 553
Measles                      0
BMI                         34
Under_Five_Deaths            0
Polio                       19
Tot_Exp                    226
Diphtheria                  19
HIV/AIDS                     0
GDP                        448
Population                 652
thinness_1to19_years        34
Income_Comp_Of_Resources   167
Schooling                  163
dtype: int64
```

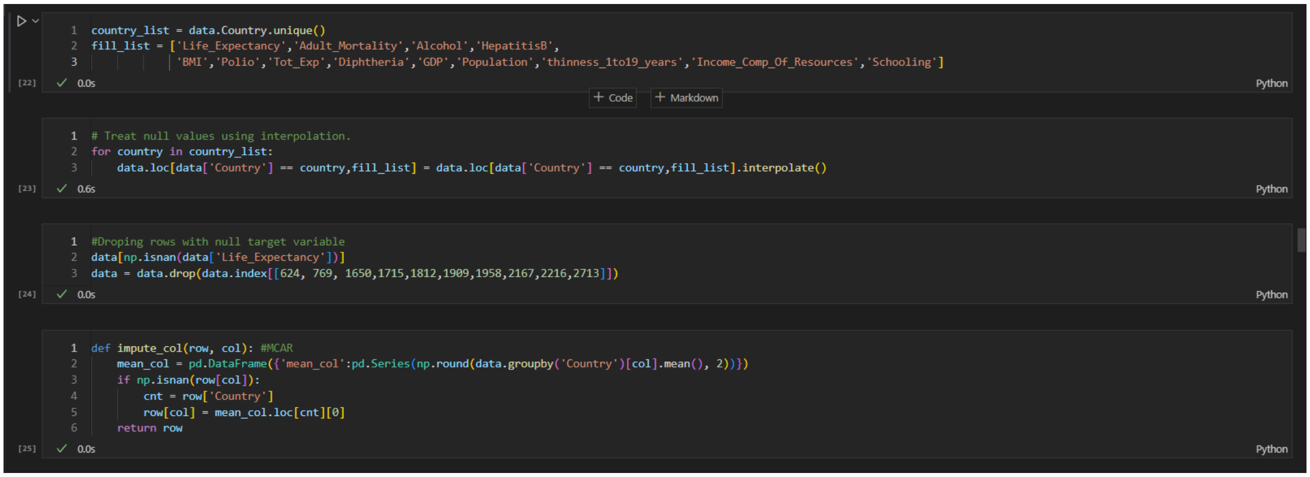
Percentage of Null Values for each column:



```python
# percentage of null values in each column.
data.isnull().sum()*100/data.isnull().count()
```

```
Country                      0.000000
Year                         0.000000
Status                       0.000000
Life_Expectancy              0.340368
Adult_Mortality              0.340368
Infant_Deaths                0.000000
Alcohol                      6.603131
Percentage_Exp               0.000000
HepatitisB                  18.822328
Measles                      0.000000
BMI                          1.157250
Under_Five_Deaths            0.000000
Polio                        0.646698
Tot_Exp                      7.692308
Diphtheria                   0.646698
HIV/AIDS                     0.000000
GDP                         15.248468
Population                  22.191967
thinness_1to19_years         1.157250
Income_Comp_Of_Resources     5.684139
Schooling                    5.547992
dtype: float64
```

Treat null values using interpolation:



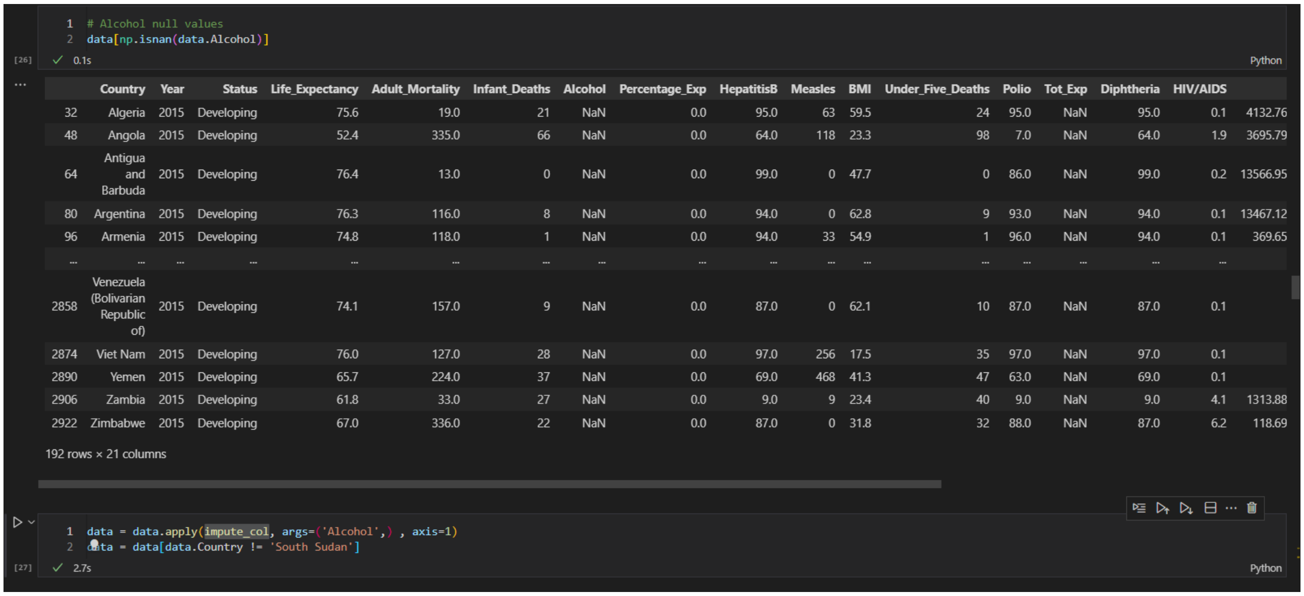
```
country_list = data.Country.unique()
fill_list = ['Life_Expectancy','Adult_Mortality','Alcohol','HepatitisB',
             'BMI','Polio','Tot_Exp','Diphtheria','GDP','Population','thinness_1to19_years','Income_Comp_Of_Resources','Schooling']
```

```
# Treat null values using interpolation.
for country in country_list:
    data.loc[data['Country'] == country,fill_list] = data.loc[data['Country'] == country,fill_list].interpolate()
```

