LIFE EXPECTANCY PREDICTION MODEL USING MACHINE LEARNING

Author - Chandan D. Chaudhari

Github - https://github.com/chandanc5525

Dataset Link :- https://raw.githubusercontent.com/chandanc5525/LifeExpectancy_ModelPipeline/mai

1. FEATURE INFORMATION

1) Country 2) Year 3) Status – Developed or Developing status 4) Life expectancy – Life Expectancy in age 5) Adult Mortality – Adult Mortality Rates of both sexes (probability of dying between 15 and 60 years per 1000 population) 6) infant deaths – Number of Infant Deaths per 1000 population 7) Alcohol – Alcohol, recorded per capita (15+) consumption (in litres of pure alcohol) 8) percentage expenditure – Expenditure on health as a percentage of Gross Domestic Product per capita(%) 9) Hepatitis B – Hepatitis B (HepB) immunization coverage among 1-year-olds (%) 10) Measles – Measles – number of reported cases per 1000 population 11) BMI – Average Body Mass Index of entire population 12) under-five deaths – Number of under-five deaths per 1000 population 13) Polio – Polio (Pol3) immunization coverage among 1-year-olds (%) 14) Total expenditure – General government expenditure on health as a percentage of total government expenditure (%) 15) Diphtheria – Diphtheria tetanus toxoid and pertussis (DTP3) immunization coverage among 1-year-olds (%) 16) HIV/AIDS – Deaths per 1 000 live births HIV/AIDS (0-4 years) 17) GDP – Gross Domestic Product per capita (in USD) 18) Population – Population of the country 19) thinness 1-19 years – Prevalence of thinness among children and adolescents for Age 10 to 19 (%) 20) thinness 5-9 years – Prevalence of thinness among children for Age 5 to 9(%) 21) Income composition – Human Development Index in terms of income composition of resources (index ranging from 0 to 1) 22) Schooling – Number of years of Schooling(years)

--- INSTALLING REQUIRED PACKAGES

In [1]:
```
!pip install klib
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: klib in c:\users\user\appdata\roaming\python\pyth
on310\site-packages (1.0.7)
Requirement already satisfied: seaborn<0.13.0,>=0.11.2 in d:\anaconda3\lib\site-
packages (from klib) (0.12.2)
Requirement already satisfied: numpy<2.0.0,>=1.16.3 in d:\anaconda3\lib\site-pac
kages (from klib) (1.23.5)
Requirement already satisfied: pandas<2.0.0,>=1.2.0 in d:\anaconda3\lib\site-pac
kages (from klib) (1.5.3)
Requirement already satisfied: scipy<2.0.0,>=1.1.0 in d:\anaconda3\lib\site-pack
ages (from klib) (1.10.0)
Requirement already satisfied: matplotlib<4.0.0,>=3.0.3 in d:\anaconda3\lib\site
-packages (from klib) (3.7.0)
Requirement already satisfied: Jinja2<4.0.0,>=3.0.3 in c:\users\user\appdata\roa
ming\python\python310\site-packages (from klib) (3.0.3)
Requirement already satisfied: MarkupSafe>=2.0 in d:\anaconda3\lib\site-packages
(from Jinja2<4.0.0,>=3.0.3->klib) (2.1.1)
Requirement already satisfied: kiwisolver>=1.0.1 in d:\anaconda3\lib\site-packag
es (from matplotlib<4.0.0,>=3.0.3->klib) (1.4.4)
Requirement already satisfied: fonttools>=4.22.0 in d:\anaconda3\lib\site-packag
es (from matplotlib<4.0.0,>=3.0.3->klib) (4.25.0)
Requirement already satisfied: python-dateutil>=2.7 in d:\anaconda3\lib\site-pac
kages (from matplotlib<4.0.0,>=3.0.3->klib) (2.8.2)
Requirement already satisfied: cycler>=0.10 in d:\anaconda3\lib\site-packages (f
rom matplotlib<4.0.0,>=3.0.3->klib) (0.11.0)
Requirement already satisfied: contourpy>=1.0.1 in d:\anaconda3\lib\site-package
s (from matplotlib<4.0.0,>=3.0.3->klib) (1.0.5)
Requirement already satisfied: pyparsing>=2.3.1 in d:\anaconda3\lib\site-package
s (from matplotlib<4.0.0,>=3.0.3->klib) (3.0.9)
Requirement already satisfied: packaging>=20.0 in d:\anaconda3\lib\site-packages
(from matplotlib<4.0.0,>=3.0.3->klib) (22.0)
Requirement already satisfied: pillow>=6.2.0 in d:\anaconda3\lib\site-packages
(from matplotlib<4.0.0,>=3.0.3->klib) (9.4.0)
Requirement already satisfied: pytz>=2020.1 in d:\anaconda3\lib\site-packages (f
rom pandas<2.0.0,>=1.2.0->klib) (2022.7)
Requirement already satisfied: six>=1.5 in d:\anaconda3\lib\site-packages (from
python-dateutil>=2.7->matplotlib<4.0.0,>=3.0.3->klib) (1.16.0)
```

--- IMPORT PACKAGES

In [2]:

```python
# Import Python Neccessories Libraries
import mlflow
import os
import numpy as np
import pandas as pd
from sklearn.base import BaseEstimator, TransformerMixin
from sklearn.utils.validation import check_array, check_is_fitted
import math
# Import Data Visualization Libraries
import seaborn as sns
import matplotlib.pyplot as plt
# Import FilterWarnings Library
import warnings
warnings.filterwarnings('ignore')
# Import EDA library
import klib
```

In [3]:

```python
# Import Data using Pandas function

df = pd.read_csv(r'https://raw.githubusercontent.com/chandanc5525/LifeExpectancy_
df.head()
```

Out[3]:

	Country	Year	Status	Adult_Mortality	Infant_Deaths	Alcohol	Percentage_Expenditure	Hepa
0	Afghanistan	2015	Developing	263.0	62	0.01	71.279624	
1	Afghanistan	2014	Developing	271.0	64	0.01	73.523582	
2	Afghanistan	2013	Developing	268.0	66	0.01	73.219243	
3	Afghanistan	2012	Developing	272.0	69	0.01	78.184215	
4	Afghanistan	2011	Developing	275.0	71	0.01	7.097109	

5 rows × 22 columns

In [4]:

```
# Checking the total Rows and Columns in the Dataset

df.shape
```

Out[4]:

```
(2938, 22)
```

In [5]:

```
# Listing the presented columns
df.columns
```

Out[5]:

```
Index(['Country', 'Year', 'Status', '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_Cresources', 'Schooling',
       'Life_expectancy'],
      dtype='object')
```

In [6]:

```
# Checking Missing Data Information
df.isnull().sum()
```

Out[6]:

```
Country                     0
Year                        0
Status                      0
Adult_Mortality            10
Infant_Deaths               0
Alcohol                   194
Percentage_Expenditure      0
Hepatitis_B               553
Measles                     0
BMI                        34
under-five deaths           0
Polio                      19
Total_Expenditure         226
Diphtheria                 19
HIV/AIDS                    0
GDP                       448
Population                652
thinness  1-19 years       34
thinness 5-9 years         34
Income_Cresources         167
Schooling                 163
Life_expectancy            10
dtype: int64
```

OBSERVATIONS 1

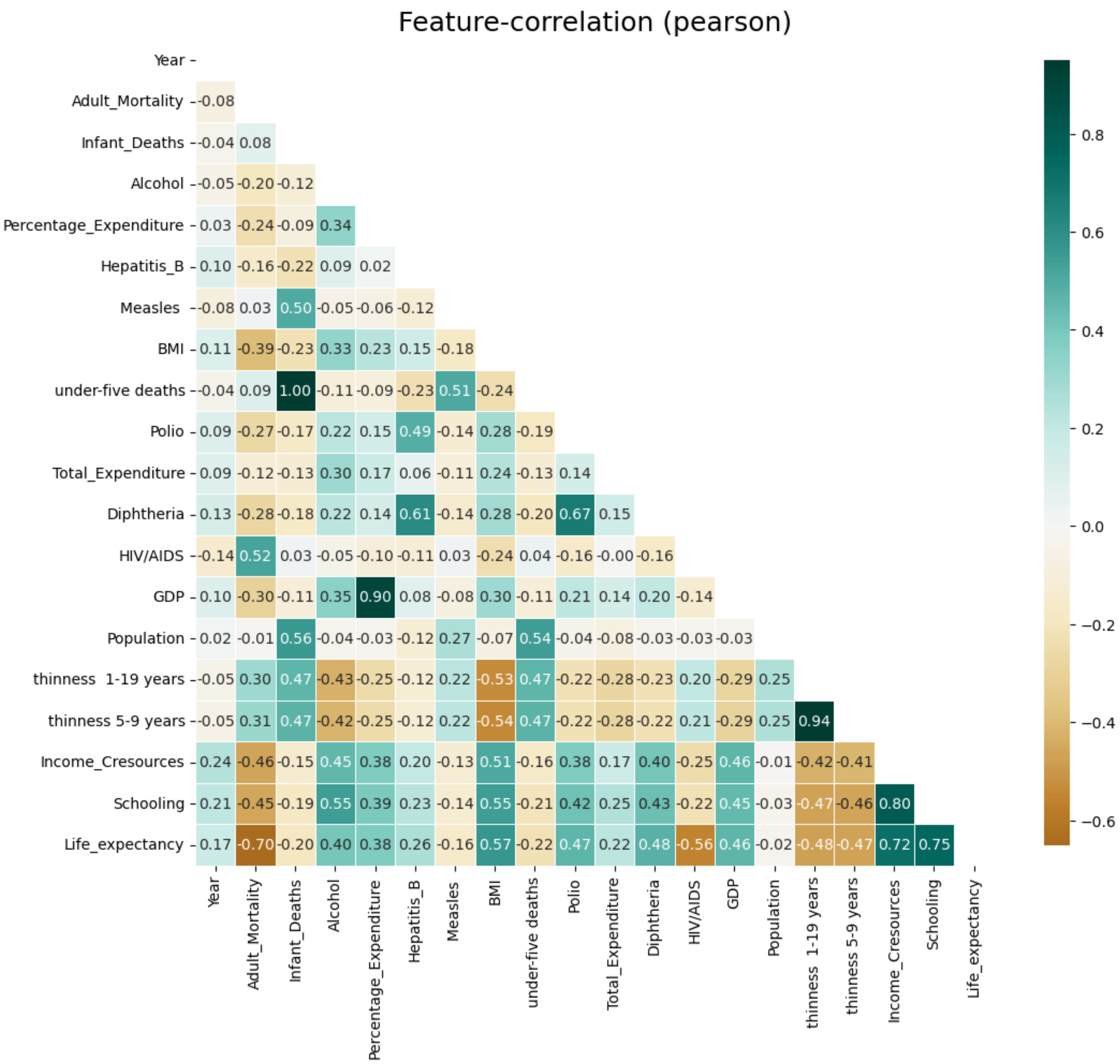
1. The Above Dataset Contains 2938 Rows and 22 Columns.
2. Out of 22 Columns, The LifeExpectancy Column acts as Target Column.

3. There are Total 133 Countries including Developed and Developing, and Few Feature Columns such as Percentage_Expenditure and Measles,Total Expenditure, Population and Schooling.
4. In Order to Evaluate Life Expectancy there are two possible ways:

   [a]. Simply drop all the rows having null values in it, Since this dataset will be for 133 different countries w.r.t Years.

   [b]. We can go for Imputing Method so that Null values can be taken care.

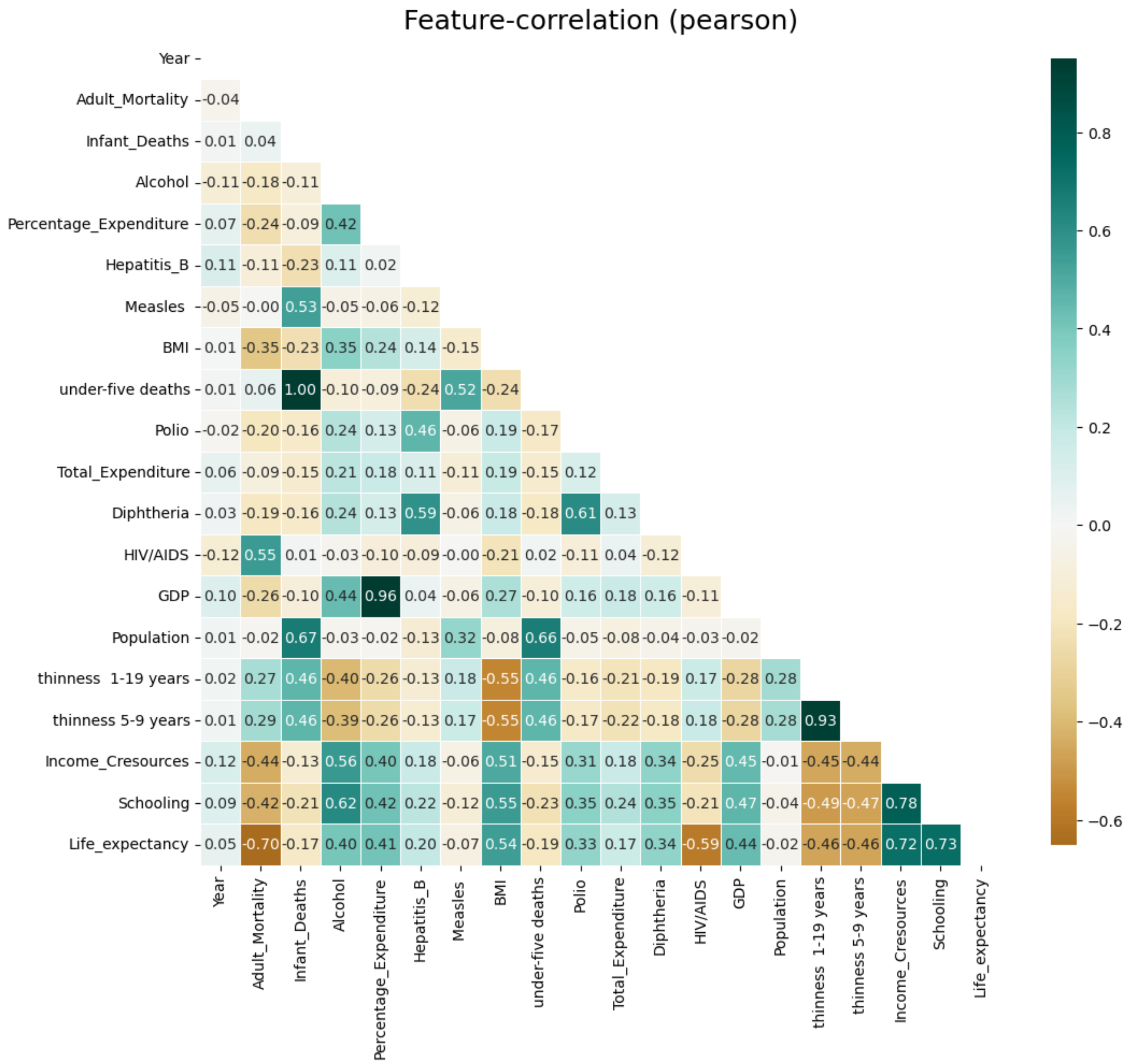
In [7]:
```python
# Correlation Plot without Dropping Null Values
klib.corr_plot(df,annot=True);
```



In [8]:
```python
# Dropping All Null Values in the Dataset
ndf = df.dropna()
ndf.shape
```

Out[8]:
```
(1649, 22)
```

In [9]:
```python
# Correlation Plot After Dropping Null Values
klib.corr_plot(ndf,annot=True);
```



OBSERVATION 2

1. Life Expectancy is found to be Positively Correlated with Following Feature Columns such as -

   [a]. Schooling with 73%

   [b]. Income Composition Resources with 72%

   [c]. GDP with 44%

   [d]. BMI with 54%

   [e]. Percentage Expenditure and Alcohol with 40%

   [f]. Immunization we consider i.e.Hepetitis_B, Polio and Diphtheria having Positive Correlation with Life Expectancy

1. Life Expectancy is found to be Negatively Correlated with Following Feature Columns such as -

   [a]. Adult Mortality with 70%

[b]. HIV/AIDS with 59%

[c]. Thiness 1-19 Years and Thiness 5-9 Years with with 46%

--- TASK 1 : Answers to All Questions Using EDA

QUESTIONARIES:-

1. Does various predicting factors which has been chosen initially really affect Life expectancy? What are the predicting variables actually affecting life expectancy?
2. Should a country having a lower life expectancy value(<65) increase its healthcare expenditure in order to improve its average lifespan?
3. How do Infant and Adult mortality rates affect life expectancy?
4. Does Life Expectancy has a positive or negative correlation with eating habits, lifestyle, exercise, smoking, drinking alcohol etc?
5. What is the impact of schooling on the lifespan of humans?
6. Does Life Expectancy have a positive or negative relationship with drinking alcohol?
7. Do densely populated countries tend to have lower life expectancy?
8. What is the impact of Immunization coverage on Life Expectancy?

Q.] Does various predicting factors which has been chosen initially really affect Life expectancy? What are the predicting variables actually affecting life expectancy?

In [10]:
```
# Evaluating Country Status, Having Life Expectancy Less Than 65 Years
ndf[ndf['Life_expectancy']<65]['Status'].value_counts()
```

Out[10]:
```
Developing    439
Name: Status, dtype: int64
```

In [11]:
```
# # Evaluating Country Status, Having Life Expectancy Greater Than 65 Years
ndf[ndf['Life_expectancy']>65]['Status'].value_counts()
```

Out[11]:
```
Developing    957
Developed     242
Name: Status, dtype: int64
```

In [12]:
```
# Evaluating Country Status, Having Life Expectancy Equal to 65 Years
ndf[ndf['Life_expectancy']==65]['Status'].value_counts()
```

Out[12]:
```
Developing    11
Name: Status, dtype: int64
```

In [13]:
```
# Name of the Country Having Life Expectancy Less Than 65 Years
a = ndf[ndf['Life_expectancy']<65]['Country']
a.unique()
```

Out[13]:
```
array(['Afghanistan', 'Angola', 'Benin', 'Bhutan', 'Botswana',
       'Burkina Faso', 'Burundi', 'Cambodia', 'Cameroon',
       'Central African Republic', 'Chad', 'Comoros', 'Djibouti',
       'Equatorial Guinea', 'Eritrea', 'Ethiopia', 'Gabon', 'Ghana',
       'Guinea', 'Guinea-Bissau', 'Haiti', 'India', 'Iraq', 'Kazakhstan',
       'Kenya', 'Kiribati', 'Lesotho', 'Liberia', 'Madagascar', 'Malawi',
       'Mali', 'Mauritania', 'Mongolia', 'Mozambique', 'Myanmar',
       'Namibia', 'Nepal', 'Niger', 'Nigeria', 'Pakistan',
       'Papua New Guinea', 'Russian Federation', 'Rwanda',
       'Sao Tome and Principe', 'Senegal', 'Sierra Leone', 'South Africa',
       'Swaziland', 'Tajikistan', 'Togo', 'Turkmenistan', 'Uganda',
       'Zambia', 'Zimbabwe'], dtype=object)
```

In [14]:
```
# Taking Mean of Feature Columns
b = ndf[ndf['Life_expectancy']<65].mean()
b
```

Out[14]:
```
Year                       2.007959e+03
Adult_Mortality            2.932005e+02
Infant_Deaths              5.905923e+01
Alcohol                    2.703007e+00
Percentage_Expenditure     9.658601e+01
Hepatitis_B                6.997950e+01
Measles                    2.824875e+03
BMI                        2.255626e+01
under-five deaths          8.657631e+01
Polio                      7.140547e+01
Total_Expenditure          5.730752e+00
Diphtheria                 7.250797e+01
HIV/AIDS                   6.770387e+00
GDP                        1.017342e+03
Population                 1.696464e+07
thinness  1-19 years       7.635763e+00
thinness 5-9 years         7.661048e+00
Income_Cresources          4.416446e-01
Schooling                  9.286788e+00
Life_expectancy            5.739408e+01
dtype: float64
```

OBSERVATION 3

1. Developing Countries having Lower Life Expectancy i.e. Less than 65.
2. Income Composition Resources for such countries found to be very poor and also if we compare Total Expenditure for such countries are very less than Percentage Expanditure, Meaning Goverment has to focus more Expenditure in order to improve life expectancy.
3. BMI value is also found to be in Normal range i.e Between 18.5 to 24.5.

Q.] How do Infant and Adult mortality rates affect life expectancy?

