Multiple Linear Regression

Data dictionary

Country - Country

Year - Year

Status - Developed or Developing status

Life expectancy - Life Expectancy in age

Adult Mortality - Adult Mortality Rates of both sexes (probability of dying between 15 and 60 years per 1000 population)

infant deaths - Number of Infant Deaths per 1000 population

Alcohol - Alcohol, recorded per capita (15+) consumption (in litres of pure alcohol)

percentage expenditure - Percentage expenditure on health as a percentage of Gross Domestic Product per capita(%)

Hepatitis B - Hepatitis B (HepB) immunization coverage among 1-year-olds (%)

Measles - number of reported cases per 1000 population

BMI - Average Body Mass Index of entire population

under-five deaths - Number of under-five deaths per 1000 population

Polio - Polio (Pol3) immunization coverage among 1-year-olds (%)

Total expenditure - General government expenditure on health as a percentage of total government expenditure (%)

Diphtheria - Diphtheria tetanus toxoid and pertussis (DTP3) immunization coverage among 1-year-olds (%)

HIV/AIDS - Deaths per 1 000 live births HIV/AIDS (0-4 years)

GDP - Gross Domestic Product per capita (in USD)

Population - Population of the country

thinness 1-19 years - Prevalence of thinness among children and adolescents for Age 10 to 19 (%)

thinness 5-9 years - Prevalence of thinness among children for Age 5 to 9(%)

Income composition of resources - Human Development Index in terms of income composition of resources (index ranging from 0 to 1)

Schooling - Number of years of Schooling(years)

Import library

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")

%matplotlib inline
```

Read data

In [2]:

```
ds = pd.read_csv('Life Expectancy Data.csv')
```

In [3]:

```
ds.columns
```

Out[3]:

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

In [4]:

```
ds.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 22 columns):
 #   Column                           Non-Null Count  Dtype  
---  ------                           --------------  -----  
 0   Country                          2938 non-null   object 
 1   Year                             2938 non-null   int64  
 2   Status                           2938 non-null   object 
 3   Life expectancy                  2928 non-null   float64
 4   Adult Mortality                  2928 non-null   float64
 5   infant deaths                    2938 non-null   int64  
 6   Alcohol                          2744 non-null   float64
 7   percentage expenditure           2938 non-null   float64
 8   Hepatitis B                      2385 non-null   float64
 9   Measles                          2938 non-null   int64  
 10   BMI                             2904 non-null   float64
 11  under-five deaths                2938 non-null   int64  
 12  Polio                            2919 non-null   float64
 13  Total expenditure                2712 non-null   float64
 14  Diphtheria                       2919 non-null   float64
 15   HIV/AIDS                        2938 non-null   float64
 16  GDP                              2490 non-null   float64
 17  Population                       2286 non-null   float64
 18   thinness  1-19 years            2904 non-null   float64
 19   thinness 5-9 years              2904 non-null   float64
 20  Income composition of resources  2771 non-null   float64
 21  Schooling                        2775 non-null   float64
dtypes: float64(16), int64(4), object(2)
memory usage: 505.1+ KB
```

Data preprocessing

In [5]:

```python
ds.isna().any()
```

Out[5]:

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

In [6]:

```python
for i in ds.columns:
    if ds[i].isna().any():
        ds[i].fillna(ds[i].mean(), inplace=True)
```

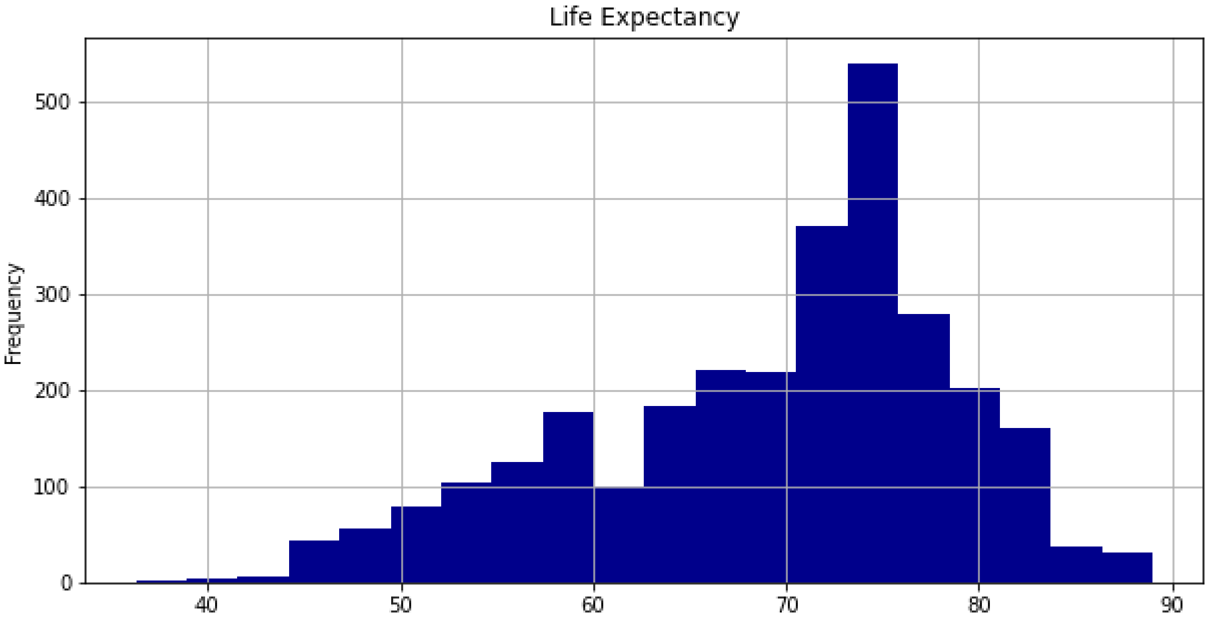
```
In [7]: ds.isna().any()
```

```
Out[7]: Country                            False
        Year                               False
        Status                             False
        Life expectancy                    False
        Adult Mortality                    False
        infant deaths                      False
        Alcohol                            False
        percentage expenditure             False
        Hepatitis B                        False
        Measles                            False
         BMI                               False
        under-five deaths                  False
        Polio                              False
        Total expenditure                  False
        Diphtheria                         False
         HIV/AIDS                          False
        GDP                                False
        Population                         False
         thinness  1-19 years              False
         thinness 5-9 years                False
        Income composition of resources    False
        Schooling                          False
        dtype: bool
```

EDA

In [8]:

