Analysis of Life Expectancy data across nations

Introduction:

Our analysis utilizes data from the World Health Organization and the United Nations regarding the life expectancies of various countries. Specifically, our dataset contains data provided by the GHO (a public health observatory established under the WHO), which provides publically available data on the health statuses of various countries. This includes immunization factors and mortality factors, to name a few. It combines this data with data provided by the United Nations, which provides economic data for said countries. In total, our dataset includes data from 193 countries between the years 2000 and 2015.

The objective of our analysis is to determine which economic and health-related factors affect life expectancy. Specifically, we want to examine several questions:

1. Whether or not increased adult and child mortality can have a negative effect on a country's life expectancy.
2. Whether or not BMI and alcohol consumption affect life expectancy.
3. Whether or not increased immunization coverage for various diseases such as diptheria, measles, and Hep. B can have a positive effect on life expectancy.
4. Whether or not economic factors such as a country's development status, GDP, and education can affect their life expectancy. We specifically want to investigate whether or not being a developed country can have a positive impact on a country's life expectancy, as well as whether or not increased GDP and education can have a positive impact on life expectancy.
5. Whether or not countries in certain regions are more prone to having higher or lower life expectancies.

In [31]:
```
import statistics
import numpy as np
import pandas as pd
from pandas import DataFrame as df
import scipy as sp
```

```
from scipy import io as spio
from scipy import stats
import matplotlib.pyplot as plt
from scipy import misc
from scipy.stats import norm
import warnings
%matplotlib inline
warnings.filterwarnings('ignore')
import seaborn as sns
sns.set(style="whitegrid")
import math
from BorutaShap import BorutaShap
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import RandomForestClassifier
import random
import statsmodels.formula.api as smf
import statsmodels.stats.api as sms
from sklearn.model_selection import train_test_split
from sklearn import linear_model
from sklearn.linear_model import LinearRegression
from sklearn import metrics
from sklearn.model_selection import cross_val_score
#!pip install fitter
from fitter import Fitter
from statsmodels.stats.outliers_influence import variance_inflation_factor
import patsy as pt
import statsmodels.api as sm
import statsmodels.stats.outliers_influence as smo

#!pip install matplotlib==3.7.3 (for some reason the correlation map wasn't displaying numbers unless I
```

In [32]:
```
dt= pd.read_csv("Life-Expectancy-Data-Updated.csv", encoding='latin-1')
dt=dt[
    ["Region","Life_expectancy","Under_five_deaths","Adult_mortality","Alcohol_consumption","Hepatitis_B
     "Diphtheria","Incidents_HIV","GDP_per_capita","Schooling","Economy_status_Developed"]]
dt.sort_values(by=['Region'], inplace=True)
dt.head()
```

Out[32]:

	Region	Life_expectancy	Under_five_deaths	Adult_mortality	Alcohol_consumption	Hepatitis_B	Measles	BMI	Diphtheria	Incidents_HIV
2296	Africa	57.8	88.5	293.6475	1.91	80	55	22.9	80	1.48
769	Africa	52.9	140.2	397.8705	3.06	86	60	22.7	86	0.85
1996	Africa	51.1	94.2	484.2665	1.64	77	28	22.1	80	4.36
771	Africa	58.9	108.3	267.2085	5.49	88	34	22.0	88	0.23
773	Africa	72.6	15.2	157.4485	3.23	97	80	24.9	97	1.05

In [33]:
```
dt.shape
```

Out[33]: (2864, 13)

In [34]:
```
dt.iloc[:,1:12]
```

Out[34]:

	Life_expectancy	Under_five_deaths	Adult_mortality	Alcohol_consumption	Hepatitis_B	Measles	BMI	Diphtheria	Incidents_HIV	GDP_p
2296	57.8	88.5	293.6475	1.91	80	55	22.9	80	1.48	
769	52.9	140.2	397.8705	3.06	86	60	22.7	86	0.85	
1996	51.1	94.2	484.2665	1.64	77	28	22.1	80	4.36	
771	58.9	108.3	267.2085	5.49	88	34	22.0	88	0.23	
773	72.6	15.2	157.4485	3.23	97	80	24.9	97	1.05	
...	...	...	...	...	...	...	...	...	...	
1856	72.3	27.7	155.1280	6.89	87	84	25.8	89	0.15	
647	76.1	16.6	142.8290	4.32	91	72	26.0	91	0.20	
645	73.5	23.4	137.5050	4.51	93	51	25.7	93	0.18	
2713	71.9	32.7	148.5315	3.93	84	52	25.2	95	0.20	
1353	75.4	13.9	122.6690	7.69	91	91	27.2	91	0.14	

2864 rows × 11 columns

Boruta Algorithm

In [35]:
```
bral_dt=dt.copy()
```

In [36]:
```
x=bral_dt.iloc[:,1:12]
y=bral_dt["Life_expectancy"]
```

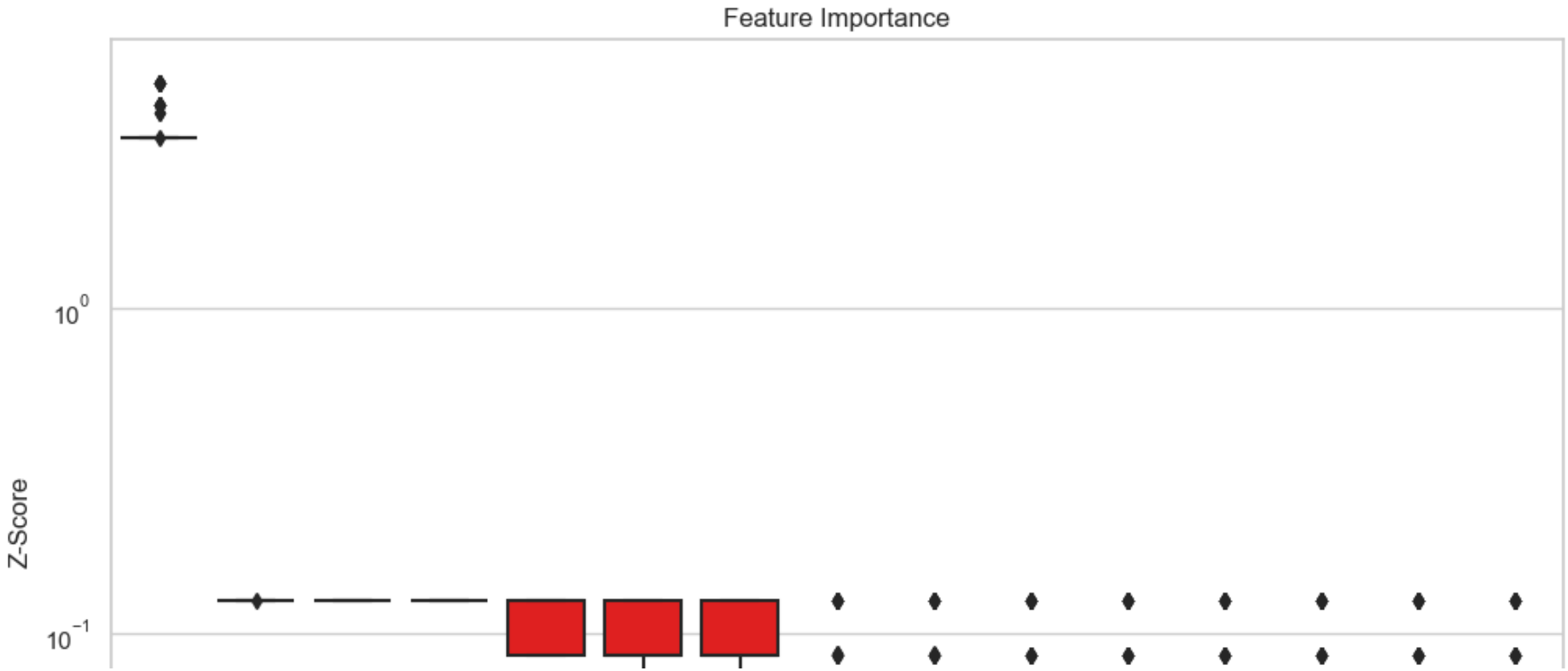
In [37]:

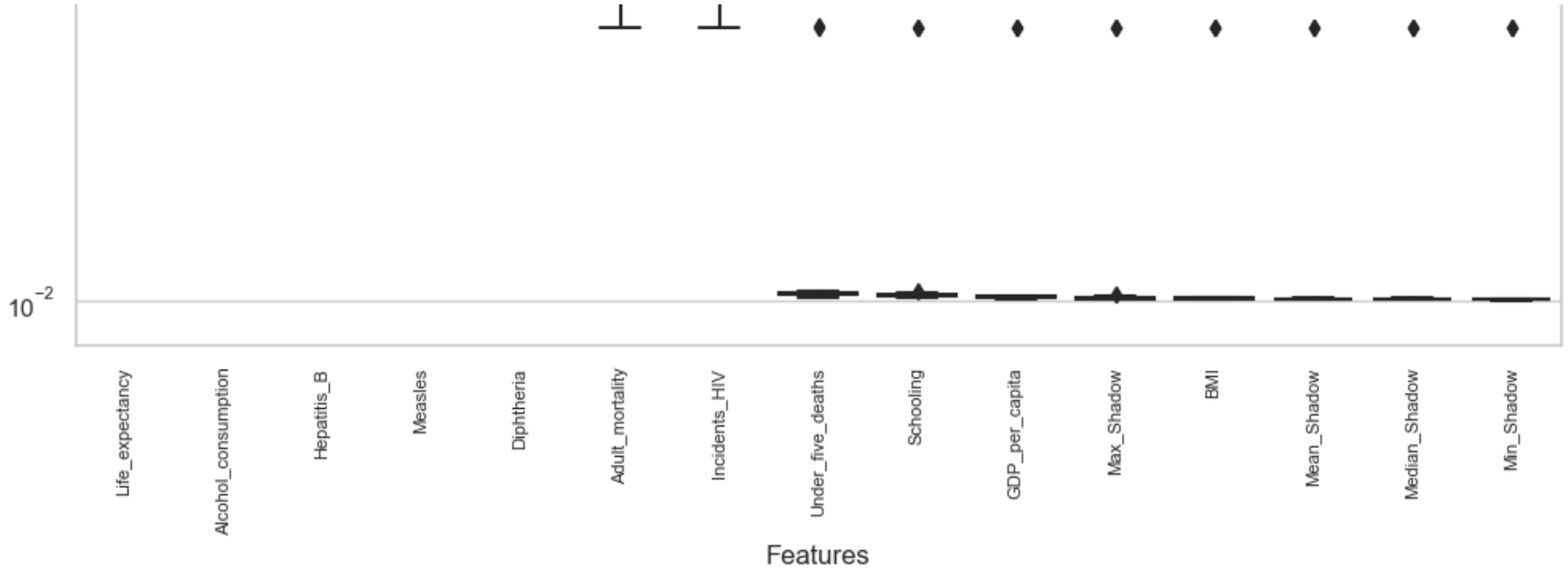
In [37]:

```
Feature_Selector= BorutaShap(importance_measure = "shap", classification = False)
Feature_Selector.fit(X=x, y=y, n_trials = 100, random_state=0)
Feature_Selector.plot(which_features="all")
```

A Jupyter widget could not be displayed because the widget state could not be found. This could happen if the kernel storing the widget is no longer available, or if the widget state was not saved in the notebook. You may be able to create the widget by running the appropriate cells.

```
4 attributes confirmed important: ['Life_expectancy', 'GDP_per_capita', 'Schooling', 'Under_five_death
s']
6 attributes confirmed unimportant: ['Diphtheria', 'Measles', 'Incidents_HIV', 'Hepatitis_B', 'Adult_mo
rtality', 'Alcohol_consumption']
1 tentative attributes remains: ['BMI']
```





Since all 15 quantitative variables are important as per our above analysis. Therefore, our variable selection will include: Under_five_deaths, Adult_mortality, GDP_per_capita, Alcohol_consumption, Schooling, BMI, Incidents_HIV, Measles, Hepatitis_B and Diptheria. We have selected these important variables based on the questions we are hoping to answer from our analysis. We are incorporating these to see dependency of Life Expectancy data on immunization, mortality rate and economic and social factors.

In [ ]:
```
dt= pd.read_csv("Life-Expectancy-Data-Updated.csv", encoding='latin-1')
dt=dt[["Region","Life_expectancy","Under_five_deaths","Adult_mortality","Alcohol_consumption","Hepatitis
dt.sort_values(by=['Region'], inplace=True)
dt.head()
```

Q1(b) Identify Factor Variables

In [8]:
```
# identify unique values in region
dt['Region'].unique()
```

Out[8]:
```
array(['Africa', 'Asia', 'Central America and Caribbean',
       'European Union', 'Middle East', 'North America', 'Oceania',
       'Rest of Europe', 'South America'], dtype=object)
```

In [9]:
```
factor1 = smf.ols('Life_expectancy~Region', data=dt).fit()
factor1.summary() #correlation between region and life expectancy
```

Out[9]:

OLS Regression Results

Dep. Variable:	Life_expectancy	**R-squared:**	0.621
Model:	OLS	**Adj. R-squared:**	0.620
Method:	Least Squares	**F-statistic:**	585.3
Date:	Wed, 22 Nov 2023	**Prob (F-statistic):**	0.00
Time:	12:57:11	**Log-Likelihood:**	-9092.2
No. Observations:	2864	**AIC:**	1.820e+04
Df Residuals:	2855	**BIC:**	1.826e+04
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	57.8473	0.203	285.071	0.000	57.449	58.245
Region[T.Asia]	11.6076	0.345	33.655	0.000	10.931	12.284
Region[T.Central America and Caribbean]	14.5902	0.389	37.459	0.000	13.826	15.354
Region[T.European Union]	19.8677	0.345	57.604	0.000	19.191	20.544

Region[T.Middle East]	16.1281	0.437	36.886	0.000	15.271	16.985
Region[T.North America]	19.9944	0.861	23.224	0.000	18.306	21.682
Region[T.Oceania]	11.6703	0.482	24.224	0.000	10.726	12.615
Region[T.Rest of Europe]	16.6781	0.426	39.182	0.000	15.843	17.513
Region[T.South America]	14.9334	0.465	32.118	0.000	14.022	15.845

Omnibus:	105.586	**Durbin-Watson:**	2.014
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	153.290
Skew:	0.359	**Prob(JB):**	5.17e-34
Kurtosis:	3.877	**Cond. No.**	8.49

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [10]:
```
factor2 = smf.ols('Life_expectancy~Economy_status_Developed', data=dt).fit()
factor2.summary() #correlation between developed countries and life expectancy
```

Out[10]:

OLS Regression Results

Dep. Variable:	Life_expectancy	**R-squared:**	0.274
Model:	OLS	**Adj. R-squared:**	0.274
Method:	Least Squares	**F-statistic:**	1082.
Date:	Wed, 22 Nov 2023	**Prob (F-statistic):**	1.41e-201
Time:	12:57:18	**Log-Likelihood:**	-10023.
No. Observations:	2864	**AIC:**	2.005e+04

Df Residuals:	2862	**BIC:**	2.006e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	66.3417	0.168	394.609	0.000	66.012	66.671
Economy_status_Developed	12.1640	0.370	32.895	0.000	11.439	12.889

Omnibus:	234.957	**Durbin-Watson:**	0.929
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	295.966
Skew:	-0.787	**Prob(JB):**	5.39e-65
Kurtosis:	3.063	**Cond. No.**	2.59

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

We identified two factor variables to include as predictors: Region and Economy_status_Developed. The linear regression results above suggest that both Region and Economy_status_Developed are correlated with life expectancy. Given that the p-values of the estimated coefficients for both variables are less than 5%, we can conclude that the relationships are statistically significant. Additionally, the p-values of the F-statistics in both regression models are less than 5%, confirming that the coefficients are not zero. Therefore, we will combine these two factor variables with the ones identified in part (a) to perform analyses for parts (2) and (3).

The regression model with combined predictors takes the form:

$$Life\ Expectancy = \beta_1 + \beta_2 Under\ Five\ Deaths + \beta_3 Adult\ Mortality + \beta_4 Alcohol\ Consumption + \beta_5 Hepatitis\ B + \beta_6 Measles + \beta_7 BMI + \beta_8 Dipthe + \beta_9 HIV + \beta_{10} GDP\ per\ capita + \beta_{11} Schooling + \delta_1 Region + \delta_2 Economy\ Status\ Developed + e.$$

Q2(a) Descriptive Analysis of Variables

In [11]:
```
print(dt.head())
```

```
      Region  Life_expectancy  Under_five_deaths  Adult_mortality  \
2296  Africa             57.8               88.5         293.6475
769   Africa             52.9              140.2         397.8705
1996  Africa             51.1               94.2         484.2665
771   Africa             58.9              108.3         267.2085
773   Africa             72.6               15.2         157.4485

      Alcohol_consumption  Hepatitis_B  Measles   BMI  Diphtheria  \
2296                 1.91           80       55  22.9          80
769                  3.06           86       60  22.7          86
1996                 1.64           77       28  22.1          80
771                  5.49           88       34  22.0          88
773                  3.23           97       80  24.9          97

      Incidents_HIV  GDP_per_capita  Schooling  Economy_status_Developed
2296           1.48            1105        6.3                         0
769            0.85             588        3.4                         0
1996           4.36            1157        5.4                         0
771            0.23             639        1.4                         0
773            1.05            6939        7.6                         0
```

In [12]:
```
#check its contents
dt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2864 entries, 2296 to 1353
Data columns (total 13 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Region                    2864 non-null   object 
 1   Life_expectancy           2864 non-null   float64
 2   Under_five_deaths         2864 non-null   float64
 3   Adult_mortality           2864 non-null   float64
 4   Alcohol_consumption       2864 non-null   float64
 5   Hepatitis_B               2864 non-null   int64  
 6   Measles                   2864 non-null   int64  
 7   BMI                       2864 non-null   float64
 8   Diphtheria                2864 non-null   int64  
 9   Incidents_HIV             2864 non-null   float64
 10  GDP_per_capita            2864 non-null   int64  
 11  Schooling                 2864 non-null   float64
 12  Economy_status_Developed  2864 non-null   int64  
dtypes: float64(7), int64(5), object(1)
memory usage: 313.2+ KB
```

In [13]:
```
# Look at the data index
print(dt.index)
```

```
Index([2296,  769, 1996,  771,  773, 1436,  777,  778, 1994,  780,
       ...
       2716, 1849, 2607,  668,  666, 1856,  647,  645, 2713, 1353],
      dtype='int64', length=2864)
```

In [14]:
```
# Check the names of variables provided
print(dt.columns)
```

```
Index(['Region', 'Life_expectancy', 'Under_five_deaths', 'Adult_mortality',
       'Alcohol_consumption', 'Hepatitis_B', 'Measles', 'BMI', 'Diphtheria',
       'Incidents_HIV', 'GDP_per_capita', 'Schooling',
       'Economy_status_Developed'],
      dtype='object')
```

In [15]:
```
# Data dimensions
dt.shape
```

Out[15]:
```
(2864, 13)
```

In [16]:
```
# Check if there are any missing observations
print(dt.isnull().any())
```

```
Region                     False
Life_expectancy            False
Under_five_deaths          False
Adult_mortality            False
Alcohol_consumption        False
Hepatitis_B                False
Measles                    False
BMI                        False
Diphtheria                 False
Incidents_HIV              False
GDP_per_capita             False
Schooling                  False
Economy_status_Developed   False
dtype: bool
```