In [15]:
```
# Dataset for Developed Countries
developed_country_data = ndf[ndf['Status']=='Developed']
developed_country_data.drop(['Year','Country'],axis=1,inplace=True)
developed_country_data
```

Out[15]:

	Status	Adult_Mortality	Infant_Deaths	Alcohol	Percentage_Expenditure	Hepatitis_B	Measles
113	Developed	6.0	1	9.71	10769.363050	91.0	340
114	Developed	61.0	1	9.87	11734.853810	91.0	158
115	Developed	61.0	1	10.03	11714.998580	91.0	199
116	Developed	63.0	1	10.30	10986.265270	92.0	190
117	Developed	64.0	1	10.52	8875.786493	92.0	70
...	...	...	...	...	...	...	..
2440	Developed	86.0	2	11.12	1934.398154	77.0	152
2506	Developed	54.0	0	7.30	1142.212403	67.0	26
2507	Developed	57.0	0	7.30	1212.666327	67.0	51
2508	Developed	57.0	0	7.40	10947.023270	53.0	30
2509	Developed	58.0	0	7.40	11477.667100	42.0	26

242 rows × 20 columns

In [16]:

```
# Dataset for Developed Countries
developing_country_data = ndf[ndf['Status']=='Developing']
developing_country_data.drop(['Year','Country'],axis=1,inplace=True)
developing_country_data
```

Out[16]:

	Status	Adult_Mortality	Infant_Deaths	Alcohol	Percentage_Expenditure	Hepatitis_B	Measle
0	Developing	263.0	62	0.01	71.279624	65.0	115
1	Developing	271.0	64	0.01	73.523582	62.0	49
2	Developing	268.0	66	0.01	73.219243	64.0	43
3	Developing	272.0	69	0.01	78.184215	67.0	278
4	Developing	275.0	71	0.01	7.097109	68.0	301
...	...	...	...	...	...	...	.
2933	Developing	723.0	27	4.36	0.000000	68.0	3
2934	Developing	715.0	26	4.06	0.000000	7.0	99
2935	Developing	73.0	25	4.43	0.000000	73.0	30
2936	Developing	686.0	25	1.72	0.000000	76.0	52
2937	Developing	665.0	24	1.68	0.000000	79.0	148

1407 rows × 20 columns

In [17]:

```
# Checking Mean of Developend Countries Dataset
developed_country_data.mean()
```

Out[17]:
```
Adult_Mortality            8.419008e+01
Infant_Deaths              8.719008e-01
Alcohol                    1.043620e+01
Percentage_Expenditure     2.656822e+03
Hepatitis_B                8.788017e+01
Measles                    4.749339e+02
BMI                        5.233678e+01
under-five deaths          1.086777e+00
Polio                      9.449174e+01
Total_Expenditure          7.023099e+00
Diphtheria                 9.464463e+01
HIV/AIDS                   1.000000e-01
GDP                        1.897693e+04
Population                 8.744688e+06
thinness  1-19 years       1.435950e+00
thinness 5-9 years         1.460744e+00
Income_Cresources          8.361612e-01
Schooling                  1.557355e+01
Life_expectancy            7.869174e+01
dtype: float64
```

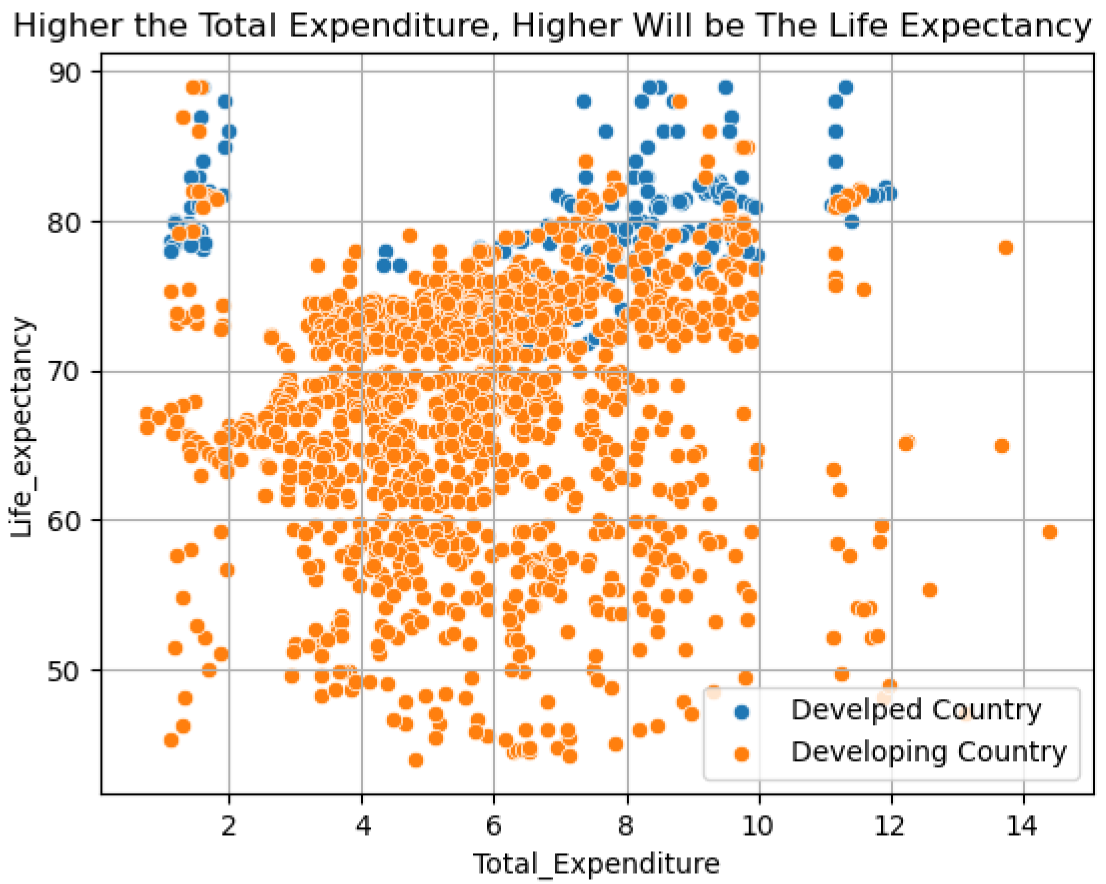
In [18]:
```
# Checking Mean of Developing Countries Dataset
developing_country_data.mean()
```

Out[18]:
```
Adult_Mortality            1.826674e+02
Infant_Deaths              3.800213e+01
Alcohol                    3.517896e+00
Percentage_Expenditure     3.622293e+02
Hepatitis_B                7.772779e+01
Measles                    2.525414e+03
BMI                        3.568486e+01
under-five deaths          5.163895e+01
Polio                      8.168515e+01
Total_Expenditure          5.772374e+00
Diphtheria                 8.235110e+01
HIV/AIDS                   2.307889e+00
GDP                        3.259395e+03
Population                 1.566995e+07
thinness  1-19 years       5.437953e+00
thinness 5-9 years         5.500640e+00
Income_Cresources          5.963589e-01
Schooling                  1.152587e+01
Life_expectancy            6.768735e+01
dtype: float64
```

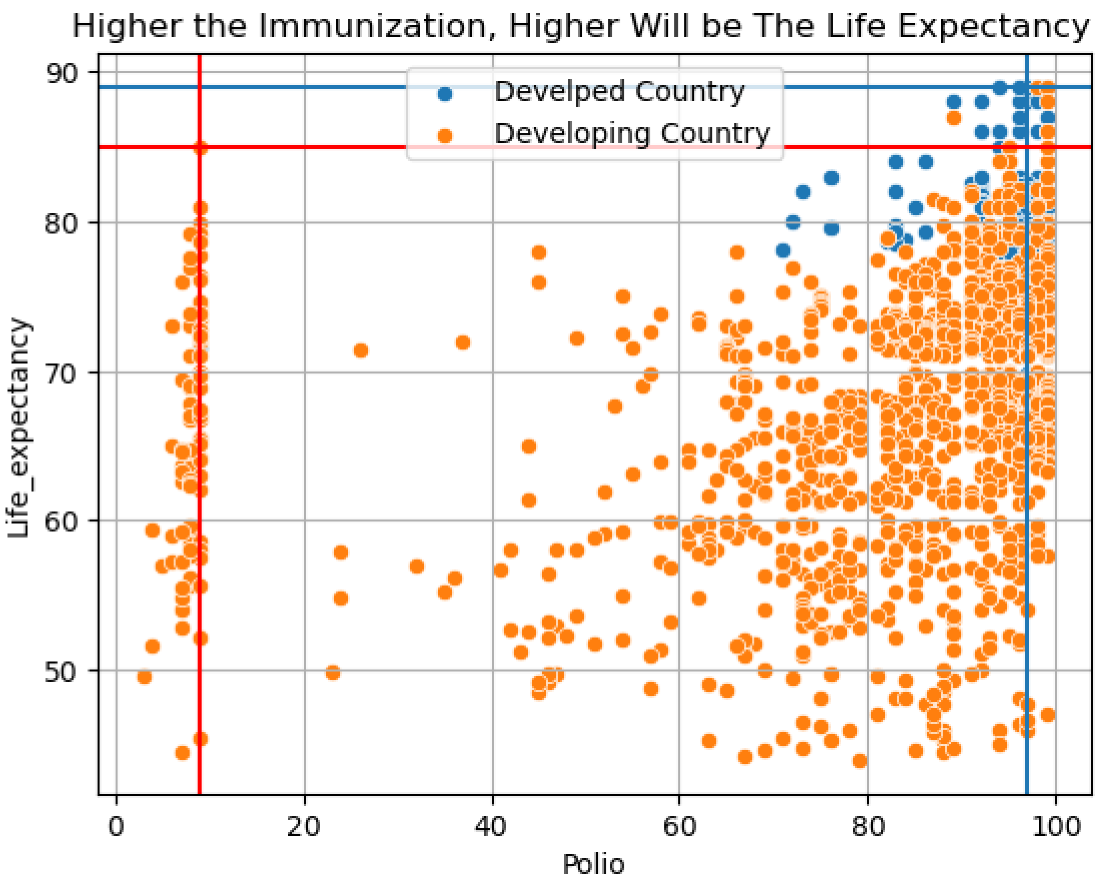
In [19]:
```
sns.scatterplot(x= developed_country_data.GDP,y= developed_country_data.Life_expe
sns.scatterplot(x= developing_country_data.GDP,y= developing_country_data.Life_ex
plt.legend(['Develped Country','Developing Country'])
plt.title('Higher the GDP, Lesser Will be The Life Expectancy',loc='right')
plt.grid()
plt.show()
```

In [20]:

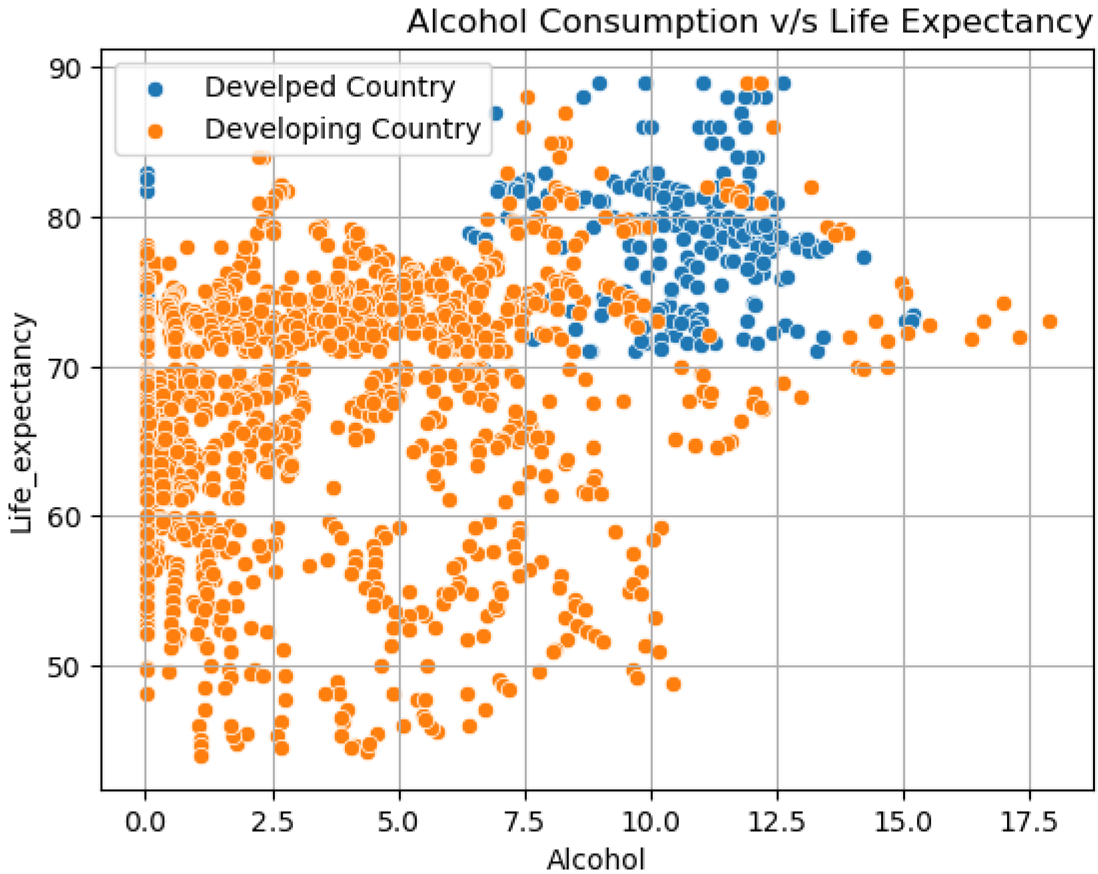
```
sns.scatterplot(x= developed_country_data.Total_Expenditure,y= developed_country_
sns.scatterplot(x= developing_country_data.Total_Expenditure,y= developing_countı
plt.legend(['Develped Country','Developing Country'])
plt.title('Higher the Total Expenditure, Higher Will be The Life Expectancy',loc=
plt.grid()
plt.show()
```



In [21]:
```
sns.scatterplot(x= developed_country_data.Polio,y= developed_country_data.Life_ex
sns.scatterplot(x= developing_country_data.Polio,y= developing_country_data.Life_
plt.axhline(89)
plt.axvline(97)
plt.axhline(85,color ='r')
plt.axvline(9,color = 'r')
plt.title('Higher the Immunization, Higher Will be The Life Expectancy',loc='righ
plt.grid()
plt.legend(['Develped Country','Developing Country'],)
plt.show()
```

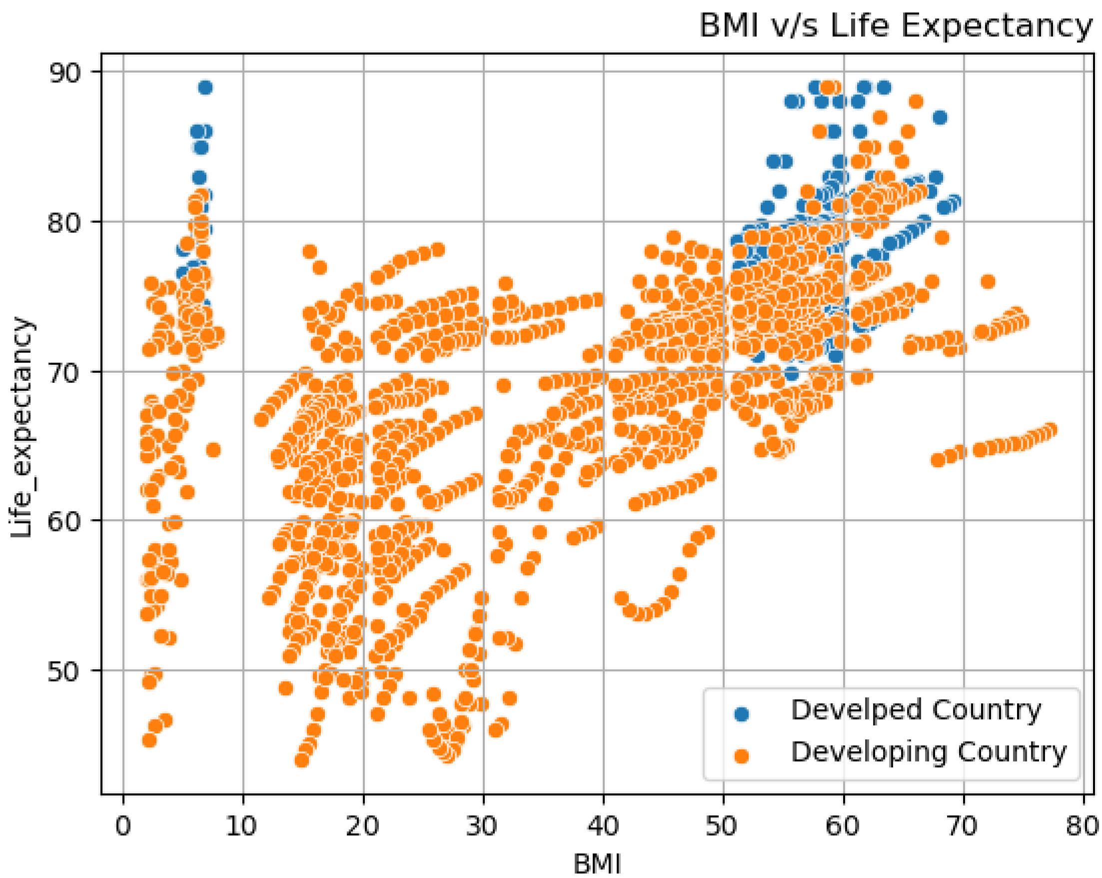


In [22]:
```
sns.scatterplot(x= developed_country_data.Alcohol,y= developed_country_data.Life_
sns.scatterplot(x= developing_country_data.Alcohol,y= developing_country_data.Li
plt.legend(['Develped Country','Developing Country'])
plt.title('Alcohol Consumption v/s Life Expectancy',loc='right')
plt.grid()
plt.show()
```



In [23]:

```
sns.scatterplot(x= developed_country_data.BMI,y= developed_country_data.Life_expe
sns.scatterplot(x= developing_country_data.BMI,y= developing_country_data.Life_ex
plt.legend(['Develped Country','Developing Country'])
plt.title('BMI v/s Life Expectancy',loc='right')
plt.grid()
plt.show()
```



OBSERVATION 4

1. Infant and Adult Mortality Rate found to be very poor for Developing Countries as Compared with Developed Countries.
2. There are Plenty of reason for the same few reasons are listed below - High GDP, Lesser Total Expenditure, Comparatively Less immunization for Developing Countries
3. Intresting Fact is Alcohol Consumption found to be very high for Developing Countries

Q.] Does Life Expectancy has a positive or negative correlation with eating habits, lifestyle, exercise, smoking, drinking alcohol etc?

OBSERVATION 5

From Above Data, We Found that the Value of BMI is Good for Developed Countries in comparison with Developing Country.

As Already Mentioned, The Alcohol Consumption is very high for Developing Countries.