```
#Droping rows with null target variable
data[np.isnan(data['Life_Expectancy'])]
data = data.drop(data.index[[624, 769, 1650,1715,1812,1909,1958,2167,2216,2713]])
```

```
def impute_col(row, col): #MCAR
    mean_col = pd.DataFrame({'mean_col':pd.Series(np.round(data.groupby('Country')[col].mean(), 2))})
    if np.isnan(row[col]):
        cnt = row['Country']
        row[col] = mean_col.loc[cnt][0]
    return row
```

Alcohol null values and imputing it:



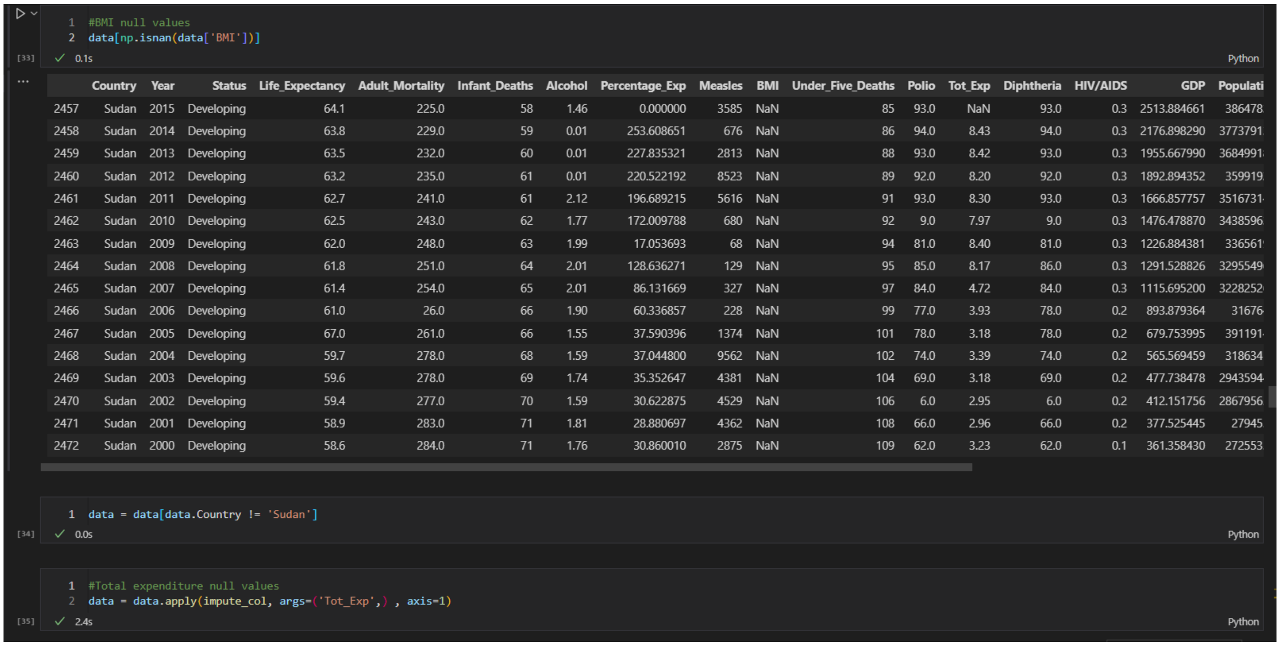
```
# Alcohol null values
data[np.isnan(data.Alcohol)]
```

	Country	Year	Status	Life_Expectancy	Adult_Mortality	Infant_Deaths	Alcohol	Percentage_Exp	HepatitisB	Measles	BMI	Under_Five_Deaths	Polio	Tot_Exp	Diphtheria	HIV/AIDS	
32	Algeria	2015	Developing	75.6	19.0	21	NaN	0.0	95.0	63	59.5	24	95.0	NaN	95.0	0.1	4132.76
48	Angola	2015	Developing	52.4	335.0	66	NaN	0.0	64.0	118	23.3	98	7.0	NaN	64.0	1.9	3695.79
64	Antigua and Barbuda	2015	Developing	76.4	13.0	0	NaN	0.0	99.0	0	47.7	0	86.0	NaN	99.0	0.2	13566.95
80	Argentina	2015	Developing	76.3	116.0	8	NaN	0.0	94.0	0	62.8	9	93.0	NaN	94.0	0.1	13467.12
96	Armenia	2015	Developing	74.8	118.0	1	NaN	0.0	94.0	33	54.9	1	96.0	NaN	94.0	0.1	369.65
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2858	Venezuela (Bolivarian Republic of)	2015	Developing	74.1	157.0	9	NaN	0.0	87.0	0	62.1	10	87.0	NaN	87.0	0.1	
2874	Viet Nam	2015	Developing	76.0	127.0	28	NaN	0.0	97.0	256	17.5	35	97.0	NaN	97.0	0.1	
2890	Yemen	2015	Developing	65.7	224.0	37	NaN	0.0	69.0	468	41.3	47	63.0	NaN	69.0	0.1	
2906	Zambia	2015	Developing	61.8	33.0	27	NaN	0.0	9.0	9	23.4	40	9.0	NaN	9.0	4.1	1313.88
2922	Zimbabwe	2015	Developing	67.0	336.0	22	NaN	0.0	87.0	0	31.8	32	88.0	NaN	87.0	6.2	118.69

192 rows × 21 columns