In [17]:
```
# Summary Statistics
dt.describe()
```

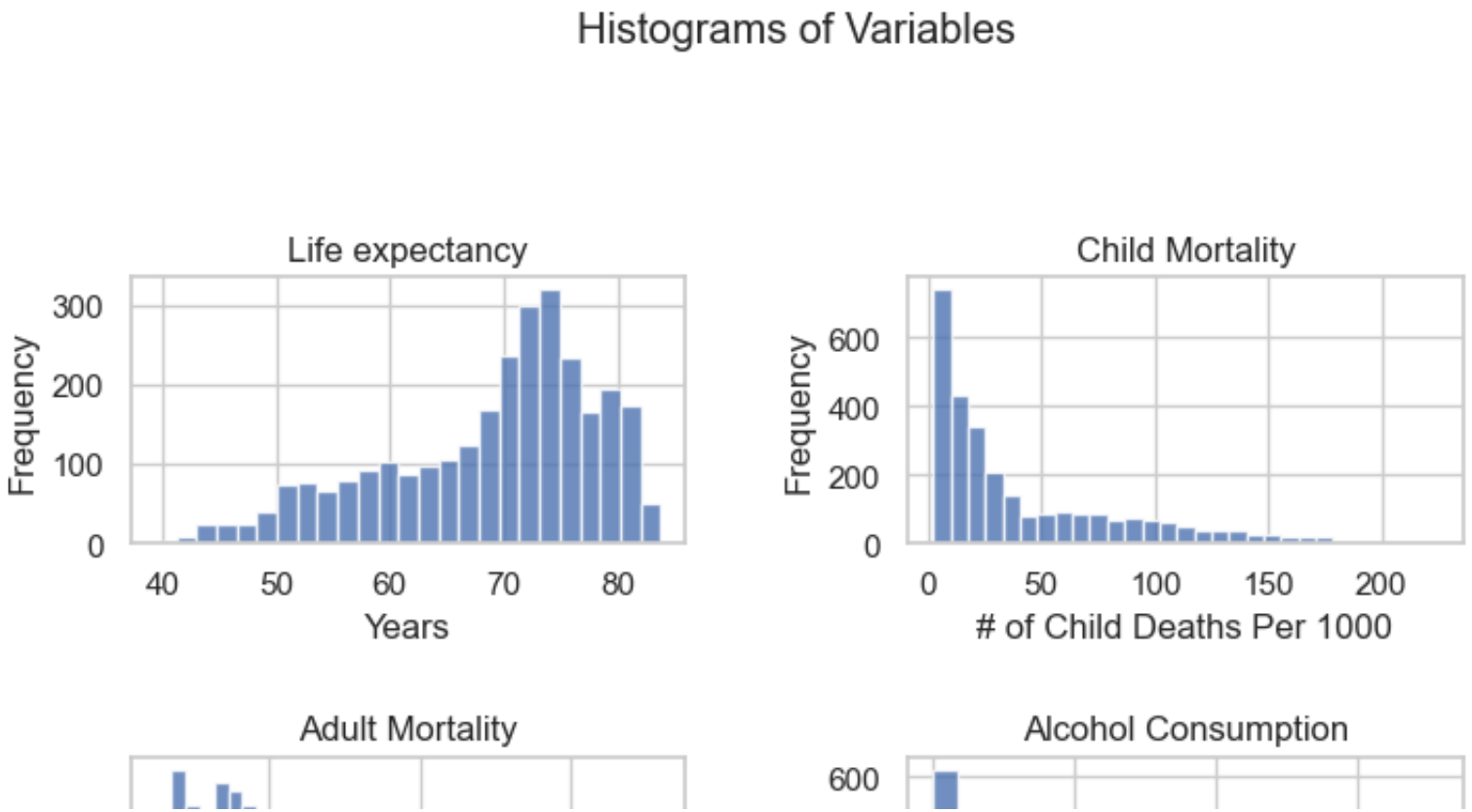
Out[17]:

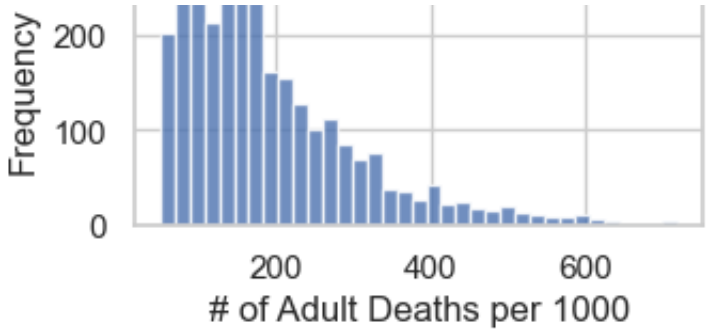
	Life_expectancy	Under_five_deaths	Adult_mortality	Alcohol_consumption	Hepatitis_B	Measles	BMI	Diphtheria	Incider
count	2864.000000	2864.000000	2864.000000	2864.000000	2864.000000	2864.000000	2864.000000	2864.000000	2864.
mean	68.856075	42.938268	192.251775	4.820882	84.292598	77.344972	25.032926	86.271648	0.
std	9.405608	44.569974	114.910281	3.981949	15.995511	18.659693	2.193905	15.534225	2.
min	39.400000	2.300000	49.384000	0.000000	12.000000	10.000000	19.800000	16.000000	0.
25%	62.700000	9.675000	106.910250	1.200000	78.000000	64.000000	23.200000	81.000000	0.
50%	71.400000	23.100000	163.841500	4.020000	89.000000	83.000000	25.500000	93.000000	0.
75%	75.400000	66.000000	246.791375	7.777500	96.000000	93.000000	26.400000	97.000000	0.
max	83.800000	224.900000	719.360500	17.870000	99.000000	99.000000	32.100000	99.000000	21.

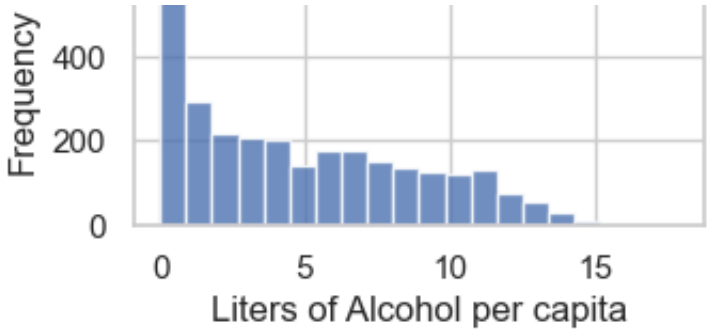
In [18]:
```
fig, ax = plt.subplots(6, 2, figsize = (8, 16))
plt.subplots_adjust(left=0.1, bottom=0.1, right=0.9, top=0.9, wspace=0.4,hspace=0.8)
fig.suptitle('Histograms of Variables')

ax[0,0].hist(dt["Life_expectancy"], alpha = 0.8, bins = "fd")
ax[0,0].set_title("Life expectancy")
ax[0,0].set_xlabel("Years")
ax[0,0].set_ylabel("Frequency")

ax[0,1].hist(dt["Under_five_deaths"], alpha = 0.8, bins = "fd")
ax[0,1].set_title("Child Mortality")
ax[0,1].set_xlabel("# of Child Deaths Per 1000")
ax[0,1].set_ylabel("Frequency")

ax[1,0].hist(dt["Adult_mortality"], alpha = 0.8, bins = "fd")
```

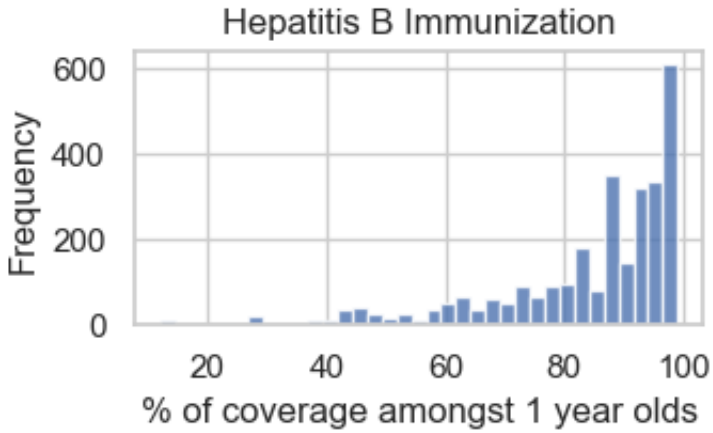
```
ax[1,0].set_title("Adult Mortality")
ax[1,0].set_xlabel("# of Adult Deaths per 1000")
ax[1,0].set_ylabel("Frequency")

ax[1,1].hist(dt["Alcohol_consumption"], alpha = 0.8, bins = "fd")
ax[1,1].set_title("Alcohol Consumption")
ax[1,1].set_xlabel("Liters of Alcohol per capita")
ax[1,1].set_ylabel("Frequency")

ax[2,0].hist(dt["Hepatitis_B"], alpha = 0.8, bins = "fd")
ax[2,0].set_title("Hepatitis B Immunization")
ax[2,0].set_xlabel("% of coverage amongst 1 year olds")
ax[2,0].set_ylabel("Frequency")

ax[2,1].hist(dt["Measles"], alpha = 0.8, bins = "fd")
ax[2,1].set_title("Measles Immunization")
ax[2,1].set_xlabel("% of coverage amongst 1 year olds")
ax[2,1].set_ylabel("Frequency")

ax[3,0].hist(dt["BMI"], alpha = 0.8, bins = "fd")
ax[3,0].set_title("BMI")
ax[3,0].set_xlabel("BMI")
ax[3,0].set_ylabel("Frequency")

ax[3,1].hist(dt["Diphtheria"], alpha = 0.8, bins = "fd")
ax[3,1].set_title("Diphtheria immunization")
ax[3,1].set_xlabel("% of coverage amongst 1 year olds")
ax[3,1].set_ylabel("Frequency")

ax[4,0].hist(dt["Incidents_HIV"], alpha = 0.8, bins = "sturges")
ax[4,0].set_title("Incidents of HIV")
ax[4,0].set_xlabel("Incidents of HIV per 1000 people aged 15-49")
ax[4,0].set_ylabel("Frequency")

ax[4,1].hist(dt["GDP_per_capita"], alpha = 0.8, bins = "sturges")
```

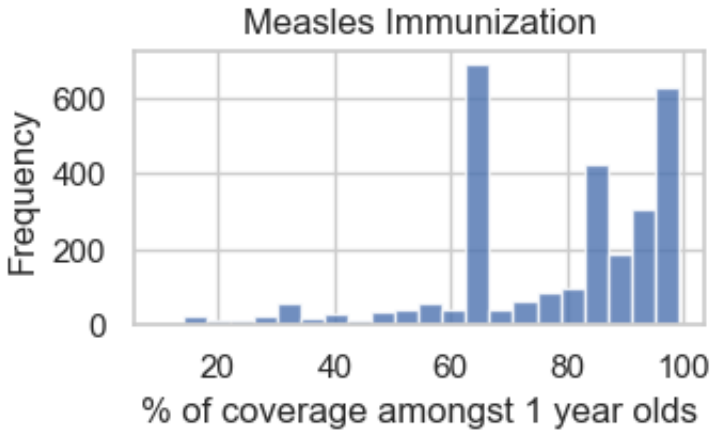
```
ax[4,1].set_title("GDP per capita")
ax[4,1].set_xlabel("GDP per capita in USD")
ax[4,1].set_ylabel("Frequency")

ax[5,0].hist(dt["Schooling"], alpha = 0.8, bins = "fd")
ax[5,0].set_title("Avg. # of years educated for people 25+")
ax[5,0].set_xlabel("Years")
ax[5,0].set_ylabel("Frequency")

fig.delaxes(ax[5,1])

plt.show()
```

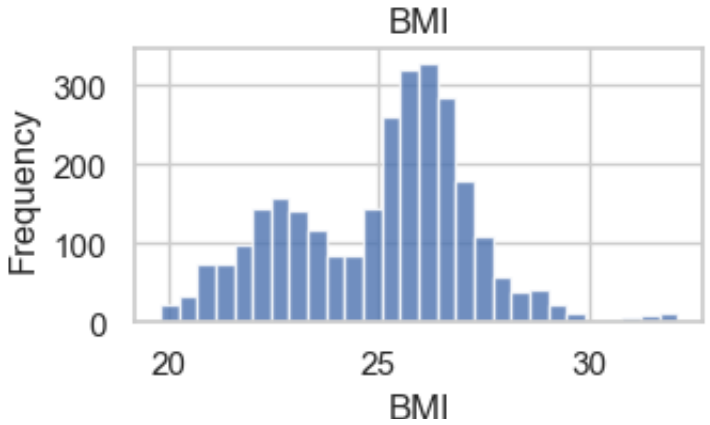


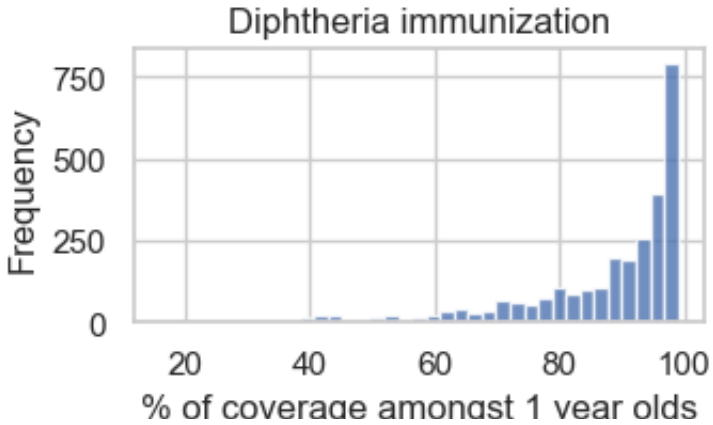

Frequency
200
100
0
200
400
600
# of Adult Deaths per 1000

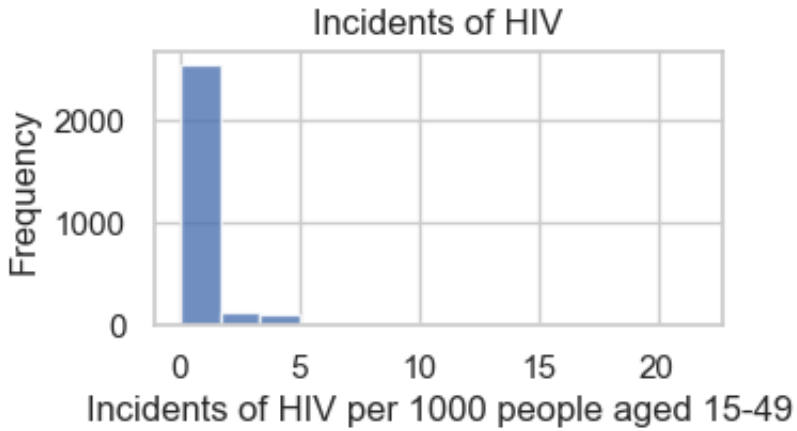

Frequency
400
200
0
0
5
10
15
Liters of Alcohol per capita

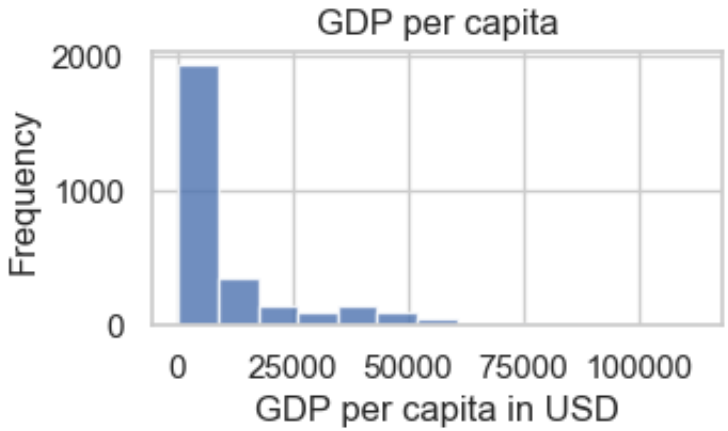

Hepatitis B Immunization
Frequency
600
400
200
0
20
40
60
80
100
% of coverage amongst 1 year olds

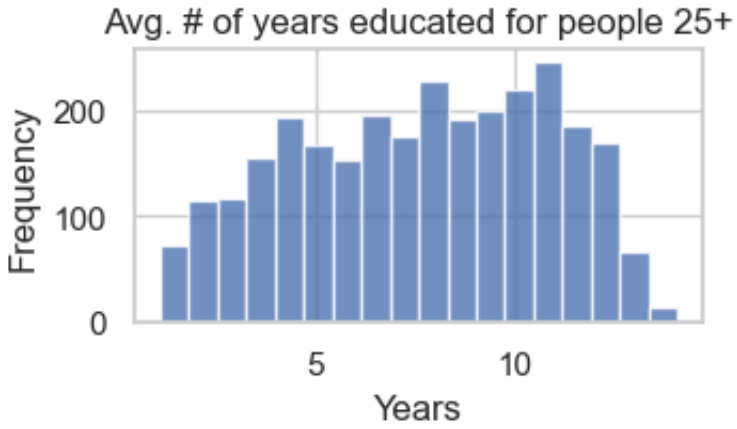

Measles Immunization
Frequency
600
400
200
0
20
40
60
80
100
% of coverage amongst 1 year olds


BMI
Frequency
300
200
100
0
20
25
30
BMI


Diphtheria immunization
Frequency
750
500
250
0
20
40
60
80
100
% of coverage amongst 1 year olds