Life Expectancy is found to be Positively Correlated with Following Feature Columns such as -

[a]. Income Composition Resources with 72%

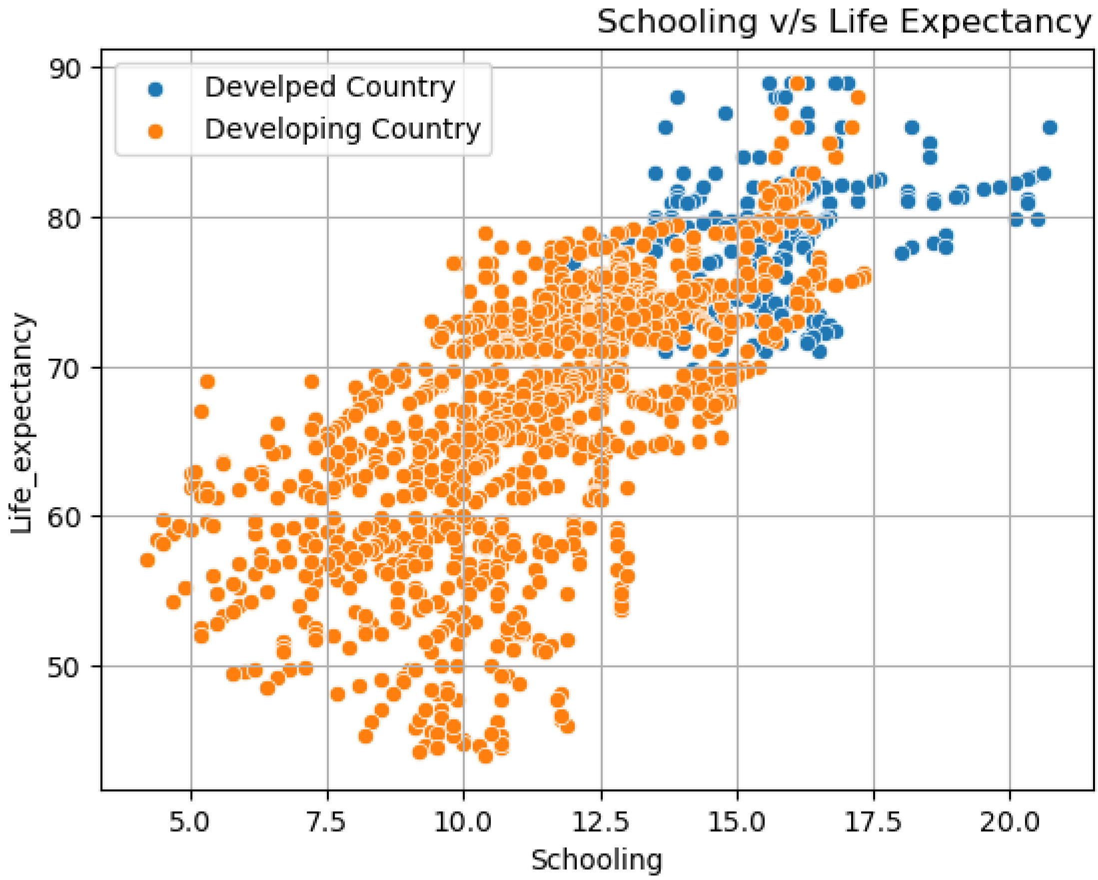
[b]. BMI with 54%

[c]. Percentage Expenditure and Alcohol with 40%

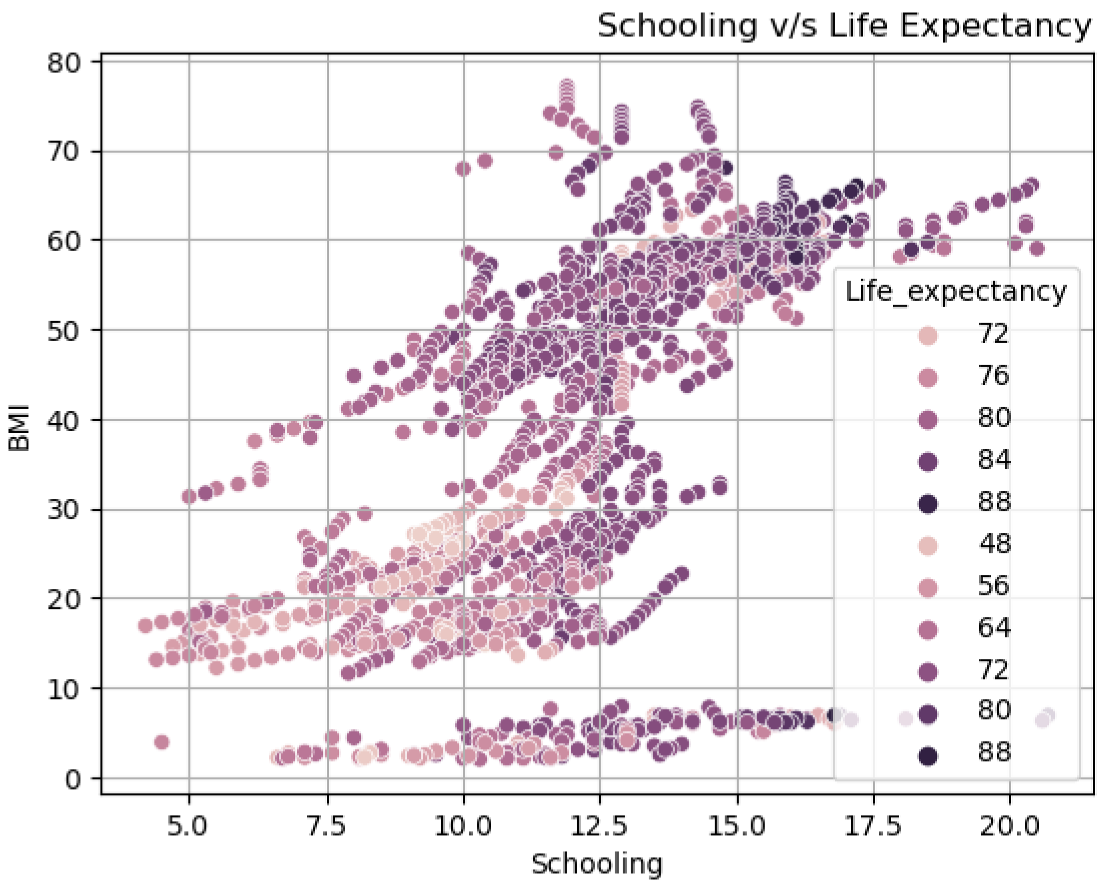
Q.] What is the impact of schooling on the lifespan of humans?

In [24]:

```
sns.scatterplot(x= developed_country_data.Schooling,y= developed_country_data.Li
sns.scatterplot(x= developing_country_data.Schooling,y= developing_country_data.L
plt.legend(['Develped Country','Developing Country'])
plt.title('Schooling v/s Life Expectancy',loc='right')
plt.grid()
plt.show()
```



In [25]:
```
sns.scatterplot(x= developed_country_data.Schooling,y= developed_country_data.BMI
sns.scatterplot(x= developing_country_data.Schooling,y= developing_country_data.B
#plt.legend(['Develped Country','Developing Country'])
plt.title('Schooling v/s Life Expectancy',loc='right')
plt.grid()
plt.show()
```



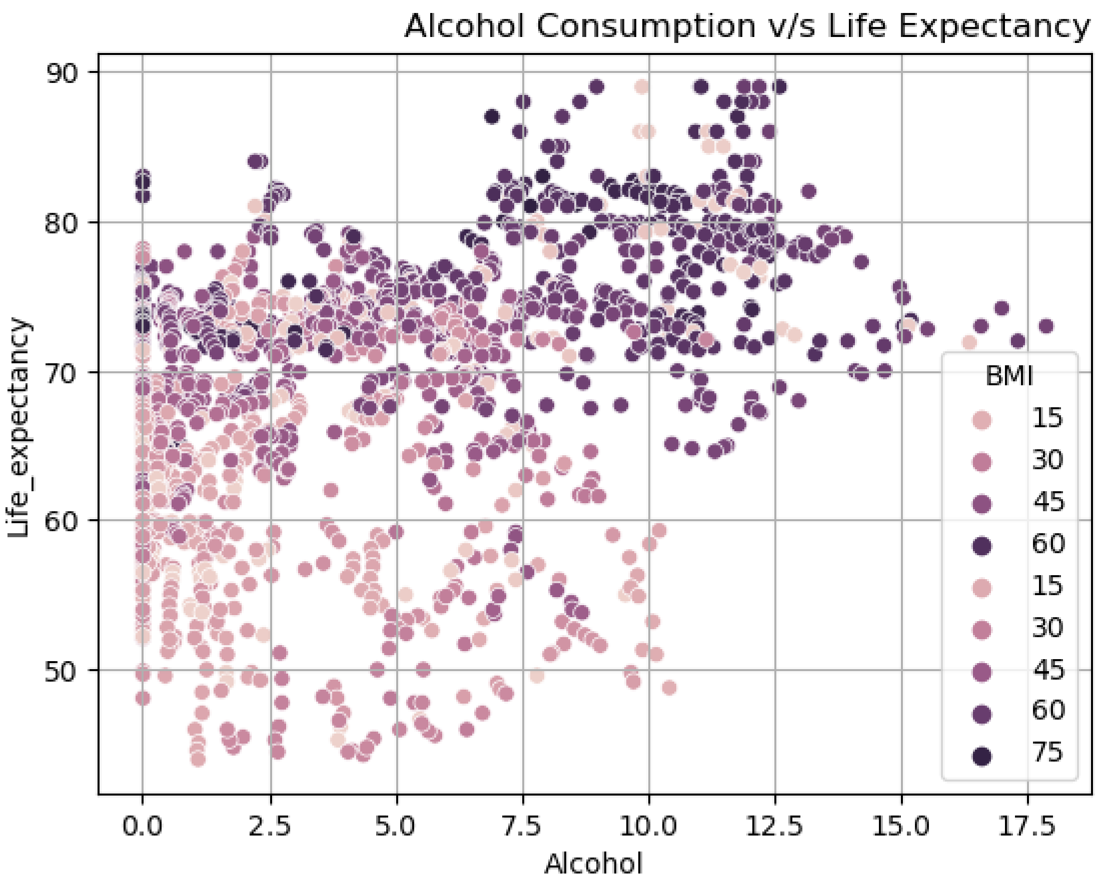
OBSERVATION 6

Schooling – Number of years of Schooling (years)

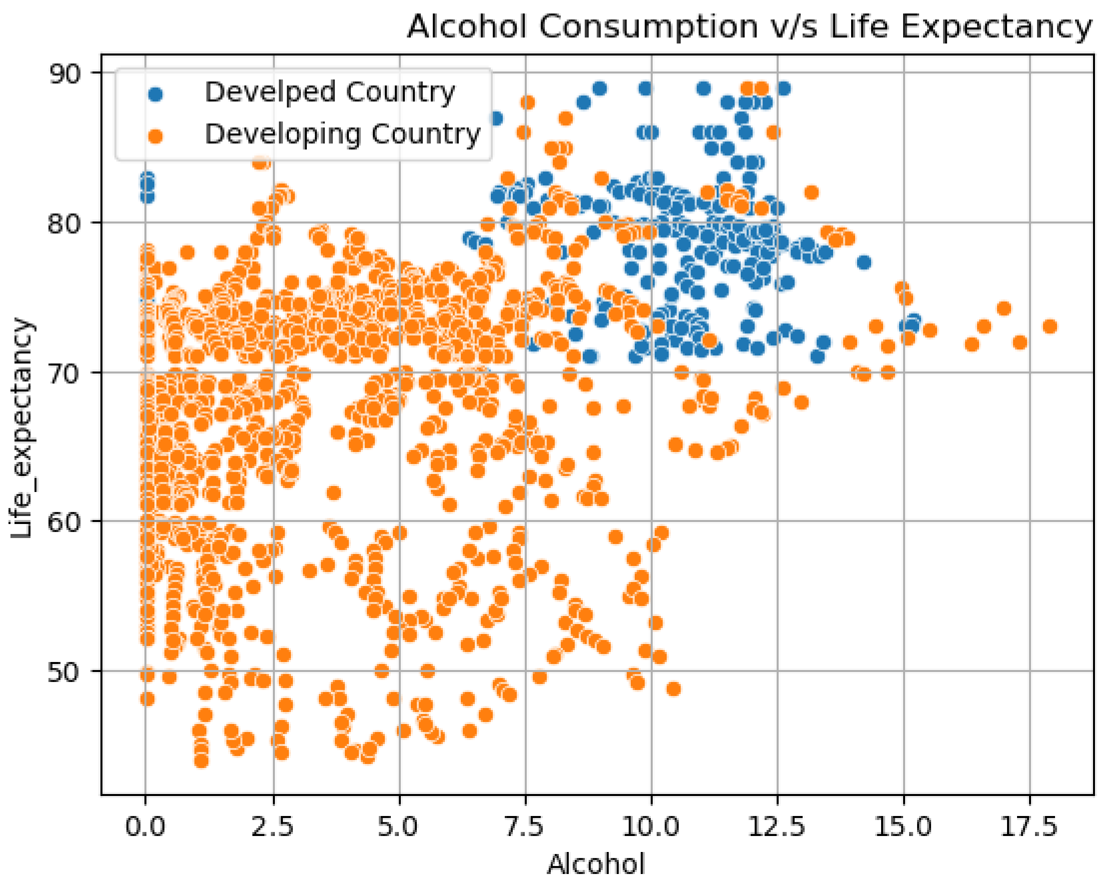
1. Schooling has positive correlation w.r.t Life Expectancy i.e. 73%
2. For Developed Countries - Schooling found to be very higher compared with Developing Countries.
3. Schooling also further related with Life style and Health Conciousness and Eating Habbits. This might be the reason for higher life expectancy found for developed countries than developing countries

Q.] Does Life Expectancy have a positive or negative relationship with drinking alcohol?

In [26]:
```
sns.scatterplot(x= developed_country_data.Alcohol,y= developed_country_data.Life_
sns.scatterplot(x= developing_country_data.Alcohol,y= developing_country_data.Lif
plt.title('Alcohol Consumption v/s Life Expectancy',loc='right')
plt.grid()
plt.show()
```



In [27]:
```
sns.scatterplot(x= developed_country_data.Alcohol,y= developed_country_data.Life_
sns.scatterplot(x= developing_country_data.Alcohol,y= developing_country_data.Lif
plt.legend(['Develped Country','Developing Country'])
plt.title('Alcohol Consumption v/s Life Expectancy',loc='right')
plt.grid()
plt.show()
```



OBSERVATION 7

Life Expectancy found to be Positively correlated with Alcohol i.e. 40%

Q.] Do densely populated countries tend to have lower life expectancy?

In [28]:
```
# Checking Maximum Population Country
ndf['Population'].max()
```
Out[28]:
```
1293859294.0
```

In [29]:
```
# Checking Minimum Population Country
ndf['Population'].min()
```
Out[29]:
```
34.0
```

In [30]:
```
# Average Population Country
ndf['Population'].mean()
```
Out[30]:
```
14653625.889484538
```

In [31]:
```
# Dense Populated Country
ndf[ndf['Population']==1293859294.0]
```
Out[31]:

	Country	Year	Status	Adult_Mortality	Infant_Deaths	Alcohol	Percentage_Expenditure	Hep
1187	India	2014	Developing	184.0	957	3.07	86.521539	

1 rows × 22 columns

In [32]:
```python
# Low populated Country
ndf[ndf['Population']==34]
```

Out[32]:

	Country	Year	Status	Adult_Mortality	Infant_Deaths	Alcohol	Percentage_Expenditure	Hep
1614	Maldives	2003	Developing	112.0	0	1.75	491.497891	

1 rows × 22 columns

In [33]:
```python
# More populated Countries i.e Population higher than Mean
ndf[ndf['Population']>14653625.889484538]
```

Out[33]:

	Country	Year	Status	Adult_Mortality	Infant_Deaths	Alcohol	Percentage_Expenditure	H
0	Afghanistan	2015	Developing	263.0	62	0.01	71.279624	
2	Afghanistan	2013	Developing	268.0	66	0.01	73.219243	
8	Afghanistan	2007	Developing	295.0	82	0.02	10.910156	
11	Afghanistan	2004	Developing	293.0	87	0.02	15.296066	
13	Afghanistan	2002	Developing	3.0	88	0.01	16.887351	
...	...	...	...	...	...	...	...	
2731	Ukraine	2014	Developing	23.0	4	8.06	5.663849	
2744	Ukraine	2001	Developing	253.0	6	4.31	8.897421	
2745	Ukraine	2000	Developing	257.0	6	4.49	7.883791	
2909	Zambia	2012	Developing	349.0	29	2.59	196.915250	
2923	Zimbabwe	2014	Developing	371.0	23	6.50	10.822595	

292 rows × 22 columns

In [34]:
```python
# More populated Countries i.e Population higher than Mean
ndf[ndf['Population']>14653625.889484538]['Country'].unique()
```

Out[34]:
```
array(['Afghanistan', 'Algeria', 'Angola', 'Argentina', 'Australia',
       'Bangladesh', 'Brazil', 'Burkina Faso', 'Cambodia', 'Cameroon',
       'Canada', 'Chile', 'Colombia', 'Ecuador', 'Ethiopia', 'France',
       'Germany', 'Ghana', 'Guatemala', 'India', 'Indonesia', 'Iraq',
       'Italy', 'Kazakhstan', 'Kenya', 'Madagascar', 'Malawi', 'Malaysia',
       'Mali', 'Mexico', 'Morocco', 'Mozambique', 'Myanmar', 'Nepal',
       'Netherlands', 'Niger', 'Nigeria', 'Pakistan', 'Peru',
       'Philippines', 'Poland', 'Romania', 'Russian Federation',
       'South Africa', 'Spain', 'Syrian Arab Republic', 'Thailand',
       'Turkey', 'Uganda', 'Ukraine', 'Zambia', 'Zimbabwe'], dtype=object)
```

In [35]:
```python
# More populated Countries i.e Population Lower than Mean
ndf[ndf['Population']<14653625.889484538]['Country'].unique()
```

Out[35]:
```
array(['Afghanistan', 'Albania', 'Algeria', 'Angola', 'Argentina',
       'Armenia', 'Australia', 'Austria', 'Azerbaijan', 'Bangladesh',
       'Belarus', 'Belgium', 'Belize', 'Benin', 'Bhutan',
       'Bosnia and Herzegovina', 'Botswana', 'Brazil', 'Bulgaria',
       'Burkina Faso', 'Burundi', 'Cabo Verde', 'Cambodia', 'Cameroon',
       'Canada', 'Central African Republic', 'Chad', 'Chile', 'China',
       'Colombia', 'Comoros', 'Costa Rica', 'Croatia', 'Cyprus',
       'Djibouti', 'Dominican Republic', 'Ecuador', 'El Salvador',
       'Equatorial Guinea', 'Eritrea', 'Estonia', 'Ethiopia', 'Fiji',
       'France', 'Gabon', 'Georgia', 'Germany', 'Ghana', 'Greece',
       'Guatemala', 'Guinea', 'Guinea-Bissau', 'Guyana', 'Haiti',
       'Honduras', 'India', 'Indonesia', 'Iraq', 'Ireland', 'Israel',
       'Italy', 'Jamaica', 'Jordan', 'Kazakhstan', 'Kenya', 'Kiribati',
       'Latvia', 'Lebanon', 'Lesotho', 'Liberia', 'Lithuania',
       'Luxembourg', 'Madagascar', 'Malawi', 'Malaysia', 'Maldives',
       'Mali', 'Malta', 'Mauritania', 'Mauritius', 'Mexico', 'Mongolia',
       'Montenegro', 'Morocco', 'Mozambique', 'Myanmar', 'Namibia',
       'Nepal', 'Netherlands', 'Nicaragua', 'Niger', 'Nigeria',
       'Pakistan', 'Panama', 'Papua New Guinea', 'Paraguay', 'Peru',
       'Philippines', 'Poland', 'Portugal', 'Romania',
       'Russian Federation', 'Rwanda', 'Samoa', 'Sao Tome and Principe',
       'Senegal', 'Serbia', 'Seychelles', 'Sierra Leone',
       'Solomon Islands', 'South Africa', 'Spain', 'Sri Lanka',
       'Suriname', 'Swaziland', 'Sweden', 'Syrian Arab Republic',
       'Tajikistan', 'Thailand', 'Timor-Leste', 'Togo', 'Tonga',
       'Trinidad and Tobago', 'Tunisia', 'Turkey', 'Turkmenistan',
       'Uganda', 'Ukraine', 'Uruguay', 'Uzbekistan', 'Vanuatu', 'Zambia',
       'Zimbabwe'], dtype=object)
```

Q.] Do densely populated countries tend to have lower life expectancy?

In [36]:
```python
# Dense Populated Country
ndf[ndf['Population']==1293859294.0]
```

Out[36]:

	Country	Year	Status	Adult_Mortality	Infant_Deaths	Alcohol	Percentage_Expenditure	Hep
1187	India	2014	Developing	184.0	957	3.07	86.521539	

1 rows × 22 columns

In [37]:
```python
# Dense Populated Country Year Wise
ndf[ndf['Year']==2000]['Population'].max()
```

Out[37]: 175287587.0

In [38]:
```python
# Dense Populated Country in Year 2000
ndf[ndf['Population']==175287587.0]
```

Out[38]:

	Country	Year	Status	Adult_Mortality	Infant_Deaths	Alcohol	Percentage_Expenditure	Hepa
367	Brazil	2000	Developing	183.0	111	7.26	179.477729	

1 rows × 22 columns

In [39]:
```python
# Dense Populated Country Year Wise
ndf[ndf['Year']==2005]['Population'].max()
```

Out[39]: 1144118674.0

In [40]:
```
# Dense Populated Country in Year 2005
ndf[ndf['Population']==1144118674.0]
```

Out[40]:

	Country	Year	Status	Adult_Mortality	Infant_Deaths	Alcohol	Percentage_Expenditure	Hep
1196	India	2005	Developing	211.0	1500	1.27	3.509637	

1 rows × 22 columns

In [41]:
```
# Dense Populated Country Year Wise
ndf[ndf['Year']==2010]['Population'].max()
```

Out[41]:
```
242524123.0
```

In [42]:
```
# Dense Populated Country in Year 2010
ndf[ndf['Population']==242524123.0]
```

Out[42]:

	Country	Year	Status	Adult_Mortality	Infant_Deaths	Alcohol	Percentage_Expenditure	He
1207	Indonesia	2010	Developing	187.0	138	0.08	190.545365	

1 rows × 22 columns

In [43]:
```
# Dense Populated Country Year Wise
ndf[ndf['Year']==2015]['Population'].max()
```

Out[43]:
```
33736494.0
```

In [44]:
```
# Dense Populated Country in Year 2010
ndf[ndf['Population']==33736494.0]
```

Out[44]:

	Country	Year	Status	Adult_Mortality	Infant_Deaths	Alcohol	Percentage_Expenditure	Hep
0	Afghanistan	2015	Developing	263.0	62	0.01	71.279624	

1 rows × 22 columns

OBSERVATION 8

1. Based on Above Observation we find, Average Life Expectancy for Dense Populated Country is 65 to 69 Years.
2. To understand the trend pattern, we have segregated dataset into %years of span length i.e. Year 2000,2005,2010,2015. Intresting Insigh is highlighted through snippet codes.
3. In Year 2000 - Brazil is found to be Mostly Popultaed Country with Avg Life Expectancy of 75 Years.

   In Year 2005 - India is found to be Mostly Popultaed Country with Avg Life Expectancy of 64.4 Years.