```
data = data.apply(impute_col, args=('Alcohol',) , axis=1)
data = data[data.Country != 'South Sudan']
```

BMI null values::



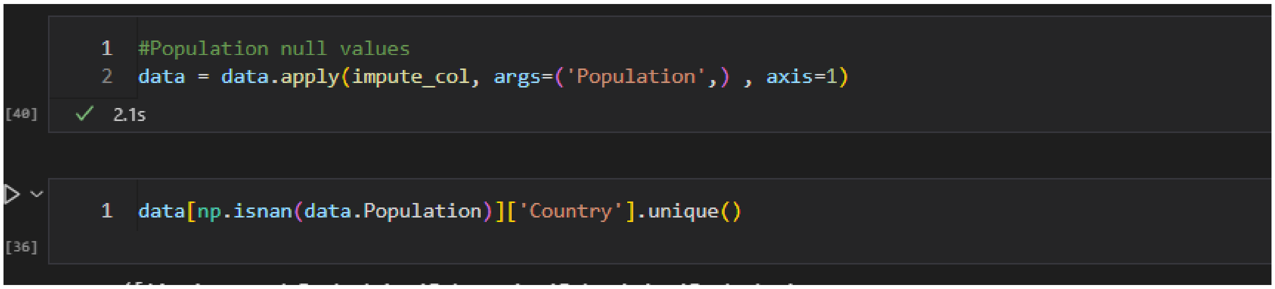
[33]
```
#BMI null values
data[np.isnan(data['BMI'])]
```
✓ 0.1s — Python

	Country	Year	Status	Life_Expectancy	Adult_Mortality	Infant_Deaths	Alcohol	Percentage_Exp	Measles	BMI	Under_Five_Deaths	Polio	Tot_Exp	Diphtheria	HIV/AIDS	GDP	Populati
2457	Sudan	2015	Developing	64.1	225.0	58	1.46	0.000000	3585	NaN	85	93.0	NaN	93.0	0.3	2513.884661	386478
2458	Sudan	2014	Developing	63.8	229.0	59	0.01	253.608651	676	NaN	86	94.0	8.43	94.0	0.3	2176.898290	3773791
2459	Sudan	2013	Developing	63.5	232.0	60	0.01	227.835321	2813	NaN	88	93.0	8.42	93.0	0.3	1955.667990	3684991
2460	Sudan	2012	Developing	63.2	235.0	61	0.01	220.522192	8523	NaN	89	92.0	8.20	92.0	0.3	1892.894352	359919
2461	Sudan	2011	Developing	62.7	241.0	61	2.12	196.689215	5616	NaN	91	93.0	8.30	93.0	0.3	1666.857757	3516731
2462	Sudan	2010	Developing	62.5	243.0	62	1.77	172.009788	680	NaN	92	9.0	7.97	9.0	0.3	1476.478870	3438596
2463	Sudan	2009	Developing	62.0	248.0	63	1.99	17.053693	68	NaN	94	81.0	8.40	81.0	0.3	1226.884381	336561
2464	Sudan	2008	Developing	61.8	251.0	64	2.01	128.636271	129	NaN	95	85.0	8.17	86.0	0.3	1291.528826	3295549
2465	Sudan	2007	Developing	61.4	254.0	65	2.01	86.131669	327	NaN	97	84.0	4.72	84.0	0.3	1115.695200	3228252
2466	Sudan	2006	Developing	61.0	26.0	66	1.90	60.336857	228	NaN	99	77.0	3.93	78.0	0.2	893.879364	31676
2467	Sudan	2005	Developing	67.0	261.0	66	1.55	37.590396	1374	NaN	101	78.0	3.18	78.0	0.2	679.753995	391191
2468	Sudan	2004	Developing	59.7	278.0	68	1.59	37.044800	9562	NaN	102	74.0	3.39	74.0	0.2	565.569459	318634
2469	Sudan	2003	Developing	59.6	278.0	69	1.74	35.352647	4381	NaN	104	69.0	3.18	69.0	0.2	477.738478	2943594
2470	Sudan	2002	Developing	59.4	277.0	70	1.59	30.622875	4529	NaN	106	6.0	2.95	6.0	0.2	412.151756	2867956
2471	Sudan	2001	Developing	58.9	283.0	71	1.81	28.880697	4362	NaN	108	66.0	2.96	66.0	0.2	377.525445	27945
2472	Sudan	2000	Developing	58.6	284.0	71	1.76	30.860010	2875	NaN	109	62.0	3.23	62.0	0.1	361.358430	272553

[34]
```
data = data[data.Country != 'Sudan']
```
✓ 0.0s — Python

[35]
```
#Total expenditure null values
data = data.apply(impute_col, args=('Tot_Exp',) , axis=1)
```
✓ 2.4s — Python

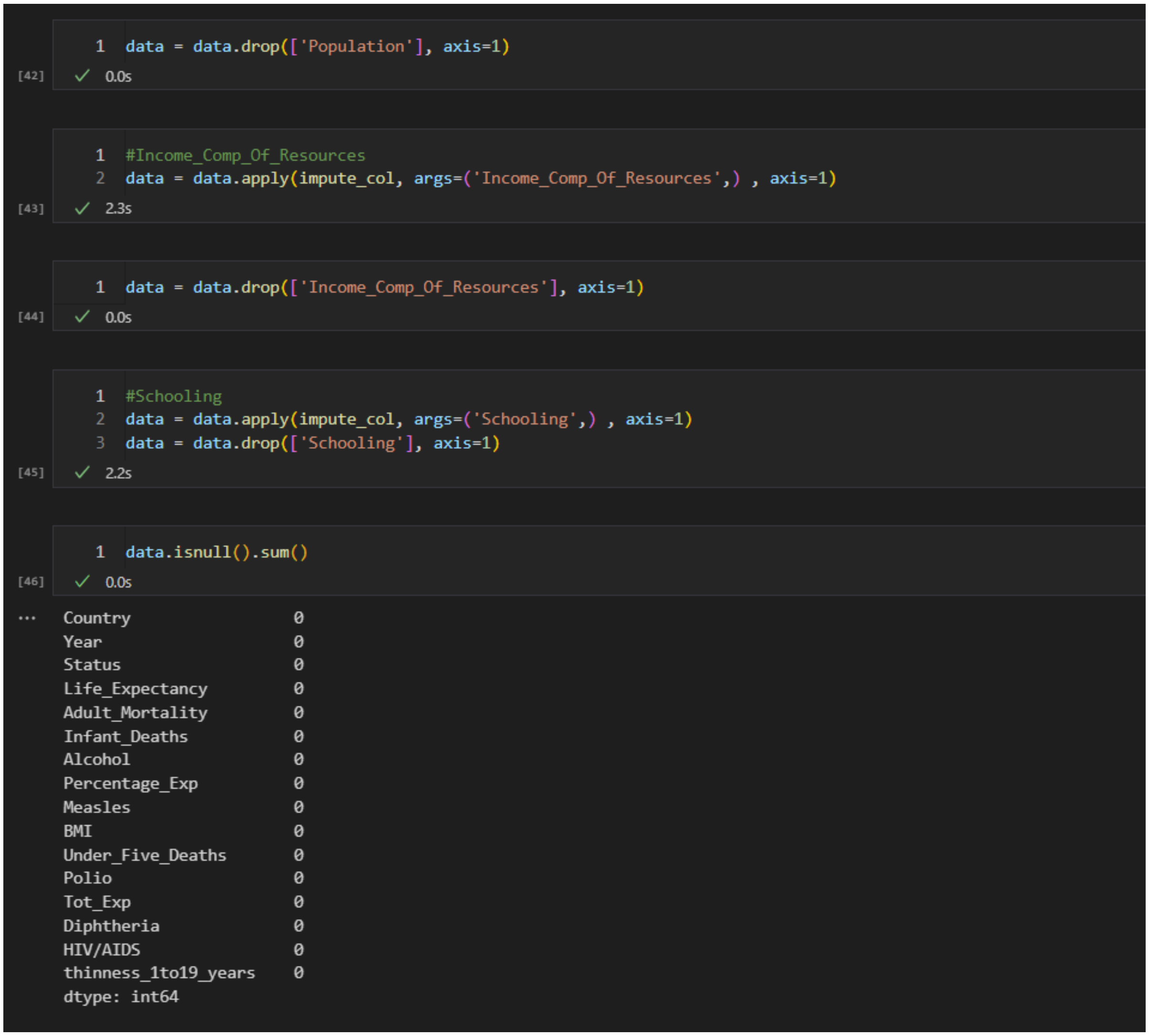
Impute the total expenditure null values
Population null values and imputing it with mean



[40]
```
#Population null values
data = data.apply(impute_col, args=('Population',) , axis=1)
```
✓ 2.1s

[36]
```
data[np.isnan(data.Population)]['Country'].unique()
```

Removing the unnecessary columns and imputing it with mean