```
plt.subplots(figsize = (10,5))
plt.hist(ds.iloc[:,3], color = 'darkblue', bins = 20)
plt.title('Life Expectancy')
plt.grid()
plt.ylabel('Frequency')
plt.show()
```



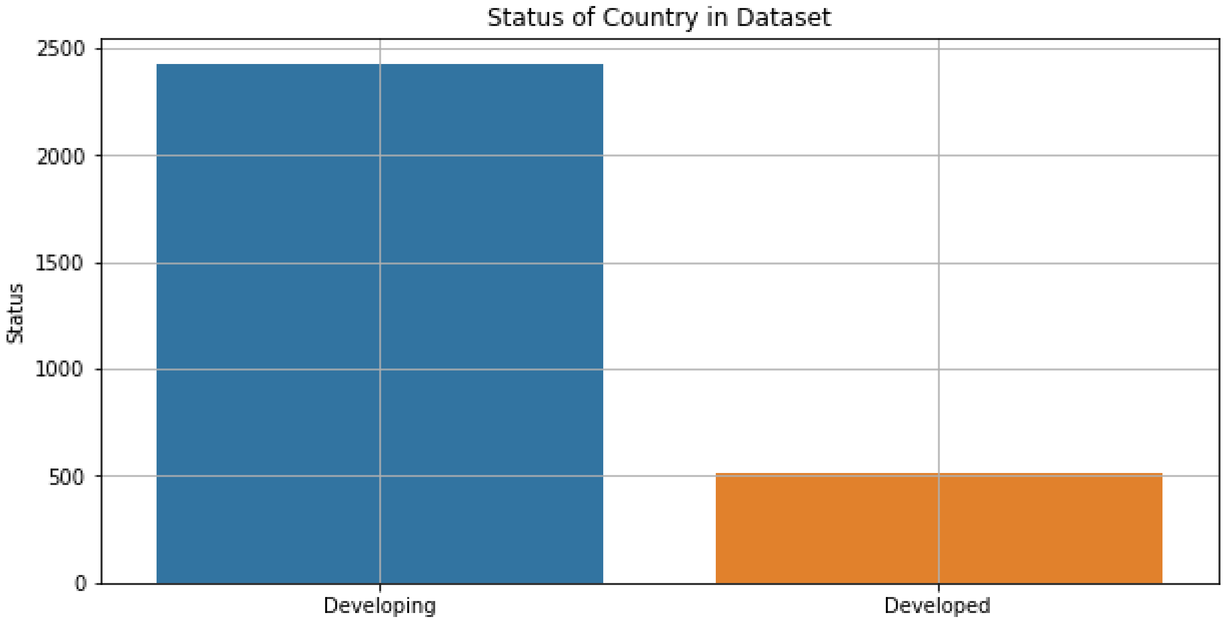
In [9]:

```
print('Maximum Life Expectancy: ', ds.iloc[:,3].max())
print('Minimum Life Expectancy: ', ds.iloc[:,3].min())
print('Most Life Expectancy: ', ds.iloc[:,3].mode())
```

```
Maximum Life Expectancy:  89.0
Minimum Life Expectancy:  36.3
Most Life Expectancy:  0    73.0
dtype: float64
```

In [10]:

```
plt.subplots(figsize = (10,5))
sns.barplot(ds.Status.value_counts().index, ds.Status.value_counts())
plt.title('Status of Country in Dataset')
plt.grid()
plt.show()
```



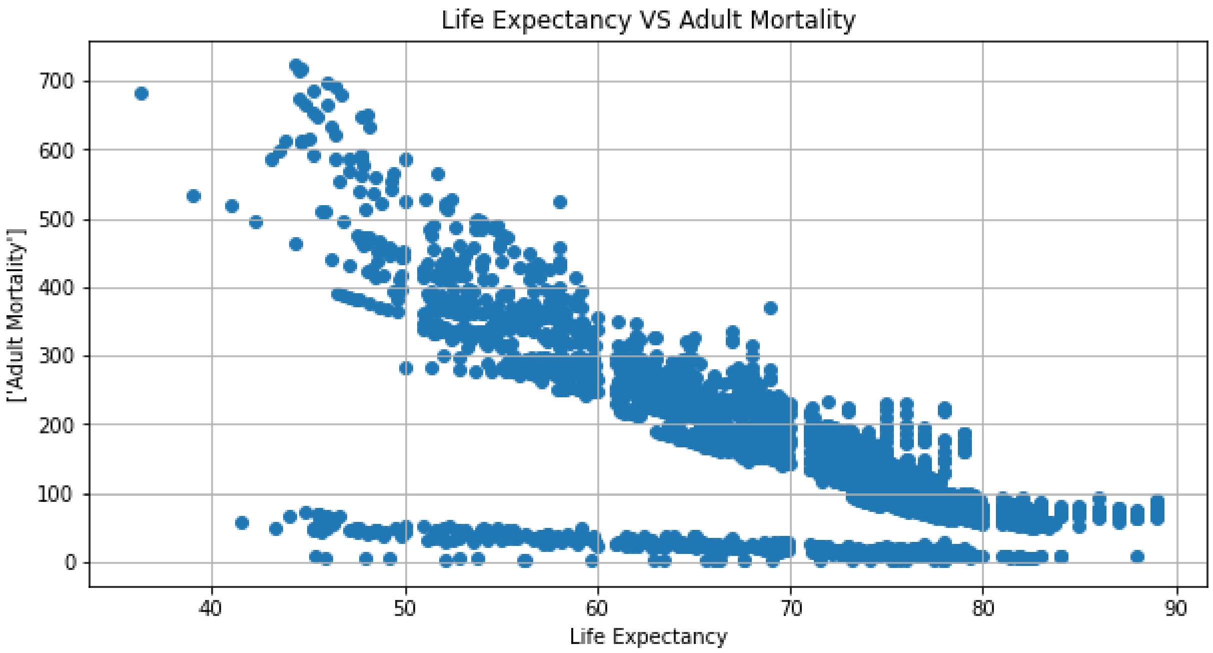
- Most of the country is a develop country in this dataset.

In [11]:

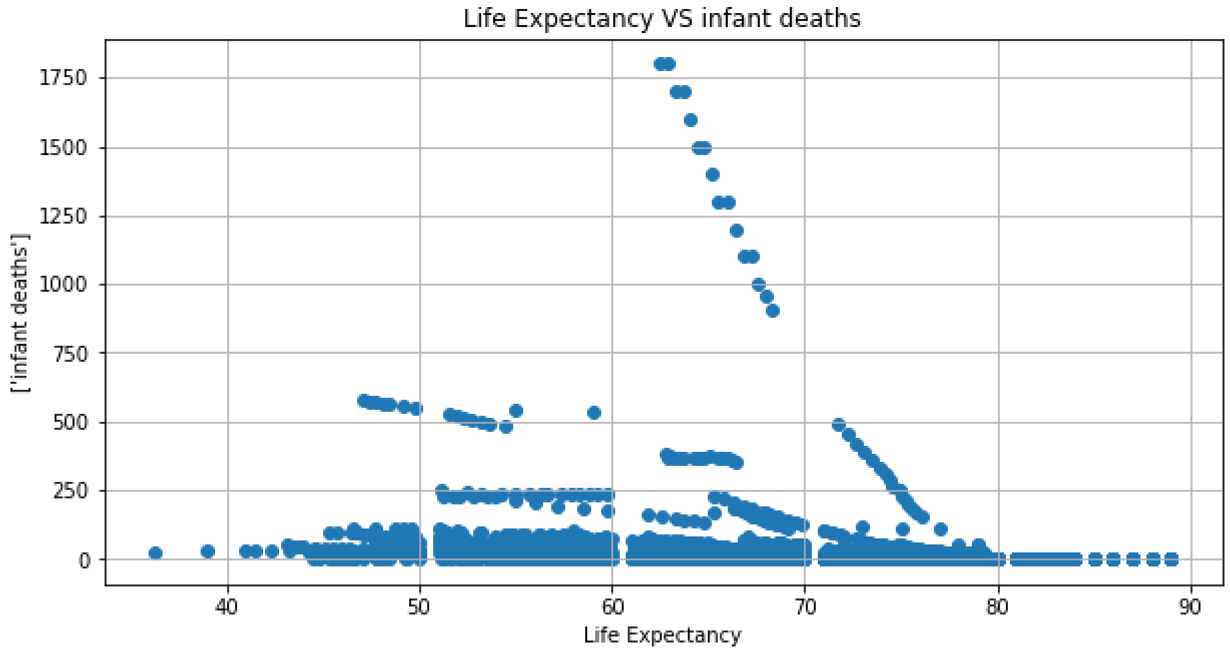
```
ds_num = ds.drop(['Country','Status', 'Life expectancy ', 'Year'], axis = 1)
```

In [12]:

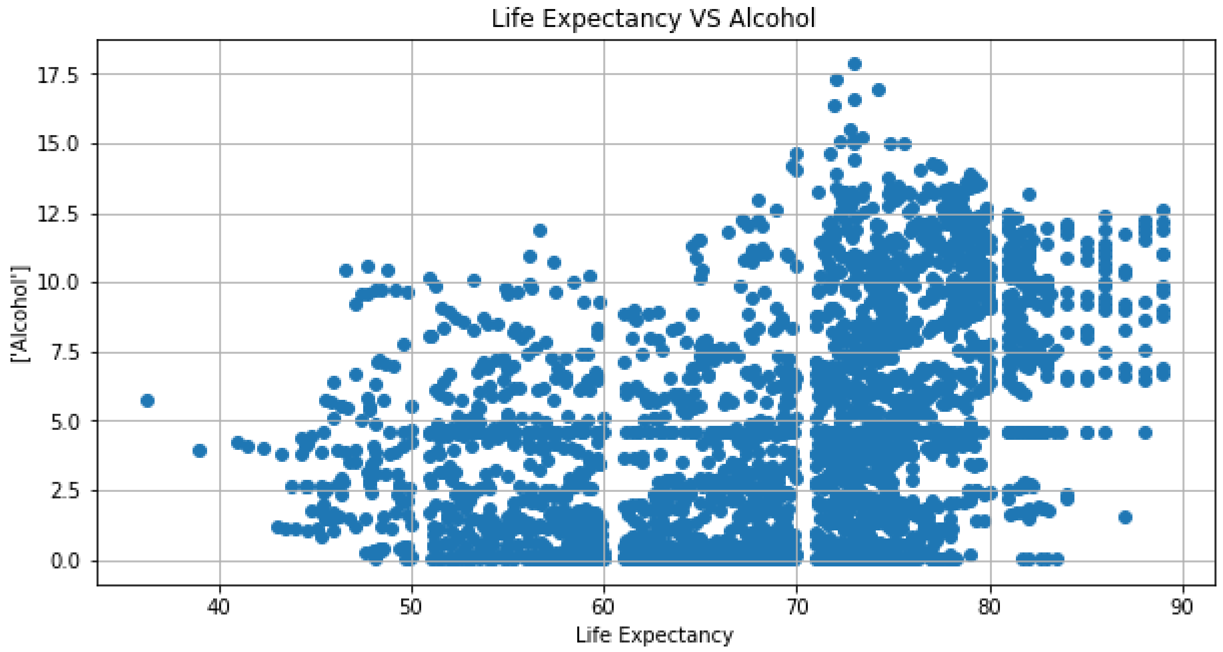
```
for i in ds_num.columns:
    plt.subplots(figsize = (10,5))
    plt.scatter(ds['Life expectancy '], ds_num[i])
    plt.title(f'Life Expectancy VS {i}')
    plt.xlabel('Life Expectancy')
    plt.ylabel([i])
    plt.grid()
    plt.show()
    corr = np.corrcoef(ds['Life expectancy '], ds_num[i])