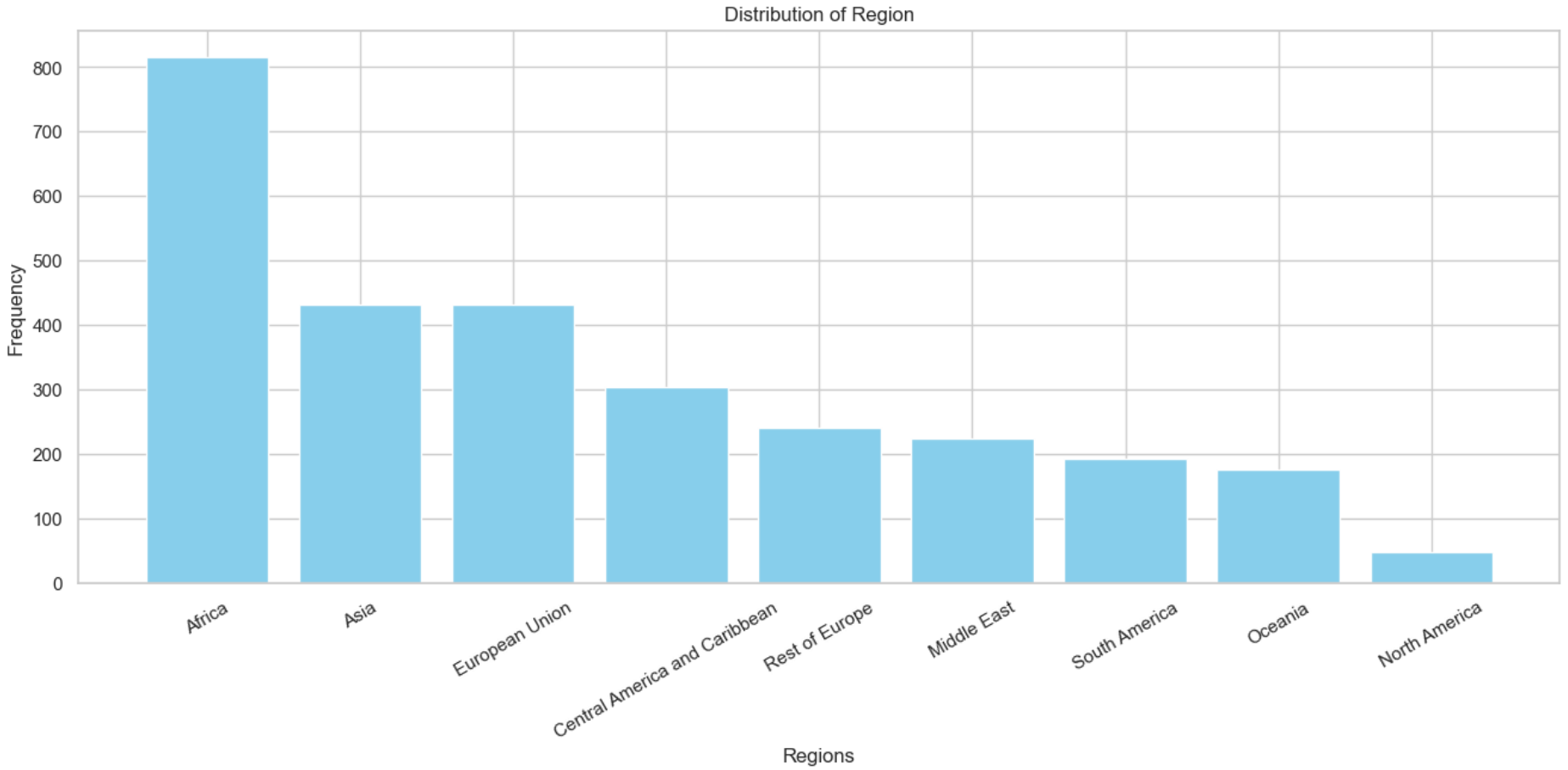






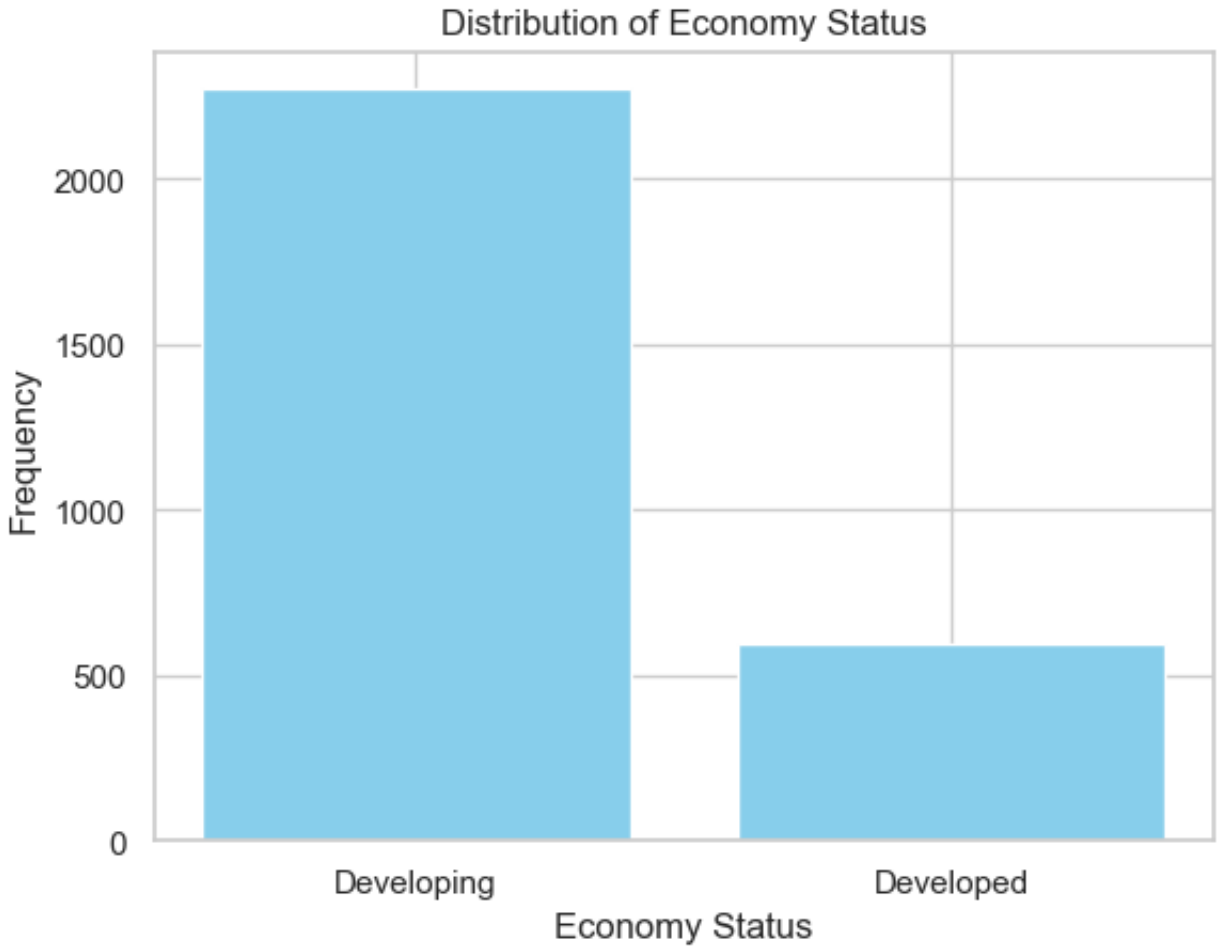
In [19]:
```
counts = dt['Region'].value_counts()
plt.figure(figsize=(16, 6))
plt.bar(counts.index.tolist(), counts, color='skyblue')

plt.title('Distribution of Region')
plt.xlabel('Regions')
```

```
plt.ylabel('Frequency')
plt.xticks(rotation = 30)
plt.show()
```



In [20]:
```
counts = dt['Economy_status_Developed'].value_counts()
plt.bar(counts.index.tolist(), counts, color='skyblue')
plt.xticks([0, 1], ['Developing', 'Developed'])

plt.title('Distribution of Economy Status')
```

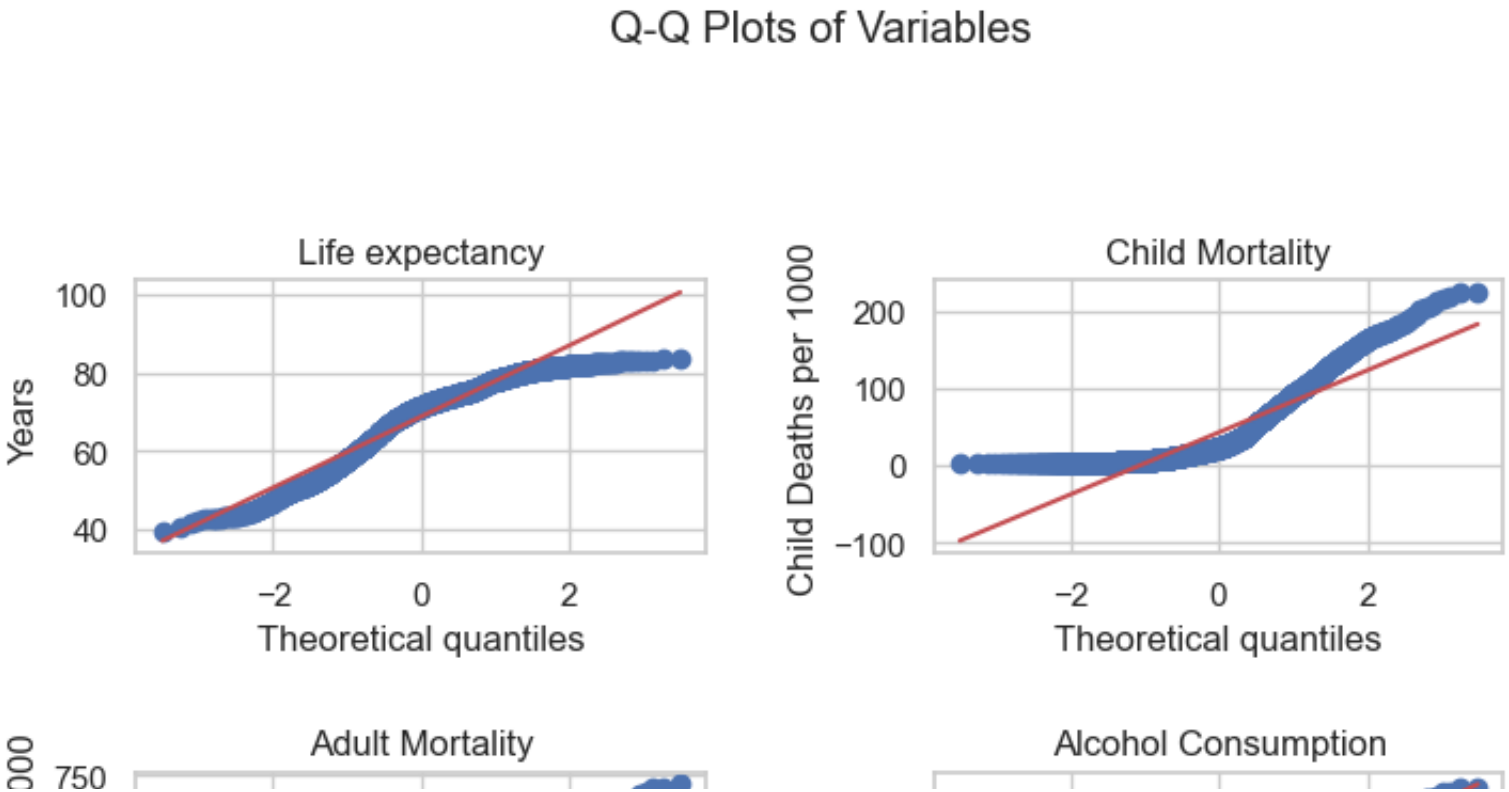
```
plt.xlabel('Economy Status')
plt.ylabel('Frequency')
plt.show()
```

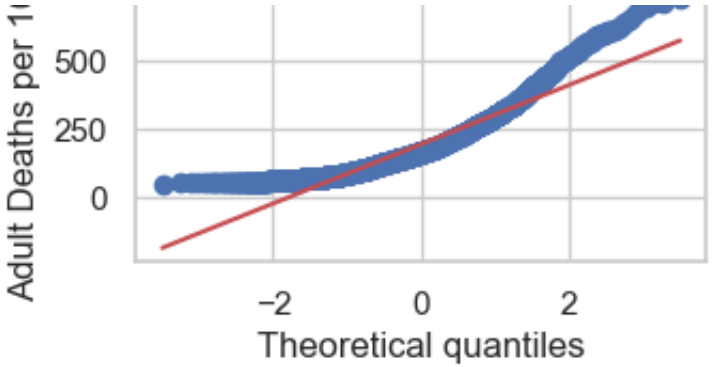


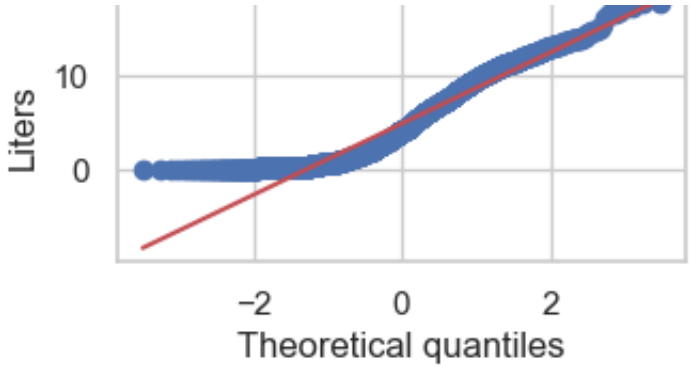
In [21]:
```
fig, ax = plt.subplots(6, 2, figsize = (8, 16))
plt.subplots_adjust(left=0.1, bottom=0.1, right=0.9, top=0.9, wspace=0.4, hspace=0.8)
fig.suptitle('Q-Q Plots of Variables')
```

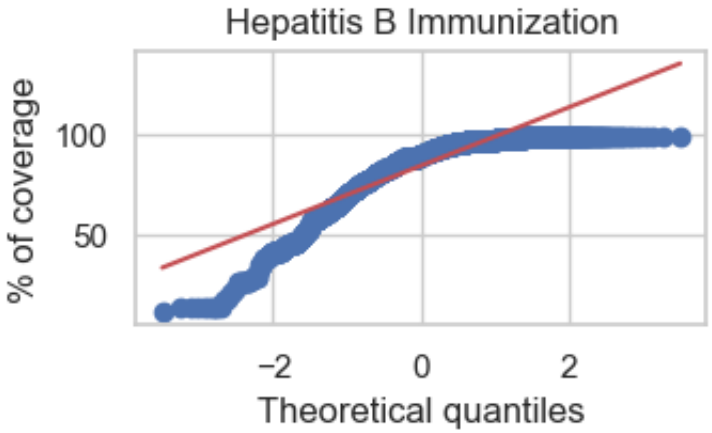
```
stats.probplot(dt["Life_expectancy"], dist = "norm", plot = ax[0,0])
ax[0,0].set_title("Life expectancy")
ax[0,0].set_ylabel("Years")

stats.probplot(dt["Under_five_deaths"], dist = "norm", plot = ax[0,1])
ax[0,1].set_title("Child Mortality")
ax[0,1].set_ylabel("Child Deaths per 1000")

stats.probplot(dt["Adult_mortality"], dist = "norm", plot = ax[1,0])
ax[1,0].set_title("Adult Mortality")
ax[1,0].set_ylabel("Adult Deaths per 1000")

stats.probplot(dt["Alcohol_consumption"], dist = "norm", plot = ax[1,1])
ax[1,1].set_title("Alcohol Consumption")
ax[1,1].set_ylabel("Liters")

stats.probplot(dt["Hepatitis_B"], dist = "norm", plot = ax[2,0])
ax[2,0].set_title("Hepatitis B Immunization")
ax[2,0].set_ylabel("% of coverage")

stats.probplot(dt["Measles"], dist = "norm", plot = ax[2,1])
ax[2,1].set_title("Measles Immunization")
ax[2,1].set_ylabel("% of coverage")

stats.probplot(dt["BMI"], dist = "norm", plot = ax[3,0])
ax[3,0].set_title("BMI")
ax[3,0].set_ylabel("BMI")

stats.probplot(dt["Diphtheria"], dist = "norm", plot = ax[3,1])
ax[3,1].set_title("Diphtheria immunization")
ax[3,1].set_ylabel("% of coverage")

stats.probplot(dt["Incidents_HIV"], dist = "norm", plot = ax[4,0])
ax[4,0].set_title("Incidents of HIV")
```

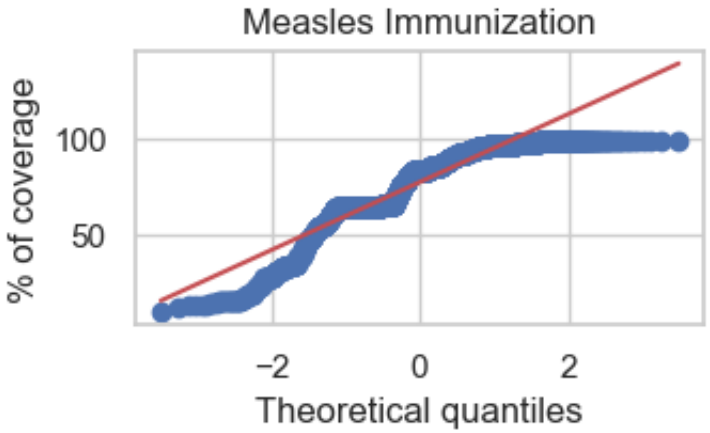
```
ax[4,0].set_ylabel("Incidents of HIV/1000")

stats.probplot(dt["GDP_per_capita"], dist = "norm", plot = ax[4,1])
ax[4,1].set_title("GDP per capita")
ax[4,1].set_ylabel("GDP/capita in USD")

stats.probplot(dt["Schooling"], dist = "norm", plot = ax[5,0])
ax[5,0].set_title("Avg. # of years educated for people 25+")
ax[5,0].set_ylabel("Years")

fig.delaxes(ax[5,1])

plt.show()
```

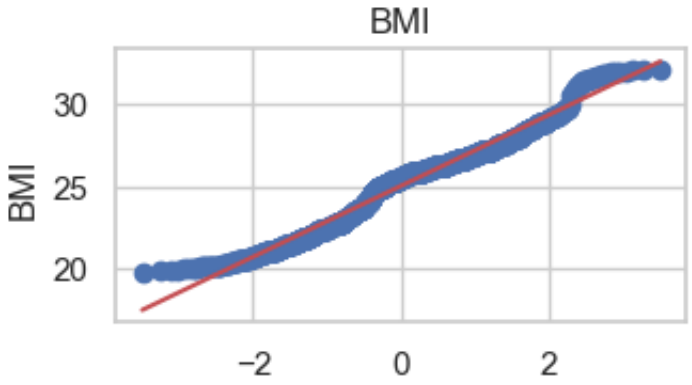


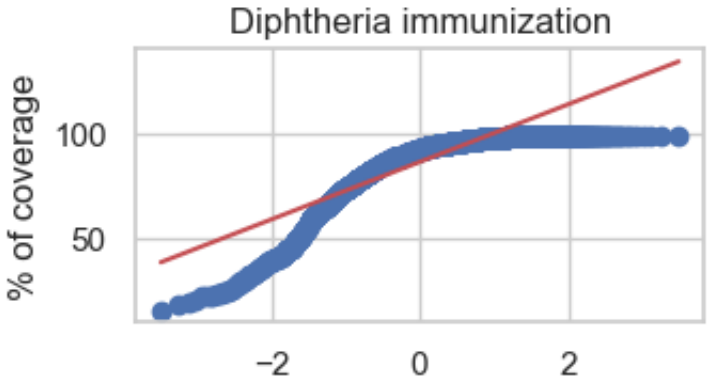

Adult Deaths per 1
500
250
0
−2
0
2
Theoretical quantiles

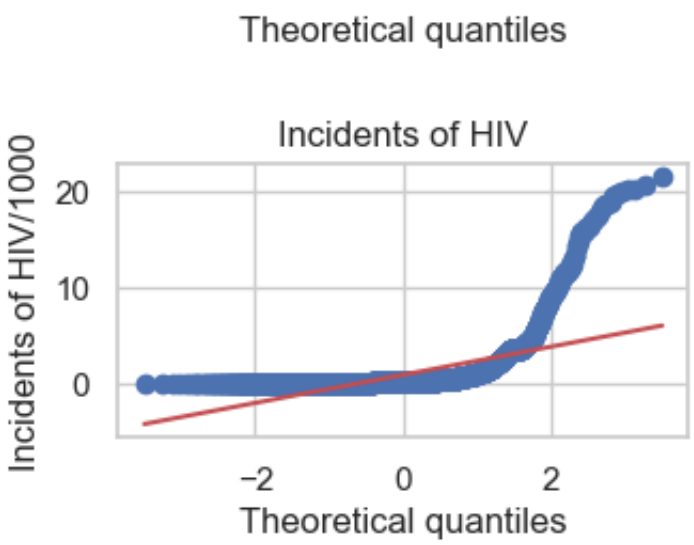

Liters
10
0
−2
0
2
Theoretical quantiles

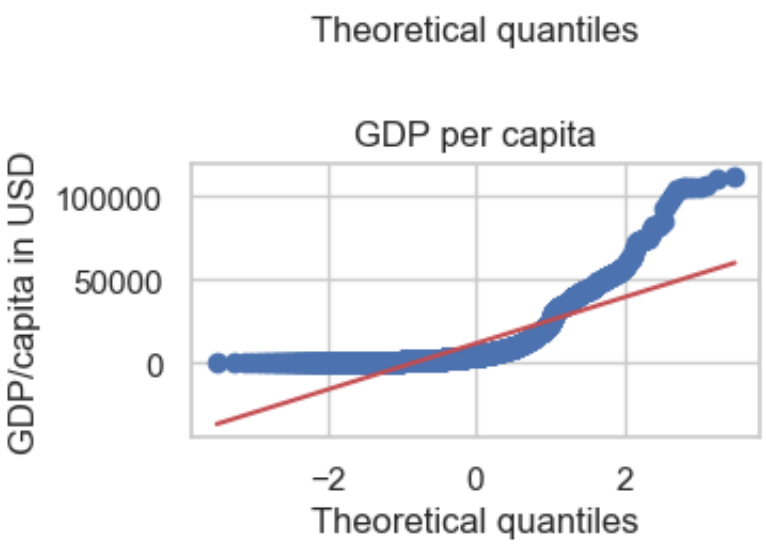

Hepatitis B Immunization
% of coverage
100
50
−2
0
2
Theoretical quantiles

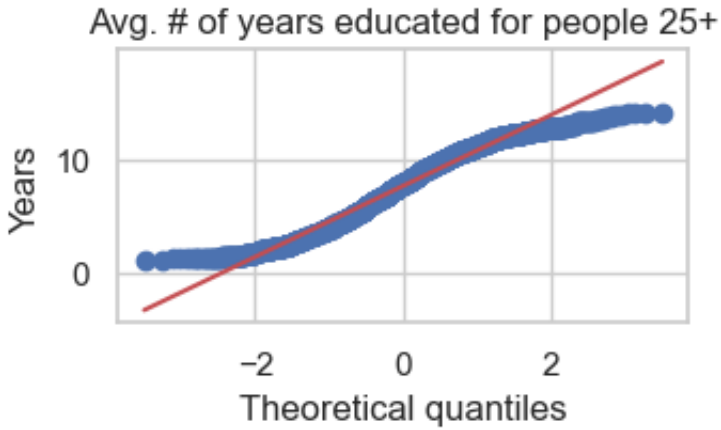

Measles Immunization
% of coverage
100
50
−2
0
2
Theoretical quantiles


BMI
BMI
30
25
20
−2
0
2


Diphtheria immunization
% of coverage
100
50
−2
0
2


Incidents of HIV
Incidents of HIV/1000
20
10
0
−2
0
2
Theoretical quantiles
Theoretical quantiles


GDP per capita
GDP/capita in USD
100000
50000
0
−2
0
2
Theoretical quantiles
Theoretical quantiles


Avg. # of years educated for people 25+
Years
10
0
−2
0
2
Theoretical quantiles

Q2(a) We can observe from the histograms above that variables such as 'Life_expectancy', 'Hepatitis_B', 'Measles', and 'Diphtheria' have left-skewed distributions, characterized by long tails in the negative direction. On the other hand, variables such as 'Child_mortality', 'Adult_mortality', 'Alcohol_consumption', and 'Incidents_HIV' have right-skewed distributions, characterized by long tails in the positive direction. Additionally, there is a significantly higher proportion of observations in 'Incidents_HIV' and 'GDP_per_capita' that fall to the left side of the axis, indicating lower values. 'BMI' and 'Schooling' exhibit approximately normal distributions.