```
[42]
data = data.drop(['Population'], axis=1)
✓ 0.0s
```

```
[43]
#Income_Comp_Of_Resources
data = data.apply(impute_col, args=('Income_Comp_Of_Resources',) , axis=1)
✓ 2.3s
```

```
[44]
data = data.drop(['Income_Comp_Of_Resources'], axis=1)
✓ 0.0s
```

```
[45]
#Schooling
data = data.apply(impute_col, args=('Schooling',) , axis=1)
data = data.drop(['Schooling'], axis=1)
✓ 2.2s
```

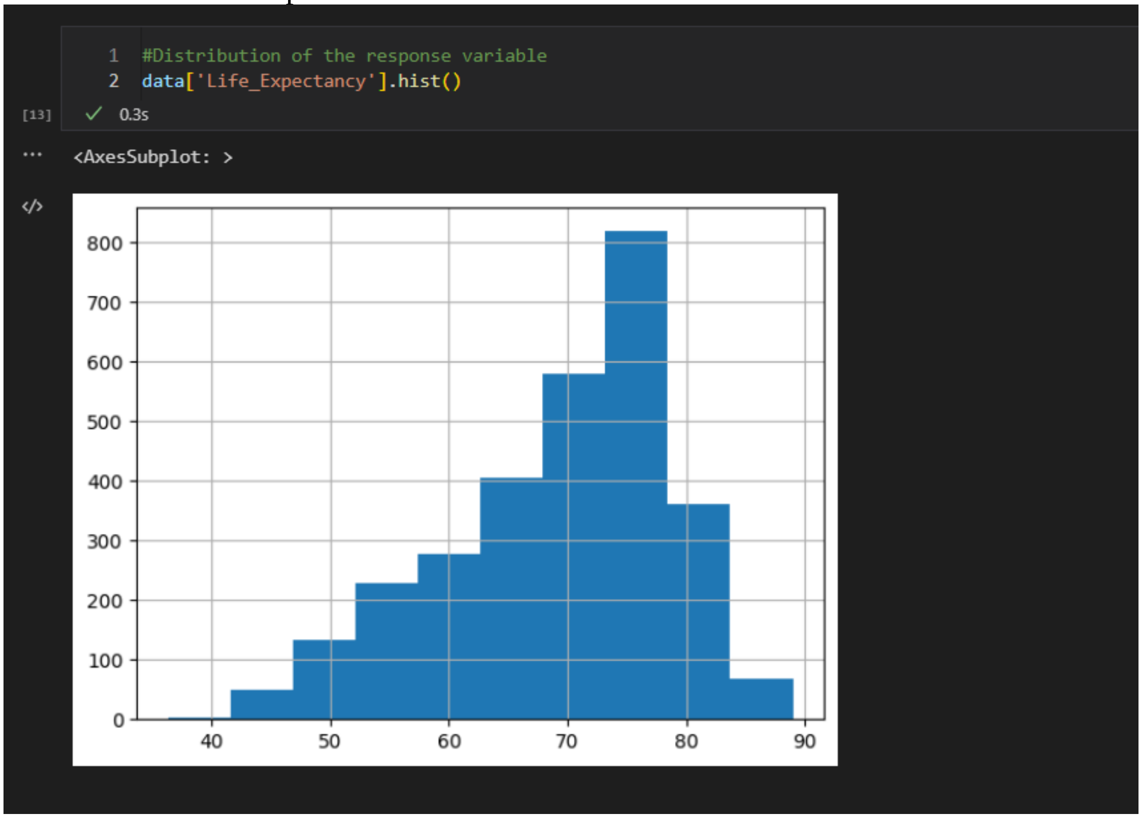
```
[46]
data.isnull().sum()
✓ 0.0s
```