    print(f'Correlation between Life Expectancy and {i}: ', corr[0,1])
```



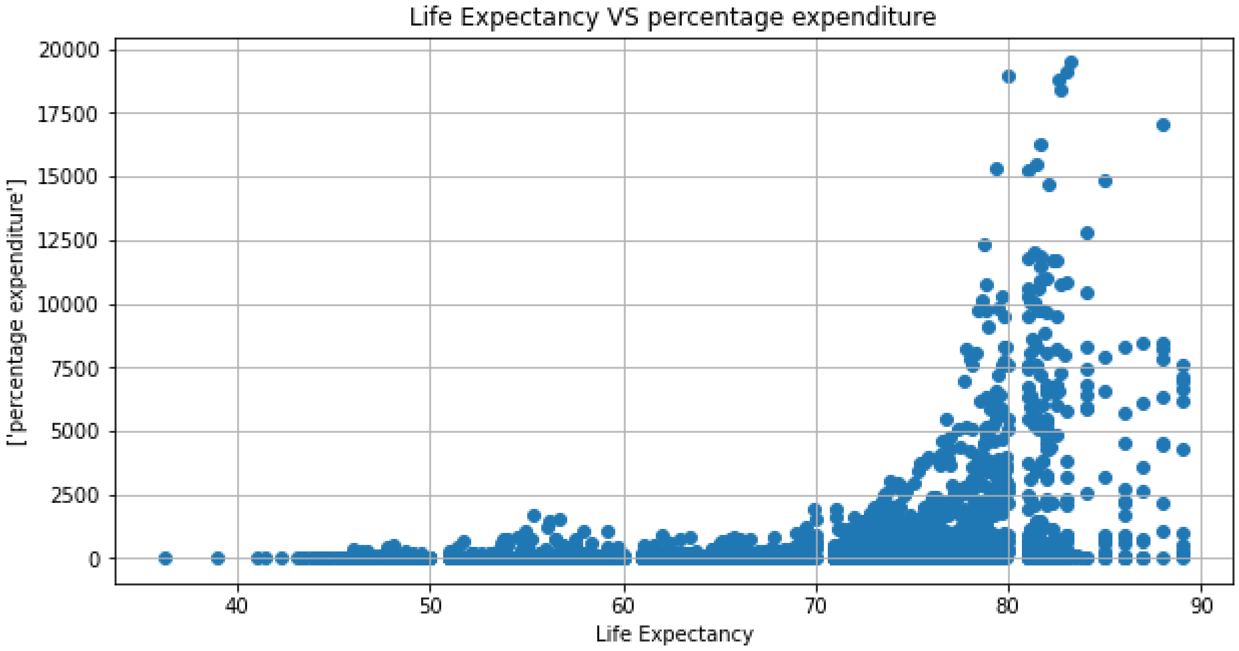
```
Correlation between Life Expectancy and Adult Mortality:  -0.6963593137699757
```



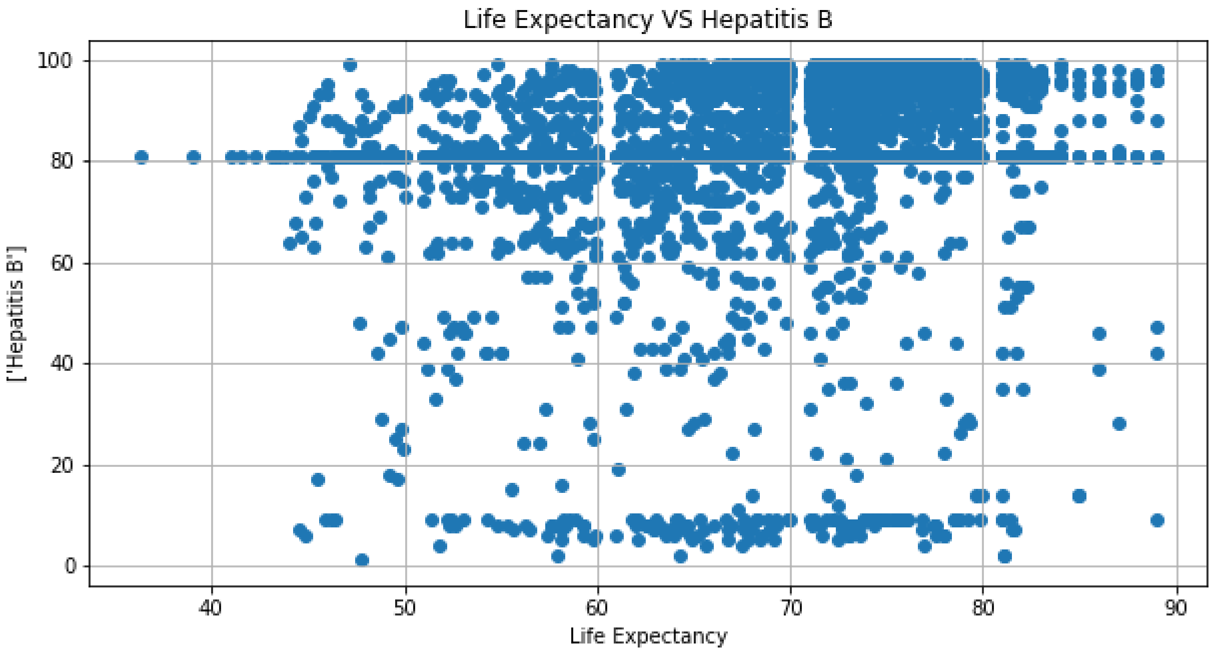
```
Correlation between Life Expectancy and infant deaths:  -0.19653500307699528
```



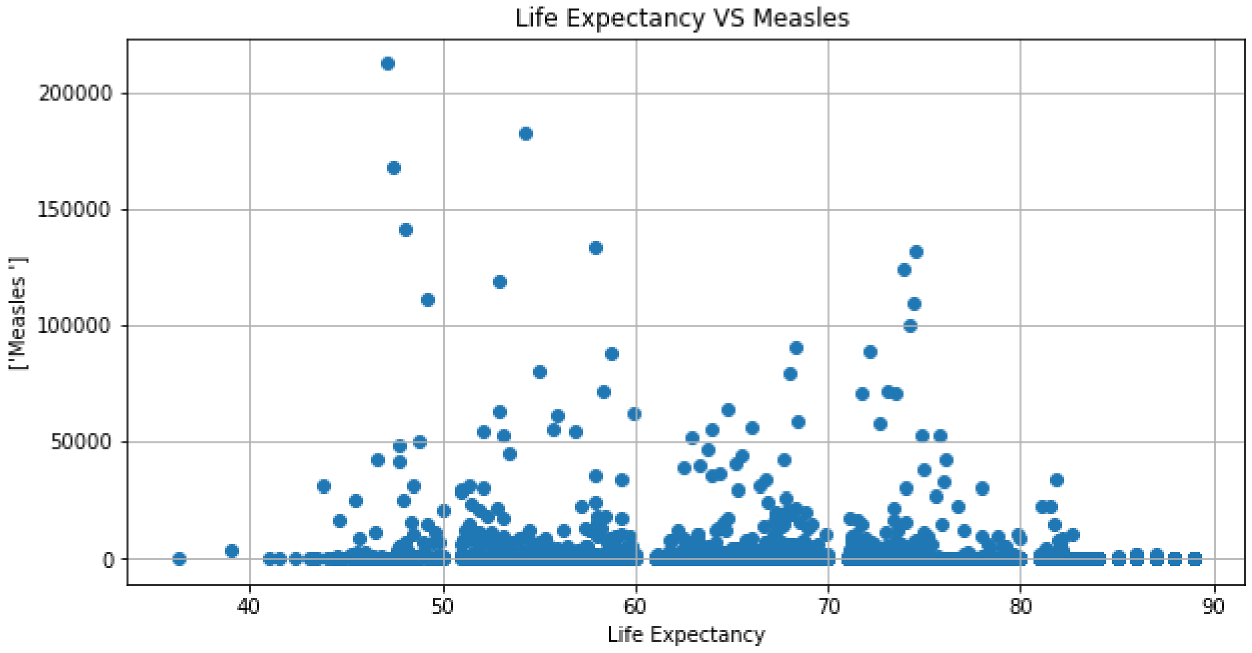
```
Correlation between Life Expectancy and Alcohol:  0.39159833938428923
```



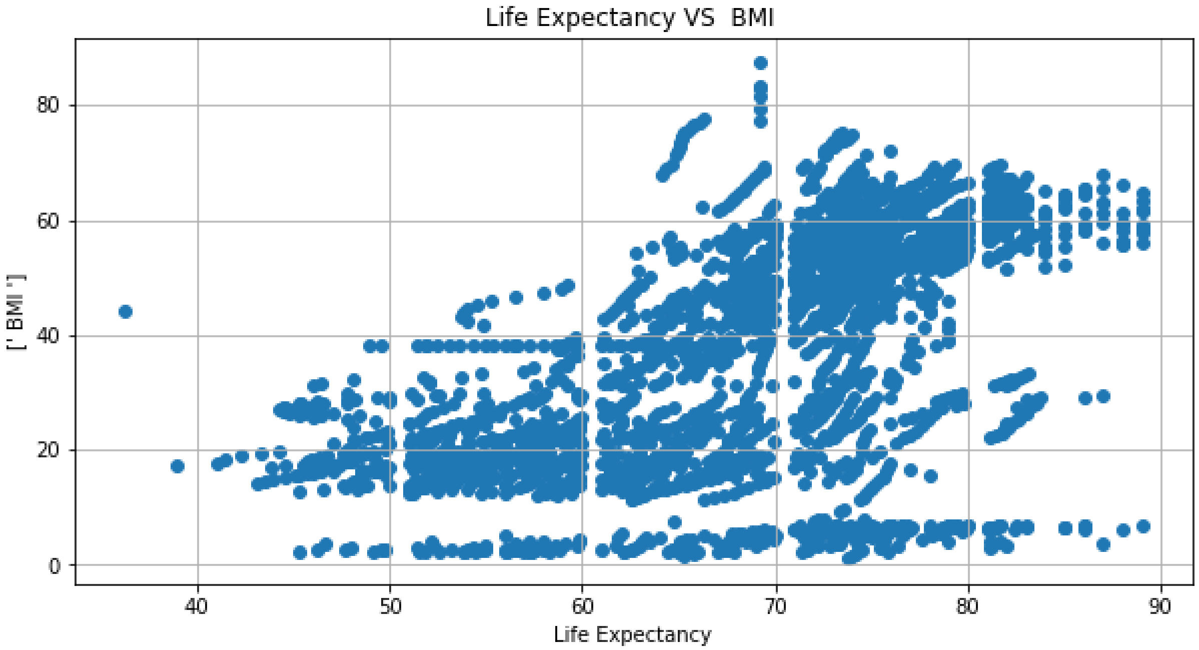
```
Correlation between Life Expectancy and percentage expenditure:  0.3817911732064308
```



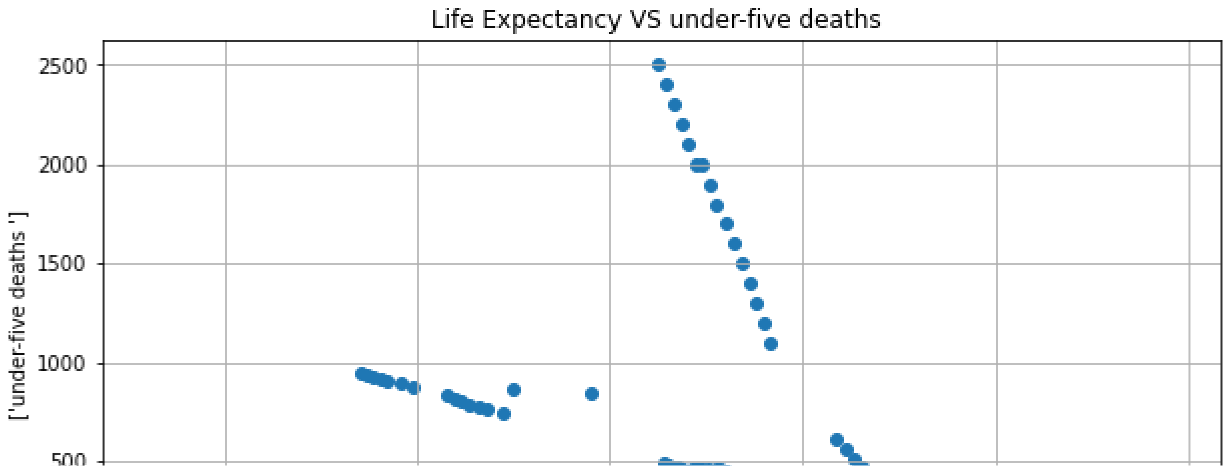
```
Correlation between Life Expectancy and Hepatitis B:  0.2037714374002677
```