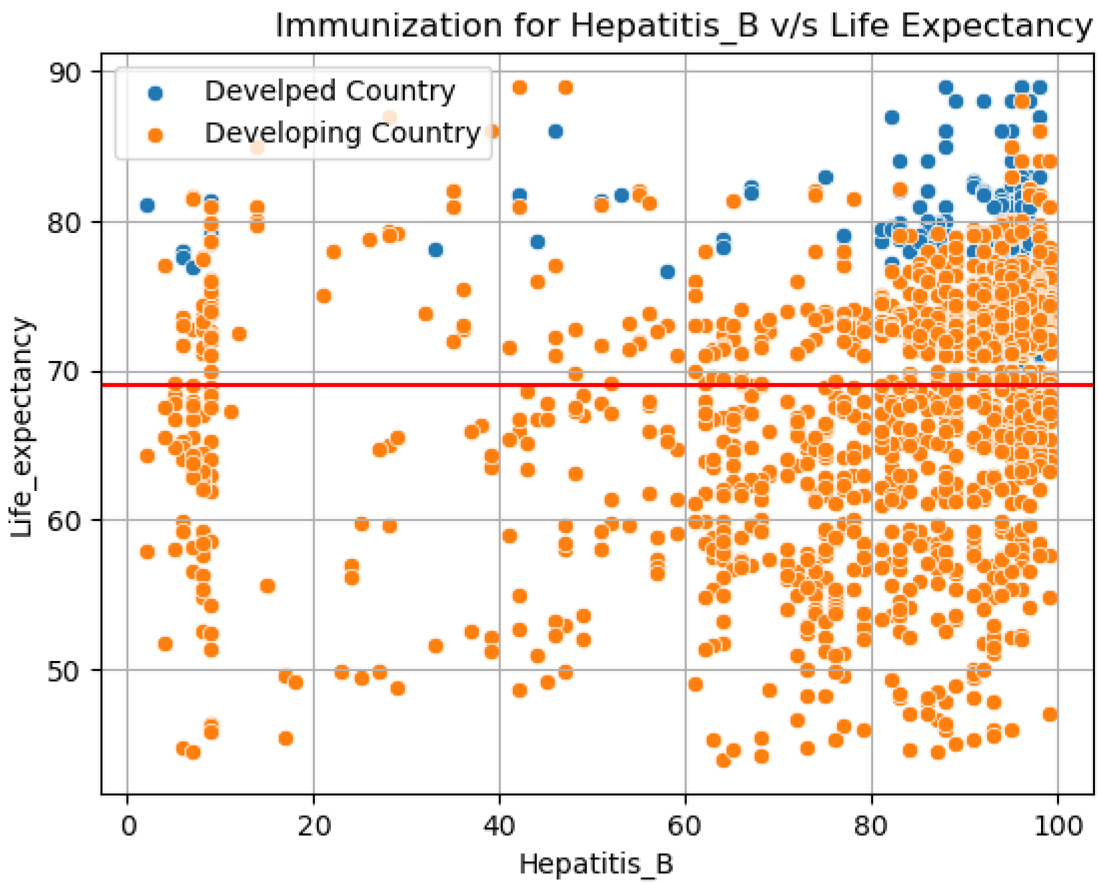
   In Year 2010 - Indonesia is found to be Mostly Popultaed Country with Avg Life Expectancy of 68.1 Years.

In Year 2015 - Afghanistan is found to be Mostly Popultaed Country with Avg Life Expectancy of 65 Years.

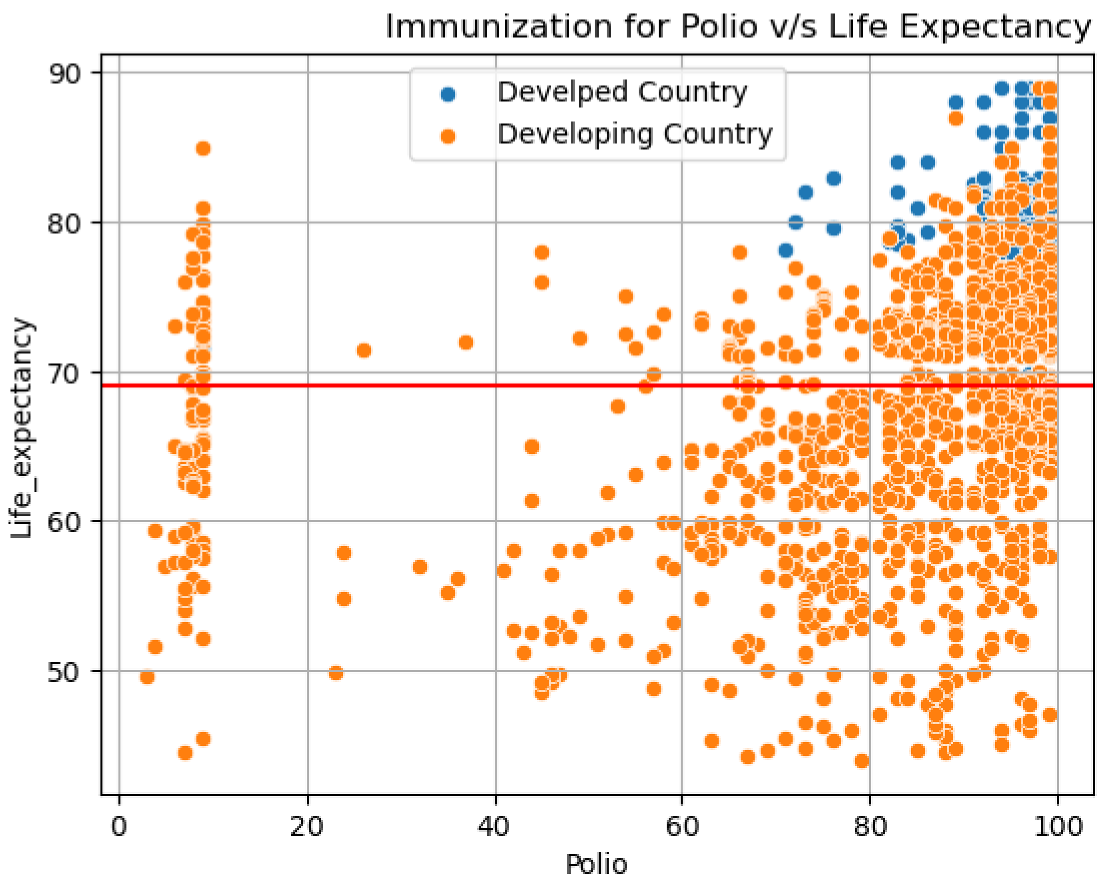
4. All Dense Populated Countries are Developing Countries Only

Q.] What is the impact of Immunization coverage on Life Expectancy?

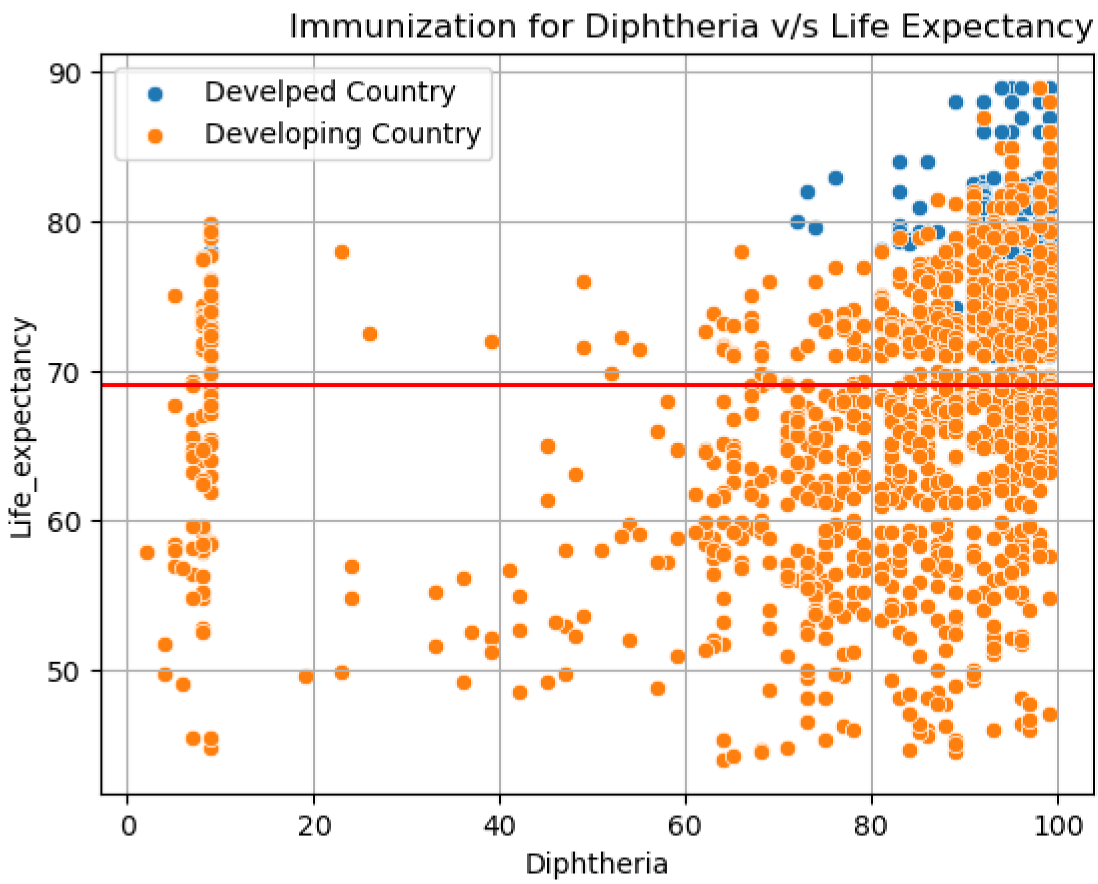
In [45]:
```
sns.scatterplot(x= developed_country_data.Hepatitis_B,y= developed_country_data.l
sns.scatterplot(x= developing_country_data.Hepatitis_B,y= developing_country_data
plt.legend(['Develped Country','Developing Country'])
plt.title('Immunization for Hepatitis_B v/s Life Expectancy',loc='right')
plt.axhline(69,color ='r')
plt.grid()
plt.show()
```



In [46]:
```
sns.scatterplot(x= developed_country_data.Polio,y= developed_country_data.Life_ex
sns.scatterplot(x= developing_country_data.Polio,y= developing_country_data.Life_
plt.legend(['Develped Country','Developing Country'])
plt.title('Immunization for Polio v/s Life Expectancy',loc='right')
plt.axhline(69,color ='r')
plt.grid()
plt.show()
```

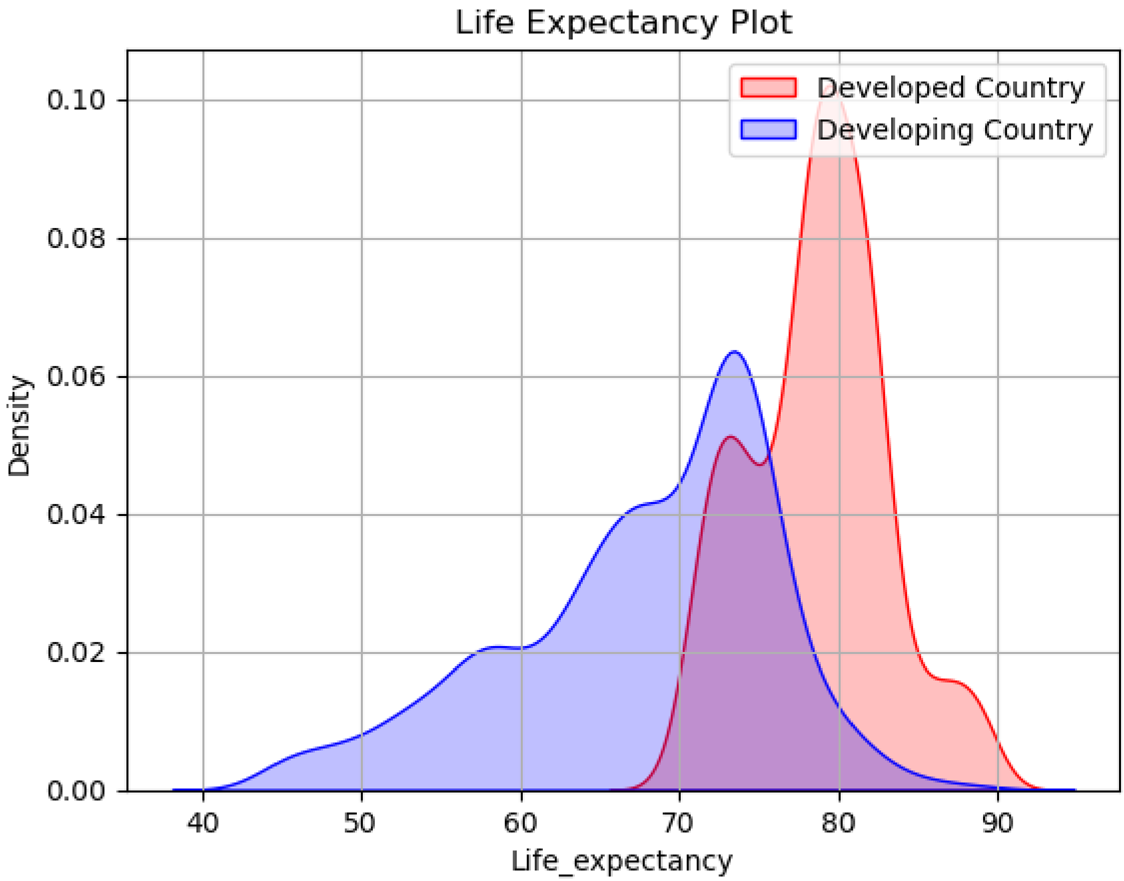


In [47]:
```
sns.scatterplot(x= developed_country_data['Diphtheria'],y= developed_country_data
sns.scatterplot(x= developing_country_data['Diphtheria'],y= developing_country_da
plt.legend(['Develped Country','Developing Country'])
plt.title('Immunization for Diphtheria v/s Life Expectancy',loc='right')
plt.axhline(69,color ='r')
plt.grid()
plt.show()
```



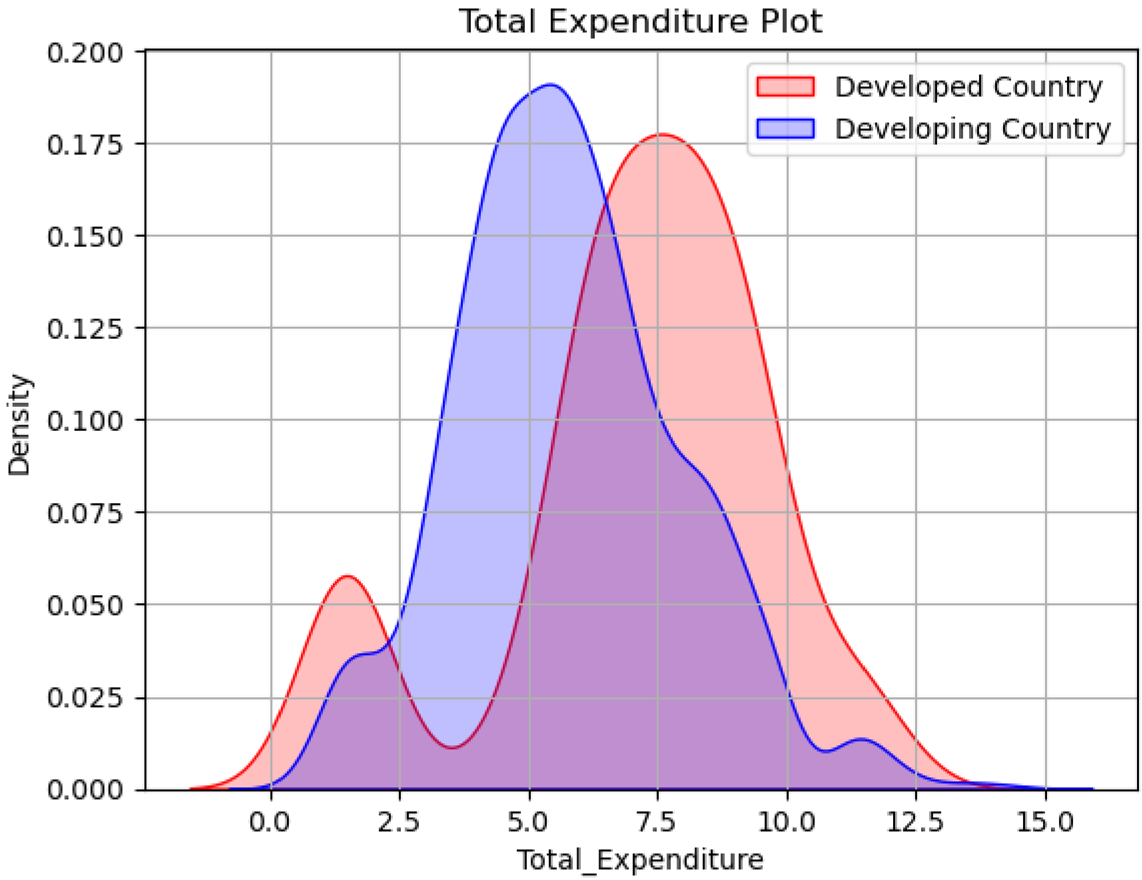
In [48]:

```
# Life Expectancy for Developed vs Developing Country
sns.kdeplot(x = developed_country_data.Life_expectancy,fill=True,color='red');
sns.kdeplot(x = developing_country_data.Life_expectancy,fill=True,color='blue');
plt.legend(['Developed Country','Developing Country'])
plt.title('Life Expectancy Plot')
plt.grid()
```



In [49]:

```
# Total Expenditure Consumption for Developed vs Developing Country
sns.kdeplot(x = developed_country_data.Total_Expenditure,fill=True,color='red');
sns.kdeplot(x = developing_country_data.Total_Expenditure,fill=True,color='blue'
plt.legend(['Developed Country','Developing Country'])
plt.title('Total Expenditure Plot')
plt.grid()
```



OBSERVATION 9

1. Higher the Immunization Higher will be the Life Expectancy.
2. From Above EDA, As Immunization Increases The Life Expectancy found to be increased mor than Avg Life Expectancy Level i.e. More than 69 Years
3. Total Expenditure plays an vital role in order to predict Life Expectancy. Since From Above Graph we can observed that higher the government expenditure on health care more will be Life Expectancy Rate for Developed Countries

--- TASK 2 : MACHINE LEARNING MODEL BUILDING

In [50]:
```
data = ndf.drop(['Country','Year'],axis=1)
data
```

Out[50]:

	Status	Adult_Mortality	Infant_Deaths	Alcohol	Percentage_Expenditure	Hepatitis_B	Measle
0	Developing	263.0	62	0.01	71.279624	65.0	115
1	Developing	271.0	64	0.01	73.523582	62.0	49
2	Developing	268.0	66	0.01	73.219243	64.0	43
3	Developing	272.0	69	0.01	78.184215	67.0	278
4	Developing	275.0	71	0.01	7.097109	68.0	301
...	...	...	...	...	...	...	.
2933	Developing	723.0	27	4.36	0.000000	68.0	3
2934	Developing	715.0	26	4.06	0.000000	7.0	99
2935	Developing	73.0	25	4.43	0.000000	73.0	30
2936	Developing	686.0	25	1.72	0.000000	76.0	52
2937	Developing	665.0	24	1.68	0.000000	79.0	148

1649 rows × 20 columns

In [51]:
```
data['Status'] = data['Status'].map({'Developing': 1, 'Developed': 0})
```

In [52]:
```
data.head()
```

Out[52]:

	Status	Adult_Mortality	Infant_Deaths	Alcohol	Percentage_Expenditure	Hepatitis_B	Measles	BMI
0	1	263.0	62	0.01	71.279624	65.0	1154	19.1
1	1	271.0	64	0.01	73.523582	62.0	492	18.6
2	1	268.0	66	0.01	73.219243	64.0	430	18.1
3	1	272.0	69	0.01	78.184215	67.0	2787	17.6
4	1	275.0	71	0.01	7.097109	68.0	3013	17.2

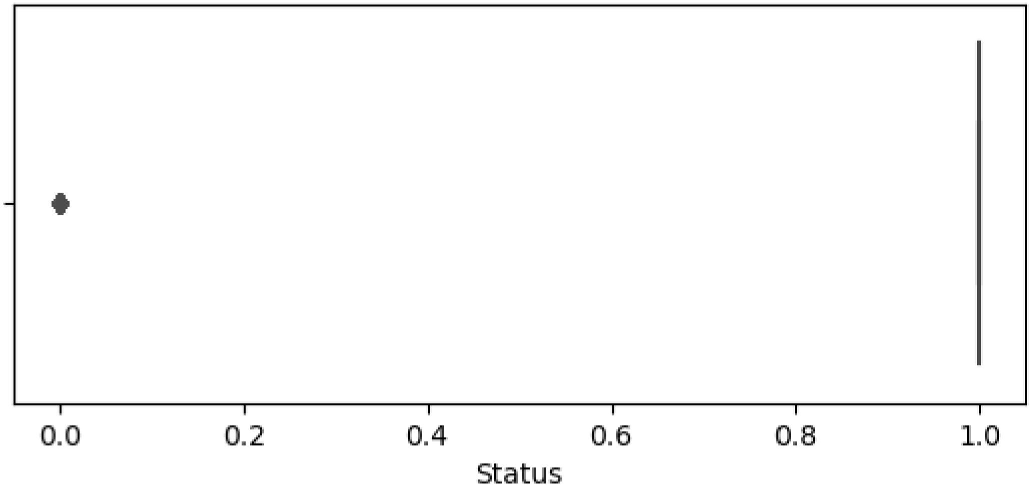
In [53]:
```
features = data.columns
features = ['Status','Adult_Mortality', 'Infant_Deaths', 'Alcohol',
       'Percentage_Expenditure', 'Hepatitis_B', 'Measles ', 'BMI',
       'under-five deaths', 'Polio', 'Total_Expenditure', 'Diphtheria',
       'HIV/AIDS', 'GDP', 'Population', 'thinness  1-19 years',
       'thinness 5-9 years', 'Income_Cresources', 'Schooling',
       'Life_expectancy']
```