```
Country                 0
Year                    0
Status                  0
Life_Expectancy         0
Adult_Mortality         0
Infant_Deaths           0
Alcohol                 0
Percentage_Exp          0
Measles                 0
BMI                     0
Under_Five_Deaths       0
Polio                   0
Tot_Exp                 0
Diphtheria              0
HIV/AIDS                0
thinness_1to19_years    0
dtype: int64
```

Above we can see that now we don’t have any null values

Data Visualizations:

Distribution of the response variables

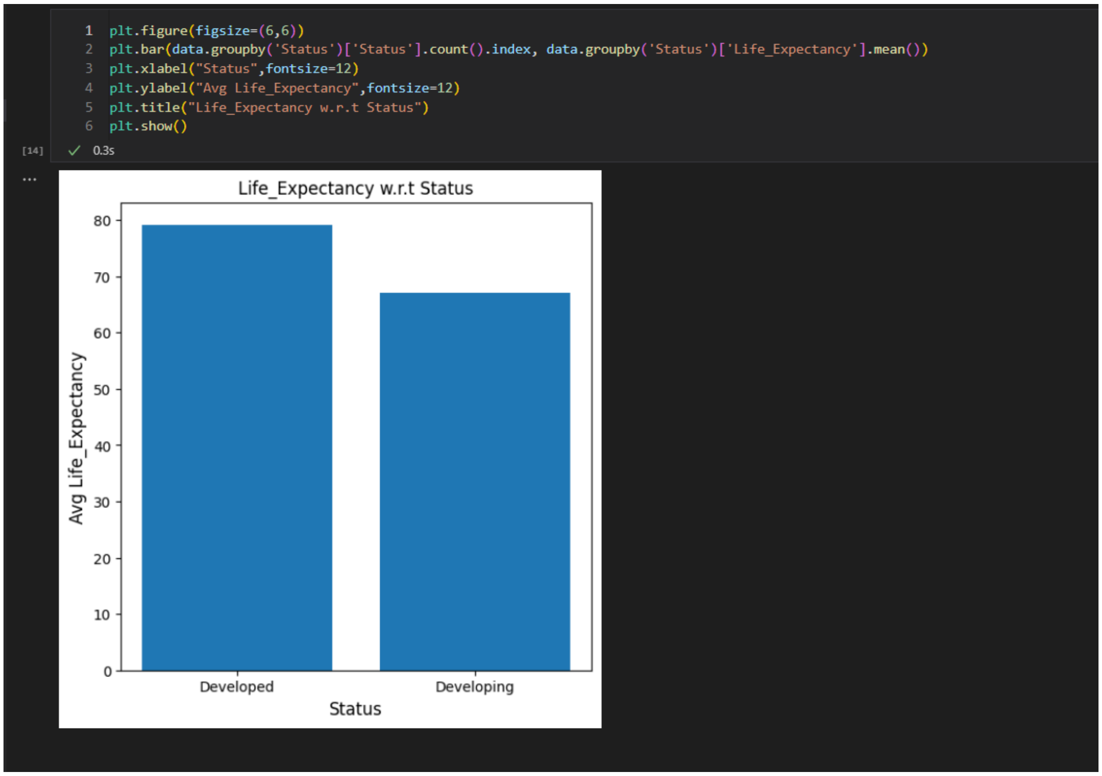


```
#Distribution of the response variable
data['Life_Expectancy'].hist()
```

```
[13] ✓ 0.3s
```

```
<AxesSubplot: >
```

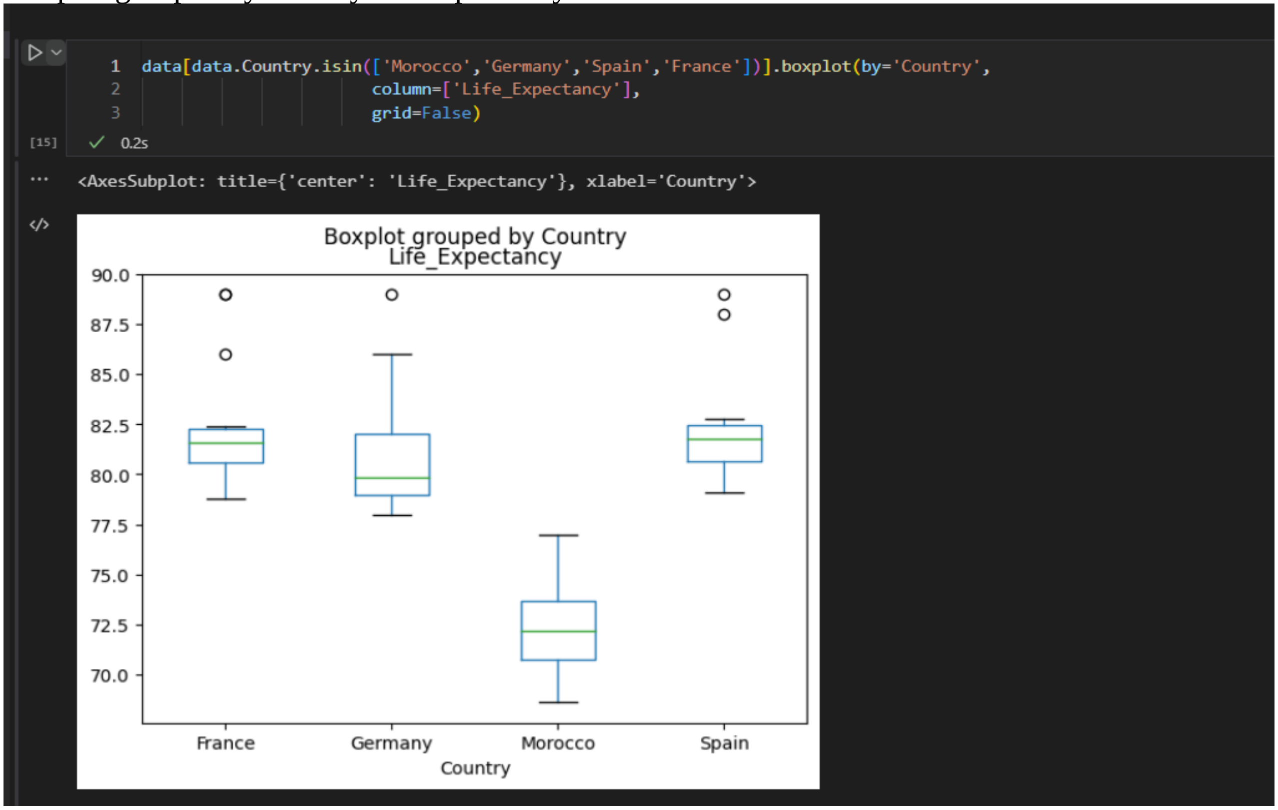
Grouping the data based on status and avg life expectancy



```
plt.figure(figsize=(6,6))
plt.bar(data.groupby('Status')['Status'].count().index, data.groupby('Status')['Life_Expectancy'].mean())
plt.xlabel("Status",fontsize=12)
plt.ylabel("Avg Life_Expectancy",fontsize=12)
plt.title("Life_Expectancy w.r.t Status")
plt.show()
```