```
Correlation between Life Expectancy and Measles :  -0.1575738185971695
```

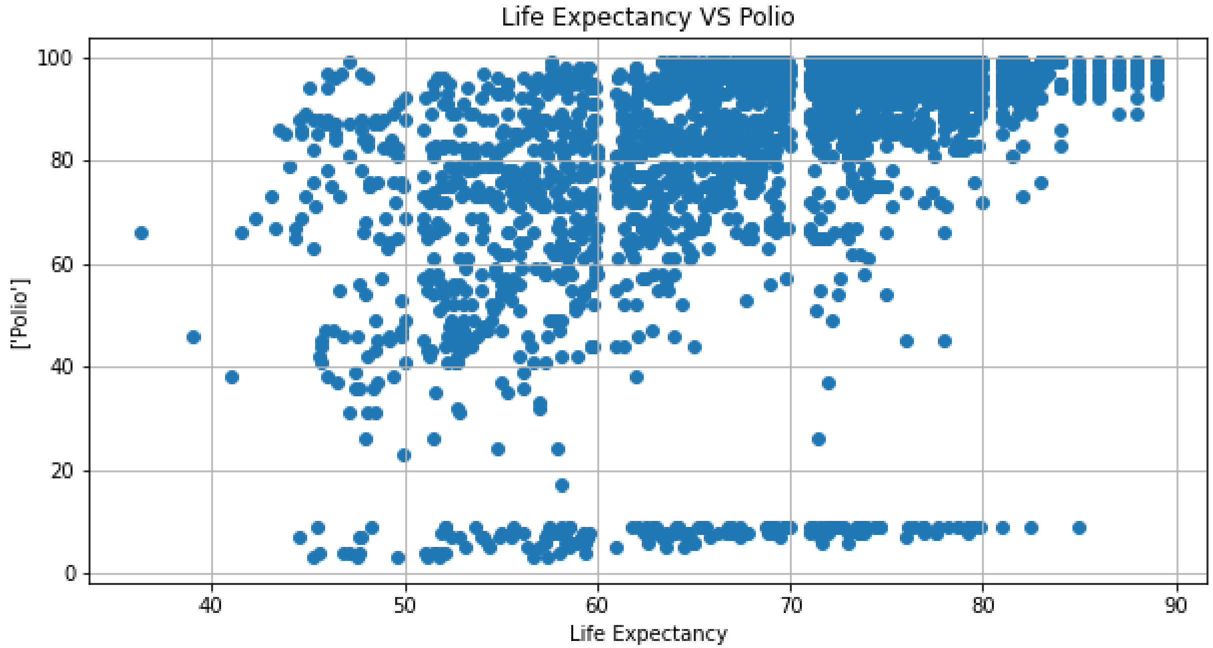