As can be seen in the above Q-Q plots for the response variable and predictors, variables such as 'BMI' and 'Schooling' are normally distributed. This may be significant in helping us determine their impact on life expectancy. The remaining predictors are not normally distributed as a number of observations deviate from the theoretical distribution. This is notable because our response variable, 'Life_expectancy', is not normally distributed, since its right tail deviates from the theoretical normal distribution.

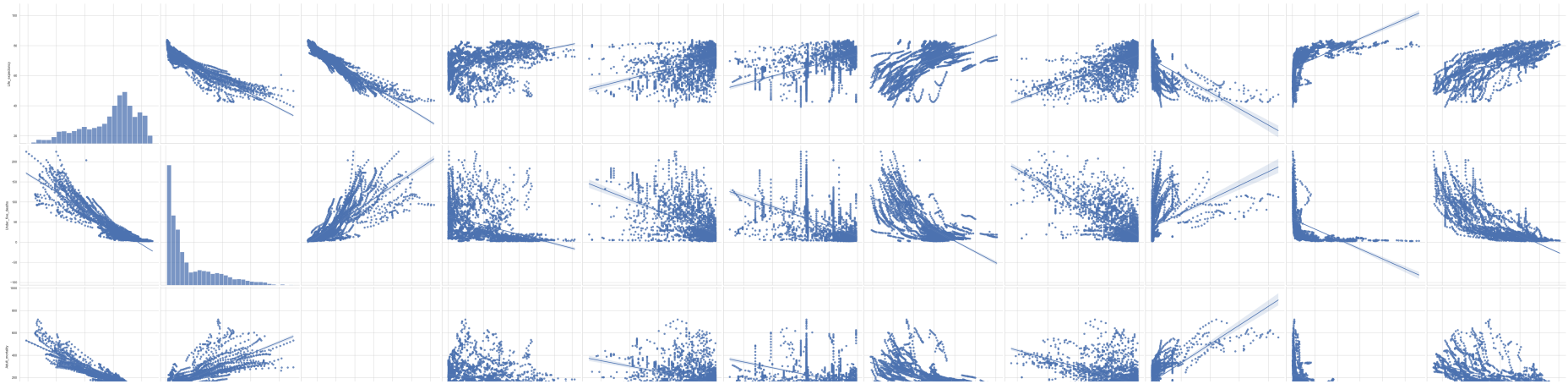
In [22]:
```
# We make a copy of the dataset with only quantitative variables...
dt_quant = dt.copy()
dt_quant = dt_quant.drop(['Region', 'Economy_status_Developed'], axis=1)
dt_quant.head()
```

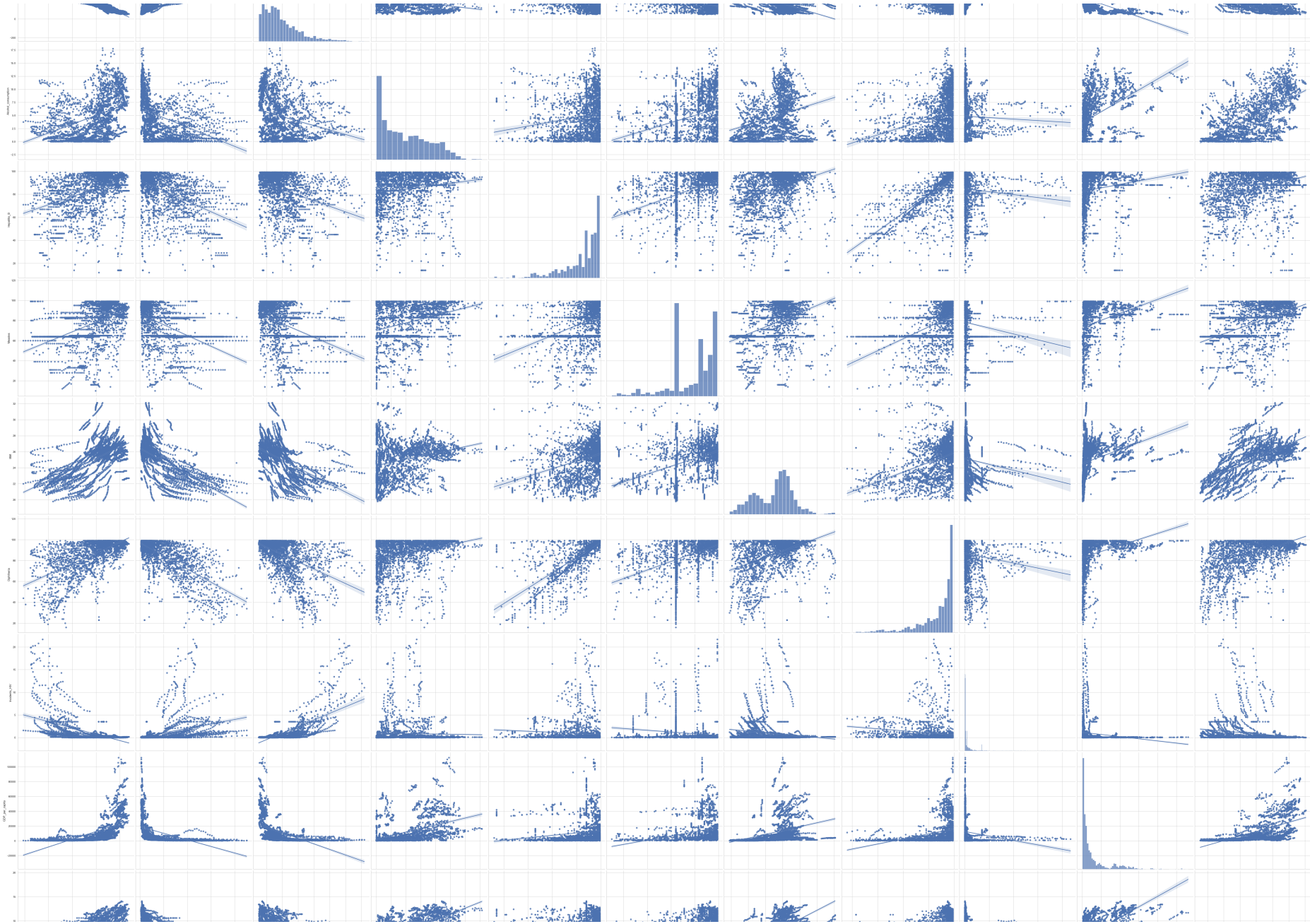
Out[22]:

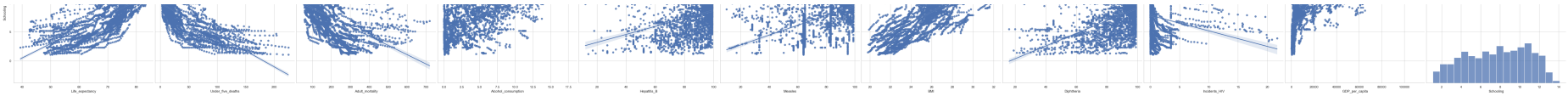
	Life_expectancy	Under_five_deaths	Adult_mortality	Alcohol_consumption	Hepatitis_B	Measles	BMI	Diphtheria	Incidents_HIV	GDP_p
2296	57.8	88.5	293.6475	1.91	80	55	22.9	80	1.48	
769	52.9	140.2	397.8705	3.06	86	60	22.7	86	0.85	
1996	51.1	94.2	484.2665	1.64	77	28	22.1	80	4.36	
771	58.9	108.3	267.2085	5.49	88	34	22.0	88	0.23	
773	72.6	15.2	157.4485	3.23	97	80	24.9	97	1.05	

In [23]:
```
plt.figure(figsize=(10,10))
sns.pairplot(dt_quant, height = 7.0, kind='reg')
```

Out[23]:
```
<seaborn.axisgrid.PairGrid at 0x16993e2c0>

<Figure size 1000x1000 with 0 Axes>
```







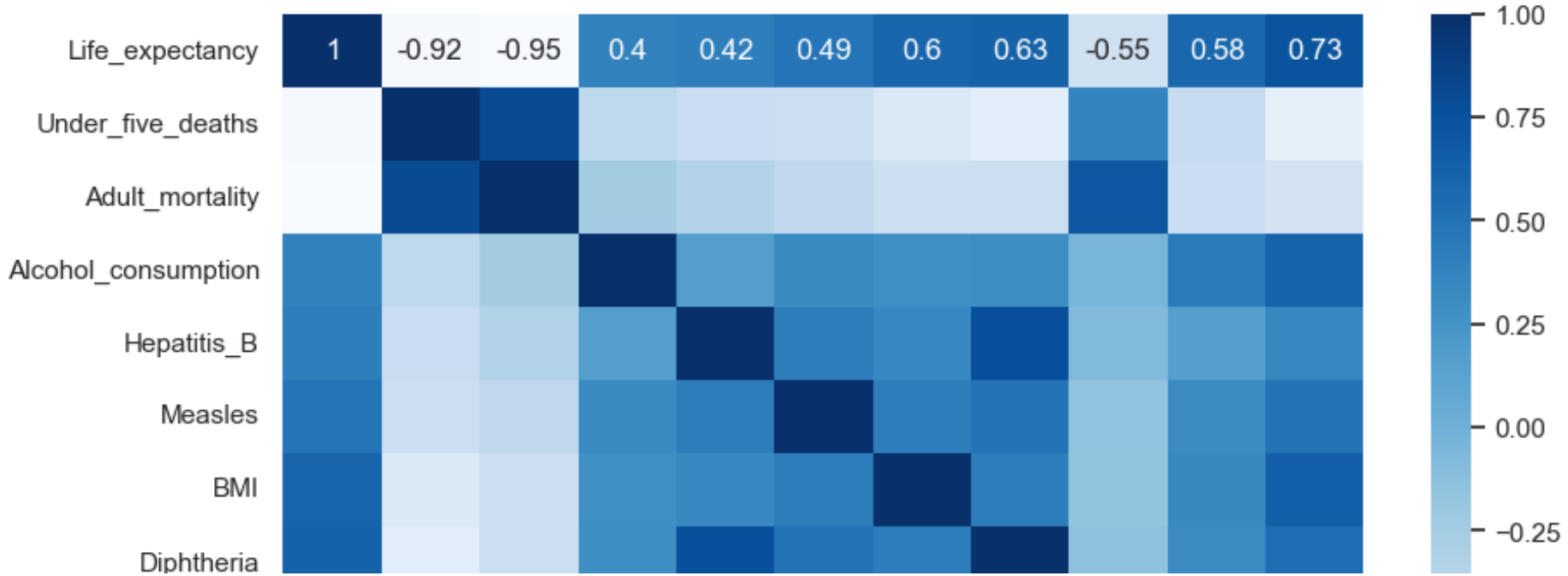
In [24]:
```
#Correlation Matrix
c = dt_quant.corr()
print(c)
```

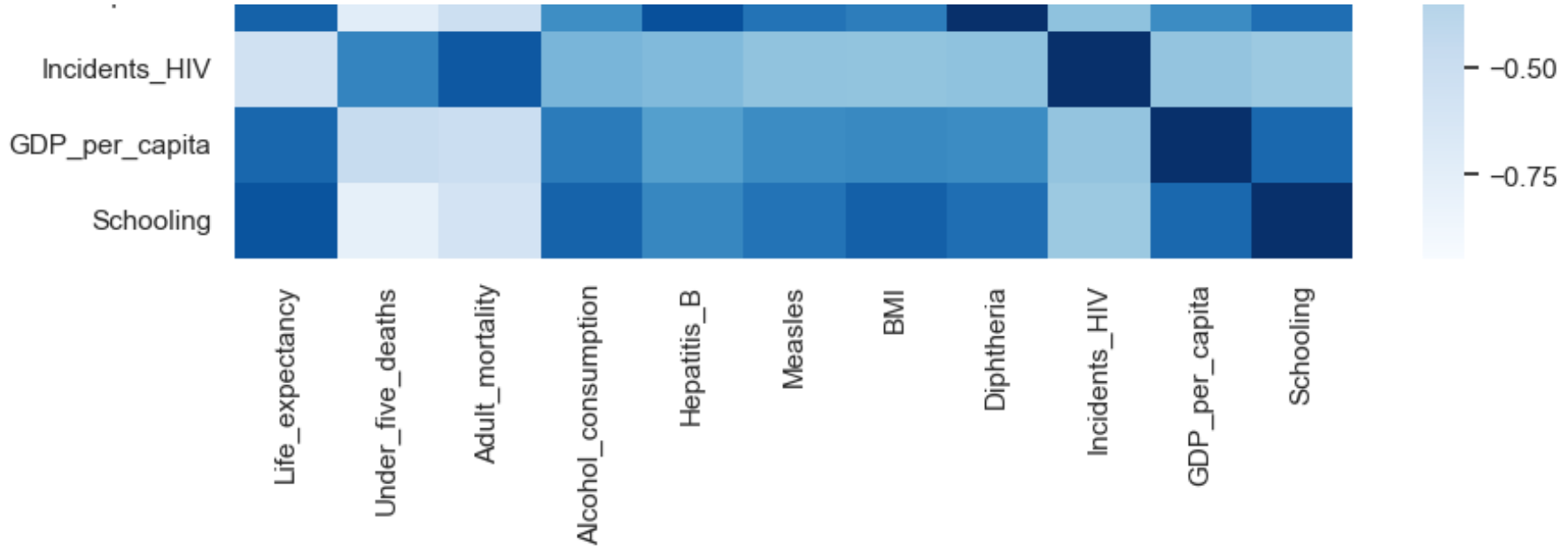
```
                    Life_expectancy  Under_five_deaths  Adult_mortality  \
Life_expectancy            1.000000          -0.920419        -0.945360
Under_five_deaths         -0.920419           1.000000         0.802361
Adult_mortality           -0.945360           0.802361         1.000000
Alcohol_consumption        0.399159          -0.409367        -0.244794
Hepatitis_B                0.417804          -0.507427        -0.344882
Measles                    0.490019          -0.512972        -0.416153
BMI                        0.598423          -0.665255        -0.522866
Diphtheria                 0.627541          -0.725355        -0.513803
Incidents_HIV             -0.553027           0.369618         0.699119
GDP_per_capita             0.583090          -0.469682        -0.510121
Schooling                  0.732484          -0.773196        -0.581035

                    Alcohol_consumption  Hepatitis_B   Measles       BMI  \
Life_expectancy                0.399159     0.417804  0.490019  0.598423
Under_five_deaths             -0.409367    -0.507427 -0.512972 -0.665255
Adult_mortality               -0.244794    -0.344882 -0.416153 -0.522866
Alcohol_consumption            1.000000     0.168436  0.318603  0.284032
Hepatitis_B                    0.168436     1.000000  0.429168  0.345421
Measles                        0.318603     0.429168  1.000000  0.416321
BMI                            0.284032     0.345421  0.416321  1.000000
Diphtheria                     0.299016     0.761780  0.494059  0.426501
Incidents_HIV                 -0.034118    -0.075782 -0.150580 -0.161142
GDP_per_capita                 0.443966     0.159375  0.313724  0.336180
Schooling                      0.615728     0.347643  0.498391  0.635475

                    Diphtheria  Incidents_HIV  GDP_per_capita  Schooling
Life_expectancy       0.627541      -0.553027        0.583090   0.732484
```

Under_five_deaths	-0.725355	0.369618	-0.469682	-0.773196
Adult_mortality	-0.513803	0.699119	-0.510121	-0.581035
Alcohol_consumption	0.299016	-0.034118	0.443966	0.615728
Hepatitis_B	0.761780	-0.075782	0.159375	0.347643
Measles	0.494059	-0.150580	0.313724	0.498391
BMI	0.426501	-0.161142	0.336180	0.635475
Diphtheria	1.000000	-0.146932	0.313321	0.535621
Incidents_HIV	-0.146932	1.000000	-0.169590	-0.201246
GDP_per_capita	0.313321	-0.169590	1.000000	0.580626
Schooling	0.535621	-0.201246	0.580626	1.000000

In [25]:
```
plt.figure(figsize=(10,6))
sns.heatmap(c,annot=True, cmap="Blues")
```

Out[25]: <Axes: >





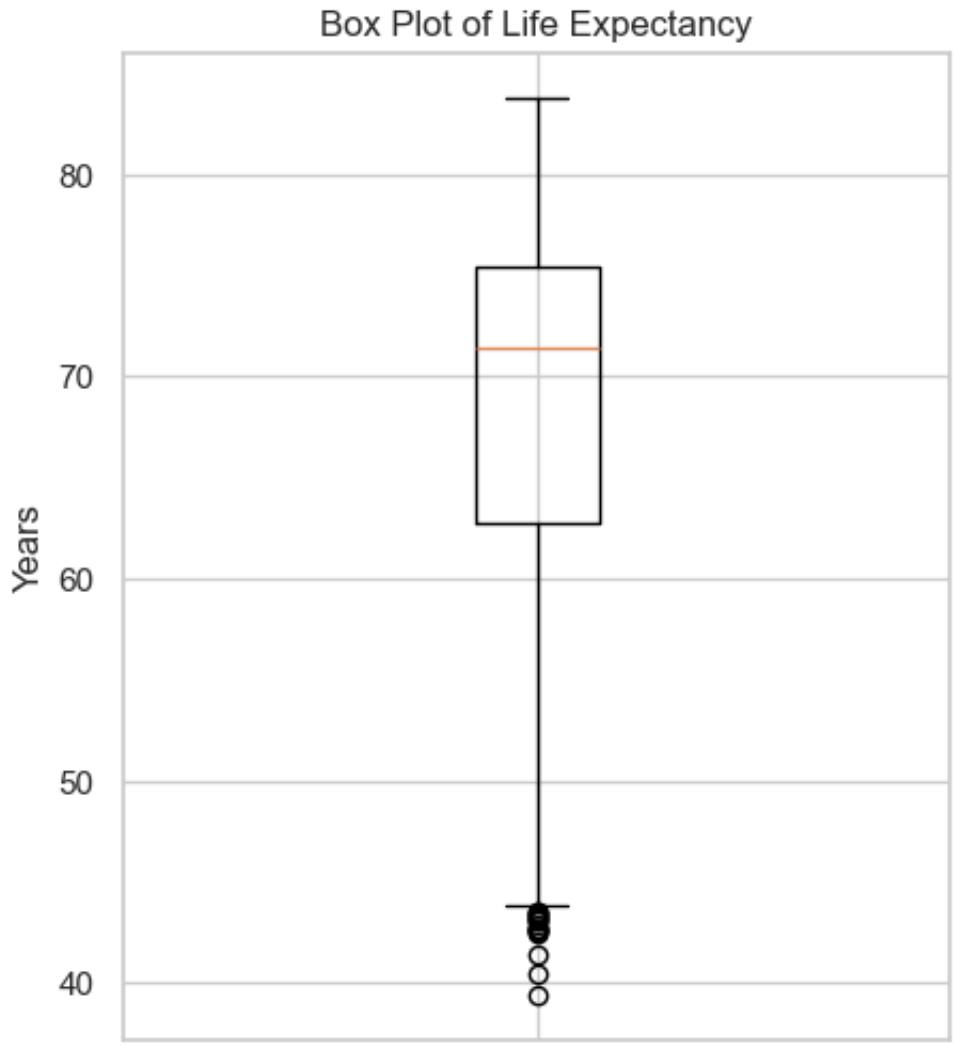
We can see from the above correlation pairplot and matrix that the response variable 'Life_expectancy' is highly correlated with 'Under_five_deaths' and 'Adult_mortality' with correlation coefficients of -0.92 and -0.95, respectively.