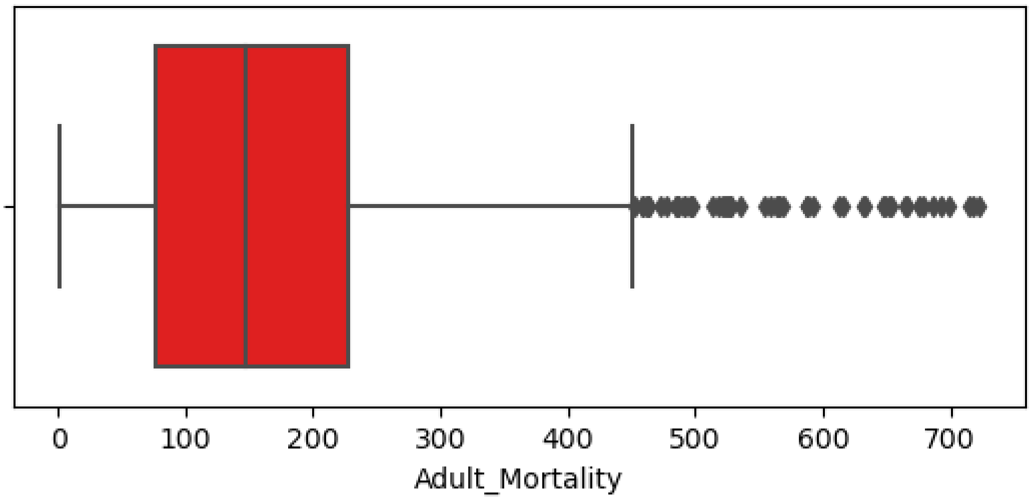
In [54]:
```
list(enumerate(features))
```

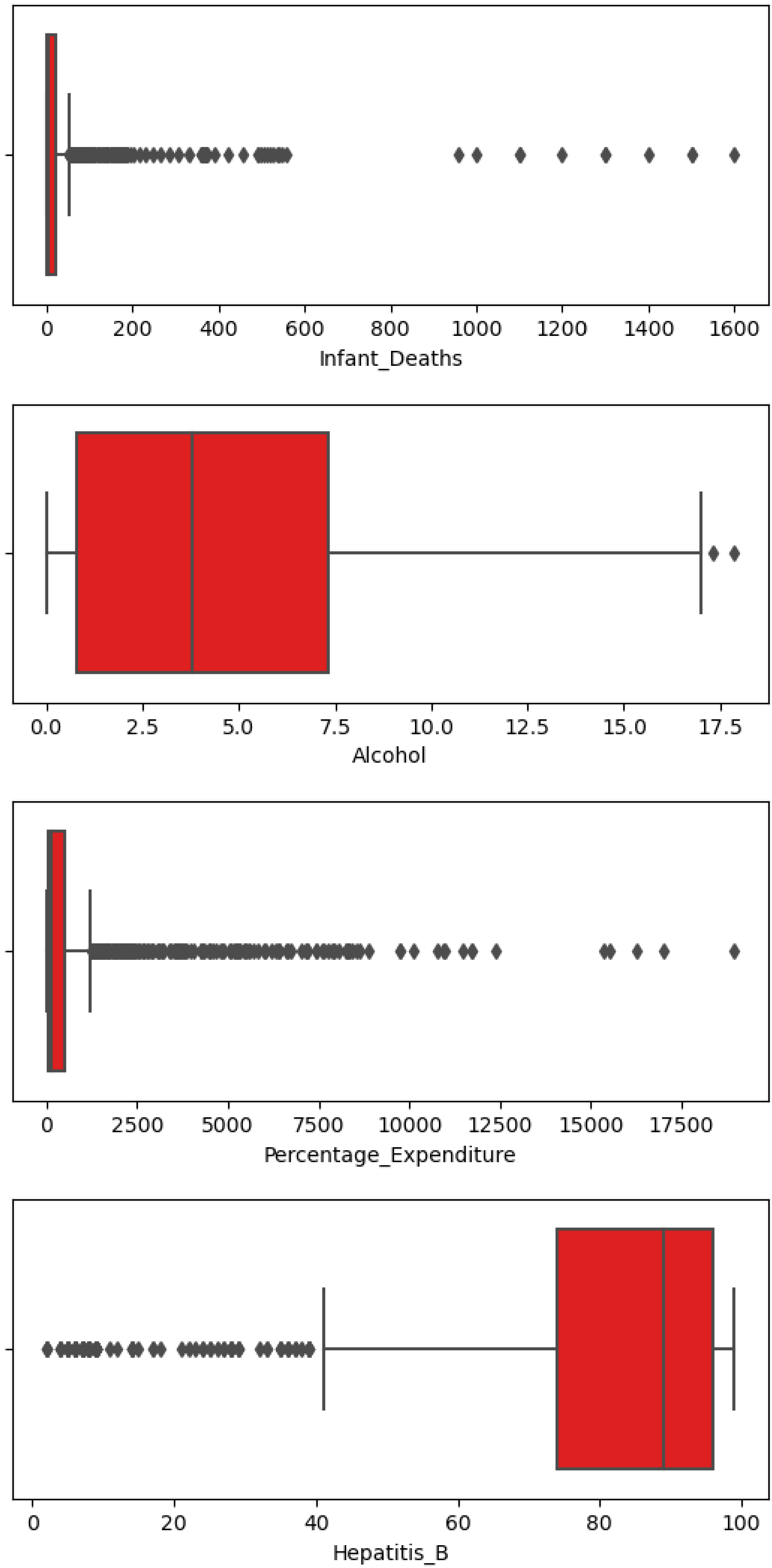
Out[54]:
```
[(0, 'Status'),
 (1, 'Adult_Mortality'),
 (2, 'Infant_Deaths'),
 (3, 'Alcohol'),
 (4, 'Percentage_Expenditure'),
 (5, 'Hepatitis_B'),
 (6, 'Measles '),
 (7, 'BMI'),
 (8, 'under-five deaths'),
 (9, 'Polio'),
 (10, 'Total_Expenditure'),
 (11, 'Diphtheria'),
 (12, 'HIV/AIDS'),
 (13, 'GDP'),
 (14, 'Population'),
 (15, 'thinness  1-19 years'),
 (16, 'thinness 5-9 years'),
 (17, 'Income_Cresources'),
 (18, 'Schooling'),
 (19, 'Life_expectancy')]
```

BOX PLOT BEFORE TREATING OUTLIERS

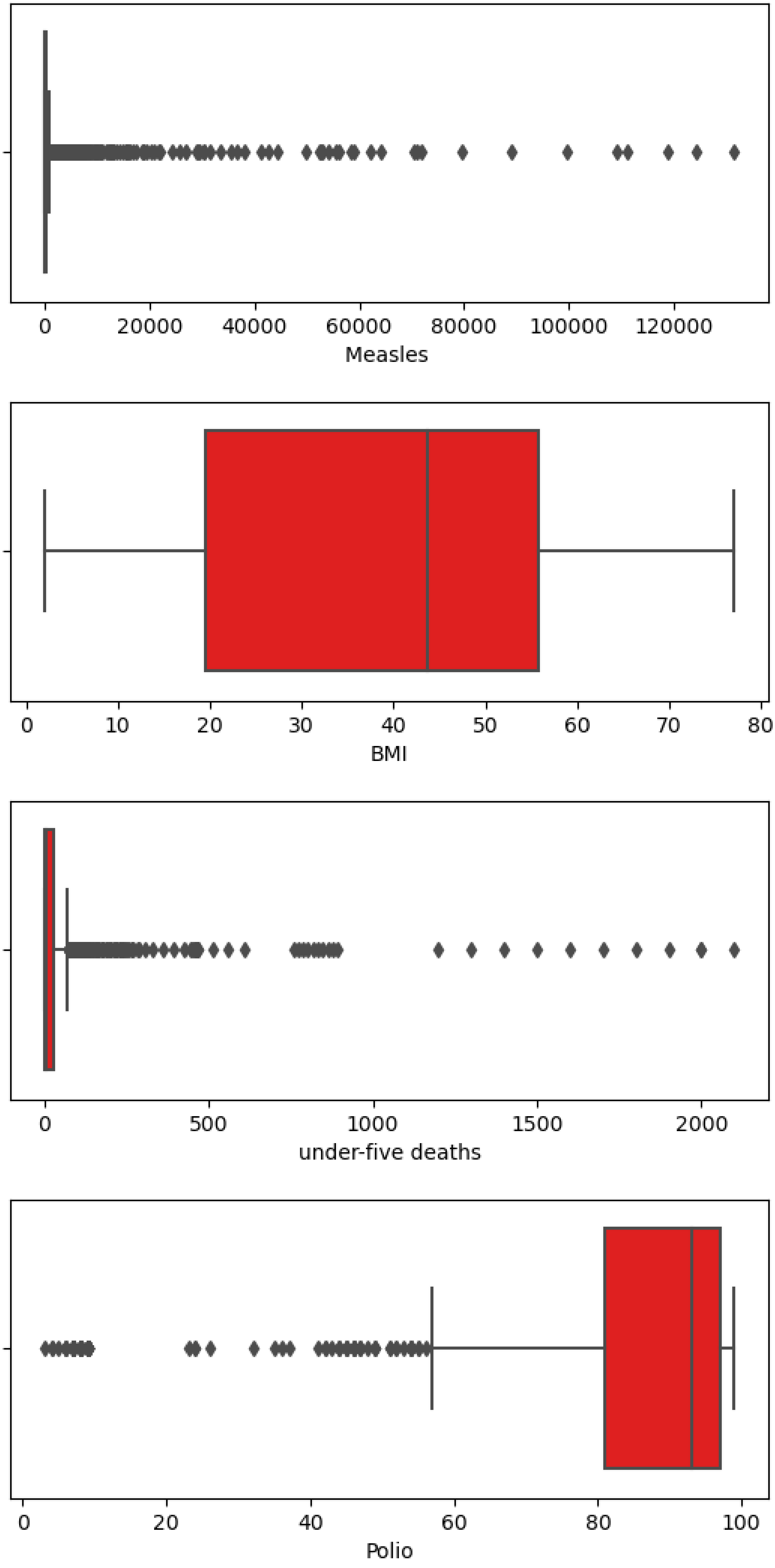
In [55]:
```
# Checking Outlier In The Datset
for col in enumerate(features):
    plt.figure(figsize=(30,15))
    plt.subplot(5,4,col[0]+1)
    sns.boxplot(x = col[1],color='red',data=data)
    plt.show()
```

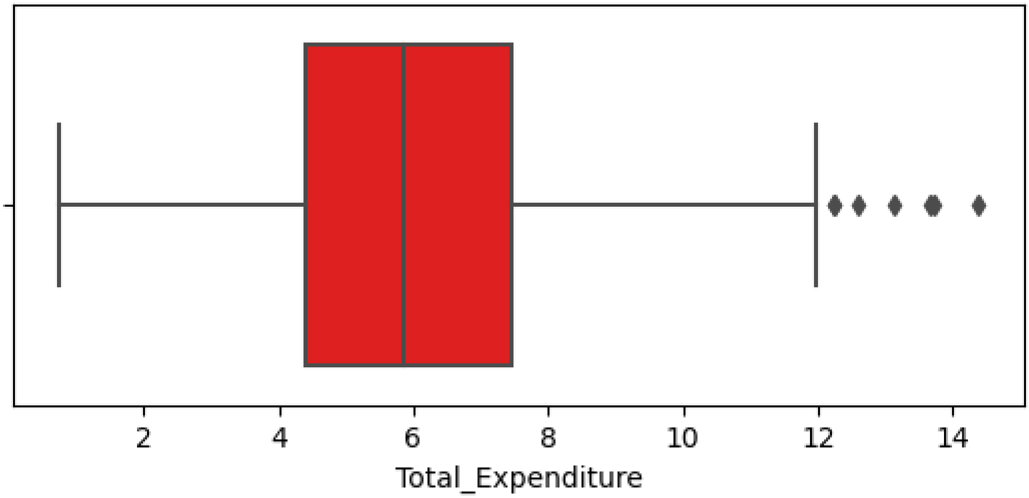


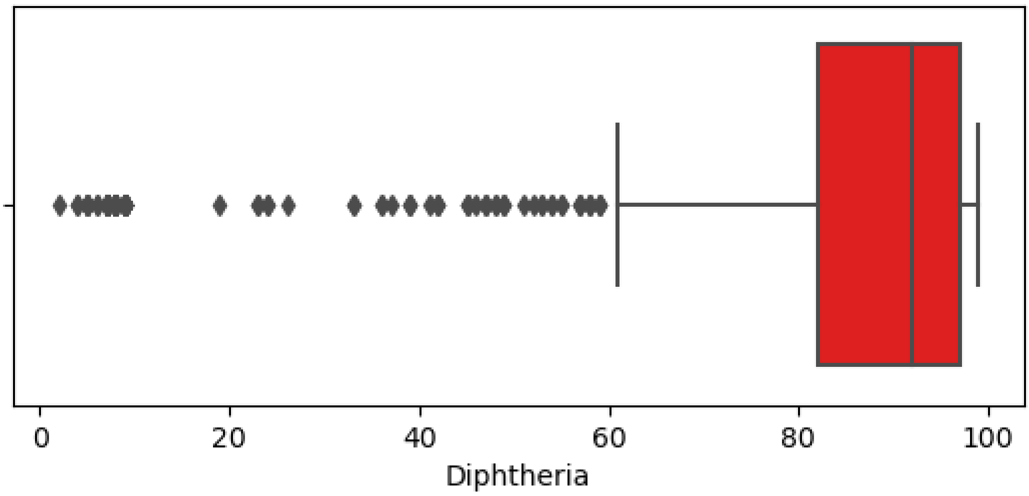


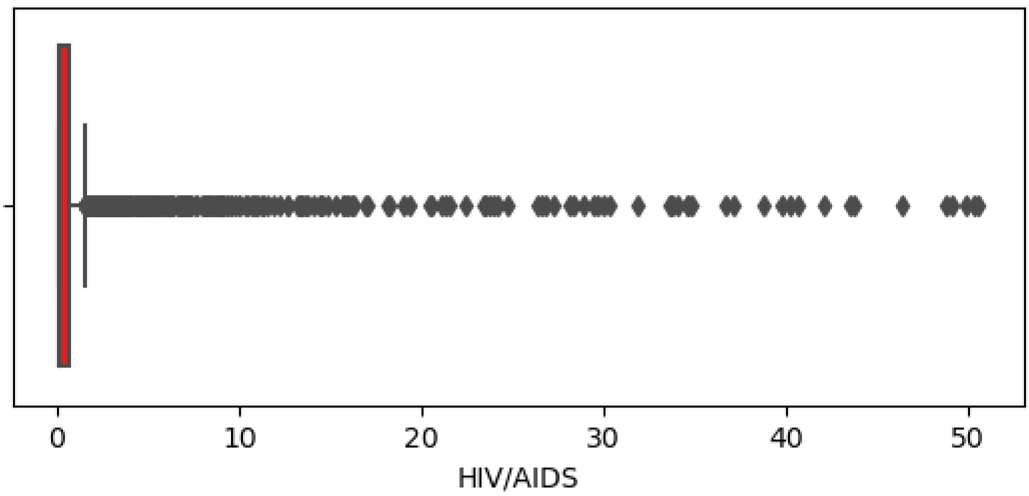


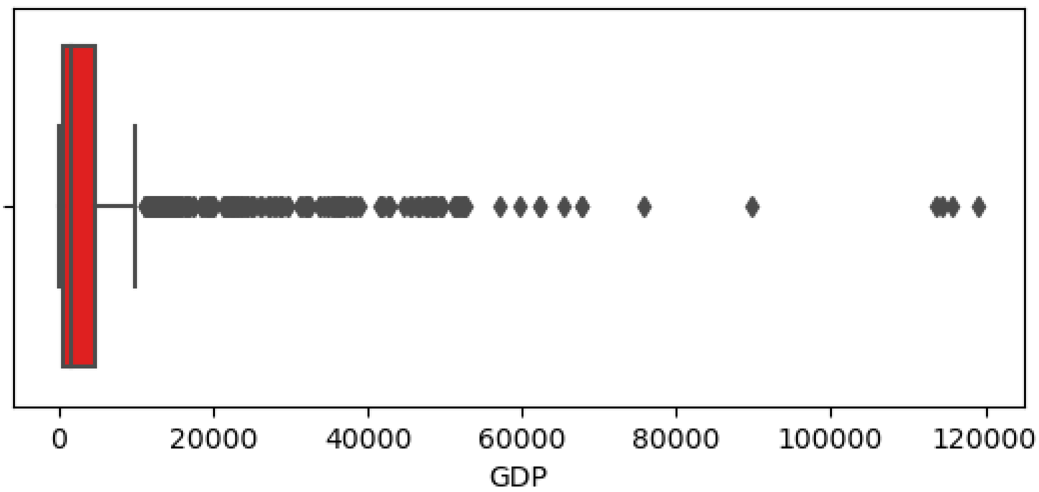
0 200 400 600 800 1000 1200 1400 1600
Infant_Deaths
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5
Alcohol
0 2500 5000 7500 10000 12500 15000 17500
Percentage_Expenditure
0 20 40 60 80 100
Hepatitis_B

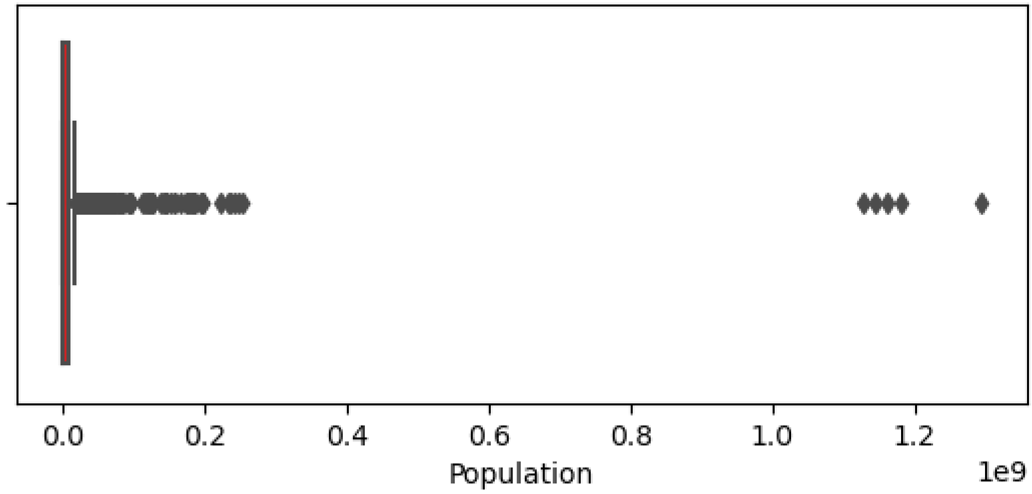

0 20000 40000 60000 80000 100000 120000
Measles
0 10 20 30 40 50 60 70 80
BMI
0 500 1000 1500 2000
under-five deaths
0 20 40 60 80 100
Polio