```
Correlation between Life Expectancy and  BMI :  0.5592553046406493
```



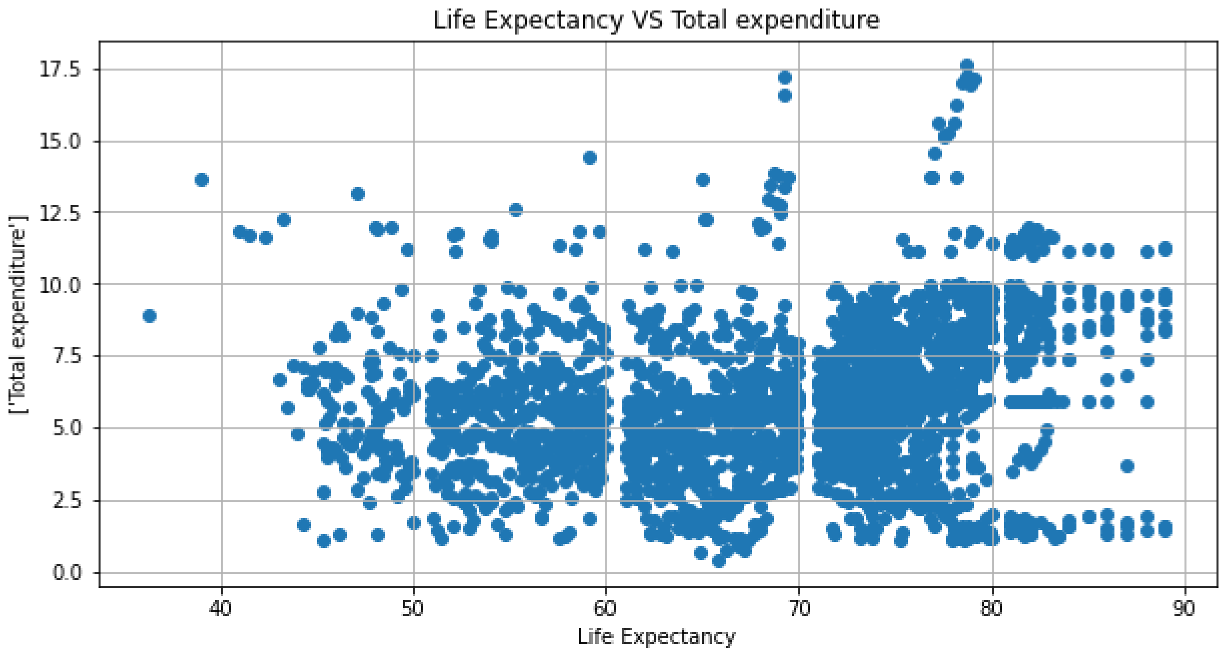
Life Expectancy VS under-five deaths