```
[14] ✓ 0.3s
```

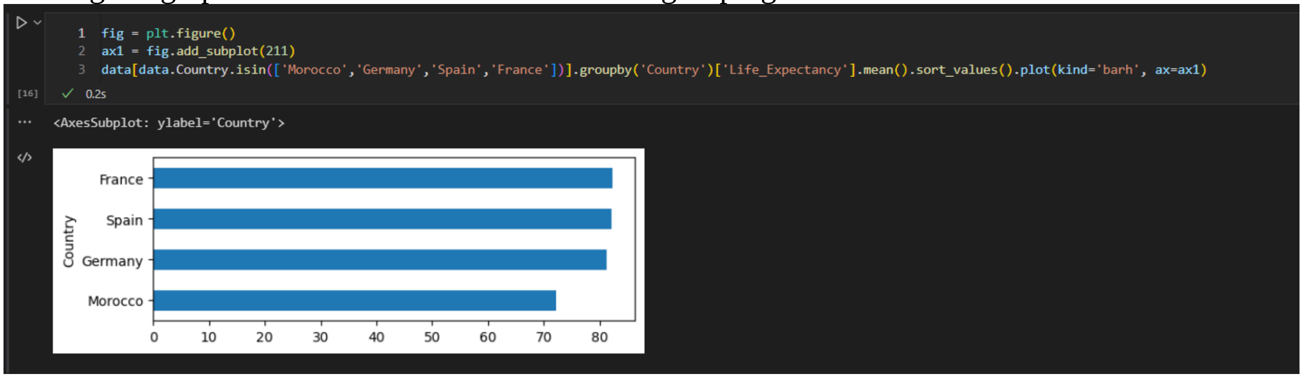
Boxplot grouped by country life expectancy



```
data[data.Country.isin(['Morocco','Germany','Spain','France'])].boxplot(by='Country',
                                                                column=['Life_Expectancy'],
                                                                grid=False)
```

```
<AxesSubplot: title={'center': 'Life_Expectancy'}, xlabel='Country'>
```

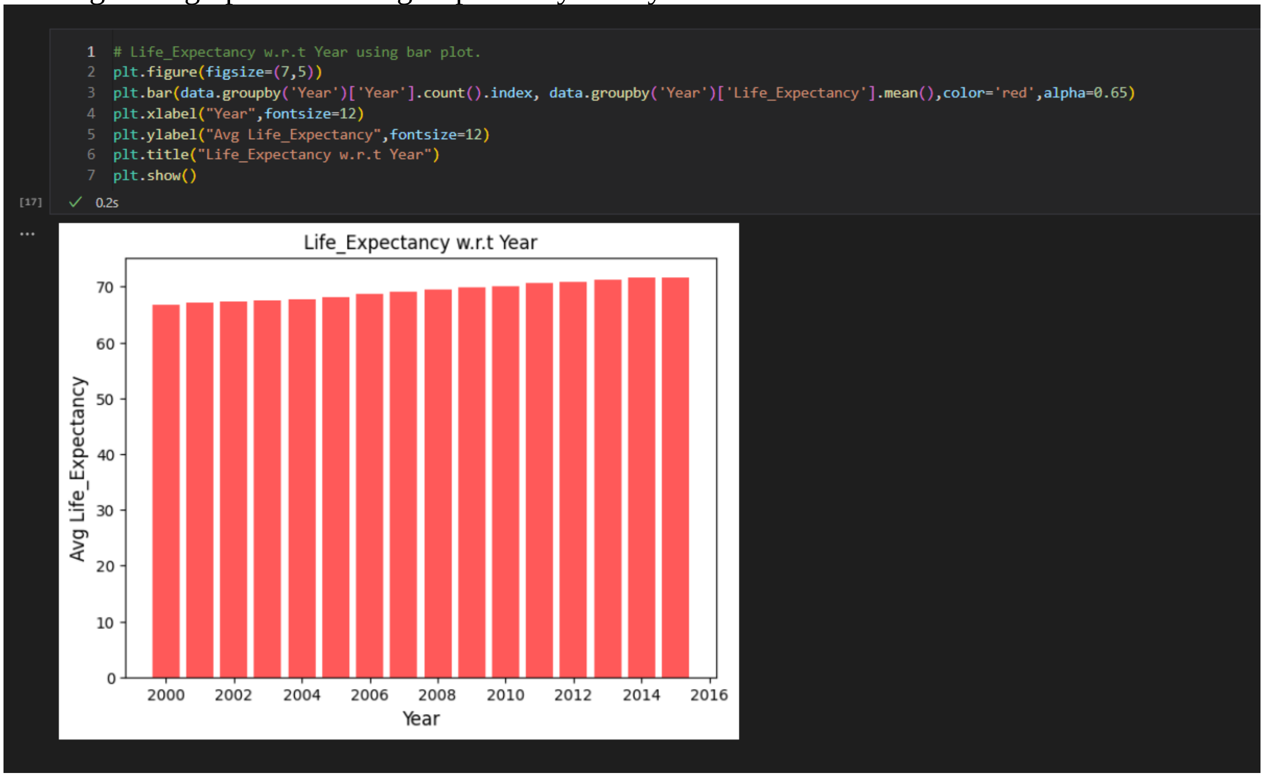
Plotting the graph for the below four countries and grouping it with mean