As for the correlation between predictors, the predictor 'Under_five_deaths' has a high correlation with 'Adult_mortality' and 'Schooling' with coefficients of 0.8 and -0.77, respectively. 'Hepatitis_B' and 'Diphtheria' are correlated with a correlation coefficient of 0.76. There is moderate to little correlation observed among the remaining predictors.

In [26]:
```
#Response Variable

plt.figure(figsize = (5, 6))
```

```
plt.boxplot(dt[["Life_expectancy"]])
plt.xticks([1], ['Life Expectancy'])
plt.ylabel("Years")
plt.title("Box Plot of Life Expectancy")
```

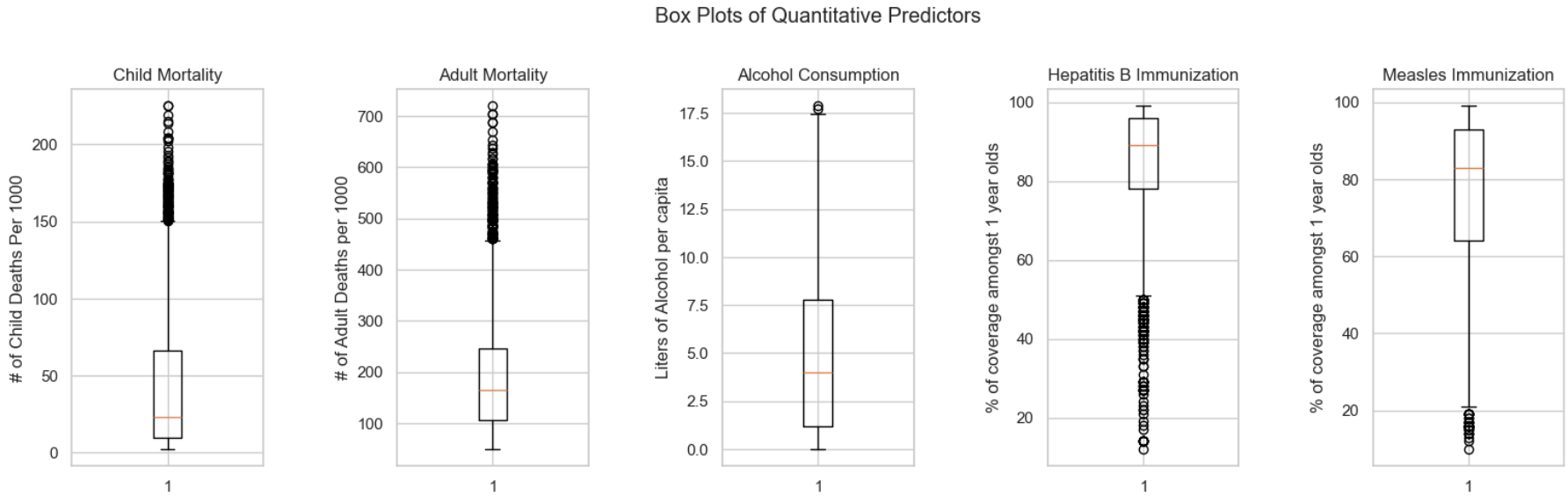
Out[26]: Text(0.5, 1.0, 'Box Plot of Life Expectancy')

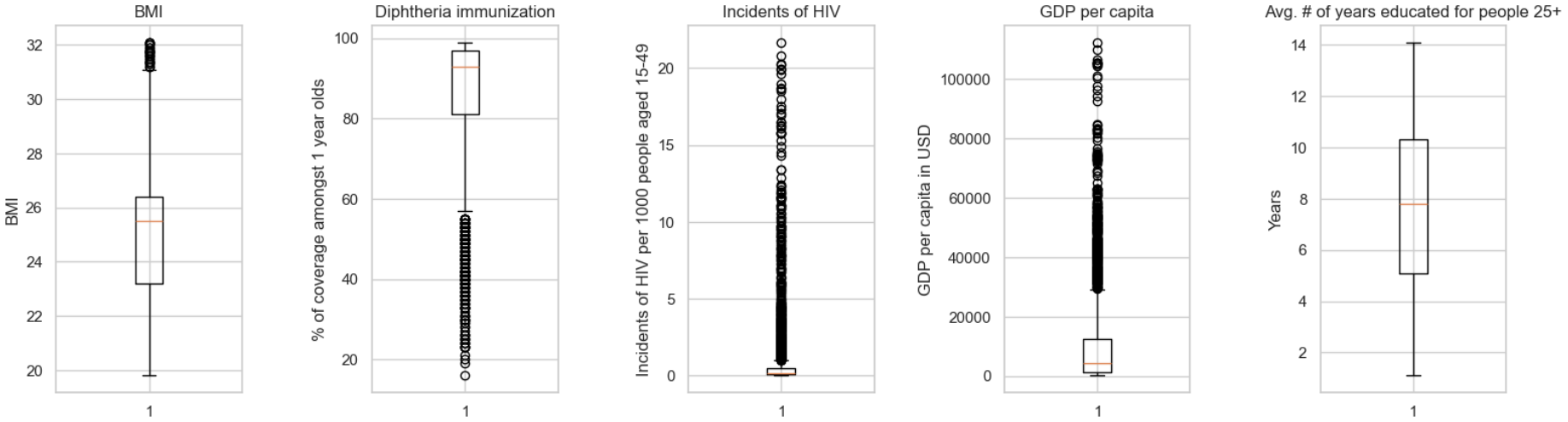


The above Box plot of Life Expectancy shows that the series is slightly left skewed with a high median value and a large spread of the data.

In [27]:
```
#Quantitative Predictors

fig, ax = plt.subplots(2, 5, figsize = (18, 10))
plt.subplots_adjust(left=0.1, bottom=0.1, right=0.9, top=0.9, wspace=0.7, hspace = 0.2)

ax[0,0].boxplot(dt["Under_five_deaths"])
ax[0,0].set_title("Child Mortality")
ax[0,0].set_ylabel("# of Child Deaths Per 1000")

ax[0,1].boxplot(dt["Adult_mortality"])
ax[0,1].set_title("Adult Mortality")
ax[0,1].set_ylabel("# of Adult Deaths per 1000")

ax[0,2].boxplot(dt["Alcohol_consumption"])
ax[0,2].set_title("Alcohol Consumption")
ax[0,2].set_ylabel("Liters of Alcohol per capita")

ax[0,3].boxplot(dt["Hepatitis_B"])
ax[0,3].set_title("Hepatitis B Immunization")
ax[0,3].set_ylabel("% of coverage amongst 1 year olds")

ax[0,4].boxplot(dt["Measles"])
ax[0,4].set_title("Measles Immunization")
ax[0,4].set_ylabel("% of coverage amongst 1 year olds")

ax[1,0].boxplot(dt["BMI"])
ax[1,0].set_title("BMI")
ax[1,0].set_ylabel("BMI")
```

```
ax[1,1].boxplot(dt["Diphtheria"])
ax[1,1].set_title("Diphtheria immunization")
ax[1,1].set_ylabel("% of coverage amongst 1 year olds")

ax[1,2].boxplot(dt["Incidents_HIV"])
ax[1,2].set_title("Incidents of HIV")
ax[1,2].set_ylabel("Incidents of HIV per 1000 people aged 15-49")

ax[1,3].boxplot(dt["GDP_per_capita"])
ax[1,3].set_title("GDP per capita")
ax[1,3].set_ylabel("GDP per capita in USD")

ax[1,4].boxplot(dt["Schooling"])
ax[1,4].set_title("Avg. # of years educated for people 25+")
ax[1,4].set_ylabel("Years")

fig.suptitle("Box Plots of Quantitative Predictors")
plt.show()
```

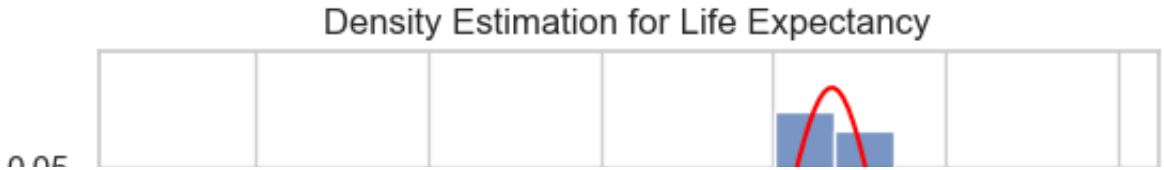


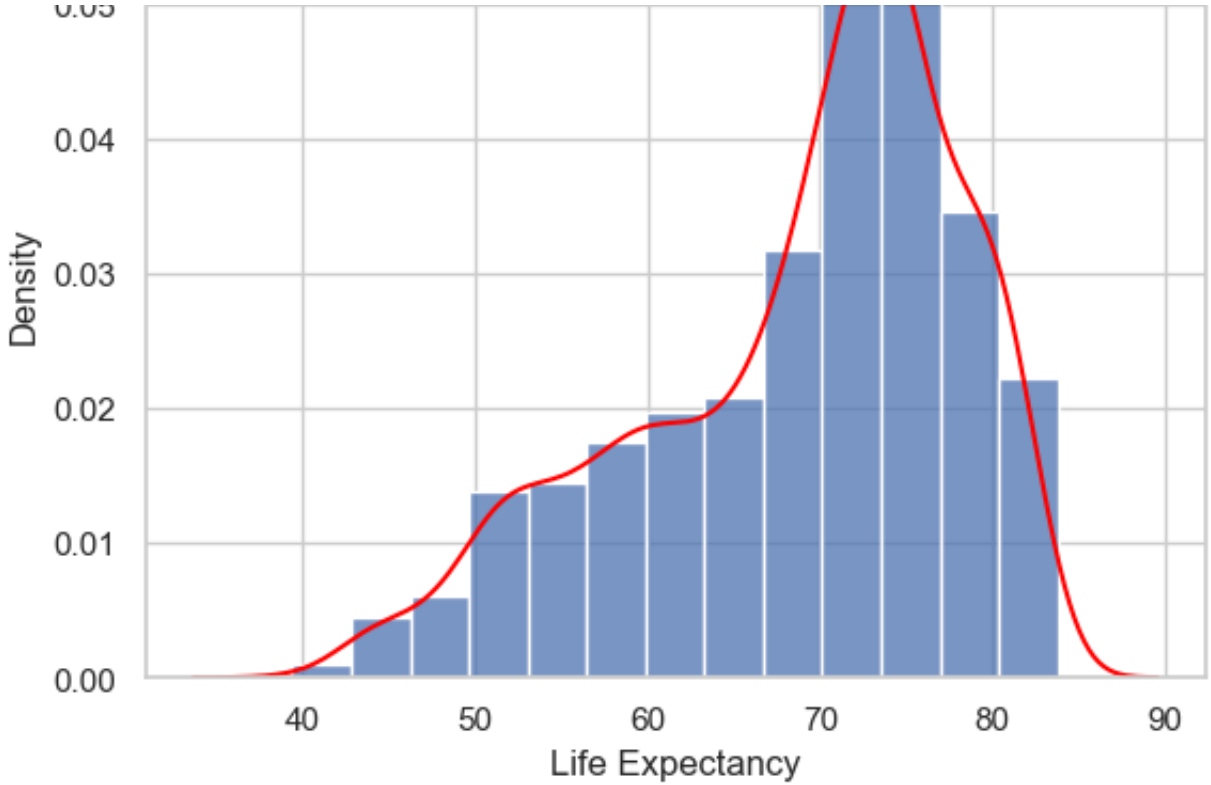


Some of the striking observations from the above analysis include, right skewness in Child Mortality and Adult Mortality data and left skewness in Diptheria and HIV Incidents data. We also observe a large spread in the data for Alcohol Consumption.

Q2(b) Density Estimation

In [28]:
```
#density estimation for life expectancy
plt.title("Density Estimation for Life Expectancy")
sns.histplot(dt.Life_expectancy, stat = "density", bins = 'sturges')
sns.kdeplot(dt.Life_expectancy, color = "red")
plt.ylabel("Density")
plt.xlabel("Life Expectancy")
plt.show()
```





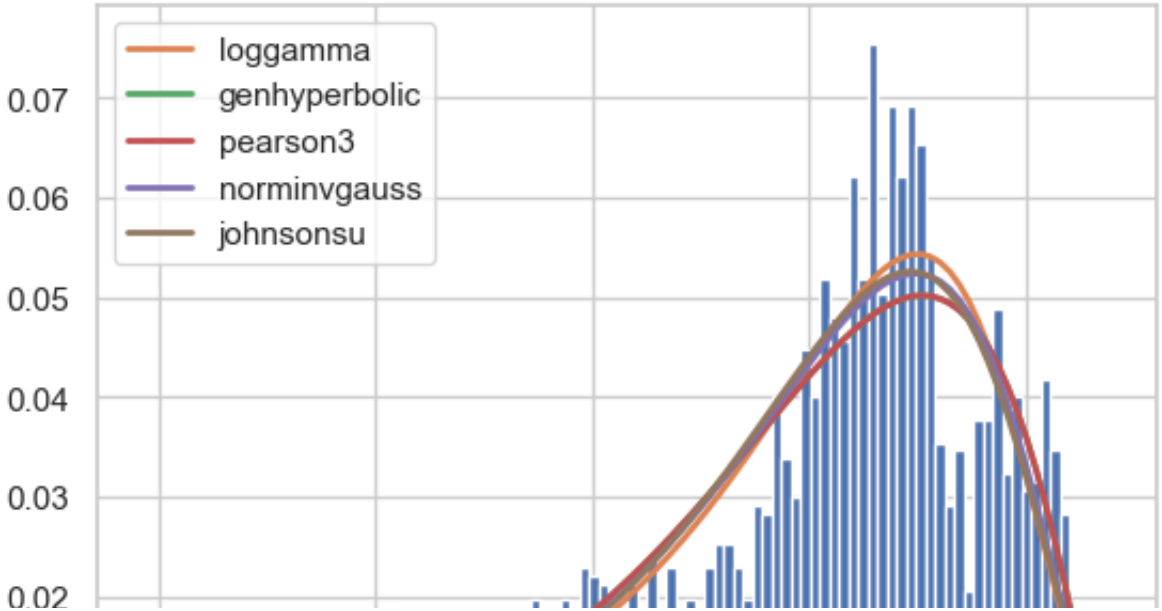
In [29]:
```
f_le = Fitter(dt['Life_expectancy'])
f_le.fit()
f_le.summary()
```

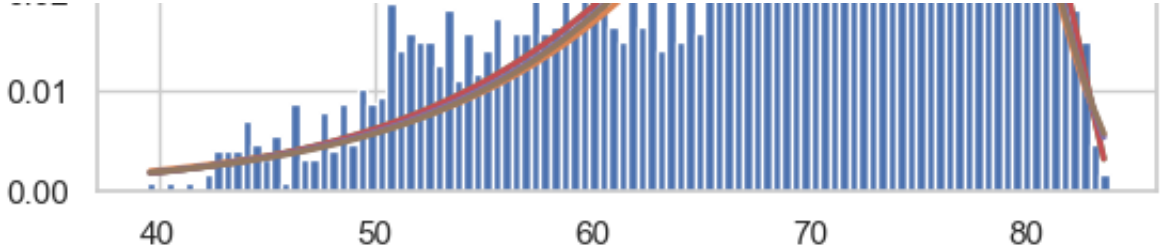
```
SKIPPED _fit distribution (taking more than 30 seconds)
SKIPPED kstwo distribution (taking more than 30 seconds)
SKIPPED rv_continuous distribution (taking more than 30 seconds)
SKIPPED rv_histogram distribution (taking more than 30 seconds)
SKIPPED nct distribution (taking more than 30 seconds)
SKIPPED recipinvgauss distribution (taking more than 30 seconds)
SKIPPED rice distribution (taking more than 30 seconds)
SKIPPED studentized_range distribution (taking more than 30 seconds)
```

```
SKIPPED t distribution (taking more than 30 seconds)
SKIPPED truncnorm distribution (taking more than 30 seconds)
SKIPPED truncpareto distribution (taking more than 30 seconds)
SKIPPED truncweibull_min distribution (taking more than 30 seconds)
SKIPPED tukeylambda distribution (taking more than 30 seconds)
SKIPPED vonmises distribution (taking more than 30 seconds)
SKIPPED weibull_min distribution (taking more than 30 seconds)
```

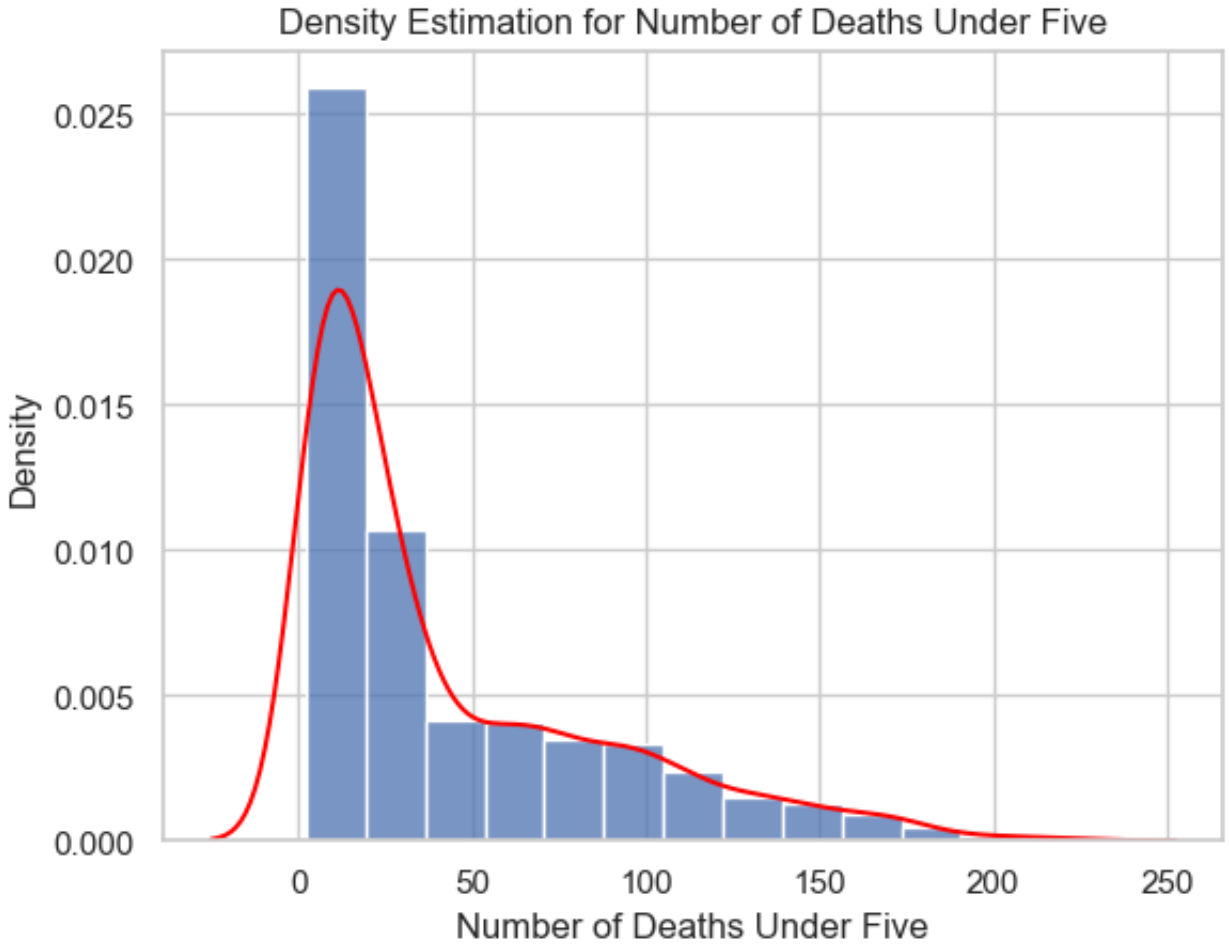
Out[29]:

	sumsquare_error	aic	bic	kl_div	ks_statistic	ks_pvalue
loggamma	0.006534	856.472320	874.352243	inf	0.043056	0.000047
genhyperbolic	0.006556	856.154645	885.954518	inf	NaN	NaN
pearson3	0.006556	852.163431	870.043354	inf	0.036947	0.000783
norminvgauss	0.006565	857.684273	881.524171	inf	0.042315	0.000068
johnsonsu	0.006579	859.214047	883.053945	inf	0.043266	0.000043




0.01
0.00
40
50
60
70
80

In [30]:
```
plt.title("Density Estimation for Number of Deaths Under Five")
sns.histplot(dt.Under_five_deaths, stat = "density", bins = 'sturges')
sns.kdeplot(dt.Under_five_deaths, color = "red")
plt.ylabel("Density")
plt.xlabel("Number of Deaths Under Five")
plt.show()
```



In [31]:
```
f_deaths = Fitter(dt['Under_five_deaths'])
f_deaths.fit()
f_deaths.summary()
```

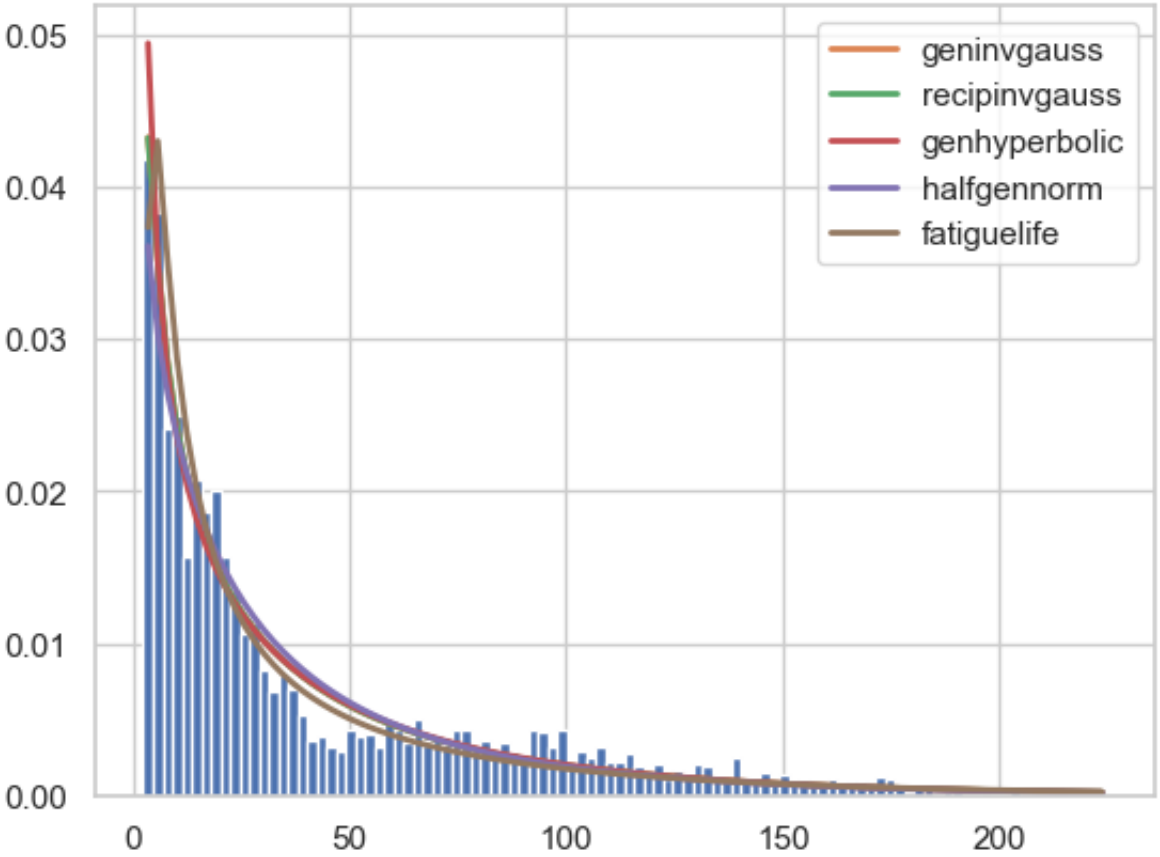
```
SKIPPED _fit distribution (taking more than 30 seconds)
SKIPPED alpha distribution (taking more than 30 seconds)
SKIPPED betaprime distribution (taking more than 30 seconds)
SKIPPED beta distribution (taking more than 30 seconds)
SKIPPED burr distribution (taking more than 30 seconds)
SKIPPED burr12 distribution (taking more than 30 seconds)
SKIPPED chi distribution (taking more than 30 seconds)
SKIPPED chi2 distribution (taking more than 30 seconds)
SKIPPED crystalball distribution (taking more than 30 seconds)
SKIPPED dweibull distribution (taking more than 30 seconds)
SKIPPED erlang distribution (taking more than 30 seconds)
SKIPPED exponpow distribution (taking more than 30 seconds)
SKIPPED exponweib distribution (taking more than 30 seconds)
SKIPPED f distribution (taking more than 30 seconds)
SKIPPED fisk distribution (taking more than 30 seconds)
SKIPPED foldcauchy distribution (taking more than 30 seconds)
SKIPPED foldnorm distribution (taking more than 30 seconds)
SKIPPED gamma distribution (taking more than 30 seconds)
SKIPPED gausshyper distribution (taking more than 30 seconds)
SKIPPED genexpon distribution (taking more than 30 seconds)
SKIPPED genextreme distribution (taking more than 30 seconds)
SKIPPED gengamma distribution (taking more than 30 seconds)
SKIPPED genhalflogistic distribution (taking more than 30 seconds)
SKIPPED genlogistic distribution (taking more than 30 seconds)
SKIPPED genpareto distribution (taking more than 30 seconds)
SKIPPED kstwo distribution (taking more than 30 seconds)
SKIPPED johnsonsb distribution (taking more than 30 seconds)
SKIPPED johnsonsu distribution (taking more than 30 seconds)
SKIPPED kappa4 distribution (taking more than 30 seconds)
SKIPPED ksone distribution (taking more than 30 seconds)
SKIPPED levy_stable distribution (taking more than 30 seconds)
```

```
SKIPPED loggamma distribution (taking more than 30 seconds)
SKIPPED lomax distribution (taking more than 30 seconds)
SKIPPED mielke distribution (taking more than 30 seconds)
SKIPPED nakagami distribution (taking more than 30 seconds)
SKIPPED ncf distribution (taking more than 30 seconds)
SKIPPED nct distribution (taking more than 30 seconds)
SKIPPED ncx2 distribution (taking more than 30 seconds)
SKIPPED norminvgauss distribution (taking more than 30 seconds)
SKIPPED rv_continuous distribution (taking more than 30 seconds)
SKIPPED rv_histogram distribution (taking more than 30 seconds)
SKIPPED pearson3 distribution (taking more than 30 seconds)
SKIPPED powerlognorm distribution (taking more than 30 seconds)
SKIPPED powernorm distribution (taking more than 30 seconds)
SKIPPED rdist distribution (taking more than 30 seconds)
SKIPPED skewnorm distribution (taking more than 30 seconds)
SKIPPED studentized_range distribution (taking more than 30 seconds)
SKIPPED t distribution (taking more than 30 seconds)
SKIPPED triang distribution (taking more than 30 seconds)
SKIPPED truncexpon distribution (taking more than 30 seconds)
SKIPPED truncnorm distribution (taking more than 30 seconds)
SKIPPED truncweibull_min distribution (taking more than 30 seconds)
SKIPPED tukeylambda distribution (taking more than 30 seconds)
SKIPPED vonmises distribution (taking more than 30 seconds)
SKIPPED vonmises_line distribution (taking more than 30 seconds)
SKIPPED weibull_max distribution (taking more than 30 seconds)
SKIPPED weibull_min distribution (taking more than 30 seconds)
```

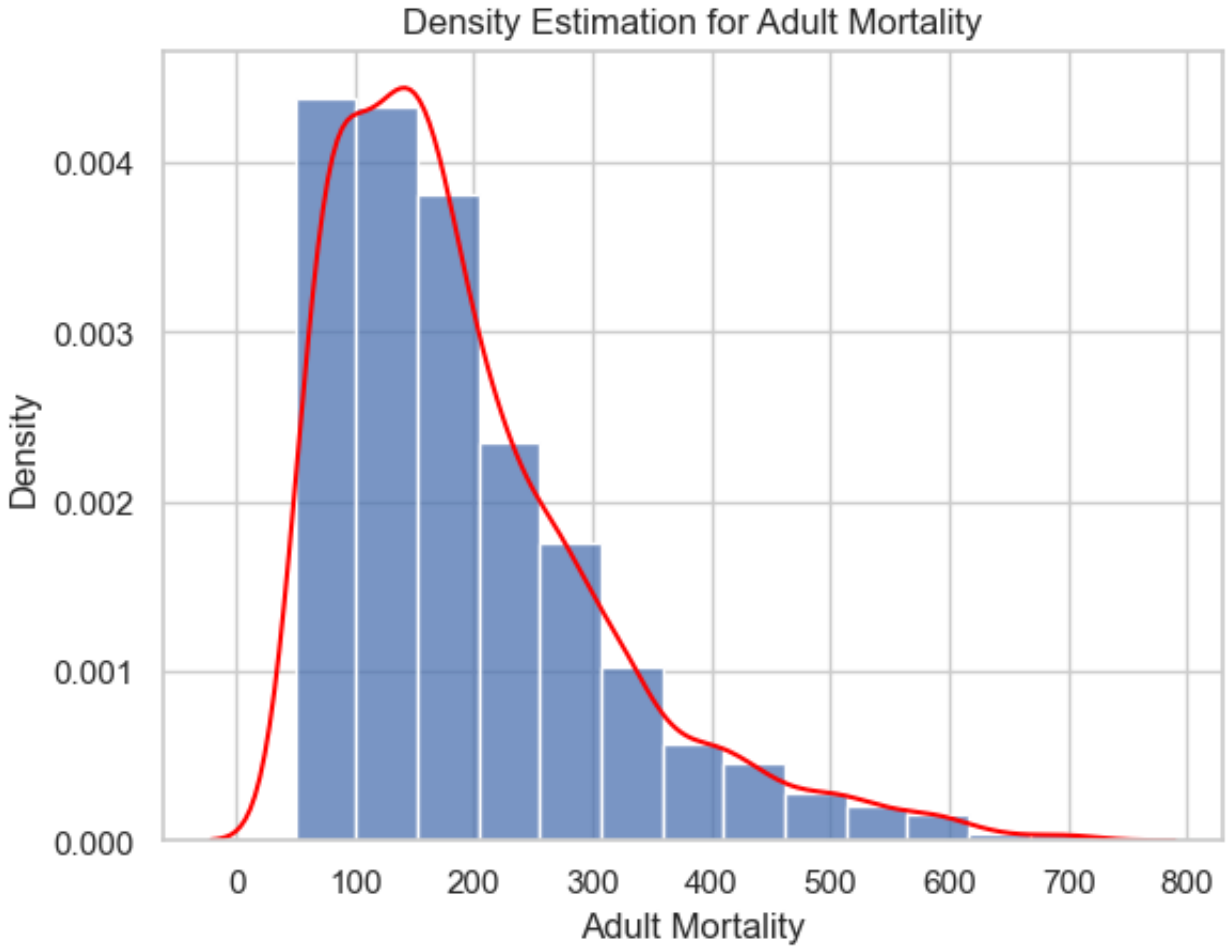
Out[31]:

	sumsquare_error	aic	bic	kl_div	ks_statistic	ks_pvalue
geninvgauss	0.000207	1277.365902	1301.205800	inf	0.047221	5.469942e-06
recipinvgauss	0.000207	1275.361950	1293.241873	inf	0.047157	5.663289e-06
genhyperbolic	0.000260	1278.487139	1308.287012	inf	0.045092	1.690296e-05
halfgennorm	0.000297	1281.330105	1299.210029	inf	0.052825	2.189389e-07

fatiguelife	0.000365	1282.755378	1300.635302	inf	0.063071	2.393940e-10



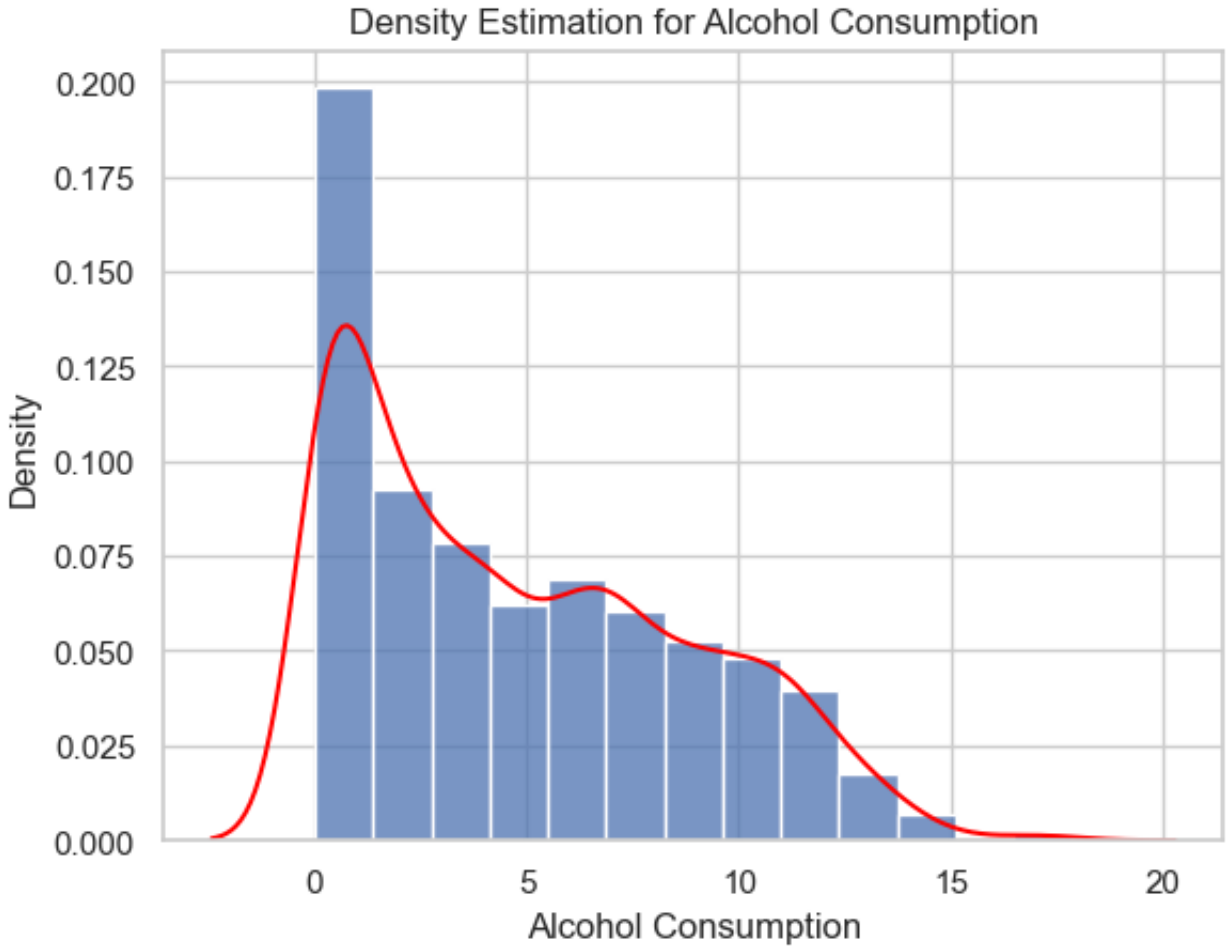
In [32]:
```
plt.title("Density Estimation for Adult Mortality")
sns.histplot(dt.Adult_mortality, stat = "density", bins = 'sturges')
sns.kdeplot(dt.Adult_mortality, color = "red")
plt.ylabel("Density")
plt.xlabel("Adult Mortality")
plt.show()
```



In [33]:
```
f_mortality = Fitter(dt['Adult_mortality'])
f_mortality.fit()
f_mortality.summary()
```

```
SKIPPED _fit distribution (taking more than 30 seconds)
SKIPPED alpha distribution (taking more than 30 seconds)
SKIPPED argus distribution (taking more than 30 seconds)
SKIPPED arcsine distribution (taking more than 30 seconds)
SKIPPED beta distribution (taking more than 30 seconds)
SKIPPED betaprime distribution (taking more than 30 seconds)
SKIPPED bradford distribution (taking more than 30 seconds)
SKIPPED burr distribution (taking more than 30 seconds)
SKIPPED burr12 distribution (taking more than 30 seconds)
SKIPPED chi2 distribution (taking more than 30 seconds)
SKIPPED crystalball distribution (taking more than 30 seconds)
SKIPPED exponpow distribution (taking more than 30 seconds)
SKIPPED exponweib distribution (taking more than 30 seconds)
SKIPPED f distribution (taking more than 30 seconds)
SKIPPED fatiguelife distribution (taking more than 30 seconds)
SKIPPED fisk distribution (taking more than 30 seconds)
SKIPPED foldcauchy distribution (taking more than 30 seconds)
SKIPPED foldnorm distribution (taking more than 30 seconds)
SKIPPED gausshyper distribution (taking more than 30 seconds)
SKIPPED genexpon distribution (taking more than 30 seconds)
```

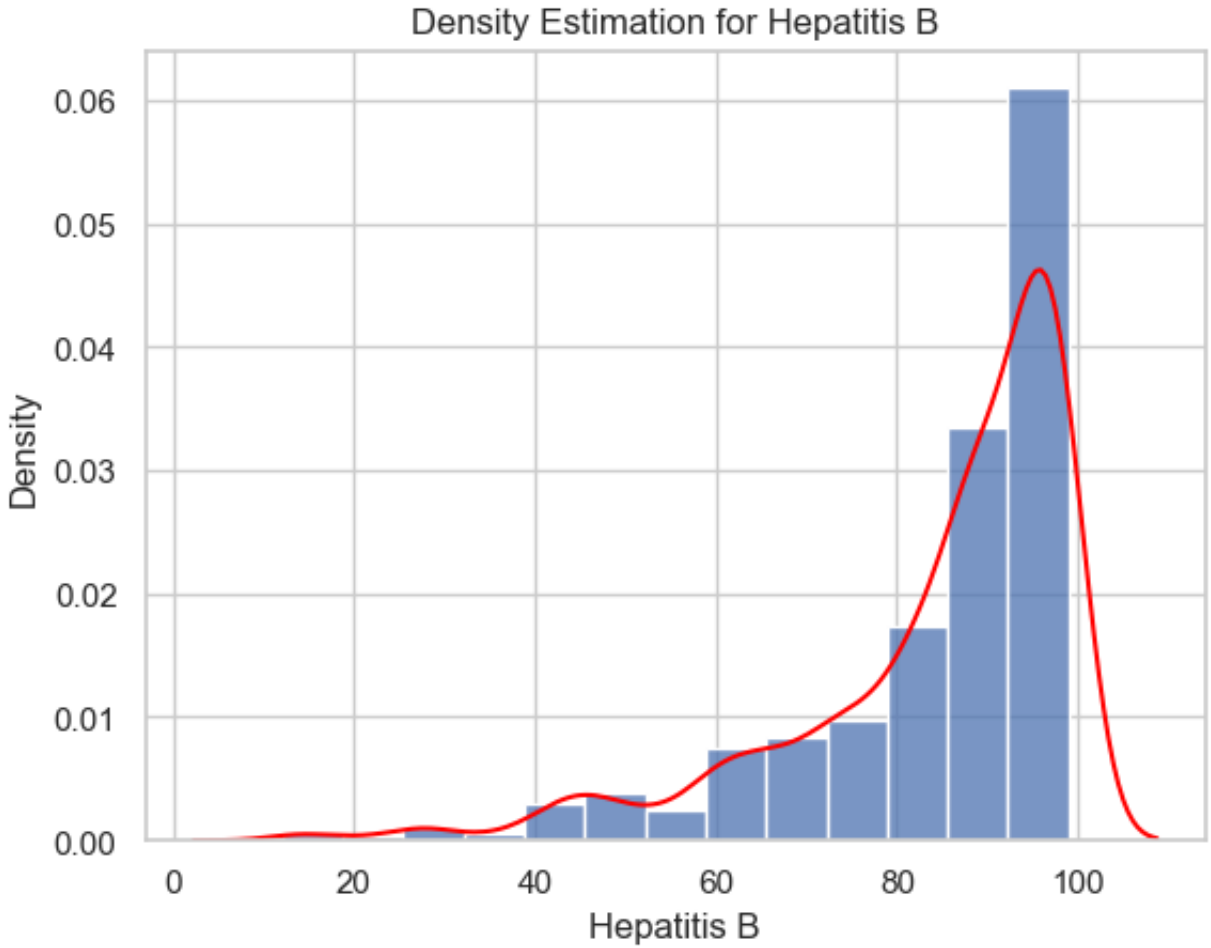
In [34]:
```
plt.title("Density Estimation for Alcohol Consumption")
sns.histplot(dt.Alcohol_consumption, stat = "density", bins = 'sturges')
sns.kdeplot(dt.Alcohol_consumption, color = "red")
plt.ylabel("Density")
plt.xlabel("Alcohol Consumption")
plt.show()
```



In [35]:
```
f_alcohol = Fitter(dt['Alcohol_consumption'])
f_alcohol.fit()
f_alcohol.summary()
```

```
SKIPPED _fit distribution (taking more than 30 seconds)
SKIPPED arcsine distribution (taking more than 30 seconds)
SKIPPED argus distribution (taking more than 30 seconds)
SKIPPED betaprime distribution (taking more than 30 seconds)
SKIPPED beta distribution (taking more than 30 seconds)
SKIPPED bradford distribution (taking more than 30 seconds)
SKIPPED burr distribution (taking more than 30 seconds)
SKIPPED burr12 distribution (taking more than 30 seconds)
SKIPPED chi distribution (taking more than 30 seconds)
SKIPPED chi2 distribution (taking more than 30 seconds)
SKIPPED crystalball distribution (taking more than 30 seconds)
SKIPPED erlang distribution (taking more than 30 seconds)
SKIPPED exponnorm distribution (taking more than 30 seconds)
SKIPPED exponpow distribution (taking more than 30 seconds)
SKIPPED exponweib distribution (taking more than 30 seconds)
SKIPPED f distribution (taking more than 30 seconds)
SKIPPED fatiguelife distribution (taking more than 30 seconds)
SKIPPED fisk distribution (taking more than 30 seconds)
SKIPPED foldcauchy distribution (taking more than 30 seconds)
SKIPPED foldnorm distribution (taking more than 30 seconds)
```

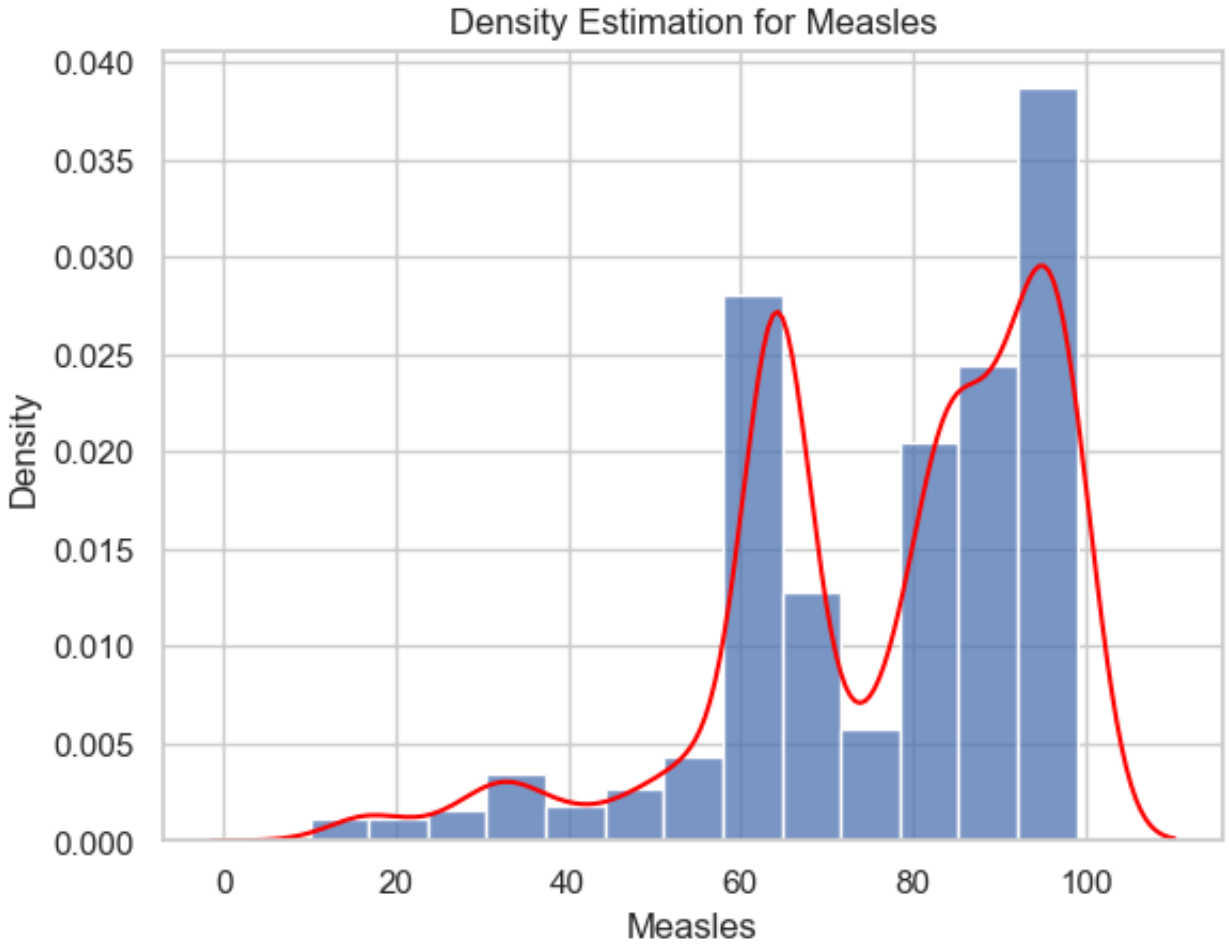
In [36]:
```
plt.title("Density Estimation for Hepatitis B")
sns.histplot(dt.Hepatitis_B, stat = "density", bins = 'sturges')
sns.kdeplot(dt.Hepatitis_B, color = "red")
plt.ylabel("Density")
plt.xlabel("Hepatitis B")
plt.show()
```



In [37]:
```
f_hep = Fitter(dt['Hepatitis_B'])
f_hep.fit()
f_hep.summary()
```

```
SKIPPED _fit distribution (taking more than 30 seconds)
SKIPPED alpha distribution (taking more than 30 seconds)
SKIPPED anglit distribution (taking more than 30 seconds)
SKIPPED arcsine distribution (taking more than 30 seconds)
SKIPPED beta distribution (taking more than 30 seconds)
SKIPPED betaprime distribution (taking more than 30 seconds)
SKIPPED bradford distribution (taking more than 30 seconds)
SKIPPED argus distribution (taking more than 30 seconds)
SKIPPED burr distribution (taking more than 30 seconds)
SKIPPED burr12 distribution (taking more than 30 seconds)
SKIPPED chi distribution (taking more than 30 seconds)
SKIPPED chi2 distribution (taking more than 30 seconds)
SKIPPED cosine distribution (taking more than 30 seconds)
SKIPPED crystalball distribution (taking more than 30 seconds)
SKIPPED dweibull distribution (taking more than 30 seconds)
SKIPPED erlang distribution (taking more than 30 seconds)
SKIPPED exponnorm distribution (taking more than 30 seconds)
SKIPPED exponpow distribution (taking more than 30 seconds)
SKIPPED exponweib distribution (taking more than 30 seconds)
SKIPPED f distribution (taking more than 30 seconds)
```

In [38]:
```
plt.title("Density Estimation for Measles")
sns.histplot(dt.Measles, stat = "density", bins = 'sturges')
sns.kdeplot(dt.Measles, color = "red")
plt.ylabel("Density")
plt.xlabel("Measles")
plt.show()
```



In [ ]:
```
f_measles = Fitter(dt['Measles'])
f_measles.fit()
f_measles.summary()
```

```
SKIPPED _fit distribution (taking more than 30 seconds)
SKIPPED alpha distribution (taking more than 30 seconds)
SKIPPED arcsine distribution (taking more than 30 seconds)
SKIPPED beta distribution (taking more than 30 seconds)
SKIPPED bradford distribution (taking more than 30 seconds)
SKIPPED argus distribution (taking more than 30 seconds)
SKIPPED betaprime distribution (taking more than 30 seconds)
SKIPPED burr distribution (taking more than 30 seconds)
SKIPPED burr12 distribution (taking more than 30 seconds)
SKIPPED chi distribution (taking more than 30 seconds)
SKIPPED chi2 distribution (taking more than 30 seconds)
SKIPPED cosine distribution (taking more than 30 seconds)
SKIPPED crystalball distribution (taking more than 30 seconds)
SKIPPED dgamma distribution (taking more than 30 seconds)
SKIPPED dweibull distribution (taking more than 30 seconds)
SKIPPED erlang distribution (taking more than 30 seconds)
SKIPPED exponnorm distribution (taking more than 30 seconds)
SKIPPED exponpow distribution (taking more than 30 seconds)
SKIPPED exponweib distribution (taking more than 30 seconds)
SKIPPED f distribution (taking more than 30 seconds)
SKIPPED fatiguelife distribution (taking more than 30 seconds)
SKIPPED fisk distribution (taking more than 30 seconds)
SKIPPED foldcauchy distribution (taking more than 30 seconds)
SKIPPED foldnorm distribution (taking more than 30 seconds)
SKIPPED gamma distribution (taking more than 30 seconds)
SKIPPED gausshyper distribution (taking more than 30 seconds)
SKIPPED genexpon distribution (taking more than 30 seconds)
SKIPPED genextreme distribution (taking more than 30 seconds)
SKIPPED gengamma distribution (taking more than 30 seconds)
SKIPPED genhalflogistic distribution (taking more than 30 seconds)
SKIPPED genhyperbolic distribution (taking more than 30 seconds)
```

```
SKIPPED geninvgauss distribution (taking more than 30 seconds)
SKIPPED genlogistic distribution (taking more than 30 seconds)
SKIPPED gennorm distribution (taking more than 30 seconds)
SKIPPED genpareto distribution (taking more than 30 seconds)
SKIPPED gibrat distribution (taking more than 30 seconds)
SKIPPED gilbrat distribution (taking more than 30 seconds)
SKIPPED gompertz distribution (taking more than 30 seconds)
SKIPPED halfcauchy distribution (taking more than 30 seconds)
SKIPPED halfgennorm distribution (taking more than 30 seconds)
SKIPPED halflogistic distribution (taking more than 30 seconds)
SKIPPED halfnorm distribution (taking more than 30 seconds)
SKIPPED invgamma distribution (taking more than 30 seconds)
SKIPPED invgauss distribution (taking more than 30 seconds)
SKIPPED invweibull distribution (taking more than 30 seconds)
SKIPPED kstwo distribution (taking more than 30 seconds)
```

In [ ]:
```
plt.title("Density Estimation for BMI")
sns.histplot(dt.BMI, stat = "density", bins = 'sturges')
sns.kdeplot(dt.BMI, color = "red")
plt.ylabel("Density")
plt.xlabel("BMI")
plt.show()
```

In [ ]:
```
f_bmi = Fitter(dt['BMI'])
f_bmi.fit()
f_bmi.summary()
```

In [ ]:
```
plt.title("Density Estimation for Diphtheria")
sns.histplot(dt.Diphtheria, stat = "density", bins = 'sturges')
sns.kdeplot(dt.Diphtheria, color = "red")
plt.ylabel("Density")
plt.xlabel("Diphtheria")
plt.show()
```

In [ ]:
```
f_Dip = Fitter(dt['Diphtheria'])
f_Dip.fit()
f_Dip.summary()
```

In [ ]:
```
plt.title("Density Estimation for HIV")
sns.histplot(dt.Incidents_HIV, stat = "density", bins = 'sturges')
sns.kdeplot(dt.Incidents_HIV, color = "red")
plt.ylabel("Density")
plt.xlabel("HIV")
plt.show()
```

In [ ]:
```
f_hiv = Fitter(dt['Incidents_HIV'])
f_hiv.fit()
f_hiv.summary()
```

In [ ]:
```
plt.title("Density Estimation for GDP Per Capita")
sns.histplot(dt.GDP_per_capita, stat = "density", bins = 'sturges')
sns.kdeplot(dt.GDP_per_capita, color = "red")
plt.ylabel("Density")
plt.xlabel("GDP Per Capita")
plt.show()
```

In [ ]:
```
f_gdp = Fitter(dt['GDP_per_capita'])
f_gdp.fit()
f_gdp.summary()
```

In [ ]:
```
plt.title("Density Estimation for Schooling")
sns.histplot(dt.Schooling, stat = "density", bins = 'sturges')
sns.kdeplot(dt.Schooling, color = "red")
plt.ylabel("Density")
plt.xlabel("Schooling")
plt.show()
```

In [ ]:
```
f_school = Fitter(dt['Schooling'])
f_school.fit()
f_school.summary()
```

In [ ]:
```
counts = dt['Region'].value_counts()
plt.figure(figsize=(16, 6))
plt.bar(counts.index.tolist(), counts, color='skyblue')

plt.title('Distribution of Region')
plt.xlabel('Regions')
plt.ylabel('Frequency')
plt.xticks(rotation = 30)
plt.show()
```

In [ ]:
```
counts = dt['Economy_status_Developed'].value_counts()
plt.bar(counts.index.tolist(), counts, color='skyblue')
plt.xticks([0, 1], ['Developing', 'Developed'])

plt.title('Distribution of Economy Status')
plt.xlabel('Economy Status')
plt.ylabel('Frequency')
plt.show()
```

Q2(c) Non-linearities

In [ ]:
```
dt['L_GDP_per_capita'] = np.log(dt['GDP_per_capita'])
dt['L_Incidents_HIV'] = np.log(dt['Incidents_HIV'])
```

The pairplot in Q2(a) indicates a strong negative linear relationship between the response variable 'Life_expectancy' and the predictors 'Under_five_deaths' and 'Adult_mortality'. Additionally, we can see from the pairplot that 'Life_expectancy' has a linear relationship with the predictors 'Alcohol_consumption', 'Hepatitis_B', 'Measles', 'BMI', 'Diphtheria', and 'Schooling', respectively. However, there seems to be a logarithmic pattern in the relationship between 'Life_expectancy' and two predictors: 'Incidents_HIV' and 'GDP_per_capita'. Therefore, we perform a log transformation on 'Incidents_HIV' and 'GDP_per_capita' to make them linear.

If a regression model incorporates non-linear variables prior to transformation, it breaches the linearity assumption, leading to an inability to precisely capture the relationships within the data. Moreover, the inclusion of non-linear variables may contribute to heteroscedasticity and biased estimates of parameter coefficients.

Q2(d) Unusual Observations

From looking at our boxplots in Q2a, we can see that our response variable and our predictors have several outliers. The life expectancy response variable has a few outliers at the lower end around < 44 years. Some predictors have relatively few outliers, such as BMI and alcohol consumption, which have a few outliers on the higher end (>31 BMI and >17.5 liters), and measles immunization, which has a few outliers on the lower end (< 20%). This can be explained by the presence of alcoholics and overweight individuals, which likely account for the presence of higher outliers for BMI and alcohol consumption. Meanwhile, measles immunization likely has outliers on the lower end because immunization is available in most parts of the world, making it relatively more uncommon to not have measles immunization.

There are also a few predictors with an extremely large number of outliers. Child and adult mortality, GDP per capita, and incidents of HIV all have an extremely large number of outliers on the higher end. This can be explained by the fact that the large majority of the world's population is not well off. Most of the data set include developing nations, meaning that GDP is likely lower and mortality rates and disease incidence is higher. Countries that are able to experience higher GDP per capita and lower mortality rates + disease incidence are less common and therefore are more likely to be considered outliers.

Diptheria immunization and hepatitis B immunization, meanwhile, have a large number of outliers on the lower end. This can be explained similarly to measles immunization: these immunizations are largely available to most people, and it's relatively uncommon to not have them.

Q2(e)

In [ ]:
```
print(dt.isnull().sum())
```

No missing values (NAs) were found in the dataset, so there's no need to remove any values.

Q3 Model Building

Evaluate transformations of variables

Predictors 'GDP_per_capita' and 'Incidents_HIV' before log transformation

In [ ]:
```
sns.pairplot(dt,vars=['Life_expectancy', 'GDP_per_capita'], diag_kind="hist", kind = "reg")
plt.suptitle('Pairplot of Life Expectancy and GDP Per Capita', y=1.02)
plt.show()
sns.pairplot(dt,vars=['Life_expectancy', 'Incidents_HIV'], diag_kind="hist", kind = "reg")
plt.suptitle('Pairplot of Life Expectancy and HIV Incidents', y=1.02)
plt.show()
```

Predictors 'GDP_per_capita' and 'Incidents_HIV' after log transformation

In [ ]:
```
sns.pairplot(dt,vars=['Life_expectancy', 'L_GDP_per_capita'], diag_kind="hist", kind = "reg")
plt.suptitle('Pairplot of Life Expectancy and Log GDP Per Capita', y=1.02)
plt.show()
sns.pairplot(dt,vars=['Life_expectancy', 'L_Incidents_HIV'], diag_kind="hist", kind = "reg")
plt.suptitle('Pairplot of Life Expectancy and Log HIV Incidents', y=1.02)
plt.show()
```

Examining the pairplots before transformation reveals a non-linear logarithmic pattern between the response variable 'Life_expectancy' and predictors 'GDP_per_capita' and 'Incidents_HIV'. Consequently, we opt to log-transform the two predictors to establish linear relationships. Subsequently, as observed in the above pairplots after the log transformation, the correlation between 'Life_expectancy' and predictors 'GDP_per_capita' and 'Incidents_HIV' is now deemed linear, and the distributions of the predictors are notably more normalized.

Test for multicollinearity, heteroskedasticity, model misspecification

In [ ]:
```
full_model = smf.ols(
    formula="Life_expectancy~Under_five_deaths+Adult_mortality+Alcohol_consumption+Hepatitis_B+Measles+B
full_model.summary()
```

In [ ]:
```
# test of multicollinearity
dt['Intercept'] = 1
X = pd.concat([dt[['Intercept', 'Under_five_deaths','Adult_mortality','Alcohol_consumption',
        'Hepatitis_B','Measles','BMI','Diphtheria','L_Incidents_HIV','L_GDP_per_capita',
        'Schooling']], dt.Economy_status_Developed], axis=1)
vif_data = pd.DataFrame()
vif_data['feature'] = X.columns
vif_data["VIF"] = [variance_inflation_factor(X.values, i) for i in range(len(X.columns))]
print(vif_data)
```

Predictors 'Under_five_deaths', 'Adult_mortality', "Diphtheria", "L_GDP_per_capita", and "Schooling" are above the VIF threshold of 4. There is multicollinearity amongst the variables in the model.

In [ ]:
```
# test for heteroskedasticity
y,X = pt.dmatrices("Life_expectancy~Under_five_deaths+Adult_mortality+Alcohol_consumption+Hepatitis_B+Me
                   data=dt, return_type = 'dataframe')
name = ["Lagrange multiplier statistic", "p-value", "f-value", "f p-value"]
sm.stats.diagnostic.het_breuschpagan(full_model.resid, X)

print("BP Results:")
print(*list(zip(name, test)), sep = "\n")
```

P-value is less than 5%, we reject the null hypothesis, and conclude that variance is not constant.

In [ ]:
```
# test for model misspecification
reset_result = smo.reset_ramsey(res = full_model, degree = 2)
reset_result
```

The p-value of Reset test is less than 5%, we reject the null hypothesis, and thus need to take higher order and/or interaction terms into account.

In [ ]:
```
# I removed Under five deaths and Log GDP per capita
reduced_1 = smf.ols(formula="Life_expectancy~Adult_mortality+Alcohol_consumption+Hepatitis_B+Measles+BMI
reduced_1.summary()
```

In [ ]:
```
dt['Intercept'] = 1
X = pd.concat([dt[['Intercept', 'Adult_mortality','Alcohol_consumption',
        'Hepatitis_B','Measles','BMI','Diphtheria','L_Incidents_HIV',
        'Schooling']], dt.Economy_status_Developed], axis=1)
vif_data = pd.DataFrame()
vif_data['feature'] = X.columns
vif_data["VIF"] = [variance_inflation_factor(X.values, i) for i in range(len(X.columns))]
print(vif_data)
```

All predictors are under the VIF threshold of 4. Multicollinearity is not observed in this reduced form of model.

In [ ]:
```
 for heteroskedasticity
pt.dmatrices("Life_expectancy~Adult_mortality+Alcohol_consumption+Hepatitis_B+Measles+BMI+L_Incidents_HI
             data=dt, return_type = 'dataframe')
 sm.stats.diagnostic.het_breuschpagan(reduced_1.resid, X)

"BP Results:")
*list(zip(name, test)), sep = "\n")
```

Since our p-value is less than 0.05, this indicates that the variance is not constant.

In [ ]:
```
# test for model misspecification
reset_result = smo.reset_ramsey(res = reduced_1, degree = 2)
reset_result
```

The p-value of Reset test is less than 5%, we reject the null hypothesis, and thus need to take higher order and/or interaction terms into account.

In [ ]:
```
# I removed Adult mortality and Under five deaths.
reduced_2 = smf.ols(
    formula="Life_expectancy~Alcohol_consumption+Hepatitis_B+Measles+BMI+Diphtheria+L_Incidents_HIV+L_GD
reduced_2.summary()
```

In [ ]:
```
# test of multicollinearity

dt['Intercept'] = 1
X = pd.concat([dt[['Intercept', 'Alcohol_consumption',
        'Hepatitis_B','Measles','BMI','Diphtheria','L_Incidents_HIV','L_GDP_per_capita',
        'Schooling']], dt.Economy_status_Developed], axis=1)
vif_data = pd.DataFrame()
vif_data['feature'] = X.columns
vif_data["VIF"] = [variance_inflation_factor(X.values, i) for i in range(len(X.columns))]
print(vif_data)
```

All predictors are under the VIF threshold of 4. Multicollinearity is not observed in this reduced form of model.

In [ ]:
```
# test for heteroskedasticity
y,X = pt.dmatrices("Life_expectancy~Alcohol_consumption+Hepatitis_B+Measles+BMI+Diphtheria+L_Incidents_H
                   data=dt, return_type = 'dataframe')
test = sm.stats.diagnostic.het_breuschpagan(reduced_2.resid, X)

print("BP Results:")
print(*list(zip(name, test)), sep = "\n")
```

The p-value is less than 5%, indicating the variance is not constant. There is heteroskedasticity in the model.

In [ ]:
```
reset_result = smo.reset_ramsey(res = reduced_2, degree = 2)
reset_result
```

The p-value of Reset test is less than 5%, we reject the null hypothesis, and thus need to take higher order and/or interaction terms into account.

In reduced_3, we have included a quadratic of the veriable, Schooling and an interaction term between Economy Status and Adult Mortality.

In [ ]:
```
reduced_3 = smf.ols(formula="Life_expectancy~Adult_mortality+Alcohol_consumption+Hepatitis_B+Measles+BMI
reduced_3.summary()
```

In [ ]:
```
dt['Intercept'] = 1
dt['Interaction'] = dt['Adult_mortality']*dt['Economy_status_Developed']
dt['Schooling**2'] = dt['Schooling']**2
X = pd.concat([dt[['Intercept','Alcohol_consumption',
        'Hepatitis_B','Measles','BMI','Diphtheria','L_Incidents_HIV', 'L_GDP_per_capita',
        'Schooling','Interaction']]], axis=1)
vif_data = pd.DataFrame()
vif_data['feature'] = X.columns
vif_data["VIF"] = [variance_inflation_factor(X.values, i) for i in range(len(X.columns))]
print(vif_data)
```

In [ ]:
```
# test for heteroskedasticity
y,X = pt.dmatrices("Life_expectancy~Adult_mortality+Alcohol_consumption+Hepatitis_B+Measles+BMI+L_Incide
                   data=dt, return_type = 'dataframe')
test = sm.stats.diagnostic.het_breuschpagan(reduced_3.resid, X)

print("BP Results:")
print(*list(zip(name, test)), sep = "\n")
```

There exists heteroscedasticity in the model.

In [ ]:
```
reset_result = smo.reset_ramsey(res = reduced_3, degree = 2)
reset_result
```

There is still a requirement of quadratic and/or interaction terms.

Reduced model 4: taking reduced model 2, taken quadratic of Schooling and Alcohol Consumption series. Also took interaction between Economy Status and GDP per capita.

In [ ]:
```
reduced_4 = smf.ols(formula="Life_expectancy~Alcohol_consumption+I(Alcohol_consumption**2)+Hepatitis_B+M
reduced_4.summary()
```

In [ ]:
```
dt['Intercept'] = 1
dt['Interaction'] = dt['Economy_status_Developed']*dt["L_GDP_per_capita"]
dt['Schooling**2'] = dt['Schooling']**2
dt["Alcohol_consumption**2"] = dt["Alcohol_consumption"]**2
X = pd.concat([dt[['Intercept', 'Adult_mortality','Alcohol_consumption', 'Alcohol_consumption**2',
        'Hepatitis_B','Measles','BMI','Diphtheria','L_Incidents_HIV',
        'Schooling','Schooling**2','Interaction']]], axis=1)
vif_data = pd.DataFrame()
vif_data['feature'] = X.columns
vif_data["VIF"] = [variance_inflation_factor(X.values, i) for i in range(len(X.columns))]
print(vif_data)
```

Reduced model 5: Taking reduced model 2, created an interaction between Adult Mortality and BMI.

In [ ]:
```
reduced_5 = smf.ols(formula="Life_expectancy~Adult_mortality+Alcohol_consumption+Hepatitis_B+Measles+BMI
reduced_5.summary()
```

In [ ]:
```
reduced_5.fittedvalues
```

In [ ]:
```
dt['Intercept'] = 1
dt["Interaction"] = dt['Adult_mortality']*dt['Schooling']
X = pd.concat([dt[['Intercept','Alcohol_consumption',
        'Hepatitis_B','Measles', 'Diphtheria','L_Incidents_HIV',
        'BMI','L_GDP_per_capita', 'Interaction']], dt.Economy_status_Developed], axis=1)
vif_data = pd.DataFrame()
vif_data['feature'] = X.columns
vif_data["VIF"] = [variance_inflation_factor(X.values, i) for i in range(len(X.columns))]
print(vif_data)
```

In [ ]:
```
# test for heteroskedasticity
y,X = pt.dmatrices("Life_expectancy~Adult_mortality+Alcohol_consumption+Hepatitis_B+Measles+BMI+Diphther
                   data=dt, return_type = 'dataframe')
name = ["Lagrange multiplier statistic", "p-value", "f-value", "f p-value"]
test = sm.stats.diagnostic.het_breuschpagan(reduced_5.resid, X)

print("BP Results:")
print(*list(zip(name, test)), sep = "\n")
```

In [ ]:
```
reset_result = smo.reset_ramsey(res = reduced_5, degree = 2)
reset_result
```

The above two tests show that, there exists Heteroscedasticity in reduced_5 model and also it further requires interaction and/or quadratic terms.

In [ ]:
```
print(reduced_1.aic)
print(reduced_1.bic)
```

In [ ]:
```
print(reduced_2.aic)
print(reduced_2.bic)
```

In [ ]:
```
print(reduced_3.aic)
print(reduced_3.bic)
```

In [ ]:
```
print(reduced_4.aic)
print(reduced_4.bic)
```

In [ ]:
```
print(reduced_5.aic)
print(reduced_5.bic)
```

##Based on the AIC and BIC value, we have finalised reduced_5 as our model since it has the lowest AIC and BIC amongst the reduced model. Although the full model has the lowest AIC and BIC among all models, it has issues with multicollinearity and has little relevance to the research question that we want to investigate. We ran 5 different models with interaction terms and quadratic terms, the reduced_5 gives us the least AIC and BIC values, strengthening our assertion to select it as our final model for further assessment.

In [ ]:
```
#Residuals plot
plt.figure(figsize=(10,6))
sns.regplot(x=reduced_5.fittedvalues,y=reduced_5.resid,data=dt,lowess=True)
plt.axhline(0, linestyle = '--', color = "red")
plt.ylabel("Residuals")
plt.xlabel("Fitted Values")
plt.title("Residuals Plot")
plt.show()
```

In [ ]:
```
#Residuals plot
plt.figure(figsize=(10,6))
sns.regplot(x=reduced_5.fittedvalues,y=dt["Life_expectancy"],data=dt,lowess=True)
plt.axhline(0, linestyle = '--', color = "red")
plt.ylabel("Observed Values")
plt.xlabel("Fitted Values")
plt.title("Residuals Plot")
plt.show()
```

In [ ]:
```
#Cook's Distance
cooks_distance = reduced_5.get_influence().cooks_distance
```

In [ ]:
```
plt.figure(figsize=(10,4))
plt.scatter(dt.index, cooks_distance[0])
plt.axhline(0, color = 'red')
plt.vlines(x = dt.index, ymin = 0, ymax = cooks_distance[0])
plt.xlabel('Index')
plt.ylabel('Cook\'s Distance')
plt.title("Diagnostic Plot")
plt.grid()
```

In [ ]:
```
fig, ax = plt.subplots(figsize=(10,5))
fig = sm.graphics.influence_plot(reduced_5, ax = ax, criterion="cooks")
```

In [ ]:
```
#Q-Q plot
figA, axA = plt.subplots(figsize=(6,4))
_, (__, ___, r) = sp.stats.probplot(reduced_5.resid, plot = axA, fit=True)
```

In [ ]:
```
cooks_distance=np.array(cooks_distance)
```

In [ ]:
```
#using loops to drop values
influential=np.where(cooks_distance>0.004)[0]

if len(influential)>0:
    dt=dt.drop(influential)
```

In [ ]:
```
dt.head(100)
```

In [ ]:
```
dt = dt.reset_index(drop=True)
```

In [ ]:
```
dt.head()
```

In [ ]:
```
dt.shape
```

In [ ]:
```
new_reg5 =smf.ols(formula="Life_expectancy~Adult_mortality+Alcohol_consumption+Hepatitis_B+Measles+BMI+D
new_reg5.summary()
```

In [ ]:
```
#Weighted Least Squares approach to mitigate Heteroscedasticity
new_reg5 =smf.ols(formula="Life_expectancy~Adult_mortality+Alcohol_consumption+Hepatitis_B+Measles+BMI+D
results= new_reg5.fit(cov_type='HC0')

table_ols = pd.DataFrame({'b': round(results.params, 4),
                          'se': round(results.bse, 4),
                          't': round(results.tvalues, 4),
                          'pval': round(results.pvalues, 4)})
print(f'table_ols: \n{table_ols}\n')


wls_weight = list(1 / dt['Adult_mortality'])
reg_wls = smf.wls(formula="Life_expectancy~Adult_mortality+Alcohol_consumption+Hepatitis_B+Measles+BMI+D
                  weights=wls_weight, data=dt)
results_wls = reg_wls.fit()

table_wls = pd.DataFrame({'b': round(results_wls.params, 4),
                          'se': round(results_wls.bse, 4),
                          't': round(results_wls.tvalues, 4),
                          'pval': round(results_wls.pvalues, 4)})
print(f'table_wls: \n{table_wls}\n')
```

In [ ]:
```
results_wls.summary()
```

In [ ]:
```
#Residuals plot
plt.figure(figsize=(10,6))
sns.regplot(x=results_wls.fittedvalues,y=results_wls.resid,data=dt,lowess=True)
plt.axhline(0, linestyle = '--', color = "red")
plt.ylabel("Residuals")
plt.xlabel("Fitted Values")
plt.title("Residuals Plot")
plt.show()
```

In [ ]:
```
# test for heteroskedasticity
y,X = pt.dmatrices("Life_expectancy~Adult_mortality+Alcohol_consumption+Hepatitis_B+Measles+BMI+Diphther
                   data=dt, return_type = 'dataframe')
name = ["Lagrange multiplier statistic", "p-value", "f-value", "f p-value"]
test = sm.stats.diagnostic.het_breuschpagan(results_wls.resid, X)

print("BP Results:")
print(*list(zip(name, test)), sep = "\n")
```

The WLS regression model takes the form:

$$Life\ Expectancy = 72.5393 - 0.0703 Adult\ Mortality + 0.0254 Alcohol\ Consu$$
$$- 0.004 Hepatitis\ B - 0.0014 Measles - 0.1389 BMI + 0.0525 Diptheria + 0.07$$
$$+ 0.6726 GDP\ per\ capita + 0.157 Schooling + \delta_1 Region$$
$$+ 1.7346 Economy\ Status\ Developed + 0.3262 Asia + 2.4157 Central\ Ameri$$
$$+ 0.1064 EuropeanUnion + 0.228 Middle\ East + 0.6418 North\ America$$
$$+ 0.2046 Oceania + 0.8527 Restof Europe + 2.1 SouthAmerica + 0.0011 AdultM$$
$$* Schooling$$

$$\#\#\frac{\partial LifeExpectancy}{\partial Schooling} = \beta_{17} + \beta_{19} AdultMortality$$

$$\#\#\frac{\partial LifeExpectancy}{\partial AdultMortality} = \beta_9 + \beta_{19} Schooling$$

The marginal effect of predictor would be the same as the estimated coefficient listed above, since they are in linear relationship with life expectancy, except for two predictors: Adult Mortality and Schooling, as they make up the interaction term. Holding everything else constant, an one year increase in schooling would lead to a (0.157+0.0011 = 0.1581) increase in life expectancy on average. Holding everything else constant, an one-unit increase in adult mortality would lead to (-0.0703+0.0011 = -0.0692) change in life expectancy on average.

In [ ]:
```
#Bootstrapping
b_slopes = []
b_intercept = []
b_adjR2 = []
b_R2 = []
n_boots = 1000
n_points = dt.shape[0]
plt.figure()
for _ in range(n_boots):
    df = dt.sample(n=n_points, replace=True)

    ols_model = smf.ols(formula = 'Life_expectancy~Adult_mortality+Alcohol_consumption+Hepatitis_B+Measl
    results = ols_model.fit()


    b_intercept.append(results.params[0])
    b_slopes.append(results.params[1])
    b_adjR2.append(results.rsquared_adj)
    b_R2.append(results.rsquared)
```

In [ ]:
```
beta_coeff = results_wls.params
```

In [ ]:
```
plt.figure(figsize=(10, 6))
sns.histplot(b_slopes, kde = True, alpha = 0.5)
for beta in beta_coeff[1:]:
  plt.axvline(x=beta, color='red', linestyle='--', label="WLS Coefficients")

plt.title('Bootstrapped Slopes')
plt.xlabel('Slope Values')
plt.ylabel('Frequency')
plt.show()
```

The above graph clearly shows that on 1000 bootstrapping over dependent variables, most of our weighted least squares estimated paraemters do not overlap with the bootstrapped slope parameters. Countably, only 3 LS parameters, overlap with the bootstrapped estimates. It shows certain pitfalls in the estimation and doesn't provide a strong case for robustness of our estimates.

In [ ]:
```
plt.figure(figsize=(10, 6))
sns.histplot(b_intercept, kde = True, alpha = 0.5)
plt.axvline(x=beta_coeff[0], color='red', linestyle='--', label="WLS Intercept")


plt.title('Bootstrapped Intercept')
plt.xlabel('Intercept Values')
plt.ylabel('Frequency')
plt.show()
```

The above graph clearly shows that the bootstrapped intercept and our estimated intercept from the WLS do not overlap, again indicating lack of robustness of estimation in our model. However, we can further investigate with other methods to see, whether this is the final conclusion for our estimates.

In [ ]:
```
#Cross-Validation
dt_encoded = pd.get_dummies(dt, columns=["Region"])
y=dt_encoded["Life_expectancy"]
x=dt_encoded.drop(columns=["Life_expectancy"])

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)


regress=LinearRegression()
regress.fit(x_train,y_train)


y_pred=regress.predict(x_test)

mse=metrics.mean_squared_error(y_test,y_pred)
rmse=np.sqrt(mse)
mae=metrics.mean_absolute_error(y_test,y_pred)

print("MSE:", mse)
print("RMSE:", rmse)
print("MAE:", mae)


regress=linear_model.LinearRegression()
score=cross_val_score(regress,x,y,cv=5,scoring="neg_root_mean_squared_error")

print("K-fold Cross Validation Result:")
print(score)
```

The above cross validation analysis shows that all MSE, RMSE and MAE have small values, comparing it to the range of dependent variables, indicating efficient prediction, since the difference between the predcited and observed values are small. The K-fold Cross Validation Scores show small values as well and no significant variation between the scores on each fold, indicating robust estimation of our parameters. The values also indicating that our model is a good fit overall.

##Conclusion: In finality, we would like to assert that our model overall is robust and a good fit as indicated by the Cross-Validation method. We also see that 96.2% of the Life Expectancy data is explained by the dependent variables as chosen from our model 5, which was selected on the basis of lowest AIC and BIC scores. ##Couple of questions, we had hoped to answer, can intepreted this way: 1.Whether or not increased adult and child mortality can have a negative effect on a country's life expectancy? We had dropped Child mortality due to increased case of multicollinearity, but in case of Adult Mortality, we see that an increase in Adult Mortality leads to a 7% decrease in Life Expectancy.

2.Whether or not BMI and alcohol consumption affect life expectancy? A unit increase in BMI, decreases Life Expectancy by almost 14% and on the other hand, Alcohol Consumption is not statiscally significant. Indicating no relationship between the Alcohol Consumption and Life Expectancy as per our model.

3.Whether or not increased immunization coverage for various diseases such as diptheria, measles, and Hep. B can have a positive effect on life expectancy? The Hepatitis B and Measles coverage data are not statistcal significance, showing that incidence of these have decreased overtime or their immunization are ineffective in determining Life Expectancy in different nations. On the other hand immunization data on Dipheria is statiscally signifcant and a one unit increase in Diptheria immunization leads to approximately 5% increase in Life Expectancy.

4.Whether or not economic factors such as a country's development status, GDP, and education can affect their life expectancy. We specifically want to investigate whether or not being a developed country can have a positive impact on a country's life expectancy, as well as whether or not increased GDP and education can have a positive impact on life expectancy? Here we wanted to assess the economic factors, we saw that GDP per capita is statically sigificant. A unit increase in GDP per capita has 67% increase in Life Expectancy, this is a very important conclusion from a policy perspective. As analysts, we would suggest further study on this result to see if steps taken to increase GDP per capita, specially in developing nations, can massively impact their life expectancy rates. In cognizance to our above result on GDP per capita, we also see that being a Developed nation can help increase Life Expectancy by 174% approximately. Both of these results show a promising picture on policy forefront, specially for developing nations with low GDP per capita. When we look at Schooling, it is also statiscally significant, and a unit increase in schooling leads to approximately 16% increase in Life Expectancy.

5.Whether or not countries in certain regions are more prone to having higher or lower life expectancies? There are 8 different regions being treated in our model with Africa as our Treatment variable. We see that all regions have a statistical significance except for European Union, Middle East and Oceania, compared to Africa. We see that compared to our reference varibale, Africa, all regions have a higher

values, with siginficant coeffieicnt estimates, indicating the poor life expectancy rates in the African region. It also condenses with the rest of our assessment that Africa being a region with more developing nations with low GDP per capita, should have relatively poor Life Expectancy outcomes.

##We observe that our model shows some need for improvement in terms of better approach to deal with pitfalls like heteroscedasticity or robustness of estimates, seeing the results of bootstrapping. But we also see a promising picture from Cross Validation and the regression result, which shows that this model can help answer more policy questions related to better life expectancy outcomes in developing nations.