```
Correlation between Life Expectancy and under-five deaths :  -0.22250302192435054
```

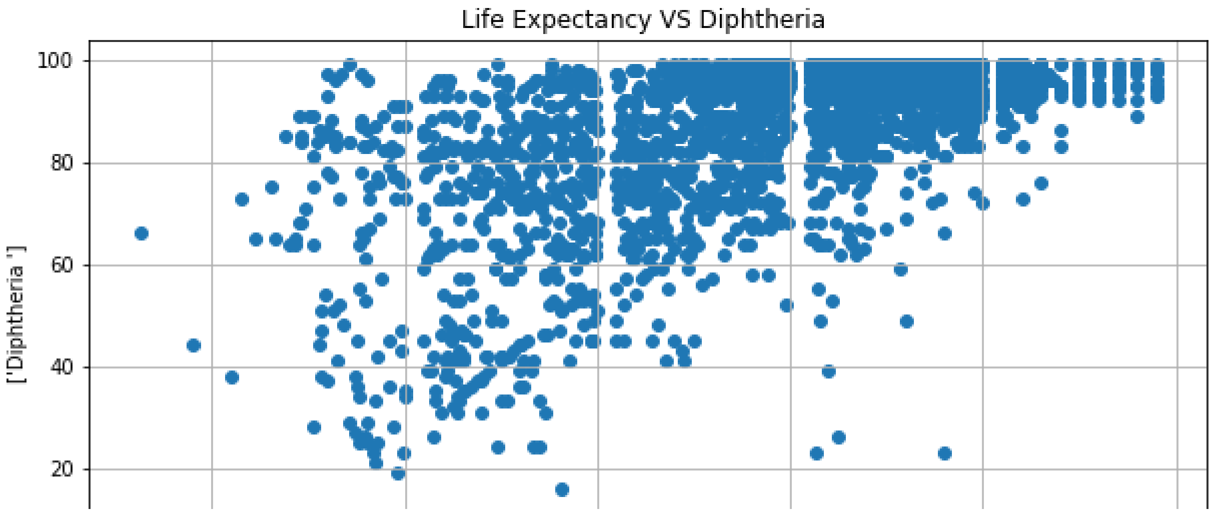


Life Expectancy VS Polio

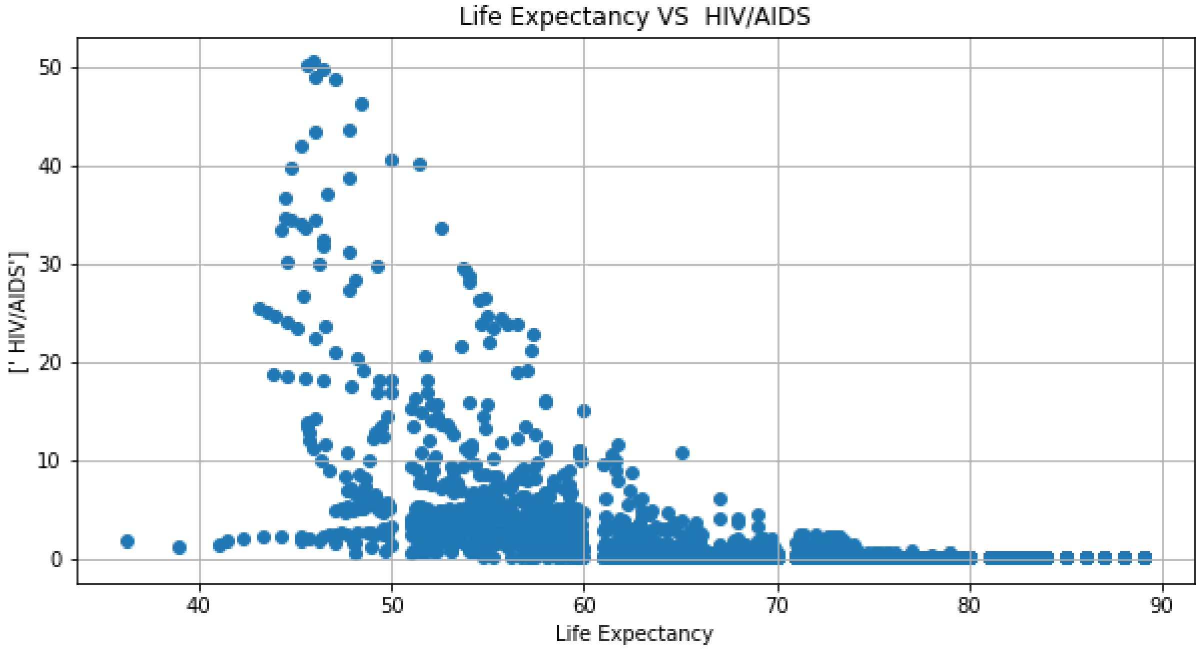
```
Correlation between Life Expectancy and Polio:  0.46157377544579
```



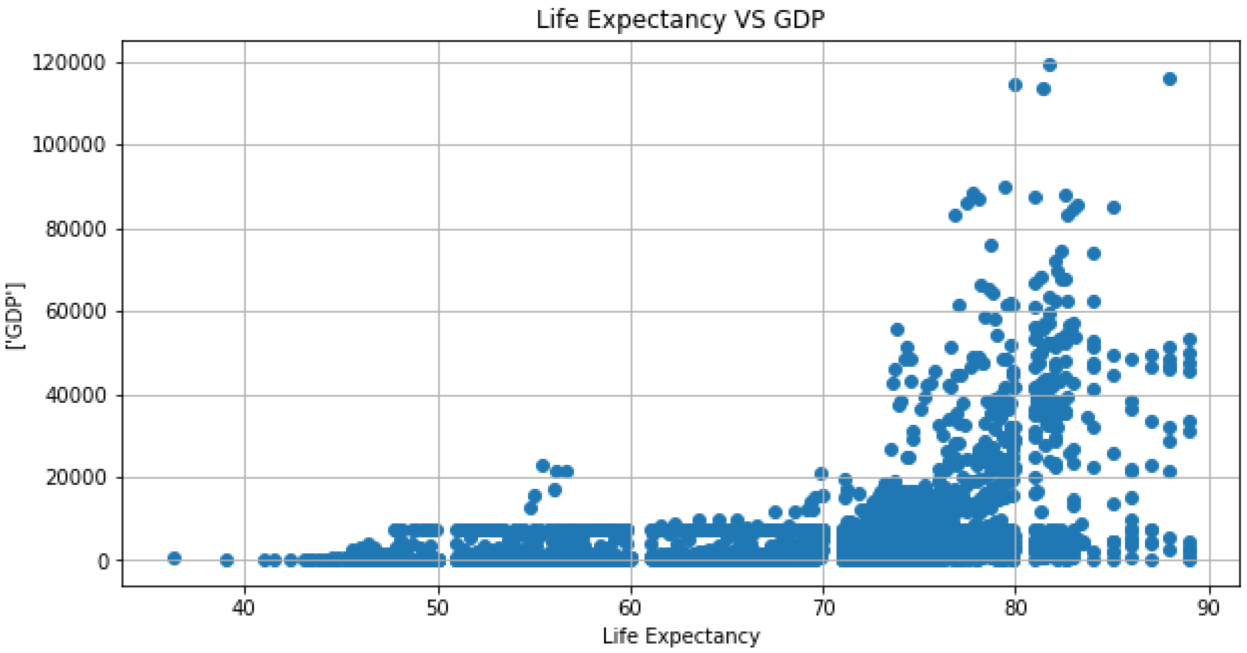
```
Correlation between Life Expectancy and Total expenditure:  0.20798062451867802
```



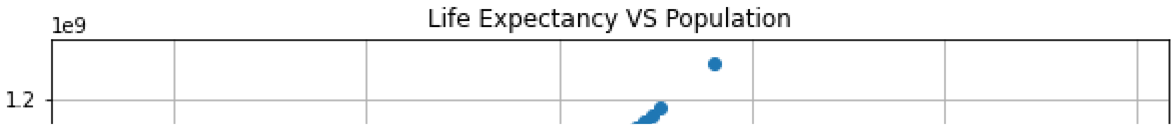
```
Correlation between Life Expectancy and Diphtheria :  0.47541838493660654
```



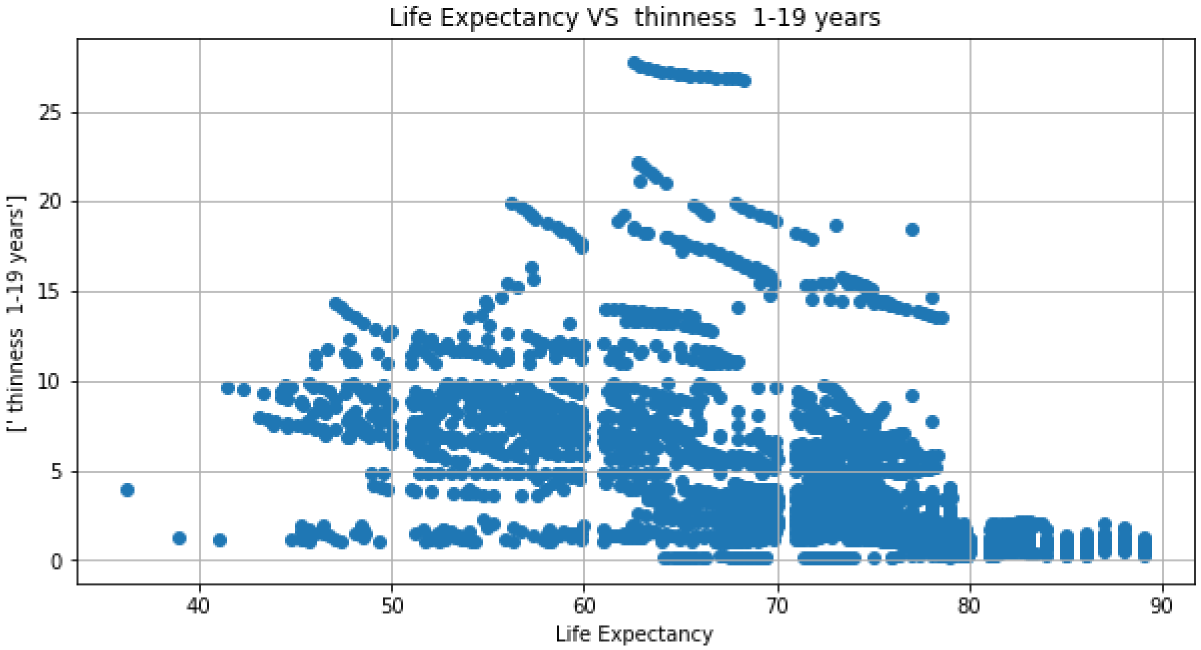
```
Correlation between Life Expectancy and  HIV/AIDS:  -0.556456816599713
```



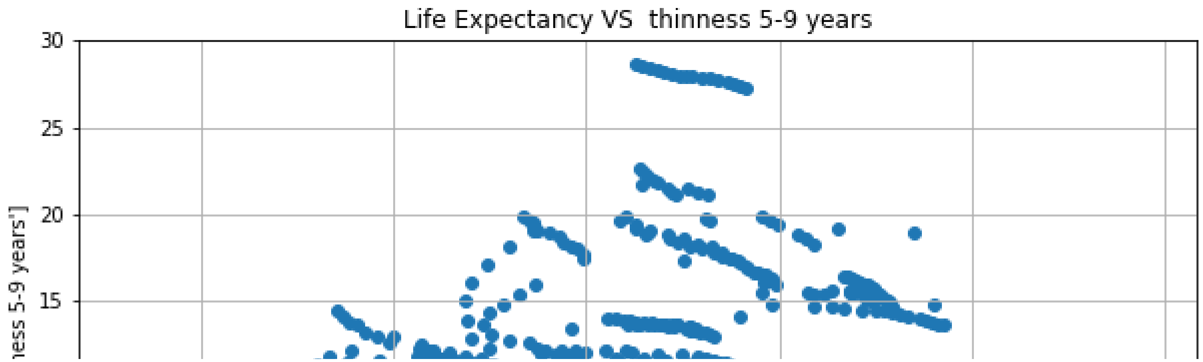
```
Correlation between Life Expectancy and GDP:  0.43049301854946415
```



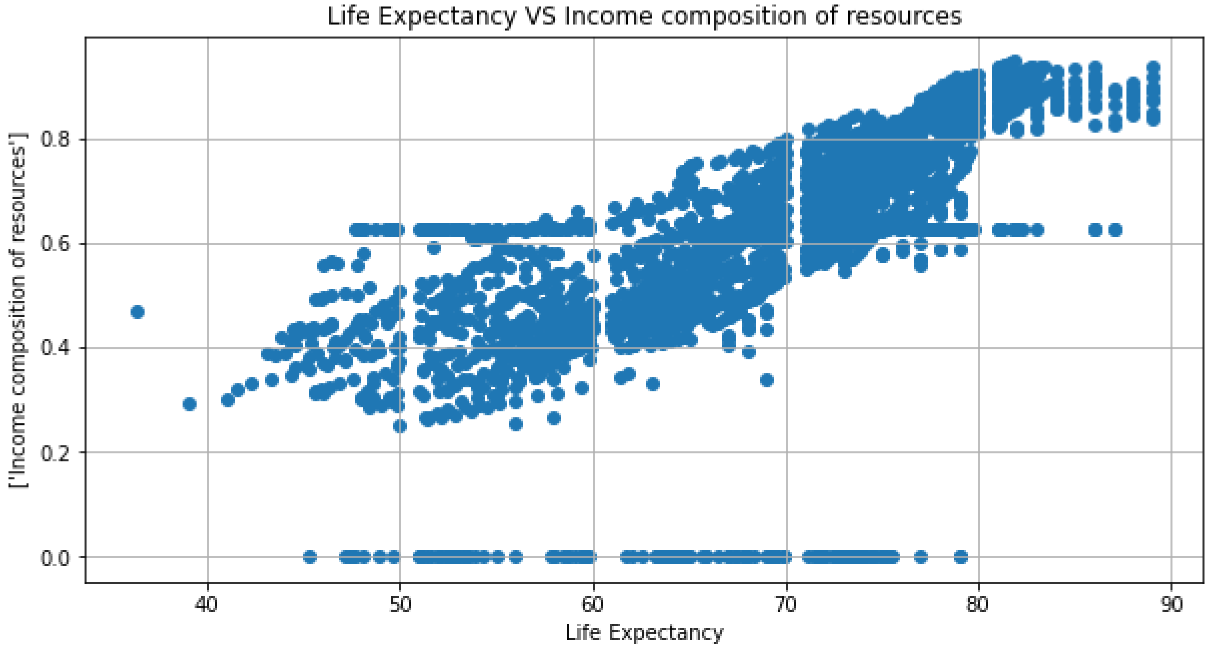
```
Correlation between Life Expectancy and Population:  -0.019637701509419594
```



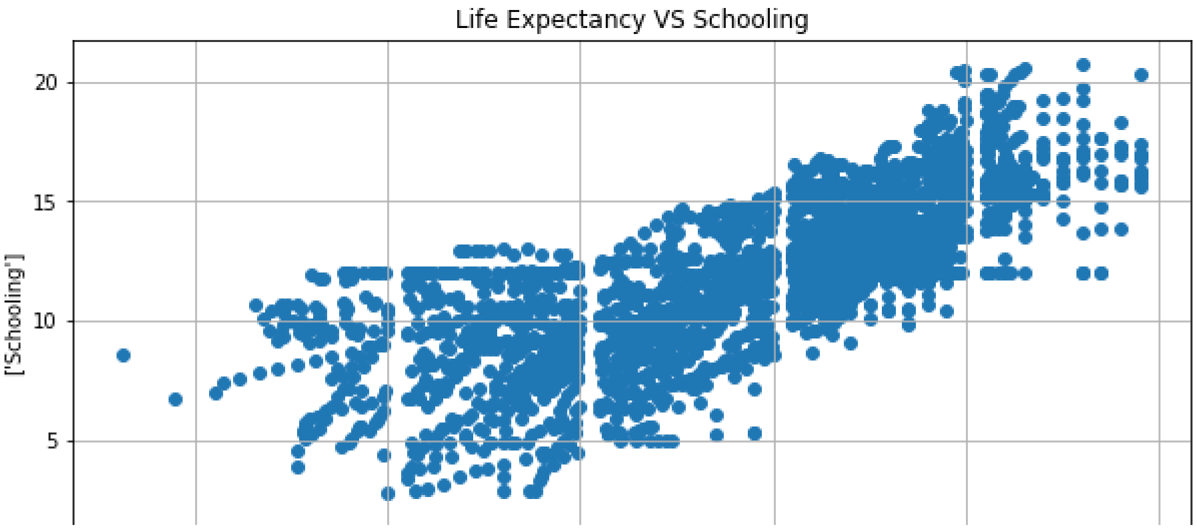
```
Correlation between Life Expectancy and  thinness  1-19 years:  -0.4721618794367624
```