```
fig = plt.figure()
ax1 = fig.add_subplot(211)
data[data.Country.isin(['Morocco','Germany','Spain','France'])].groupby('Country')['Life_Expectancy'].mean().sort_values().plot(kind='barh', ax=ax1)
```

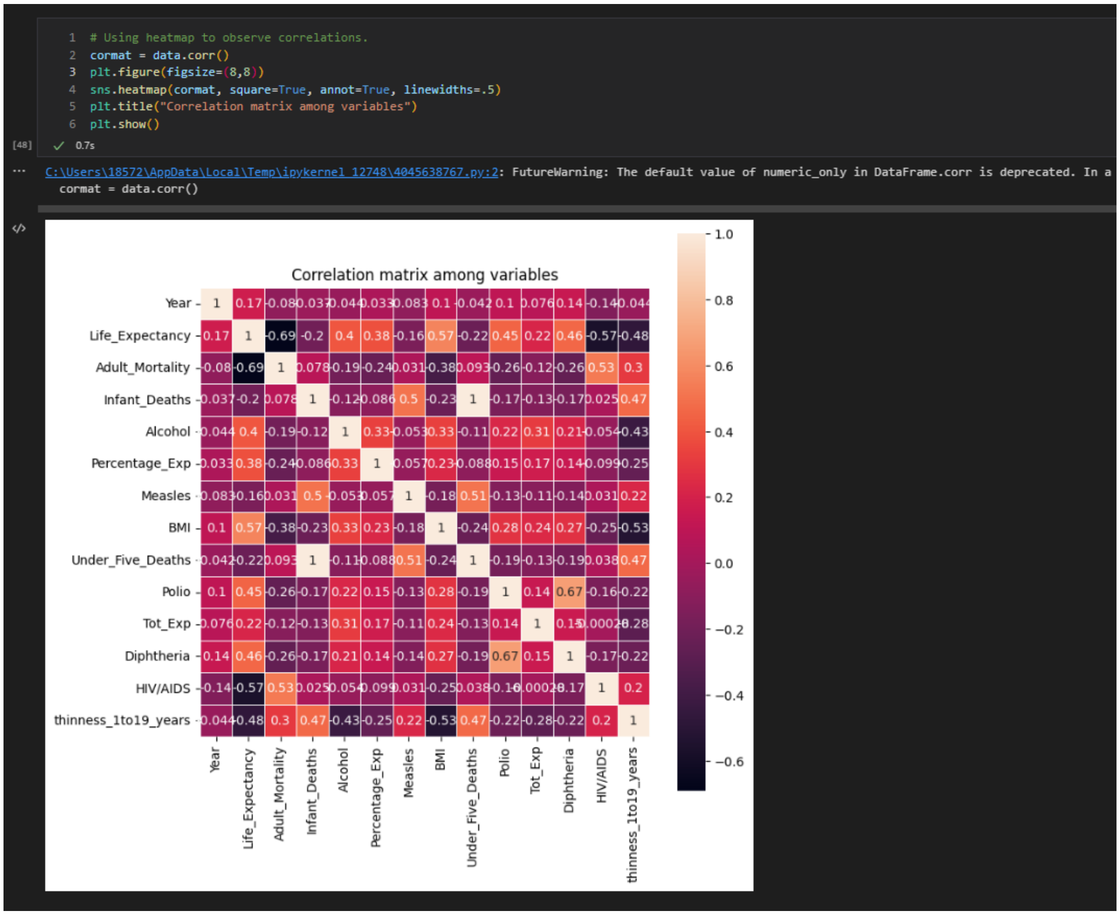
```
<AxesSubplot: ylabel='Country'>
```

Plotting a bar graph where we group data by each year



```
# Life_Expectancy w.r.t Year using bar plot.
plt.figure(figsize=(7,5))
plt.bar(data.groupby('Year')['Year'].count().index, data.groupby('Year')['Life_Expectancy'].mean(),color='red',alpha=0.65)
plt.xlabel("Year",fontsize=12)
plt.ylabel("Avg Life_Expectancy",fontsize=12)
plt.title("Life_Expectancy w.r.t Year")
plt.show()
```

Finding a covariance and plotting a heatmap among variables

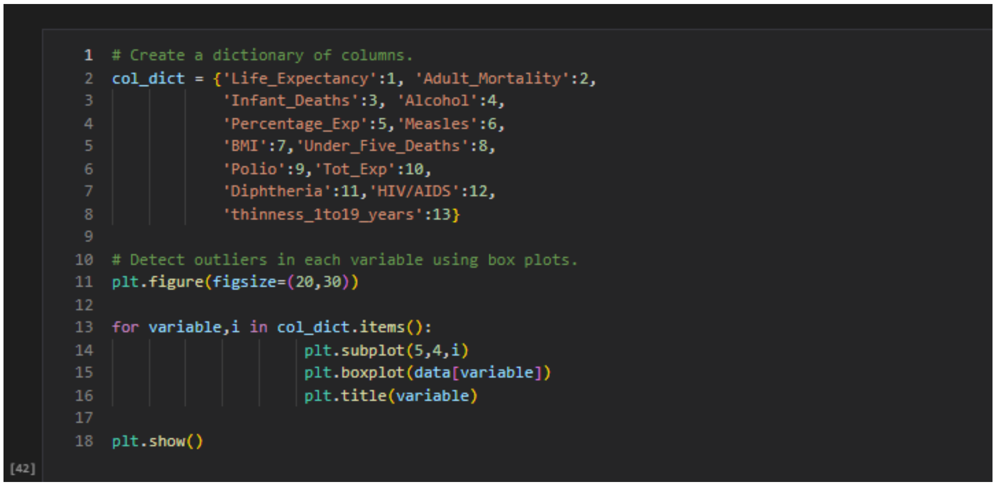


```
# Using heatmap to observe correlations.
cormat = data.corr()
plt.figure(figsize=(8,8))
sns.heatmap(cormat, square=True, annot=True, linewidths=.5)
plt.title("Correlation matrix among variables")
plt.show()
```

```
C:\Users\18572\AppData\Local\Temp\ipykernel_12748\4045638767.py:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a
  cormat = data.corr()
```

[48]
✓ 0.7s
Correlation matrix among variables
Year 1 0.17 -0.08 0.037 0.044 0.033 0.083 0.1 -0.042 0.1 0.076 0.14 -0.14 0.044
Life_Expectancy 0.17 1 -0.69 -0.2 0.4 0.38 -0.16 0.57 -0.22 0.45 0.22 0.46 -0.57 -0.48
Adult_Mortality -0.08 -0.69 1 0.078 -0.19 -0.24 0.031 -0.38 0.093 -0.26 -0.12 -0.26 0.53 0.3
Infant_Deaths 0.037 -0.2 0.078 1 -0.12 0.086 0.5 -0.23 1 -0.17 -0.13 -0.17 0.025 0.47
Alcohol 0.044 0.4 -0.19 -0.12 1 0.33 -0.053 0.33 -0.11 0.22 0.31 0.21 -0.054 -0.43
Percentage_Exp 0.033 0.38 -0.24 0.086 0.33 1 0.057 0.23 0.088 0.15 0.17 0.14 -0.099 -0.25
Measles 0.083 -0.16 0.031 0.5 -0.053 0.057 1 -0.18 0.51 -0.13 -0.11 -0.14 0.031 0.22
BMI 0.1 0.57 -0.38 -0.23 0.33 0.23 -0.18 1 -0.24 0.28 0.24 0.27 -0.25 -0.53
Under_Five_Deaths 0.042 -0.22 0.093 1 -0.11 0.088 0.51 -0.24 1 -0.19 -0.13 -0.19 0.038 0.47
Polio 0.1 0.45 -0.26 -0.17 0.22 0.15 -0.13 0.28 -0.19 1 0.14 0.67 -0.16 -0.22
Tot_Exp 0.076 0.22 -0.12 -0.13 0.31 0.17 -0.11 0.24 -0.13 0.14 1 0.15 -0.00028 -0.28
Diphtheria 0.14 0.46 -0.26 -0.17 0.21 0.14 -0.14 0.27 -0.19 0.67 0.15 1 -0.17 -0.22
HIV/AIDS -0.14 -0.57 0.53 0.025 0.054 0.099 0.031 -0.25 0.038 -0.16 0.00028 -0.17 1 0.2
thinness_1to19_years 0.044 -0.48 0.3 0.47 -0.43 -0.25 0.22 -0.53 0.47 -0.22 -0.28 -0.22 0.2 1
Year
Life_Expectancy
Adult_Mortality
Infant_Deaths
Alcohol
Percentage_Exp
Measles
BMI
Under_Five_Deaths
Polio
Tot_Exp
Diphtheria
HIV/AIDS
thinness_1to19_years
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

Outliers:

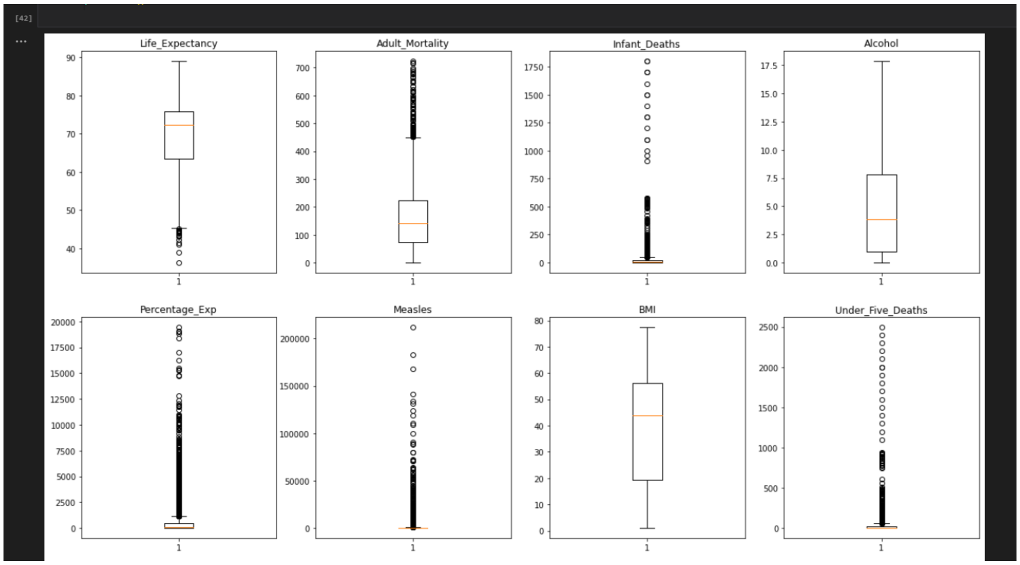


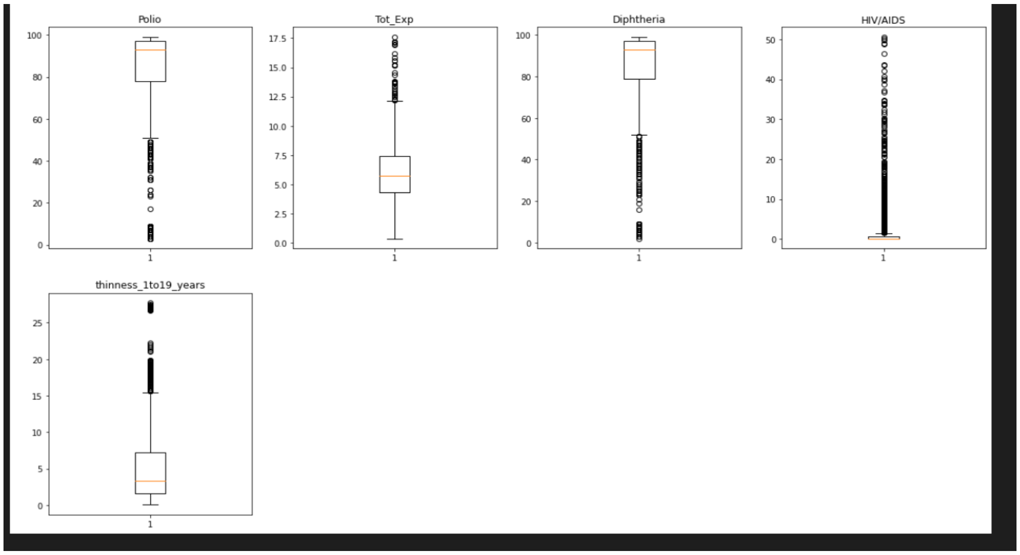
```
# Create a dictionary of columns.
col_dict = {'Life_Expectancy':1, 'Adult_Mortality':2,
            'Infant_Deaths':3, 'Alcohol':4,
            'Percentage_Exp':5,'Measles':6,
            'BMI':7,'Under_Five_Deaths':8,
            'Polio':9,'Tot_Exp':10,
            'Diphtheria':11,'HIV/AIDS':12,
            'thinness_1to19_years':13}

# Detect outliers in each variable using box plots.
plt.figure(figsize=(20,30))

for variable,i in col_dict.items():
                     plt.subplot(5,4,i)
                     plt.boxplot(data[variable])
                     plt.title(variable)

plt.show()
```


[42]
Life_Expectancy
Adult_Mortality
Infant_Deaths
Alcohol
Percentage_Exp
Measles
BMI
Under_Five_Deaths


Polio
Tot_Exp
Diphtheria
HIV/AIDS
thinness_1to19_years