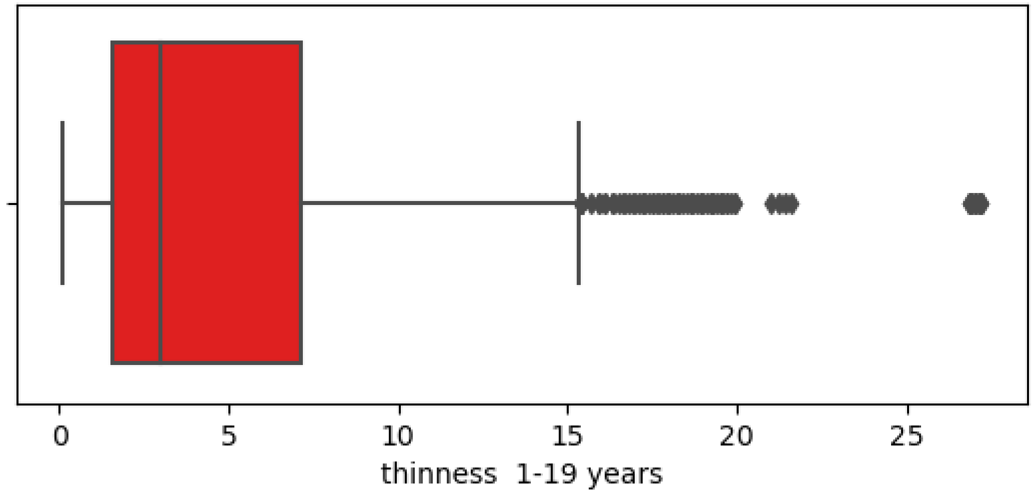

2 4 6 8 10 12 14
Total_Expenditure

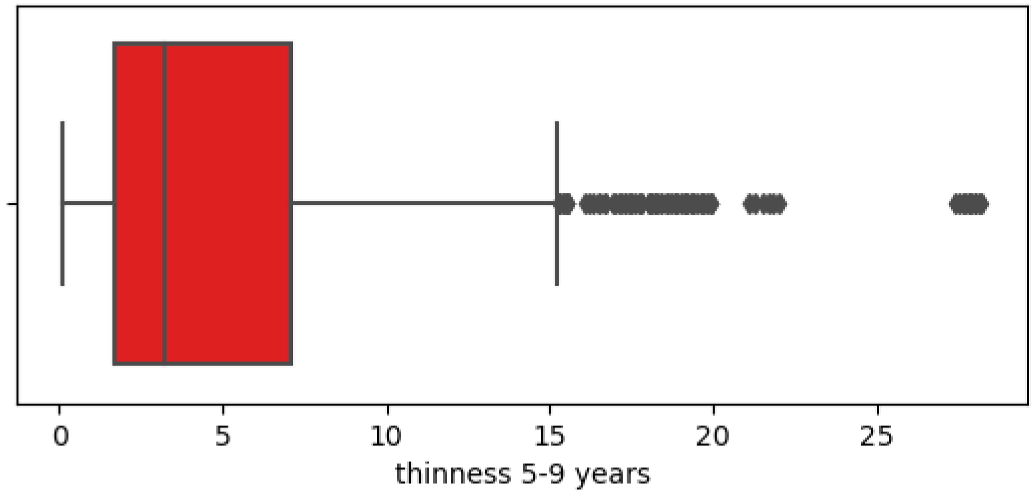

0 20 40 60 80 100
Diphtheria

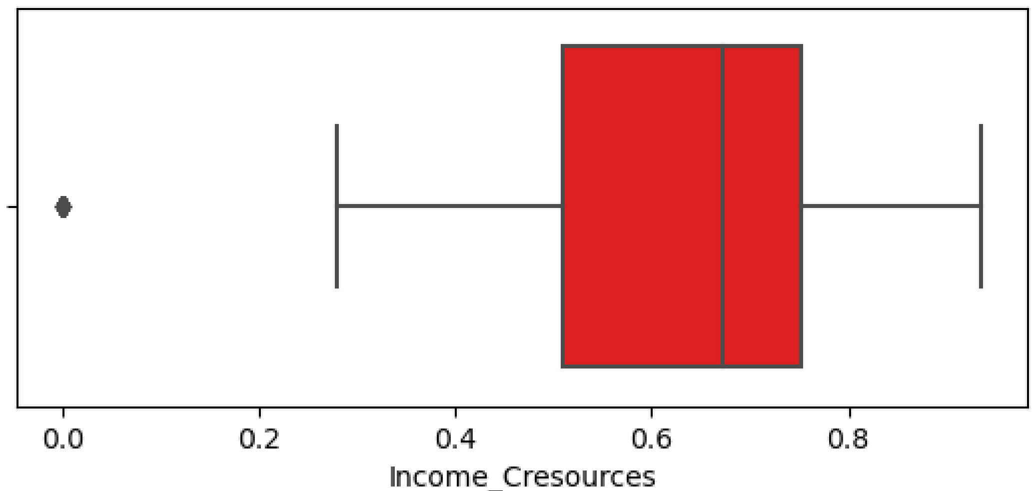

0 10 20 30 40 50
HIV/AIDS

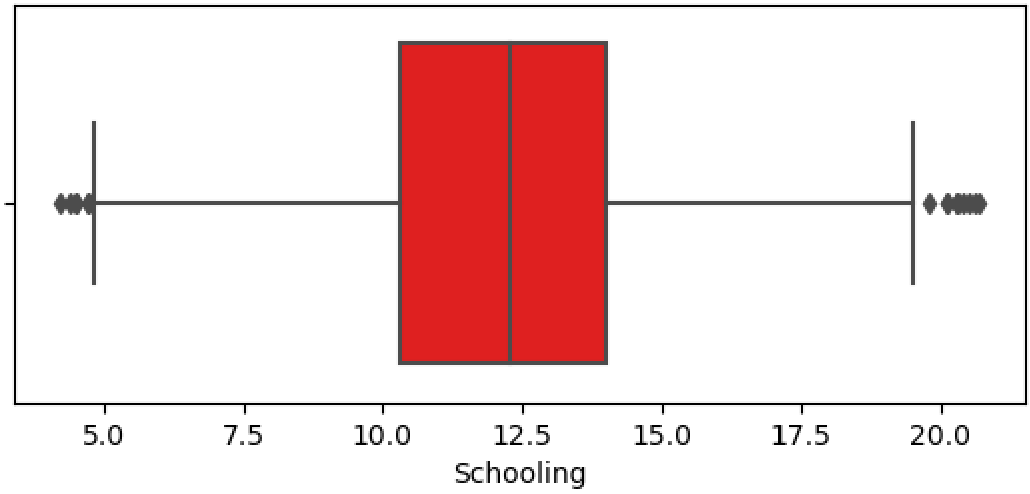

0 20000 40000 60000 80000 100000 120000
GDP

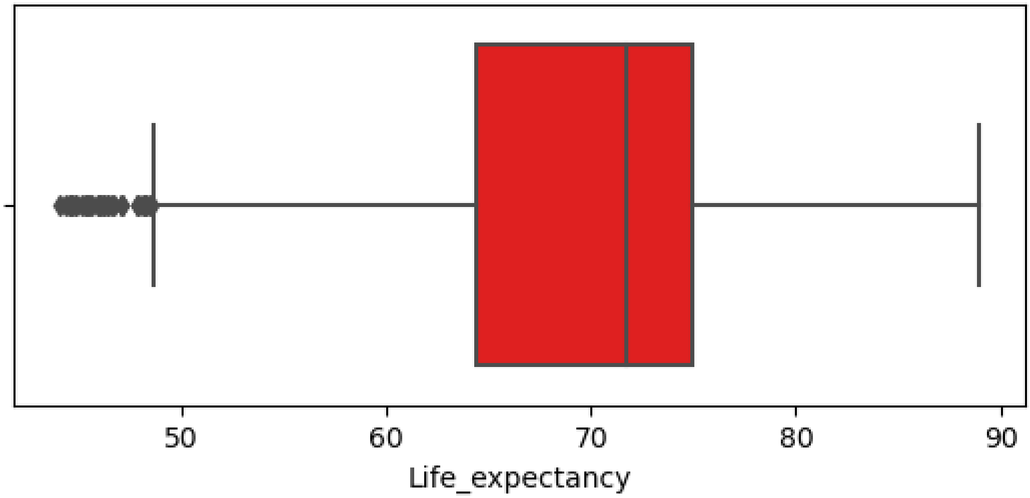

0.0
0.2
0.4
0.6
0.8
1.0
1.2
Population
1e9


0
5
10
15
20
25
thinness 1-19 years


0
5
10
15
20
25
thinness 5-9 years


0.0
0.2
0.4
0.6
0.8
Income_Cresources





From Above Observation.....

1. All Most All The Features Contains Outliers, Except Few Feature Columns such as BMI,IncomeCResource.

TREATING OUTLIERS

In [56]:

```
IQR = data['Life_expectancy'].quantile(0.75) - data['Life_expectancy'].quantile((

lower = data['Life_expectancy'].quantile(0.25) - 1.5* IQR
upper = data['Life_expectancy'].quantile(0.75) + 1.5* IQR

outliers = np.where(data['Life_expectancy']>upper,True, np.where(data['Life_expec

data = data.loc[~(outliers)]
```

In [57]:

```
IQR = data['Schooling'].quantile(0.75) - data['Schooling'].quantile(0.25)

lower = data['Schooling'].quantile(0.25) - 1.5* IQR
upper = data['Schooling'].quantile(0.75) + 1.5* IQR

outliers = np.where(data['Schooling']>upper,True, np.where(data['Schooling']<lowe

data = data.loc[~(outliers)]
```

In [58]:

```
IQR = data['thinness 5-9 years'].quantile(0.75) - data['thinness 5-9 years'].quan

lower = data['thinness 5-9 years'].quantile(0.25) - 1.5* IQR
upper = data['thinness 5-9 years'].quantile(0.75) + 1.5* IQR
```

```
outliers = np.where(data['thinness 5-9 years']>upper,True, np.where(data['thinnes

data = data.loc[~(outliers)]
```

In [59]:
```
IQR = data['Population'].quantile(0.75) - data['Population'].quantile(0.25)

lower = data['Population'].quantile(0.25) - 1.5* IQR
upper = data['Population'].quantile(0.75) + 1.5* IQR

outliers = np.where(data['Population']>upper,True, np.where(data['Population']<lo

data = data.loc[~(outliers)]
```

In [60]:
```
IQR = data['GDP'].quantile(0.75) - data['GDP'].quantile(0.25)

lower = data['GDP'].quantile(0.25) - 1.5* IQR
upper = data['GDP'].quantile(0.75) + 1.5* IQR

outliers = np.where(data['GDP']>upper,True, np.where(data['GDP']<lower,True,False

data = data.loc[~(outliers)]
```

In [61]:
```
IQR = data['HIV/AIDS'].quantile(0.75) - data['HIV/AIDS'].quantile(0.25)

lower = data['HIV/AIDS'].quantile(0.25) - 1.5* IQR
upper = data['HIV/AIDS'].quantile(0.75) + 1.5* IQR

outliers = np.where(data['HIV/AIDS']>upper,True, np.where(data['HIV/AIDS']<lower,

data = data.loc[~(outliers)]
```

In [62]:
```
IQR = data['Diphtheria'].quantile(0.75) - data['Diphtheria'].quantile(0.25)

lower = data['Diphtheria'].quantile(0.25) - 1.5* IQR
upper = data['Diphtheria'].quantile(0.75) + 1.5* IQR

outliers = np.where(data['Diphtheria']>upper,True, np.where(data['Diphtheria']<lo

data = data.loc[~(outliers)]
```

In [63]:
```
IQR = data['Total_Expenditure'].quantile(0.75) - data['Total_Expenditure'].quant

lower = data['Total_Expenditure'].quantile(0.25) - 1.5* IQR
upper = data['Total_Expenditure'].quantile(0.75) + 1.5* IQR

outliers = np.where(data['Total_Expenditure']>upper,True, np.where(data['Total_Ex

data = data.loc[~(outliers)]
```

In [64]:
```
IQR = data['Polio'].quantile(0.75) - data['Polio'].quantile(0.25)

lower = data['Polio'].quantile(0.25) - 1.5* IQR
upper = data['Polio'].quantile(0.75) + 1.5* IQR

outliers = np.where(data['Polio']>upper,True, np.where(data['Polio']<lower,True,F

data = data.loc[~(outliers)]
```

In [65]:
```
IQR = data['under-five deaths'].quantile(0.75) - data['under-five deaths'].quant

lower = data['under-five deaths'].quantile(0.25) - 1.5* IQR
upper = data['under-five deaths'].quantile(0.75) + 1.5* IQR

outliers = np.where(data['under-five deaths']>upper,True, np.where(data['under-fi
```

```
data = data.loc[~(outliers)]
```

In [66]:
```
IQR = data['Measles '].quantile(0.75) - data['Measles '].quantile(0.25)

lower = data['Measles '].quantile(0.25) - 1.5* IQR
upper = data['Measles '].quantile(0.75) + 1.5* IQR

outliers = np.where(data['Measles ']>upper,True, np.where(data['Measles ']<lower,

data = data.loc[~(outliers)]
```

In [67]:
```
IQR = data['Hepatitis_B'].quantile(0.75) - data['Hepatitis_B'].quantile(0.25)

lower = data['Hepatitis_B'].quantile(0.25) - 1.5* IQR
upper = data['Hepatitis_B'].quantile(0.75) + 1.5* IQR

outliers = np.where(data['Hepatitis_B']>upper,True, np.where(data['Hepatitis_B']<

data = data.loc[~(outliers)]
```

In [68]:
```
IQR = data['Percentage_Expenditure'].quantile(0.75) - data['Percentage_Expenditu

lower = data['Percentage_Expenditure'].quantile(0.25) - 1.5* IQR
upper = data['Percentage_Expenditure'].quantile(0.75) + 1.5* IQR

outliers = np.where(data['Percentage_Expenditure']>upper,True, np.where(data['Pe

data = data.loc[~(outliers)]
```

In [69]:
```
IQR = data['Alcohol'].quantile(0.75) - data['Alcohol'].quantile(0.25)

lower = data['Alcohol'].quantile(0.25) - 1.5* IQR
upper = data['Alcohol'].quantile(0.75) + 1.5* IQR

outliers = np.where(data['Alcohol']>upper,True, np.where(data['Alcohol']<lower,Tr

data = data.loc[~(outliers)]
```

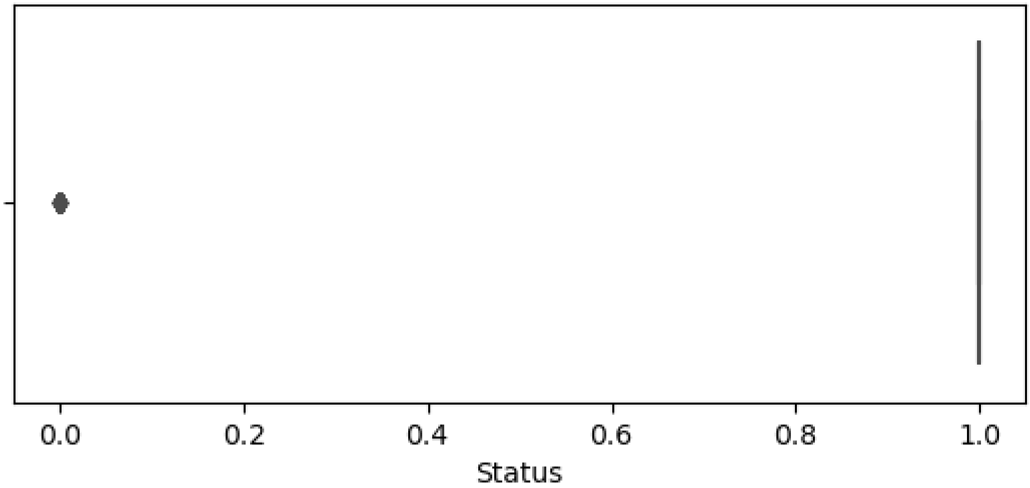
In [70]:
```
IQR = data['Infant_Deaths'].quantile(0.75) - data['Infant_Deaths'].quantile(0.25

lower = data['Infant_Deaths'].quantile(0.25) - 1.5* IQR
upper = data['Infant_Deaths'].quantile(0.75) + 1.5* IQR

outliers = np.where(data['Infant_Deaths']>upper,True, np.where(data['Infant_Deat

data = data.loc[~(outliers)]
```

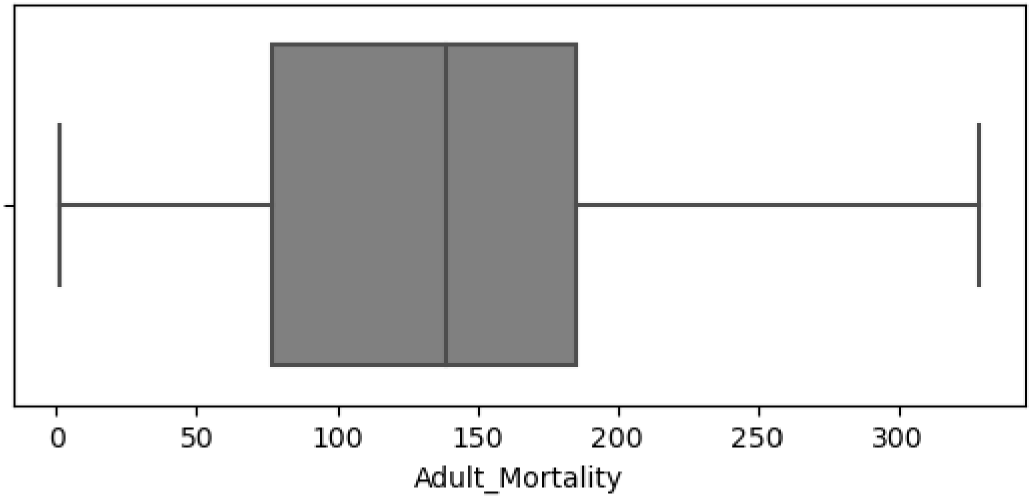
In [71]:
```
IQR = data['Adult_Mortality'].quantile(0.75) - data['Adult_Mortality'].quantile(

lower = data['Adult_Mortality'].quantile(0.25) - 1.5* IQR
upper = data['Adult_Mortality'].quantile(0.75) + 1.5* IQR

outliers = np.where(data['Adult_Mortality']>upper,True, np.where(data['Adult_Mort

data = data.loc[~(outliers)]
```

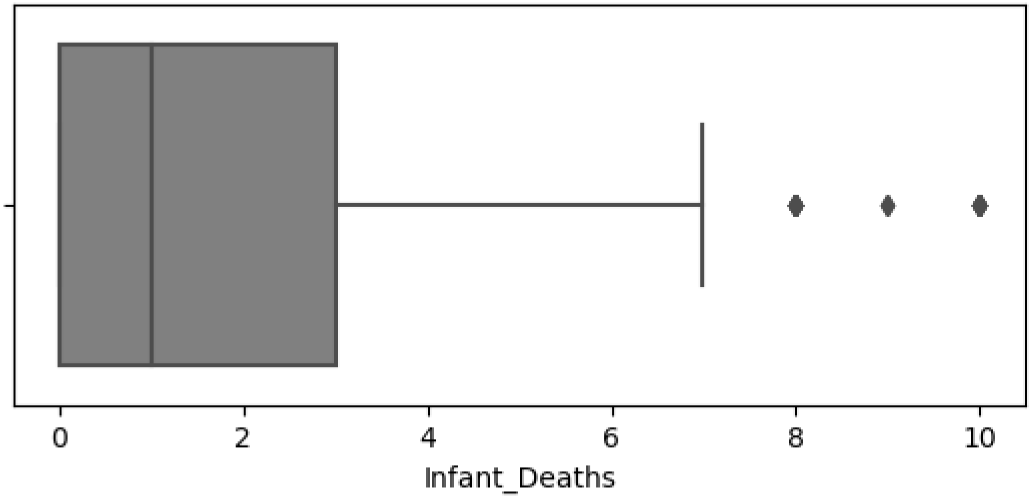
BOX PLOT AFTER TREATING OUTLIERS

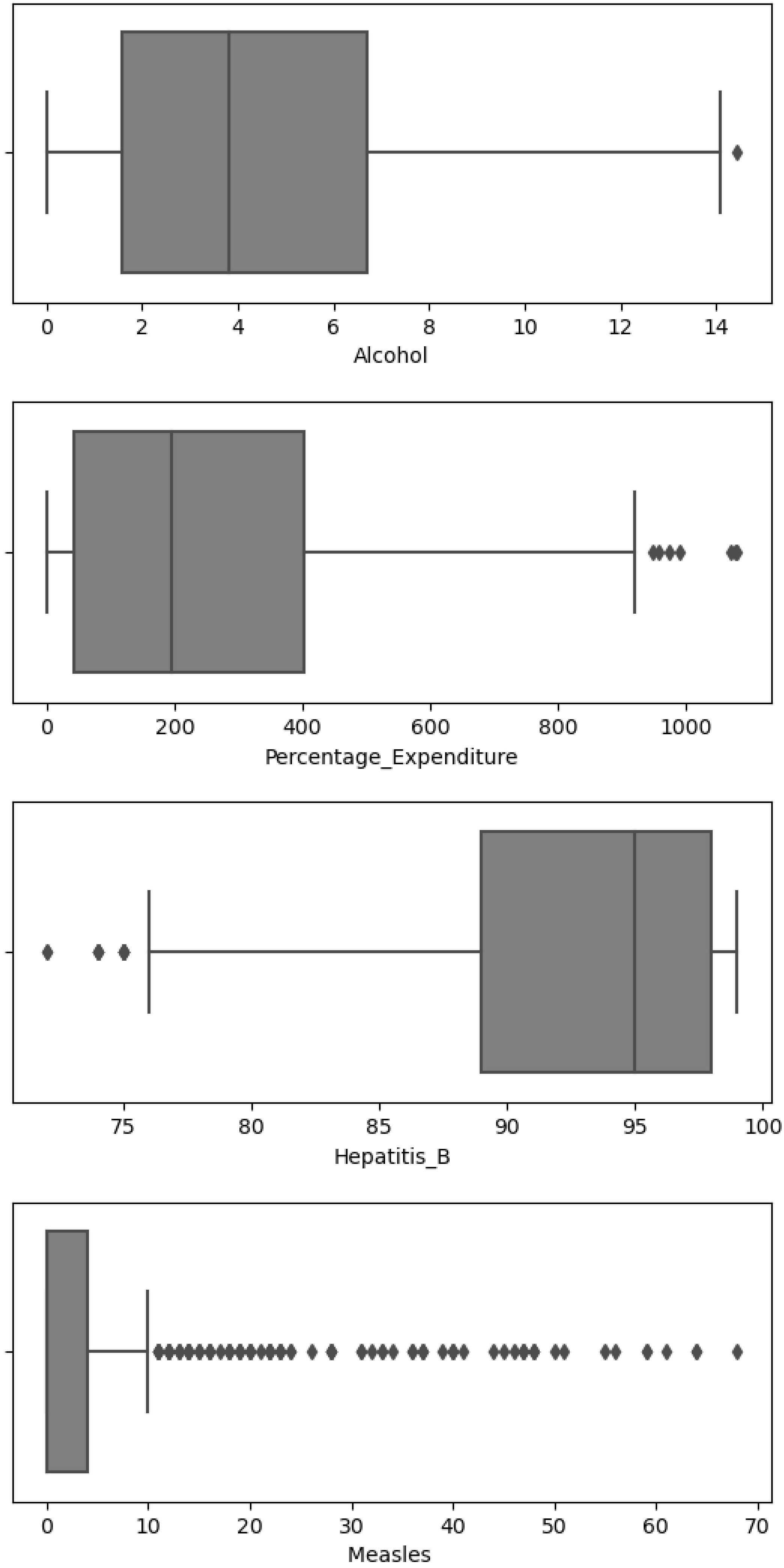
In [72]:
```
# Checking Outlier In The Datset
for col in enumerate(features):
    plt.figure(figsize=(30,15))
    plt.subplot(5,4,col[0]+1)
```

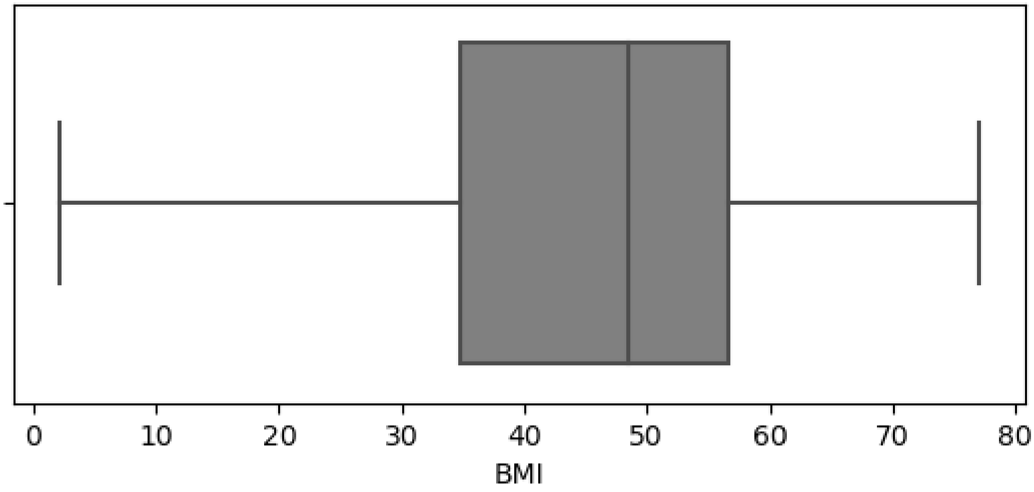
```
    sns.boxplot(x = col[1],color='gray',data=data)
    plt.show()
```

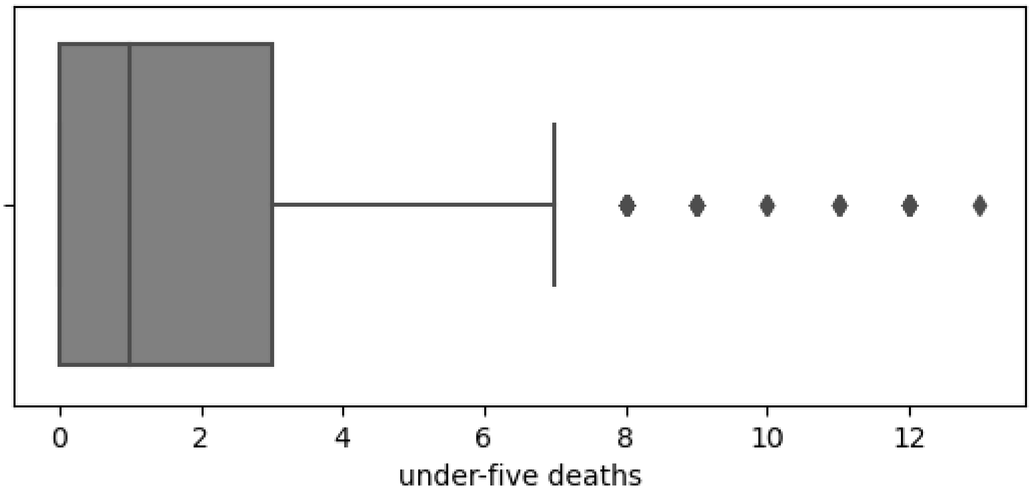


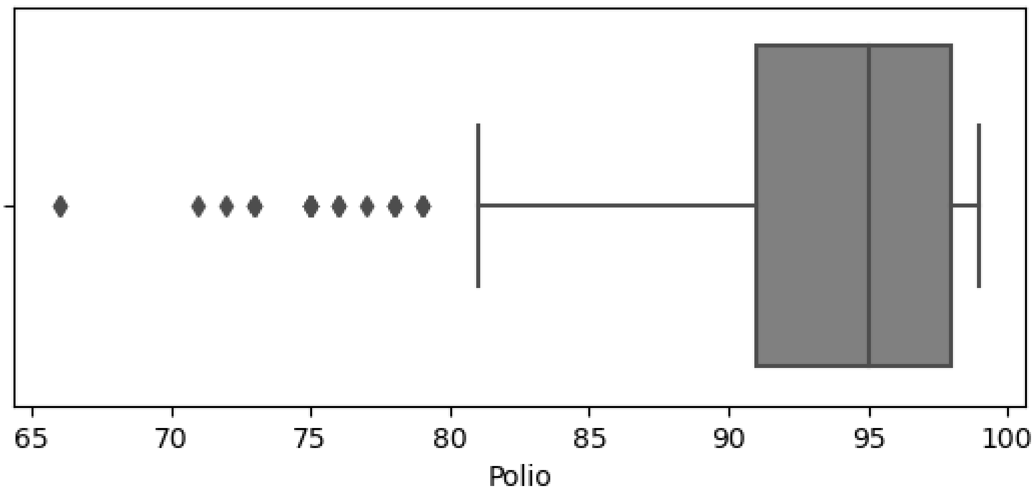


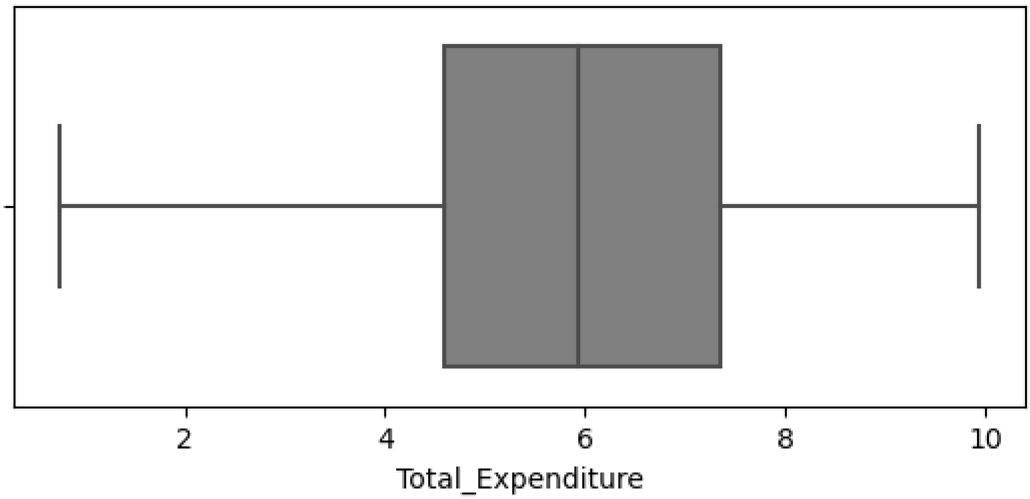


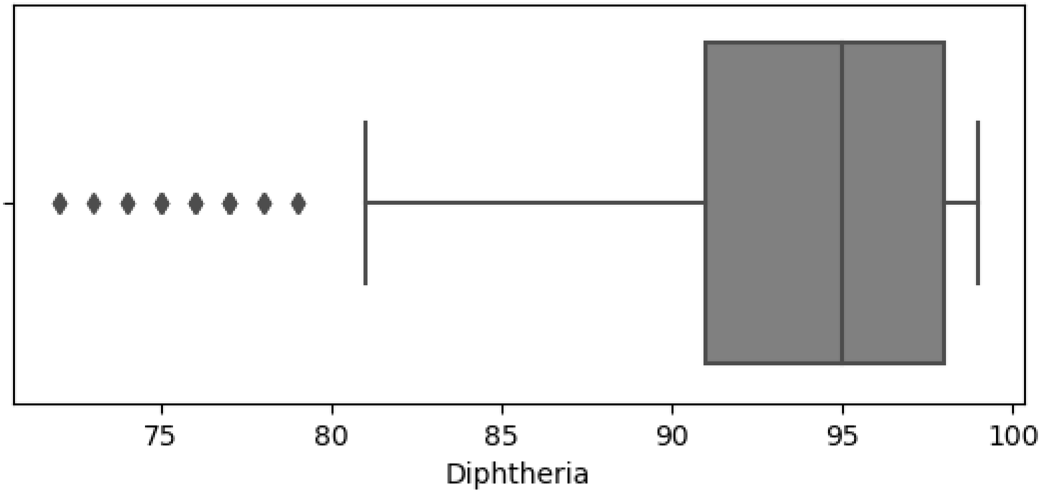

0 2 4 6 8 10 12 14
Alcohol
0 200 400 600 800 1000
Percentage_Expenditure
75 80 85 90 95 100
Hepatitis_B
0 10 20 30 40 50 60 70
Measles

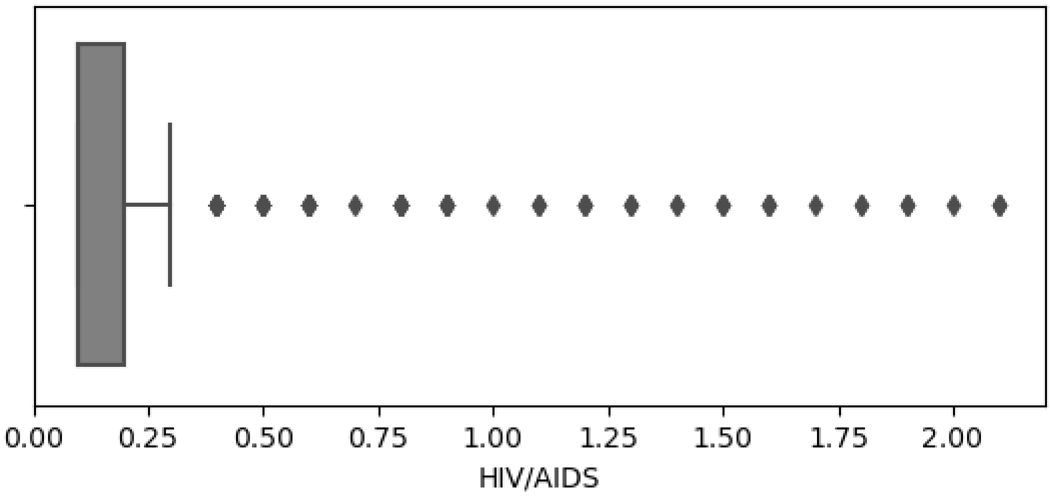

0 10 20 30 40 50 60 70 80
BMI

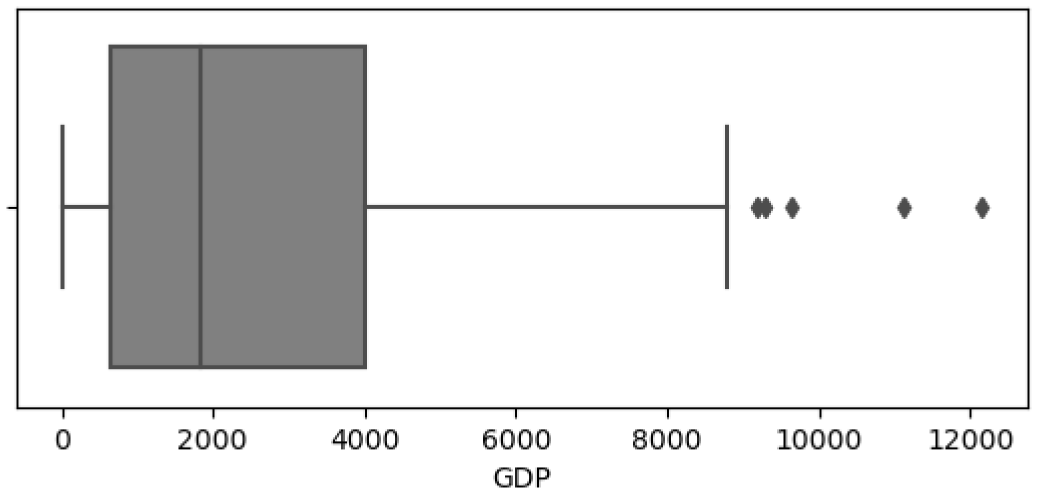

0 2 4 6 8 10 12
under-five deaths

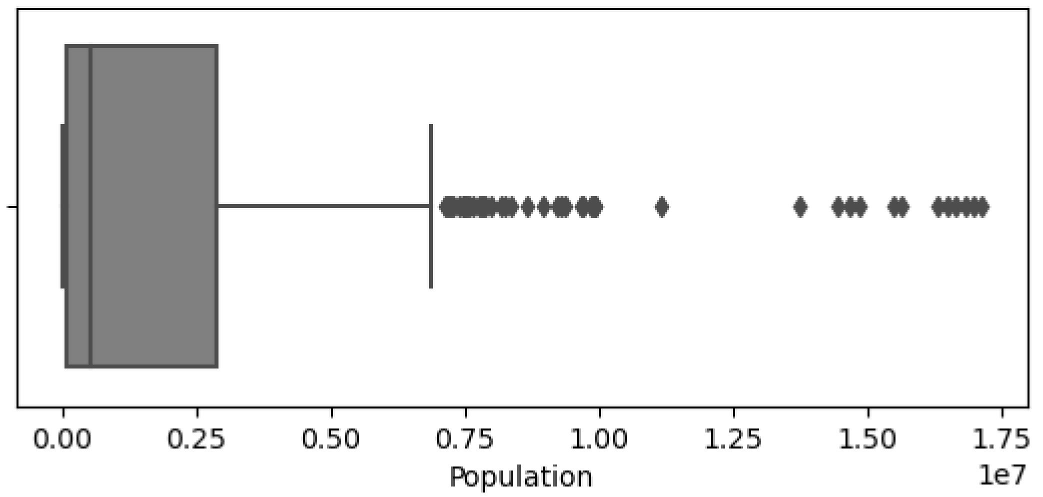

65 70 75 80 85 90 95 100
Polio

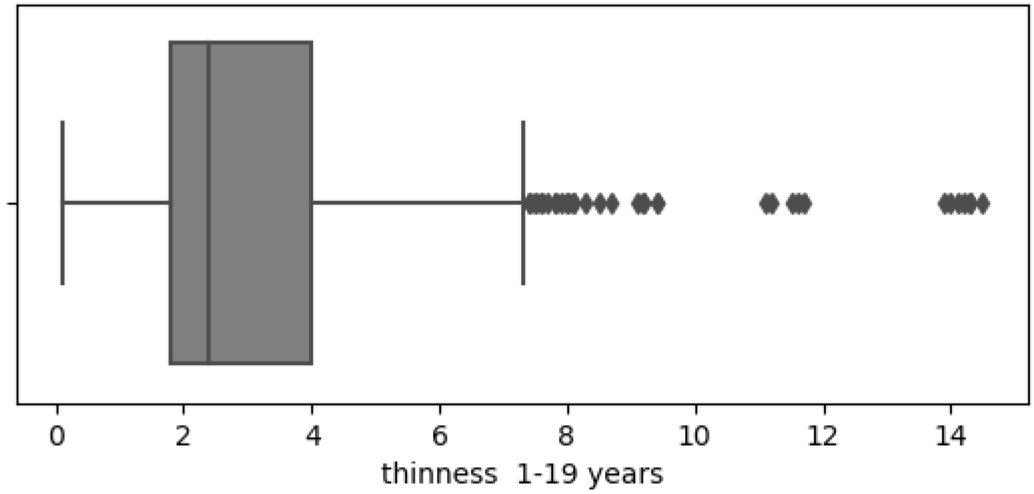

2 4 6 8 10
Total_Expenditure

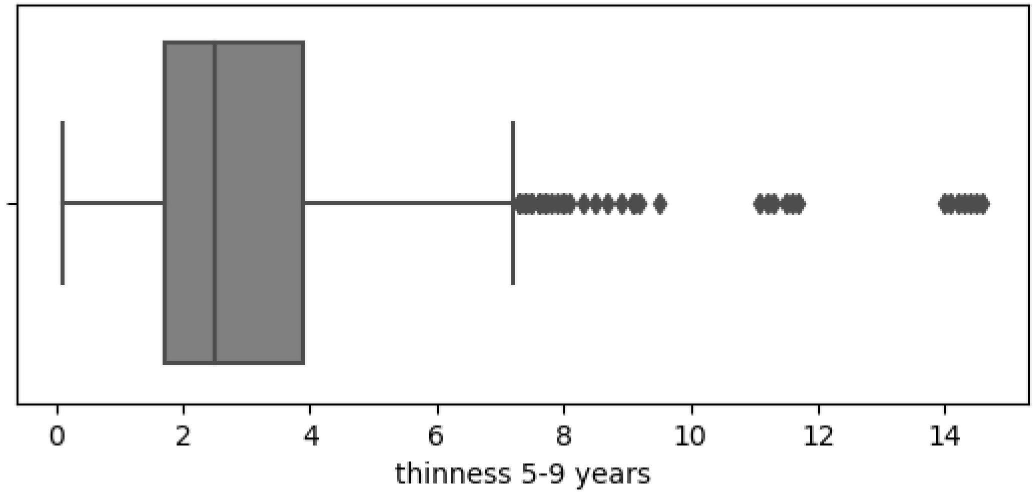

75
80
85
90
95
100
Diphtheria

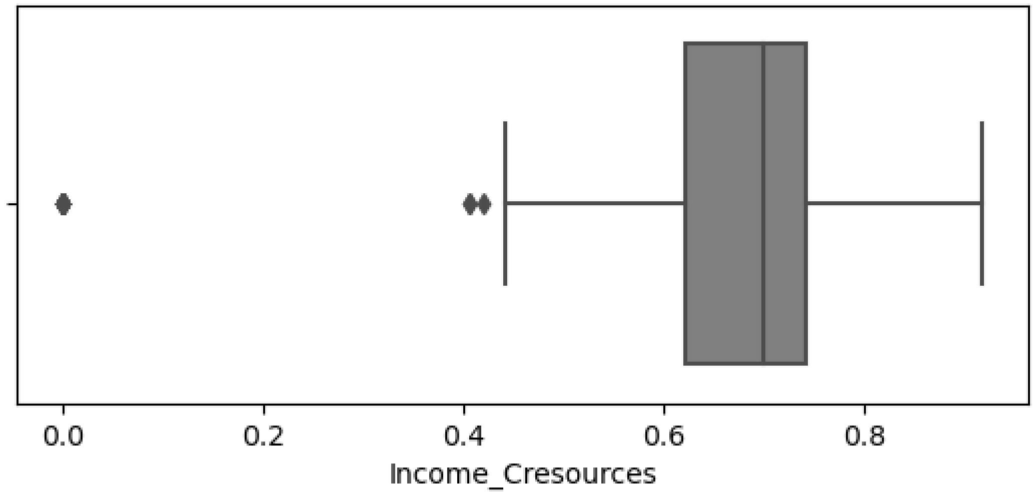

0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
HIV/AIDS

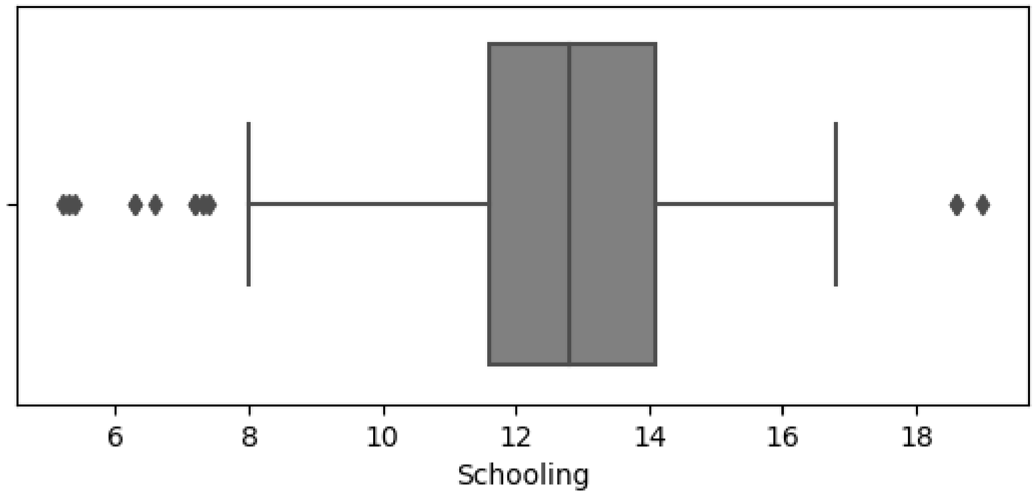

0
2000
4000
6000
8000
10000
12000
GDP

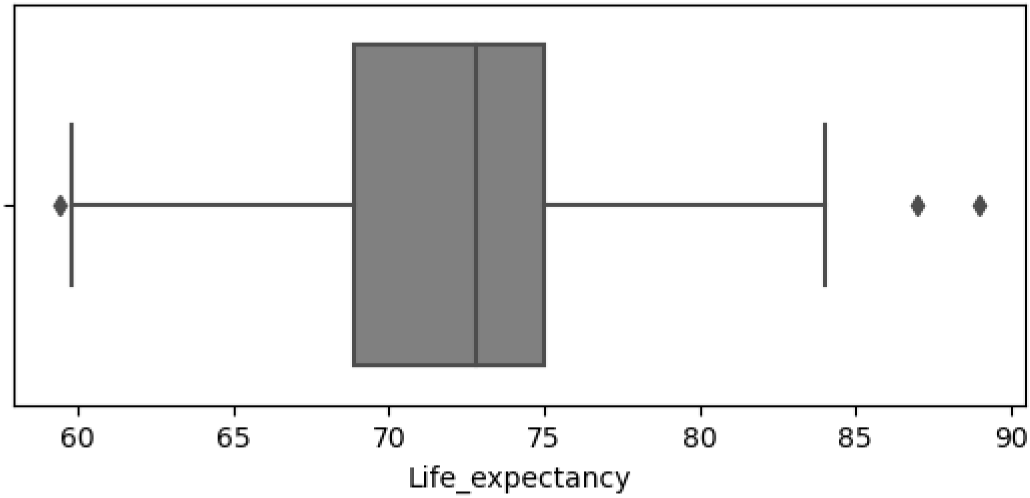

0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
1e7
Population


0
2
4
6
8
10
12
14
thinness 1-19 years


0
2
4
6
8
10
12
14
thinness 5-9 years


0.0
0.2
0.4
0.6
0.8
Income_Cresources


6
8
10
12
14
16
18
Schooling



In [73]:
```
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split

# Split Data set into Independent Features and Dependent Feature

X = data.iloc[:,:-1]
y = data.iloc[:,-1:]

X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.30,random_state=

scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [74]:
```
!pip install XGBoost
from sklearn.linear_model import LinearRegression, Ridge, Lasso, ElasticNet
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.model_selection import KFold, StratifiedKFold, cross_val_score
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: XGBoost in c:\users\user\appdata\roaming\python\p
ython310\site-packages (1.7.5)
Requirement already satisfied: numpy in d:\anaconda3\lib\site-packages (from XGB
oost) (1.23.5)
Requirement already satisfied: scipy in d:\anaconda3\lib\site-packages (from XGB
oost) (1.10.0)
```

In [75]:
```
def CVFold(models):
    score = cross_val_score(model,X_train,y_train,cv=CV,scoring = 'r2')
    print("Baseline mean R-squared from K-fold CV of {} is {}".format(model, rou
```

In [76]:
```
CV = KFold(n_splits=5, shuffle=True, random_state=23)
```

In [77]:
```
models = [LinearRegression(), Ridge(), Lasso(), DecisionTreeRegressor(), RandomFo
for model in models:
    CVFold(models)
```

```
Baseline mean R-squared from K-fold CV of LinearRegression() is 0.4412
Baseline mean R-squared from K-fold CV of Ridge() is 0.5905
Baseline mean R-squared from K-fold CV of Lasso() is -0.0056
Baseline mean R-squared from K-fold CV of DecisionTreeRegressor() is 0.6741
Baseline mean R-squared from K-fold CV of RandomForestRegressor() is 0.8551
Baseline mean R-squared from K-fold CV of XGBRegressor(base_score=None, booster=
None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             n_estimators=100, n_jobs=None, num_parallel_tree=None,
             predictor=None, random_state=None, ...) is 0.8279
Baseline mean R-squared from K-fold CV of RandomForestRegressor() is 0.8547
```

In [78]:

```
def TestXGBParams(**params):
    score = cross_val_score(XGBRegressor(**params, n_jobs=-1, random_state=23), X
    print("Mean R-squared from K-fold CV with {} is {}".format(params, round(np.r
```

In [79]:

```
estimators = [1,2,4,8,16,32,64,120,125,127,130,133,140,150,200,256]
for n in estimators:
    TestXGBParams(n_estimators=n)
```

```
Mean R-squared from K-fold CV with {'n_estimators': 1} is -108.6501
Mean R-squared from K-fold CV with {'n_estimators': 2} is -53.413
Mean R-squared from K-fold CV with {'n_estimators': 4} is -12.7212
Mean R-squared from K-fold CV with {'n_estimators': 8} is -0.1118
Mean R-squared from K-fold CV with {'n_estimators': 16} is 0.8105
Mean R-squared from K-fold CV with {'n_estimators': 32} is 0.8265
Mean R-squared from K-fold CV with {'n_estimators': 64} is 0.8279
Mean R-squared from K-fold CV with {'n_estimators': 120} is 0.8279
Mean R-squared from K-fold CV with {'n_estimators': 125} is 0.8279
Mean R-squared from K-fold CV with {'n_estimators': 127} is 0.8279
Mean R-squared from K-fold CV with {'n_estimators': 130} is 0.8279
Mean R-squared from K-fold CV with {'n_estimators': 133} is 0.8279
Mean R-squared from K-fold CV with {'n_estimators': 140} is 0.8279
Mean R-squared from K-fold CV with {'n_estimators': 150} is 0.8279
Mean R-squared from K-fold CV with {'n_estimators': 200} is 0.8279
Mean R-squared from K-fold CV with {'n_estimators': 256} is 0.8279
```

In [80]:

```
depths = [1, 2, 4,6,8,10,12,14,16,20,25]
for n in depths:
    TestXGBParams(n_estimators = 120, max_depth = n)
```

```
Mean R-squared from K-fold CV with {'n_estimators': 120, 'max_depth': 1} is 0.82
75
Mean R-squared from K-fold CV with {'n_estimators': 120, 'max_depth': 2} is 0.84
97
Mean R-squared from K-fold CV with {'n_estimators': 120, 'max_depth': 4} is 0.85
58
Mean R-squared from K-fold CV with {'n_estimators': 120, 'max_depth': 6} is 0.82
79
Mean R-squared from K-fold CV with {'n_estimators': 120, 'max_depth': 8} is 0.81
99
Mean R-squared from K-fold CV with {'n_estimators': 120, 'max_depth': 10} is 0.8
26
Mean R-squared from K-fold CV with {'n_estimators': 120, 'max_depth': 12} is 0.8
231
Mean R-squared from K-fold CV with {'n_estimators': 120, 'max_depth': 14} is 0.8
239
Mean R-squared from K-fold CV with {'n_estimators': 120, 'max_depth': 16} is 0.8
228
Mean R-squared from K-fold CV with {'n_estimators': 120, 'max_depth': 20} is 0.8
227
Mean R-squared from K-fold CV with {'n_estimators': 120, 'max_depth': 25} is 0.8
227
```

In [81]:

```
rates = [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1]
for n in rates:
    TestXGBParams(n_estimators = 128, max_depth = 4, learning_rate = n)
```

```
Mean R-squared from K-fold CV with {'n_estimators': 128, 'max_depth': 4, 'learni
ng_rate': 0.1} is 0.8815
Mean R-squared from K-fold CV with {'n_estimators': 128, 'max_depth': 4, 'learni
ng_rate': 0.2} is 0.8682
Mean R-squared from K-fold CV with {'n_estimators': 128, 'max_depth': 4, 'learni
ng_rate': 0.3} is 0.8558
Mean R-squared from K-fold CV with {'n_estimators': 128, 'max_depth': 4, 'learni
ng_rate': 0.4} is 0.8316
Mean R-squared from K-fold CV with {'n_estimators': 128, 'max_depth': 4, 'learni
ng_rate': 0.5} is 0.8441
Mean R-squared from K-fold CV with {'n_estimators': 128, 'max_depth': 4, 'learni
ng_rate': 0.6} is 0.8186
Mean R-squared from K-fold CV with {'n_estimators': 128, 'max_depth': 4, 'learni
ng_rate': 0.7} is 0.7853
Mean R-squared from K-fold CV with {'n_estimators': 128, 'max_depth': 4, 'learni
ng_rate': 0.8} is 0.7632
Mean R-squared from K-fold CV with {'n_estimators': 128, 'max_depth': 4, 'learni
ng_rate': 0.9} is 0.8004
Mean R-squared from K-fold CV with {'n_estimators': 128, 'max_depth': 4, 'learni
ng_rate': 1} is 0.7916
```

In [82]:

```
model = XGBRegressor(n_estimators = 256, max_depth = 4, learning_rate = .2, n_jol
```

In [83]:

```
model.fit(X_train, y_train)
```

Out[83]:

▾ XGBRegressor

```
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=N
one,
             gamma=None, gpu_id=None, grow_policy=None, importance_type=
None,
             interaction_constraints=None, learning_rate=0.2, max_bin=No
ne,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=4, max_leaves=None,
```

In [84]:

```
y_pred = model.predict(X_test)
r_squared = r2_score(y_test, y_pred)
MSE = mean_squared_error(y_test,y_pred)
RMSE = np.sqrt(mean_squared_error(y_test, y_pred))
MAE = mean_absolute_error(y_test,y_pred)

print('Our Optimized XGBRegressor got the following scores on the test set:')
print('R-squared: {}'.format(r_squared))
print('MSE: {}'.format(MSE))
print('RMSE: {}'.format(RMSE))
print('MAE: {}'.format(MAE))
```

```
Our Optimized XGBRegressor got the following scores on the test set:
R-squared: 0.8897756796050695
MSE: 2.179334571879111
RMSE: 1.4762569464287412
MAE: 1.0218386278084828
```

In [85]:

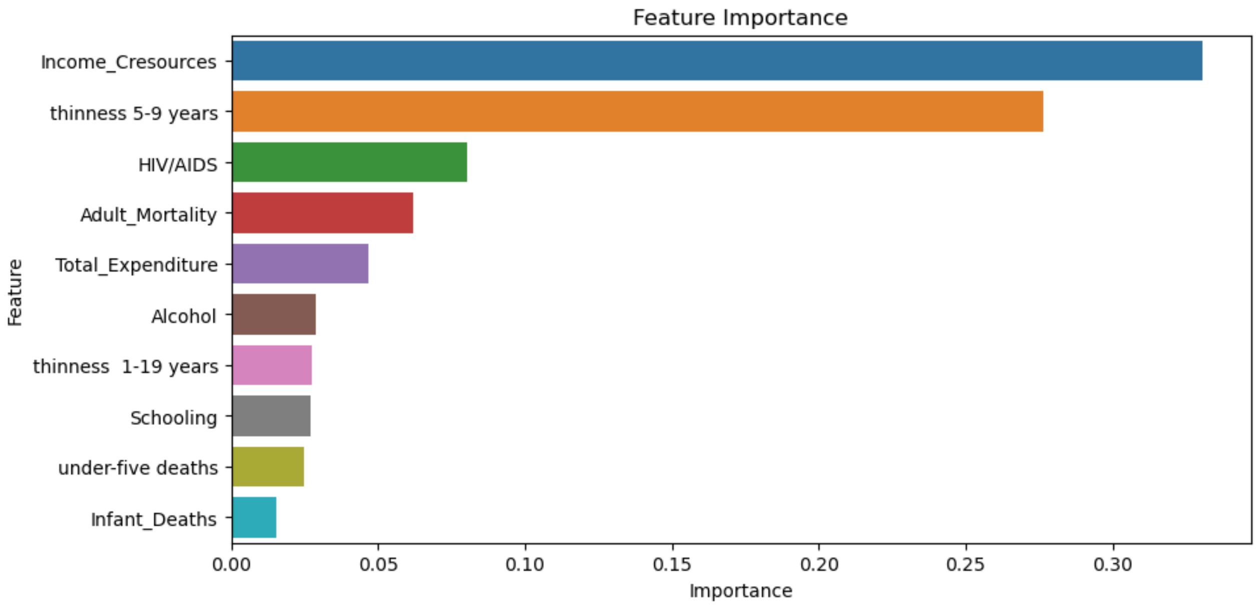
```
# Feature Importance
importances = pd.DataFrame({
    'Feature': X.columns,
    'Importance': model.feature_importances_
}).sort_values('Importance', ascending=False)
importances
```

Out[85]:

	Feature	Importance
17	Income_Cresources	0.330454
16	thinness 5-9 years	0.276460
12	HIV/AIDS	0.080364
1	Adult_Mortality	0.061961
10	Total_Expenditure	0.046582
3	Alcohol	0.028856
15	thinness 1-19 years	0.027336
18	Schooling	0.026996
8	under-five deaths	0.024626
2	Infant_Deaths	0.015404
5	Hepatitis_B	0.014784
7	BMI	0.014746
11	Diphtheria	0.012641
0	Status	0.012038
14	Population	0.007276
13	GDP	0.006005
9	Polio	0.005891
4	Percentage_Expenditure	0.004228
6	Measles	0.003352

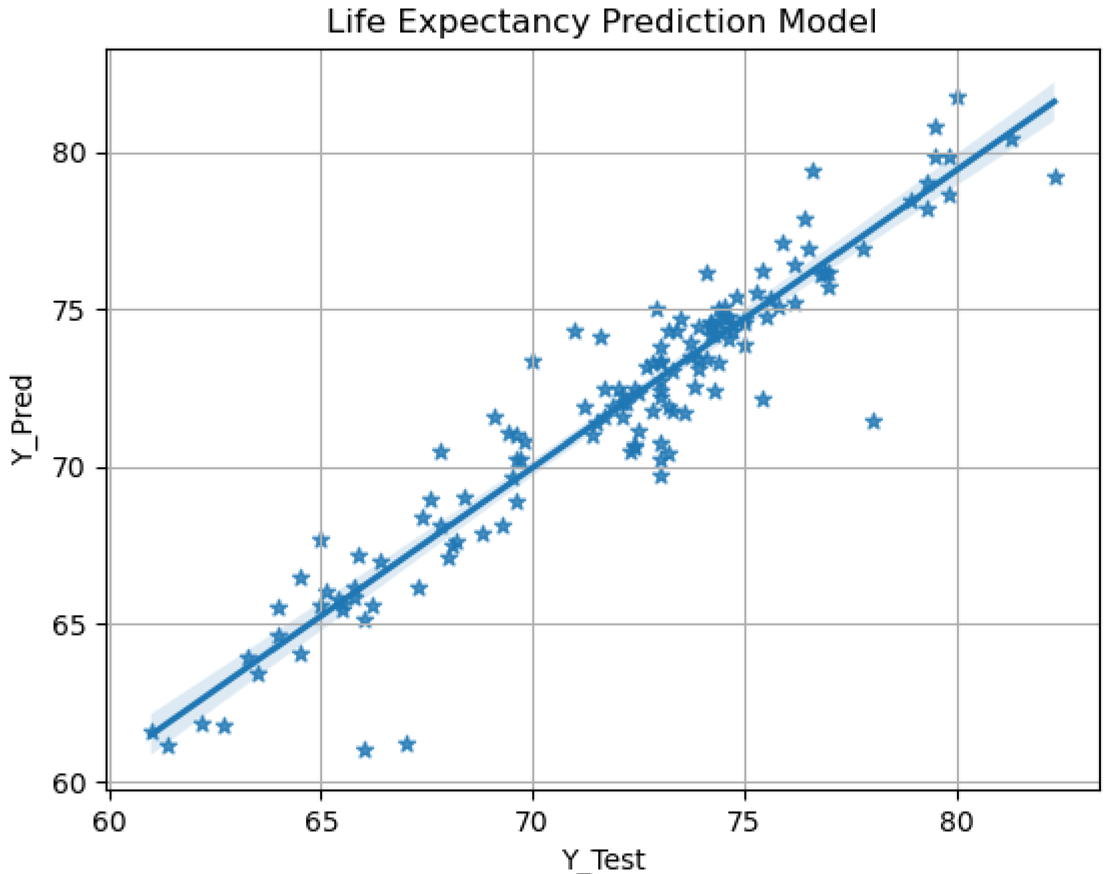
In [86]:

```
plt.figure(figsize=(10,5))
plt.title('Feature Importance')
sns.barplot(data=importances.head(10), x='Importance', y='Feature');
```

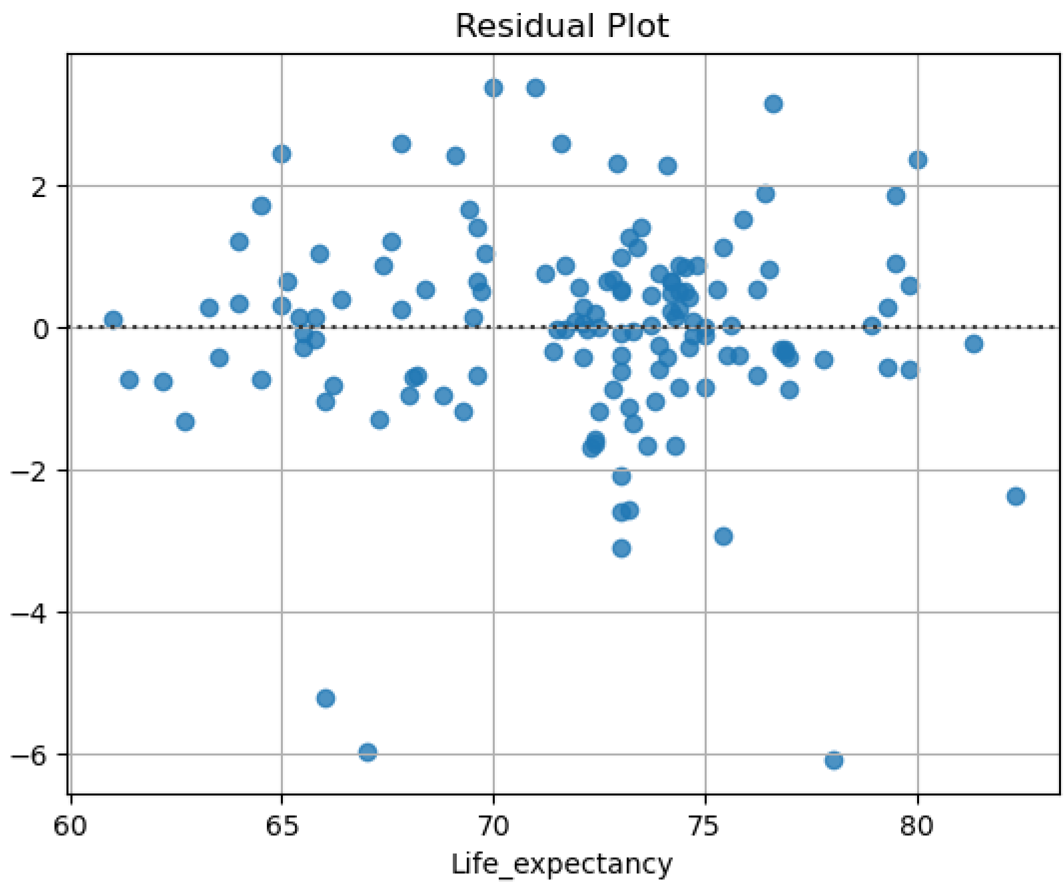


In [87]:

```
sns.regplot(x=y_test,y=y_pred,marker='*')
plt.xlabel('y_test')
plt.title('Life Expectancy Prediction Model')
plt.xlabel('Y_Test')
plt.ylabel('Y_Pred')
plt.grid()
plt.show()
```



In [88]:
```
sns.residplot(x=y_test,y=y_pred)
plt.title('Residual Plot')
plt.grid()
plt.show()
```



Saving Model as Pickle File

In [89]:

```python
import pickle
filename = "LifeExpectancy_RegressionModel.pkl"
```

In [90]:

```python
# Searialize Process
pickle.dump(data,open(filename,'wb'))
```

In [91]:

```python
# UnSearialize Process
pickle.load(open('LifeExpectancy_RegressionModel.pkl','rb'))
```

Out[91]:

	Status	Adult_Mortality	Infant_Deaths	Alcohol	Percentage_Expenditure	Hepatitis_B	Measles	E
16	1	74.0	0	4.60	364.975229	99.0	0	5
17	1	8.0	0	4.51	428.749067	98.0	0	5
18	1	84.0	0	4.76	430.876979	99.0	0	5
19	1	86.0	0	5.14	412.443356	99.0	9	5
20	1	88.0	0	5.37	437.062100	99.0	28	5
...	...	...	...	...	...	...	...	
2817	1	119.0	1	6.76	24.731423	94.0	0	
2818	1	124.0	1	6.67	14.473059	94.0	0	5
2822	1	121.0	1	5.11	160.840014	91.0	0	5
2823	1	124.0	1	5.86	27.468810	95.0	0	5
2824	1	123.0	1	6.48	421.480428	94.0	0	5

469 rows × 20 columns

Thank you