```
Correlation between Life Expectancy and  thinness 5-9 years:  -0.4666292081443012
```



```
Correlation between Life Expectancy and Income composition of resources:  0.6924828049608566
```



```
Correlation between Life Expectancy and Schooling:  0.7150663398620059
```

- There is low correlation between life expectancy and healthcare expenditure thus increasing the total healthcare expenditure does not increase the life expectancy.
- There is a negative correlation between life expectancy and adult mortality rate. If adult mortality rate decrease, the life expectancy will be increase.
- There is low correlation between life expectancy and infant deaths.
- There is a high correlation between life expectancy and income composition of resources. The higher the income composition, the higher the life expectancy.
- There is also a high correlation between life expectancy and schooling. The higher the schooling, the higher the life expectancy.
- The normal BMI range is 18.5 until 24.9. According to life expectancy and BMI graph, most of people who have higher than normal BMI range have higher life expectancy than people who have normal BMI.

Multiple Linear Regression Model

In [13]:

```
for i in ds_num.columns:
        corr = np.corrcoef(ds['Life expectancy '], ds_num[i])
        if corr[0,1] >= 0.5 or corr[0,1] <= -0.5:
            print([i])
```

```
['Adult Mortality']
[' BMI ']
[' HIV/AIDS']
['Income composition of resources']
['Schooling']
```

In [14]:

```
ds_num.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 18 columns):
 #   Column                           Non-Null Count  Dtype  
---  ------                           --------------  -----  
 0   Adult Mortality                  2938 non-null   float64
 1   infant deaths                    2938 non-null   int64  
 2   Alcohol                          2938 non-null   float64
 3   percentage expenditure           2938 non-null   float64
 4   Hepatitis B                      2938 non-null   float64
 5   Measles                          2938 non-null   int64  
 6    BMI                             2938 non-null   float64
 7   under-five deaths                2938 non-null   int64  
 8   Polio                            2938 non-null   float64
 9   Total expenditure                2938 non-null   float64
 10  Diphtheria                       2938 non-null   float64
 11   HIV/AIDS                        2938 non-null   float64
 12  GDP                              2938 non-null   float64
 13  Population                       2938 non-null   float64
 14   thinness  1-19 years            2938 non-null   float64
 15   thinness 5-9 years              2938 non-null   float64
 16  Income composition of resources  2938 non-null   float64
 17  Schooling                        2938 non-null   float64
dtypes: float64(15), int64(3)
memory usage: 413.3 KB
```

In [15]:

```
X = ds_num.iloc[:,[0,6,11,16,17]].values
y = ds['Life expectancy '].values
```

In [16]:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

In [17]:

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train,y_train)
```

Out[17]:

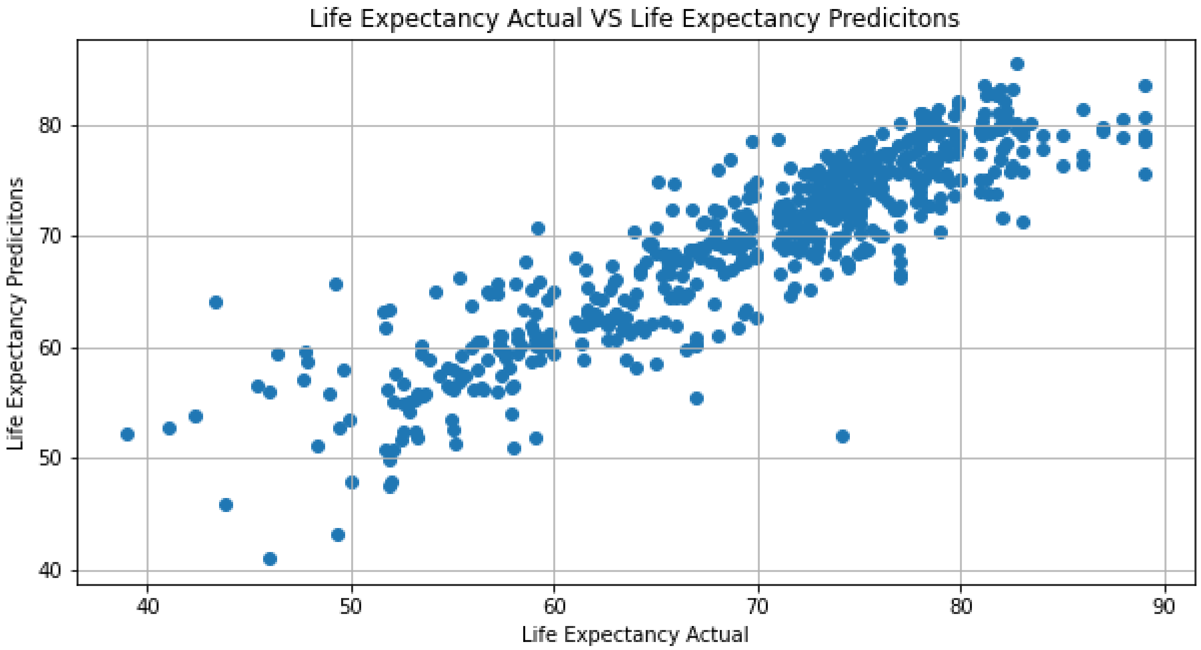
```
LinearRegression()
```

In [18]:

```
y_pred = regressor.predict(X_test)
```

In [19]:

```
plt.subplots(figsize = (10,5))
plt.scatter(y_test, y_pred)
plt.title('Life Expectancy Actual VS Life Expectancy Predicitons')
plt.xlabel('Life Expectancy Actual')
plt.ylabel('Life Expectancy Predicitons')
plt.grid()
plt.show()
```



In [20]:

```
corr = np.corrcoef(y_test,y_pred)
corr
```

Out[20]:

```
array([[1.        , 0.89185049],
       [0.89185049, 1.        ]])
```

The multiple linear regression model have high correlation value which is 0.89 which shows it can predict the Life Expectancy value with higher accurancy.

In [ ]: