LIFE EXPECTANCY PREDICTION

NAME : PIYUSHA PARDESHI

BATCH NO : PGA21

INSTITUTE : IMARTICUS`

CAPSTONE PROJECT 1

Problem Statement :

To predict life expectancy of a person using machine learning algorithms based on multiple factors.

OBJECTIVE AND SYNOPSIS

Description

Context

Although there have been lot of studies undertaken in the past on factors affecting life expectancy considering demographic variables, income composition and mortality rates. It was found that affect of immunization and human development index was not taken into account in the past. Also, some of the past research was done considering multiple linear regression based on data set of one year for all the countries. Hence, this gives motivation to resolve both the factors stated previously by formulating a regression model based on mixed effects model and multiple linear regression while considering data from a period of 2000 to 2015 for all the countries. Important immunization like Hepatitis B, Polio and Diphtheria will also be considered. In a nutshell, this study will focus on immunization factors, mortality factors, economic factors, social factors and other health related factors as well. Since the observations this dataset are based on different countries, it will be easier for a country to determine the predicting factor which is contributing to lower value of life expectancy. This will help in suggesting a country which area should be given importance in order to efficiently improve the life expectancy of its population.

Content

The project relies on accuracy of data. The Global Health Observatory (GHO) data repository under World Health Organization (WHO) keeps track of the health status as well as many other related factors for all countries The data-sets are made available to public for the purpose of health data analysis. The data-set related to life expectancy, health factors for 193 countries has been collected from the same WHO data repository website and its corresponding economic data was collected from United Nation website. Among all categories of health-related factors only those critical factors were chosen which are more representative. It has been observed that in the past 15 years , there has been a huge development in health sector resulting in improvement of human mortality rates especially in the developing nations in comparison to the past 30 years. Therefore, in this project we have considered data from year 2000-2015 for 193 countries for further analysis. The individual data files have been merged together into a single data-set. On initial visual inspection of the data showed some missing values. As the data-sets were from WHO, we found no evident errors. Missing data was handled in R software by using Missmap command. The result indicated that most of the missing data was for population, Hepatitis B and GDP. The missing data were from less known countries like Vanuatu, Tonga, Togo, Cabo Verde etc. Finding all data for these countries was difficult and hence, it was decided that we exclude these countries from the final model data-set. The final merged file(final dataset) consists of 22 Columns and 2938 rows which meant 20 predicting variables. All predicting variables was then divided into several broad categories:Immunization related factors, Mortality factors, Economical factors and Social factors.

Acknowledgements

The data was collected from WHO and United Nations website with the help of Deeksha Russell and Duan Wang.

Inspiration

The data-set aims to answer the following key questions:

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

Columns with description -

1) Country- Country

2) Year- Year

3) Status- Developed or Developing status

4) Life expectancy- Life Expectancy in age

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

6) infant deaths- Number of Infant Deaths per 1000 population

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

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

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

10) Measles- number of reported cases per 1000 population

11) BMI- Average Body Mass Index of entire population

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

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

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

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

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

17) GDP- Gross Domestic Product per capita (in USD)

18) Population- Population of the country

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

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

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

22) Schooling- Number of years of Schooling(years)

I HAVE CHOOSEN THIS TOPIC AS IT IS A REGRESSION REAL TIME PROBLEM WHICH INVOLVES MANY MEDICAL FACTORS AND BASIC METRICS WHICH CAN HELP IMPROVE ITS LIFE EXPECTANCY BY STUDYING THE REGION AND THE FACTORS AFFECTING THOSE REGIONSTHIS CAN BE USED IN VARIOUS MEDICAL AND NO MEDICAL FIELDS BOTH.

IMPORT LIBRARIES

In [1]:

```
# import numpy and panda libraries
import pandas as pd
import numpy as np
```

```
# scikit library for linear regression
import statsmodels.api as sm
from statsmodels.formula.api import ols # for anova test
import statsmodels.stats.api as stats
from sklearn.linear_model import LinearRegression,Ridge,Lasso
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor,AdaBoostRegressor,GradientBoostingRegressor
from xgboost import XGBRegressor
from sklearn.naive_bayes import GaussianNB
from sklearn import svm,neighbors,preprocessing

# scaling libraries
from sklearn.preprocessing import MinMaxScaler,StandardScaler

from sklearn.model_selection import train_test_split,cross_val_score,KFold,GridSearchCV
from datetime import date

# PCA dimension reduction library
from sklearn.decomposition import PCA

# visualisation
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib import pyplot
import pylab

from sklearn.metrics import classification_report,precision_recall_curve,recall_score,mean_squared_error,r2_score
from scipy.stats import skew
from sklearn import metrics
from sklearn import tree
from imblearn.over_sampling import SMOTE
import statsmodels.api as smapi

# feature selection
# RFE (recursive feature elimination)
from sklearn.feature_selection import RFE
from scipy import stats

import warnings
warnings.filterwarnings('ignore')
#pd.set_option('display.width',None)
#pd.set_option('display.max_rows',None)
#pd.set_option('display.max_columns',None)
```

- TO VERIFY WHICH ALGORITHM PERFORMS BETTER TO PREDICT LIFE EXPECTANCY IMPORTED LIBRARIES FOR FEW REGRESSION ALGORITHMS

READ DATA

In [2]:
```
path="C:/Users/Admin/Capstone Project 1/Life_Expectancy_Data_Regression.csv"
life_expectancy_data =pd.read_csv(path)
```

In [3]:
```
life_expectancy_data.head()
```

Out[3]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	Polio	Total expenditure	Diphtheria	HIV
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	1154	...	6.0	8.16	65.0	
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	492	...	58.0	8.18	62.0	
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	430	...	62.0	8.13	64.0	
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	2787	...	67.0	8.52	67.0	
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	3013	...	68.0	7.87	68.0	

5 rows × 22 columns

In [4]:
```
life_expectancy_data.tail()
```

Out[4]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	...	Polio	Total expenditure	Diphtheria	H
2933	Zimbabwe	2004	Developing	44.3	723.0	27	4.36	0.0	68.0	31	...	67.0	7.13	65.0	
2934	Zimbabwe	2003	Developing	44.5	715.0	26	4.06	0.0	7.0	998	...	7.0	6.52	68.0	

2935	Zimbabwe	2002	Developing	44.8	73.0	25	4.43	0.0	73.0	304	...	73.0	6.53	71.0
2936	Zimbabwe	2001	Developing	45.3	686.0	25	1.72	0.0	76.0	529	...	76.0	6.16	75.0
2937	Zimbabwe	2000	Developing	46.0	665.0	24	1.68	0.0	79.0	1483	...	78.0	7.10	78.0

5 rows × 22 columns

In [5]:
```
len(life_expectancy_data.columns)
```

Out[5]: 22

In [6]:
```
life_expectancy_data.shape
```

Out[6]: (2938, 22)

In [7]:
```
life_expectancy_data.columns
```

Out[7]:
```
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')
```

-THIS DATA HAS 2938 ROWS AND 22 COLUMNS

DISPLAY DATA INFORMATION

i .GETTING THE DATA INFORMATION

In [8]:
```
life_expectancy_data.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
```

ii . GETTING THE STATISTICAL INFORMATION

In [9]:
```
life_expectancy_data.describe()
```

Out[9]:

	Year	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths
count	2938.000000	2928.000000	2928.000000	2938.000000	2744.000000	2938.000000	2385.000000	2938.000000	2904.000000	2938.000000
mean	2007.518720	69.224932	164.796448	30.303948	4.602861	738.251295	80.940461	2419.592240	38.321247	42.035739
std	4.613841	9.523867	124.292079	117.926501	4.052413	1987.914858	25.070016	11467.272489	20.044034	160.445548
min	2000.000000	36.300000	1.000000	0.000000	0.010000	0.000000	1.000000	0.000000	1.000000	0.000000
25%	2004.000000	63.100000	74.000000	0.000000	0.877500	4.685343	77.000000	0.000000	19.300000	0.000000
50%	2008.000000	72.100000	144.000000	3.000000	3.755000	64.912906	92.000000	17.000000	43.500000	4.000000
75%	2012.000000	75.700000	228.000000	22.000000	7.702500	441.534144	97.000000	360.250000	56.200000	28.000000
max	2015.000000	89.000000	723.000000	1800.000000	17.870000	19479.911610	99.000000	212183.000000	87.300000	2500.000000

DATA CLEANING

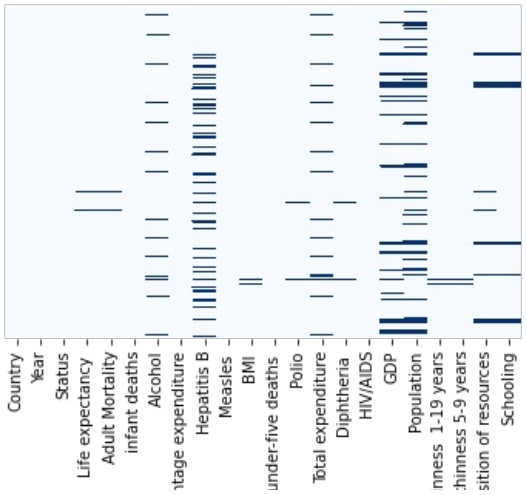
i.CHECK FOR MISSING VALUES

In [10]:
```
life_expectancy_data.isnull().sum()
```

Out[10]:
```
Country                            0
Year                               0
Status                             0
Life expectancy                   10
Adult Mortality                   10
infant deaths                      0
Alcohol                          194
percentage expenditure             0
Hepatitis B                      553
Measles                            0
 BMI                              34
under-five deaths                  0
Polio                             19
Total expenditure                226
Diphtheria                        19
 HIV/AIDS                          0
GDP                              448
Population                       652
 thinness  1-19 years             34
 thinness 5-9 years               34
Income composition of resources  167
Schooling                        163
dtype: int64
```

In [11]:
```
sns.heatmap(life_expectancy_data.isnull(),yticklabels=False,cbar=False,cmap='Blues')
```

Out[11]:
```
<AxesSubplot:>
```



percer – thi t Income compos

In [12]:
```
missing_value_columns = life_expectancy_data.isnull().sum() / life_expectancy_data.count()*100
```

In [13]:
```
round(missing_value_columns.sort_values(),2)
```

Out[13]:
```
Country                            0.00
Year                               0.00
Status                             0.00
infant deaths                      0.00
percentage expenditure             0.00
Measles                            0.00
 HIV/AIDS                          0.00
under-five deaths                  0.00
Life expectancy                    0.34
Adult Mortality                    0.34
Polio                              0.65
Diphtheria                         0.65
 thinness 5-9 years                1.17
 thinness  1-19 years              1.17
 BMI                               1.17
Schooling                          5.87
Income composition of resources    6.03
Alcohol                            7.07
Total expenditure                  8.33
GDP                               17.99
Hepatitis B                       23.19
Population                        28.52
dtype: float64
```

- GDP ,Hepatitis_B ,Population HAVE MORE NUMBER OF MISSING VALUES IN THE DATA WHICH NEEDS TO BE HANDLED REST COLUMNS HAVE NEGLIGIBLE MISSING VALUES OBSERVED.

ii. RENAME COLUMN NAMES

In [14]:
```
life_expectancy_data.columns
```

Out[14]:
```
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 [15]:
```
life_expectancy_data.rename(columns= {'Life expectancy ':'Life_Expectancy','Adult Mortality':'Adult_Mortality','i
```

In [16]:
```
life_expectancy_data.columns
```

Out[16]:
```
Index(['Country', 'Year', 'Status', 'Life_Expectancy', 'Adult_Mortality',
       'Infant_Deaths', 'Alcohol', 'Percentage_Expenditure', 'Hepatitis_B',
       'Measles', 'BMI', 'Under_5_Deaths', 'Polio', 'Total_Expenditure',
       'Diphtheria', 'HIV/AIDS', 'GDP', 'Population', 'Thinness_1to19_Years',
       'Thinness_5to9_Years', 'Income_Composition_Of_Resources', 'Schooling'],
      dtype='object')
```

iii. HANDLING ALL MISSING VALUES

In [17]:
```
missing_data_cols = ['Life_Expectancy','Adult_Mortality','Polio','Diphtheria','Thinness_5to9_Years','Thinness_1to
```

In [18]:
```
def handle_missing_values(data,cols):
```

```
    for c in cols:
        data[c]=data[c].fillna(life_expectancy_data[c].median())
    return data
```

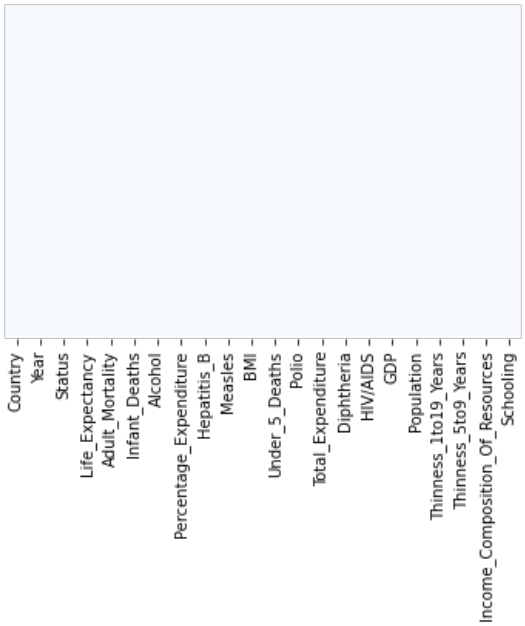
In [19]:
```
life_expectancy_data=handle_missing_values(life_expectancy_data,missing_data_cols)
```

In [20]:
```
life_expectancy_data.isnull().sum()
```

Out[20]:
```
Country                          0
Year                             0
Status                           0
Life_Expectancy                  0
Adult_Mortality                  0
Infant_Deaths                    0
Alcohol                          0
Percentage_Expenditure           0
Hepatitis_B                      0
Measles                          0
BMI                              0
Under_5_Deaths                   0
Polio                            0
Total_Expenditure                0
Diphtheria                       0
HIV/AIDS                         0
GDP                              0
Population                       0
Thinness_1to19_Years             0
Thinness_5to9_Years              0
Income_Composition_Of_Resources  0
Schooling                        0
dtype: int64
```

In [21]:
```
sns.heatmap(life_expectancy_data.isnull(),yticklabels=False,cbar=False,cmap='Blues')
```

Out[21]:
```
<AxesSubplot:>
```



- HANDLED ALL MISSING DATA IN ALL 14 COLUMNS

DATA PREPROCESSING

i. CHECK DATATYPES

In [22]:
```
life_expectancy_data.dtypes
```

Out[22]:
```
Country                          object
```

```
Year                                int64
Status                             object
Life_Expectancy                   float64
Adult_Mortality                   float64
Infant_Deaths                       int64
Alcohol                           float64
Percentage_Expenditure            float64
Hepatitis_B                       float64
Measles                             int64
BMI                               float64
Under_5_Deaths                      int64
Polio                             float64
Total_Expenditure                 float64
Diphtheria                        float64
HIV/AIDS                          float64
GDP                               float64
Population                        float64
Thinness_1to19_Years              float64
Thinness_5to9_Years               float64
Income_Composition_Of_Resources   float64
Schooling                         float64
dtype: object
```

- ALL DATA TYPES ARE IN PLACE NO NEED TO CONVERT THE DATA TYPES INTO ANOTHER FORM

- THIS DATA HAS TWO CATEGORICAL COLUMNS AND 20 NUMERICAL COLUMNS

ii .SPLIT DATA INTO NUMERIC AND CATEGORIC

In [23]:
```
def split_cols(data):
    nc=data.select_dtypes(exclude='object').columns.values
    fc=data.select_dtypes(include='object').columns.values

    return(nc,fc)
```

In [24]:
```
numeric_cols,factor_cols=split_cols(life_expectancy_data)
```

In [25]:
```
numeric_cols
```

Out[25]:
```
array(['Year', 'Life_Expectancy', 'Adult_Mortality', 'Infant_Deaths',
       'Alcohol', 'Percentage_Expenditure', 'Hepatitis_B', 'Measles',
       'BMI', 'Under_5_Deaths', 'Polio', 'Total_Expenditure',
       'Diphtheria', 'HIV/AIDS', 'GDP', 'Population',
       'Thinness_1to19_Years', 'Thinness_5to9_Years',
       'Income_Composition_Of_Resources', 'Schooling'], dtype=object)
```

In [26]:
```
factor_cols
```

Out[26]:
```
array(['Country', 'Status'], dtype=object)
```

iii. CHECK FOR ZEROS

In [27]:
```
life_expectancy_data[numeric_cols][life_expectancy_data[numeric_cols]==0].count()
```

Out[27]:
```
Year                                0
Life_Expectancy                     0
Adult_Mortality                     0
Infant_Deaths                     848
Alcohol                             0
Percentage_Expenditure            611
Hepatitis_B                         0
Measles                           983
BMI                                 0
Under_5_Deaths                    785
Polio                               0
Total_Expenditure                   0
```

```
Diphtheria                          0
HIV/AIDS                            0
GDP                                 0
Population                          0
Thinness_1to19_Years                0
Thinness_5to9_Years                 0
Income_Composition_Of_Resources    130
Schooling                           28
dtype: int64
```

In [28]:
```
def calculate_zeroth_percentile(data,cols):
    for c in cols:
        percentage = data[c][data[c]==0].count()/len(data[cols])*100
        print(c,":",round(percentage,2))
```

In [29]:
```
calculate_zeroth_percentile(life_expectancy_data,numeric_cols)
```

```
Year : 0.0
Life_Expectancy : 0.0
Adult_Mortality : 0.0
Infant_Deaths : 28.86
Alcohol : 0.0
Percentage_Expenditure : 20.8
Hepatitis_B : 0.0
Measles : 33.46
BMI : 0.0
Under_5_Deaths : 26.72
Polio : 0.0
Total_Expenditure : 0.0
Diphtheria : 0.0
HIV/AIDS : 0.0
GDP : 0.0
Population : 0.0
Thinness_1to19_Years : 0.0
Thinness_5to9_Years : 0.0
Income_Composition_Of_Resources : 4.42
Schooling : 0.95
```

- We can observe that there are 28.86 % of no infant deaths,20.80 % of no percent expenditure on health,33.46 % have no measles,26.72 % no under 5 deaths,4.42 % of no income composition of resources and 0.95 % have done no schooling.

iv. PRINT UNIQUE VALUES OF DATA WITH DATA TYPES

In [30]:
```
def check_unique_values(data,cols):
    for c in cols:
        print("Column name:",c,data[c].dtypes)
        print(pd.unique(data[c]))
        print("length:")
        print(len(pd.unique(data[c])))
        print("-----------------------------------------------")
```

In [31]:
```
check_unique_values(life_expectancy_data,factor_cols)
```

```
Column name: Country object
['Afghanistan' 'Albania' 'Algeria' 'Angola' 'Antigua and Barbuda'
 'Argentina' 'Armenia' 'Australia' 'Austria' 'Azerbaijan' 'Bahamas'
 'Bahrain' 'Bangladesh' 'Barbados' 'Belarus' 'Belgium' 'Belize' 'Benin'
 'Bhutan' 'Bolivia (Plurinational State of)' 'Bosnia and Herzegovina'
 'Botswana' 'Brazil' 'Brunei Darussalam' 'Bulgaria' 'Burkina Faso'
 'Burundi' "Côte d'Ivoire" 'Cabo Verde' 'Cambodia' 'Cameroon' 'Canada'
 'Central African Republic' 'Chad' 'Chile' 'China' 'Colombia' 'Comoros'
 'Congo' 'Cook Islands' 'Costa Rica' 'Croatia' 'Cuba' 'Cyprus' 'Czechia'
 "Democratic People's Republic of Korea"
 'Democratic Republic of the Congo' 'Denmark' 'Djibouti' 'Dominica'
 'Dominican Republic' 'Ecuador' 'Egypt' 'El Salvador' 'Equatorial Guinea'
 'Eritrea' 'Estonia' 'Ethiopia' 'Fiji' 'Finland' 'France' 'Gabon' 'Gambia'
 'Georgia' 'Germany' 'Ghana' 'Greece' 'Grenada' 'Guatemala' 'Guinea'
 'Guinea-Bissau' 'Guyana' 'Haiti' 'Honduras' 'Hungary' 'Iceland' 'India'
 'Indonesia' 'Iran (Islamic Republic of)' 'Iraq' 'Ireland' 'Israel'
 'Italy' 'Jamaica' 'Japan' 'Jordan' 'Kazakhstan' 'Kenya' 'Kiribati'
 'Kuwait' 'Kyrgyzstan' "Lao People's Democratic Republic" 'Latvia'
 'Lebanon' 'Lesotho' 'Liberia' 'Libya' 'Lithuania' 'Luxembourg'
```

```
 'Madagascar' 'Malawi' 'Malaysia' 'Maldives' 'Mali' 'Malta'
 'Marshall Islands' 'Mauritania' 'Mauritius' 'Mexico'
 'Micronesia (Federated States of)' 'Monaco' 'Mongolia' 'Montenegro'
 'Morocco' 'Mozambique' 'Myanmar' 'Namibia' 'Nauru' 'Nepal' 'Netherlands'
 'New Zealand' 'Nicaragua' 'Niger' 'Nigeria' 'Niue' 'Norway' 'Oman'
 'Pakistan' 'Palau' 'Panama' 'Papua New Guinea' 'Paraguay' 'Peru'
 'Philippines' 'Poland' 'Portugal' 'Qatar' 'Republic of Korea'
 'Republic of Moldova' 'Romania' 'Russian Federation' 'Rwanda'
 'Saint Kitts and Nevis' 'Saint Lucia' 'Saint Vincent and the Grenadines'
 'Samoa' 'San Marino' 'Sao Tome and Principe' 'Saudi Arabia' 'Senegal'
 'Serbia' 'Seychelles' 'Sierra Leone' 'Singapore' 'Slovakia' 'Slovenia'
 'Solomon Islands' 'Somalia' 'South Africa' 'South Sudan' 'Spain'
 'Sri Lanka' 'Sudan' 'Suriname' 'Swaziland' 'Sweden' 'Switzerland'
 'Syrian Arab Republic' 'Tajikistan' 'Thailand'
 'The former Yugoslav republic of Macedonia' 'Timor-Leste' 'Togo' 'Tonga'
 'Trinidad and Tobago' 'Tunisia' 'Turkey' 'Turkmenistan' 'Tuvalu' 'Uganda'
 'Ukraine' 'United Arab Emirates'
 'United Kingdom of Great Britain and Northern Ireland'
 'United Republic of Tanzania' 'United States of America' 'Uruguay'
 'Uzbekistan' 'Vanuatu' 'Venezuela (Bolivarian Republic of)' 'Viet Nam'
 'Yemen' 'Zambia' 'Zimbabwe']
length:
193
------------------------------------------------
Column name: Status object
['Developing' 'Developed']
length:
2
------------------------------------------------
```

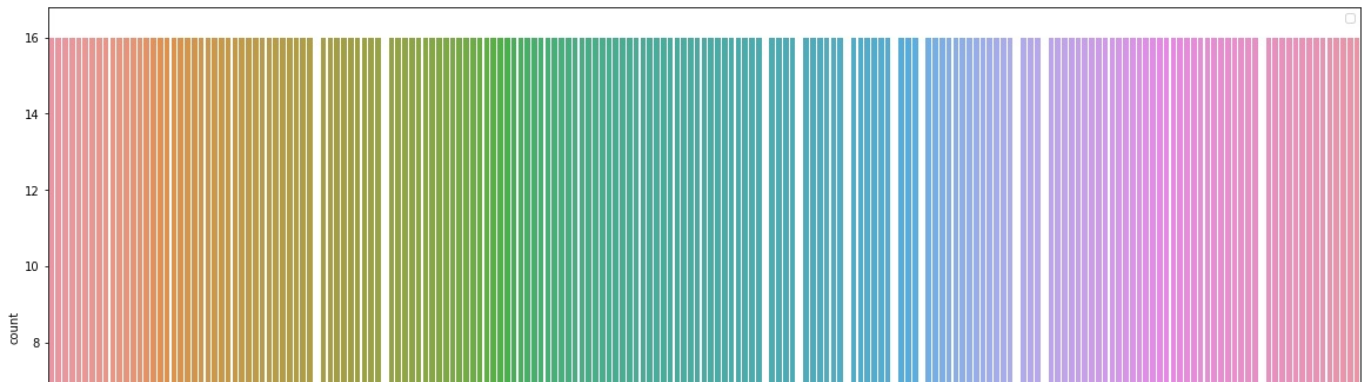
v. CHECK FOR SINGULARITY AND DATA IMBALANCE IN DATA

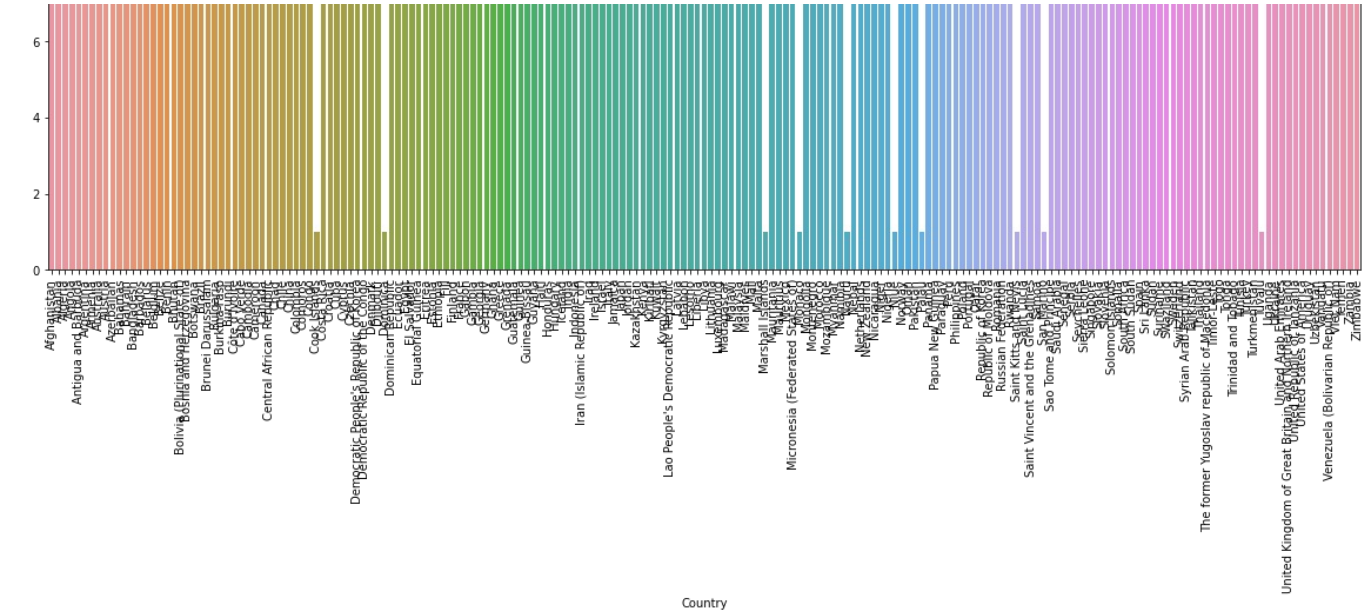
In [32]:
```
def checksingularity(data,factor_cols):
    for c in factor_cols:
        print("Column name :",c)
        counts = (data[c].value_counts()/len(data[c])*100)
        print("\n",round(counts,2))
        plt.figure(figsize=(20,10))
        sns.countplot(x=data[c],data = data)
        plt.xticks(rotation=90)
        plt.legend()
        plt.show()
        print("-------------------------------------------------")
```

In [33]:
```
checksingularity(life_expectancy_data,factor_cols)
```

```
Column name : Country

 Sri Lanka           0.54
Belgium             0.54
India               0.54
Guatemala           0.54
Gabon               0.54
                    ...
Niue                0.03
Marshall Islands    0.03
Dominica            0.03
Tuvalu              0.03
Cook Islands        0.03
Name: Country, Length: 193, dtype: float64
```

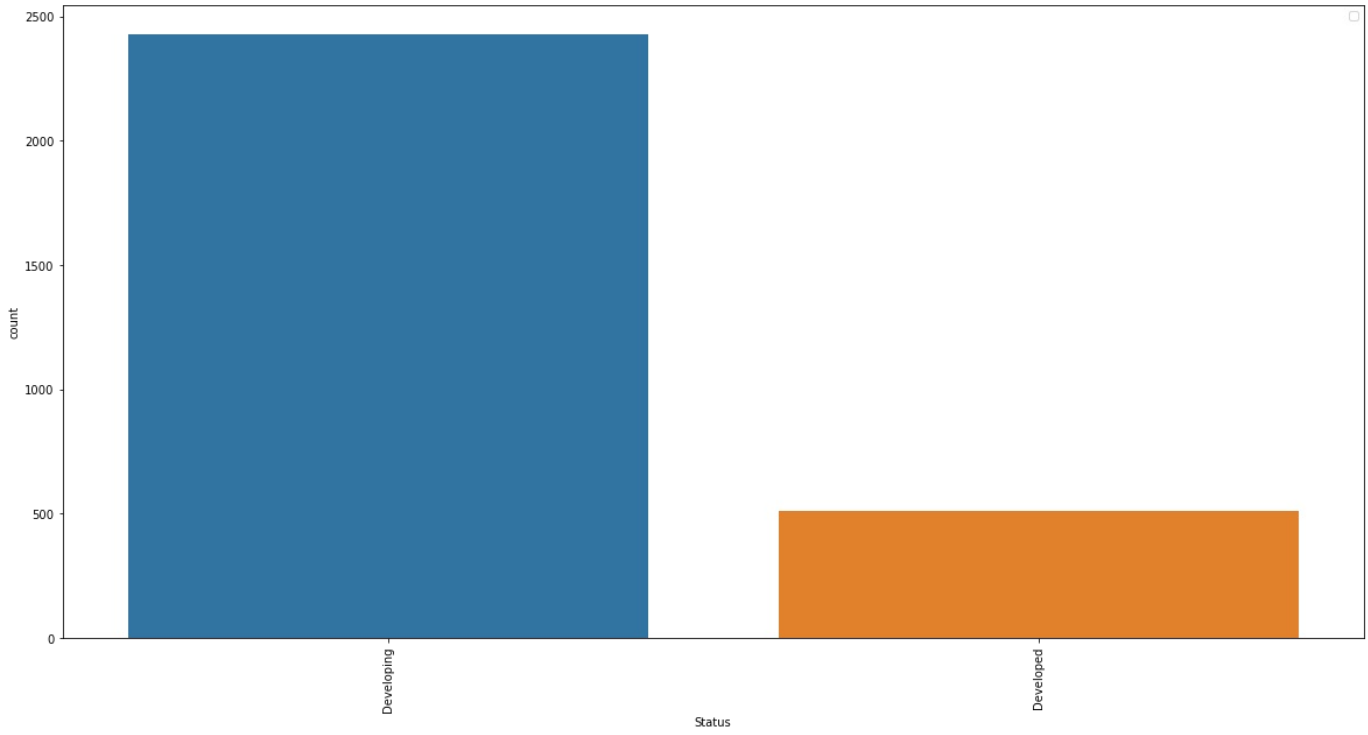
```
No handles with labels found to put in legend.
```






```
No handles with labels found to put in legend.
```

```
--------------------------------------------------
Column name : Status

 Developing    82.57
Developed     17.43
Name: Status, dtype: float64
```




```
--------------------------------------------------
```

-THERE ARE FEW COUNTRIES HAVING ONLY CONTRIBUTION OF 0.034 % WE WILL DELETE THOSE COUNTRIES AND THERE ARE ONLY 17.42 % OF DEVELOPED COUNTRIES LESSER COMPARED TO DEVELOPING COUNTRIES HAVING 82.57 % DATA

In [34]:
```
life_expectancy_data.Country.nunique()
```

Out[34]: 193

FEATURE ENGINEERING

vi. CREATING NEW COLUMN FROM EXISTING DATA

In [35]:
```
life_expectancy_data.Country=life_expectancy_data.Country.replace("Côte d'Ivoire",'Côte dIvoire')
life_expectancy_data.Country=life_expectancy_data.Country.replace("Democratic People's Republic of Korea",'Democr
life_expectancy_data.Country=life_expectancy_data.Country.replace("Lao People's Democratic Republic",'Lao Peoples
```

In [36]:
```
def group_countries_into_regions(data,col):
    count = 0
    for c in data[col]:
        if (c in (['Afghanistan','Bangladesh','Kiribati','Nepal','Lao Peoples Democratic Republic','Nauru','Syria
            data.at[count,"Region"] = 'Asia'
        elif (c in (['Algeria','Bahrain','Egypt','Iran (Islamic Republic of)','Iraq','Israel','Jordan','Kuwait','
            data.at[count,"Region"] = 'Western Asia and North Africa'
        elif (c in (['Angola','Benin','Cabo Verde','Burkina Faso','Côte dIvoire','Democratic Republic of the Cong
            data.at[count,"Region"] = 'Africa'
        elif (c in (['Armenia','Azerbaijan','Belarus','Georgia','Kazakhstan','Kyrgyzstan','Republic of Moldova','
            data.at[count,"Region"] = 'the European and Central Asian Region'
        elif (c in (['Albania','Andorra','Russian Federation','Austria','United Kingdom of Great Britain and Nort
            data.at[count,"Region"] = 'the European Region'
        elif (c in (['Australia','Canada','Cook Islands','Fiji','Micronesia (Federated States of)','Niue','New Ze
            data.at[count,"Region"] = 'North America and Oceania'
        elif (c in (['El Salvador','Saint Lucia','Guatemala','Haiti','Saint Vincent and the Grenadines','Nicaragu
            data.at[count,"Region"] = 'Latin America'
        count =count+1
    return data
```

In [37]:
```
life_expectancy_data= group_countries_into_regions(life_expectancy_data,"Country")
```

In [38]:
```
life_expectancy_data[life_expectancy_data.Region.isna()==True]
```

Out[38]:

Country	Year	Status	Life_Expectancy	Adult_Mortality	Infant_Deaths	Alcohol	Percentage_Expenditure	Hepatitis_B	Measles	...	Total_Expe

0 rows × 23 columns

In [39]:
```
pd.unique(life_expectancy_data.Region)
```

Out[39]:
```
array(['Asia', 'the European Region', 'Western Asia and North Africa',
       'Africa', 'Latin America', 'the European and Central Asian Region',
       'North America and Oceania'], dtype=object)
```

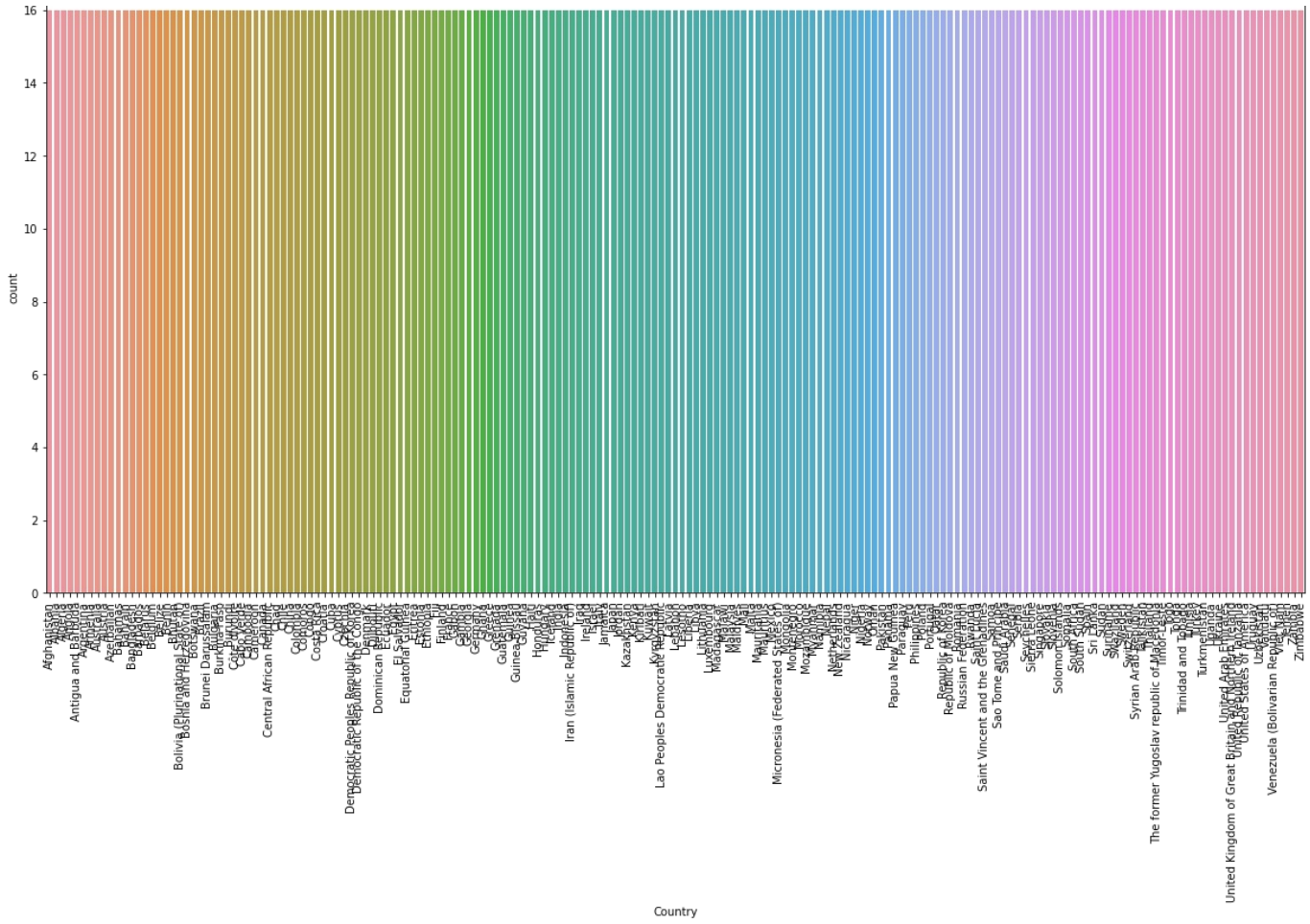
In [40]:
```
life_expectancy_data.shape
```

Out[40]:
```
(2938, 23)
```

DROP DATA IN COLUMN

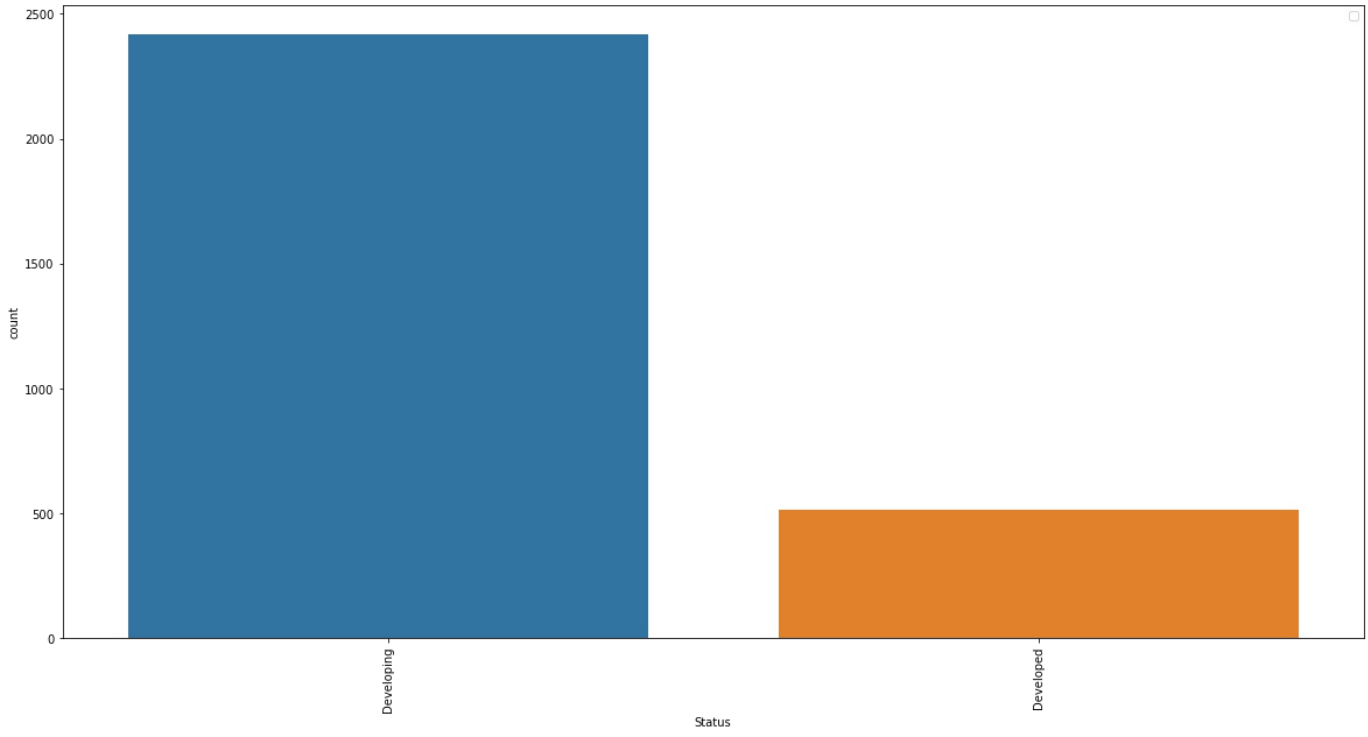
In [41]:
```
drop_countries = ['Nauru', 'Monaco','Saint Kitts and Nevis','Marshall Islands','Tuvalu','Palau','San Marino','Dom
life_expectancy_data= life_expectancy_data[~life_expectancy_data.Country.isin(drop_countries)]
checksingularity(life_expectancy_data,factor_cols)
```

```
Column name : Country

 Sri Lanka            0.55
Samoa                 0.55
Guatemala             0.55
Gabon                 0.55
Uruguay               0.55
                      ...
Azerbaijan            0.55
Afghanistan           0.55
Republic of Korea     0.55
Barbados              0.55
Malaysia              0.55
Name: Country, Length: 183, dtype: float64
No handles with labels found to put in legend.
```




```
No handles with labels found to put in legend.
```

```
--------------------------------------------------
Column name : Status

 Developing    82.51
Developed     17.49
Name: Status, dtype: float64
```




```
--------------------------------------------------
```

In [42]:
```
life_expectancy_data.shape
```

Out[42]: (2928, 23)

- WE AGAIN NEED TO SPLIT COLUMNS BEFORE EDA

In [43]:
```
numeric_cols,factor_cols=split_cols(life_expectancy_data)
```

VISUALISATIONS (EXPLORATORY DATA ANALYSIS)

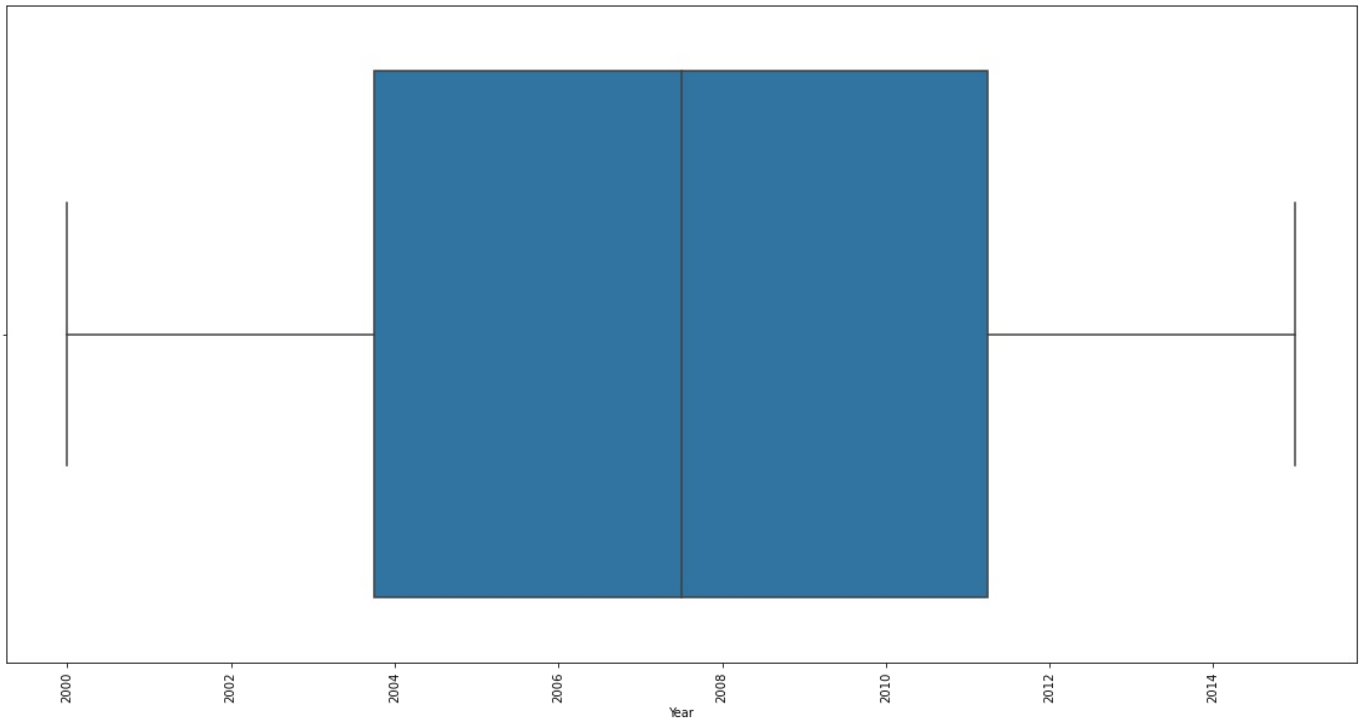
I. UNIVARIATE ANALYSIS - ANALYSING A SINGLE VARIABLE

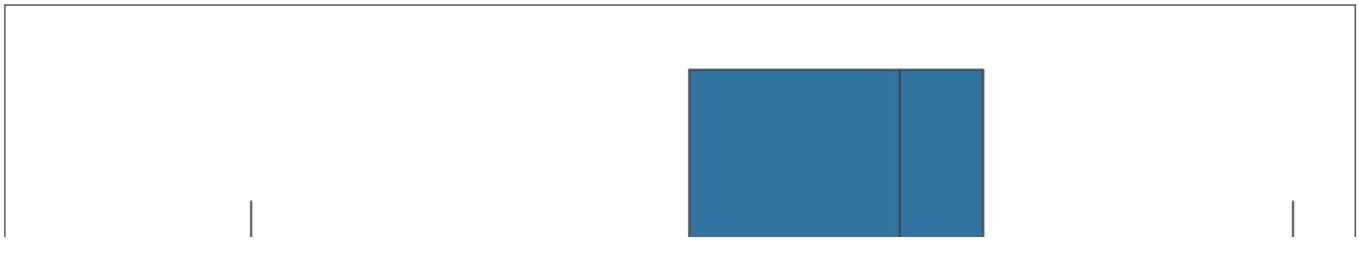
In [44]:
```
def plot_univariate_graphs(data,cols,type_graph):
    for c in cols:
        plt.figure(figsize = (20,10))
        if type_graph == 'h':
            sns.histplot(x=c,data = data,kde=True,label = "Count of "+c,stat = 'count')
            plt.legend(loc = 'best')
            plt.style.use('ggplot')
            plt.title(c+' Distribution')
        if type_graph == 'bo':
            sns.boxplot(x=c,data = data)
        if type_graph == 'ba':
            sns.barplot(x=c,data = data)
        elif type_graph == 'c':
            sns.countplot(x = c,data = data)
        plt.xticks(rotation=90)
        plt.show()
```

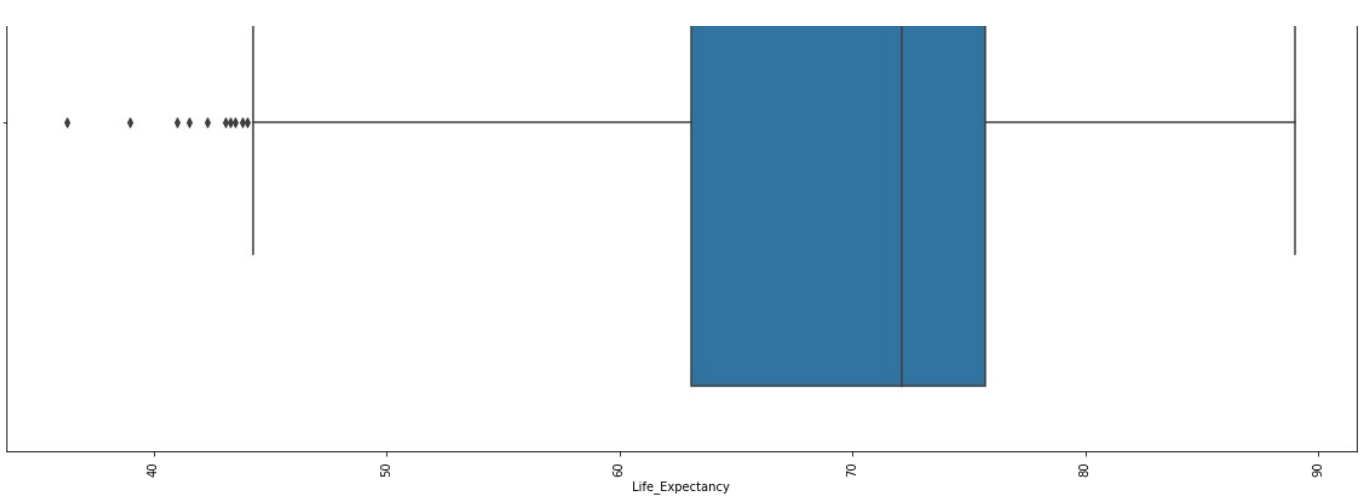
* NUMERIC COLUMNS UNIVARIATE ANALYSIS

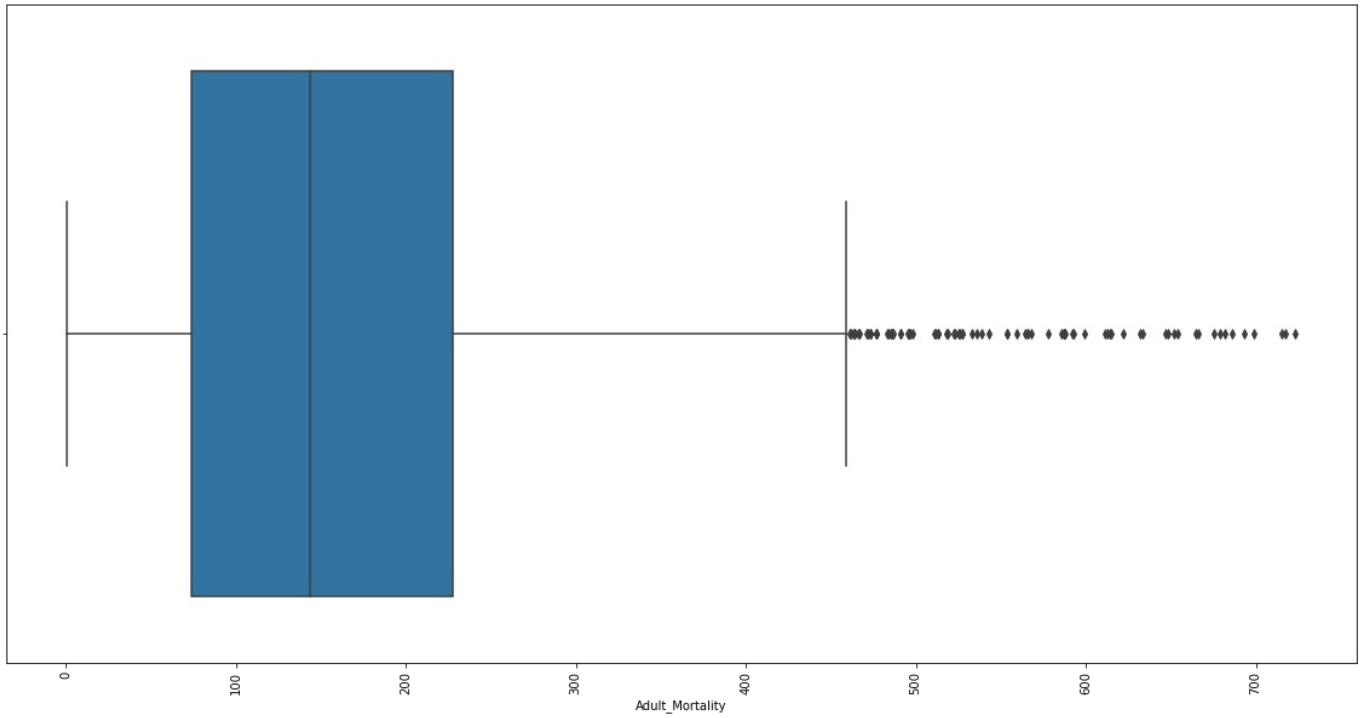
i.BOXPLOT

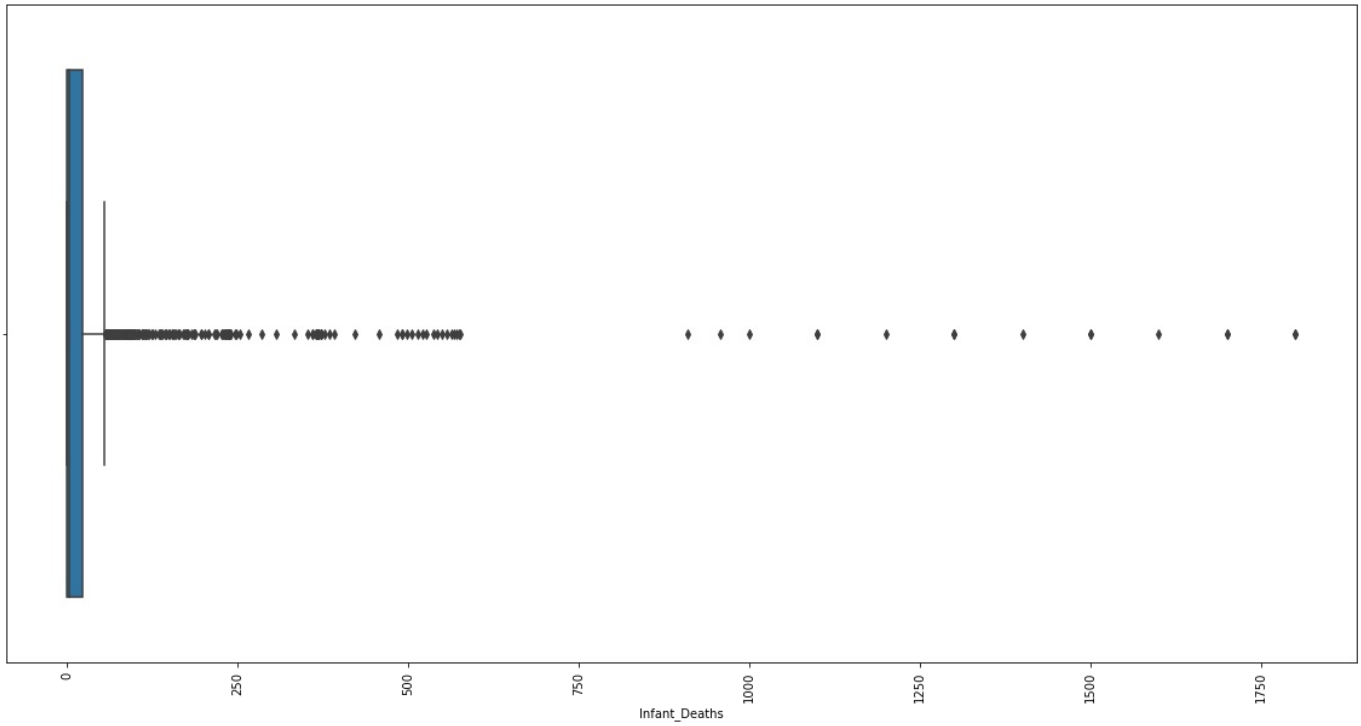
In [45]:
```
plot_univariate_graphs(life_expectancy_data,numeric_cols,'bo')
```

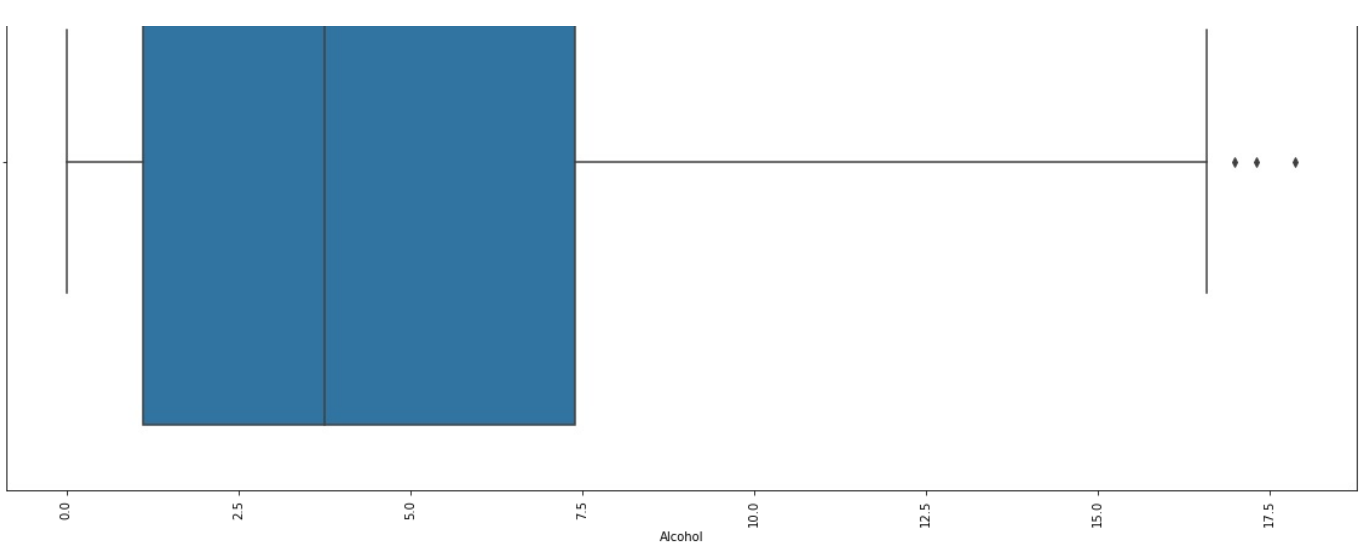


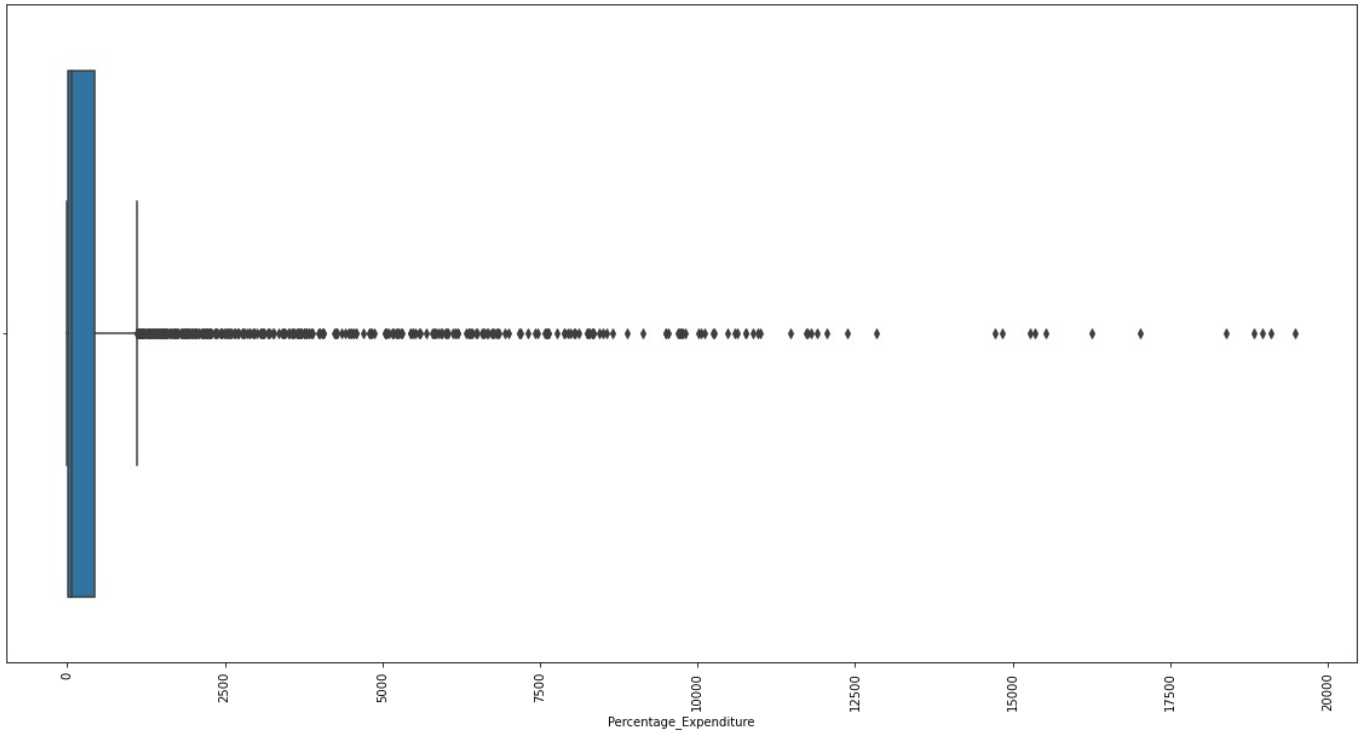


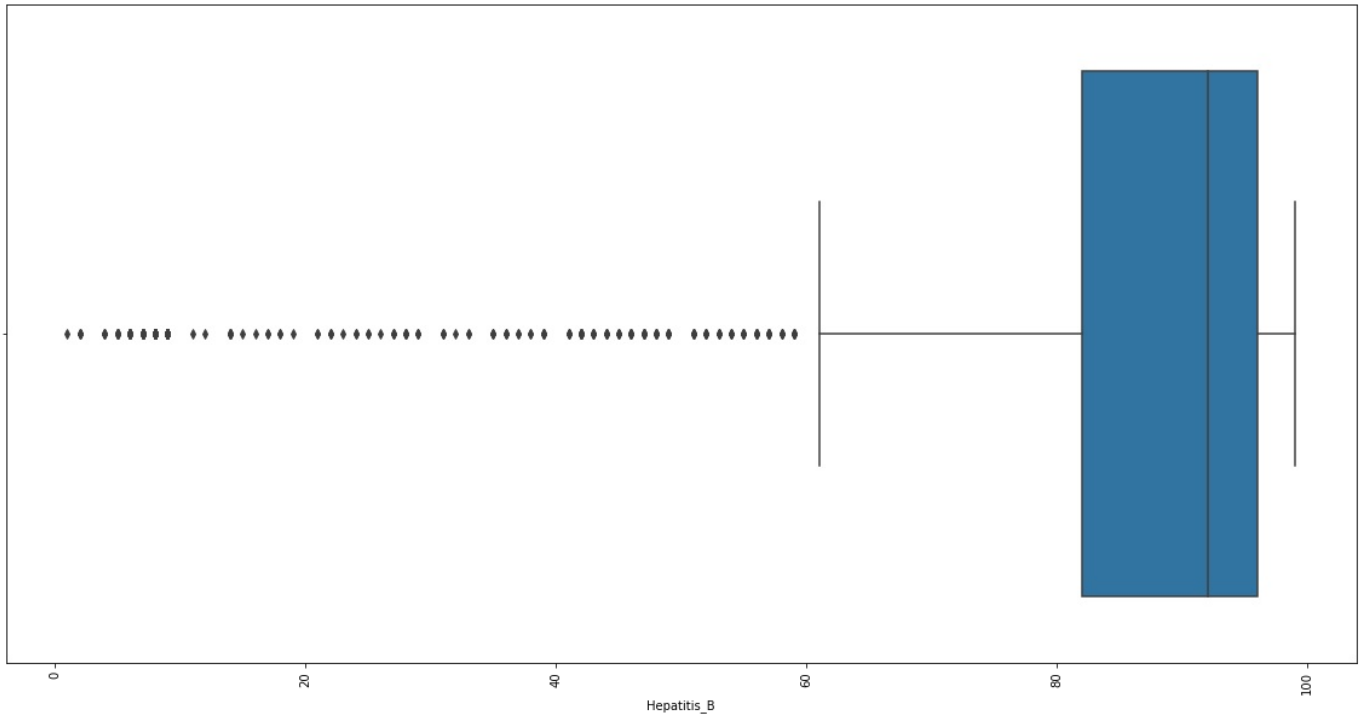

40
50
60
70
80
90
Life_Expectancy

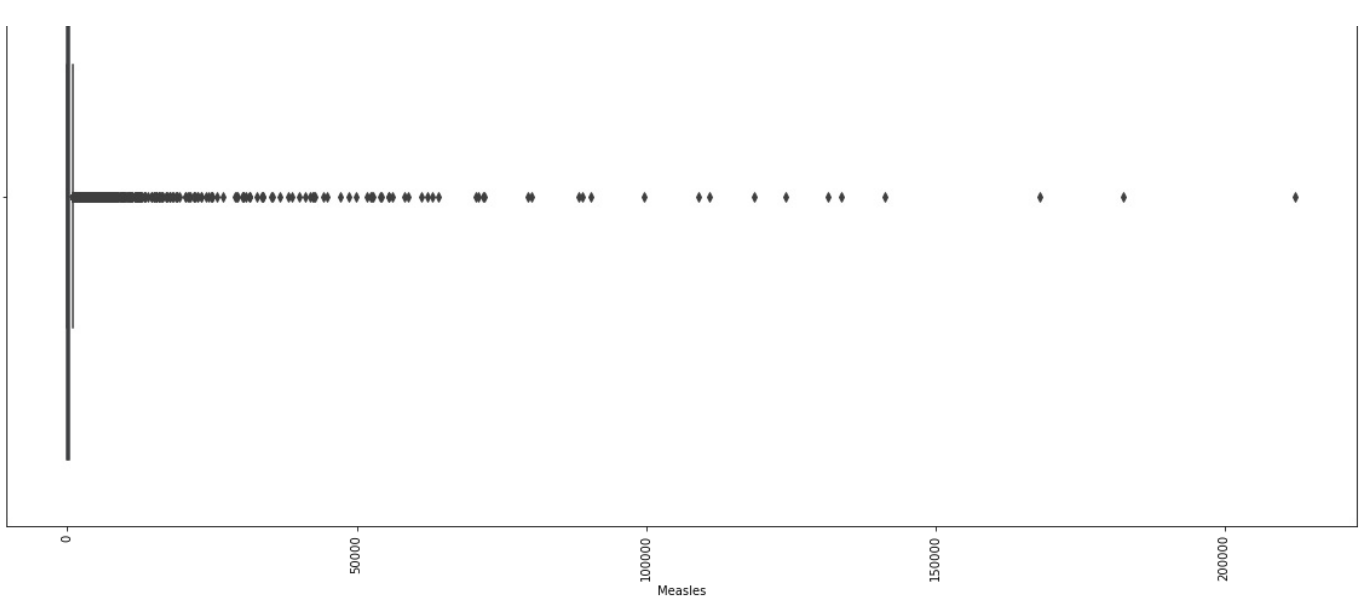

0
100
200
300
400
500
600
700
Adult_Mortality

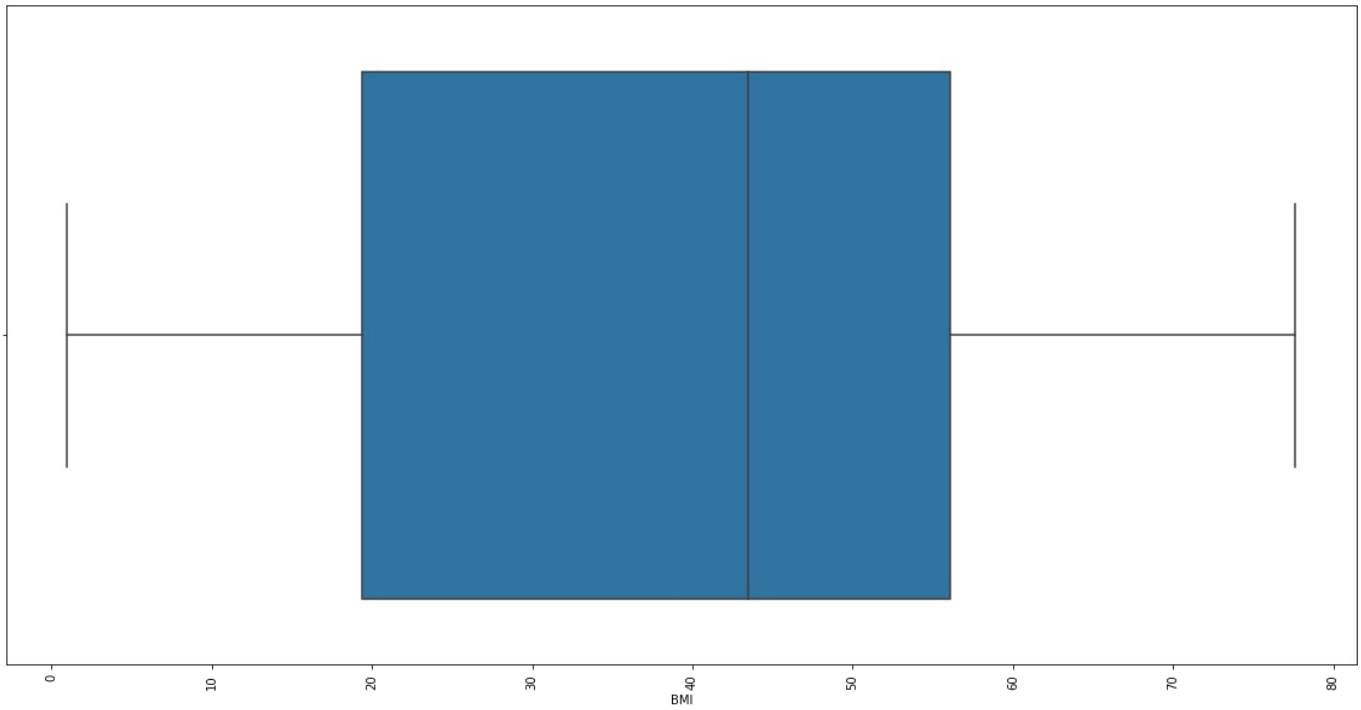

0
250
500
750
1000
1250
1500
1750
Infant_Deaths

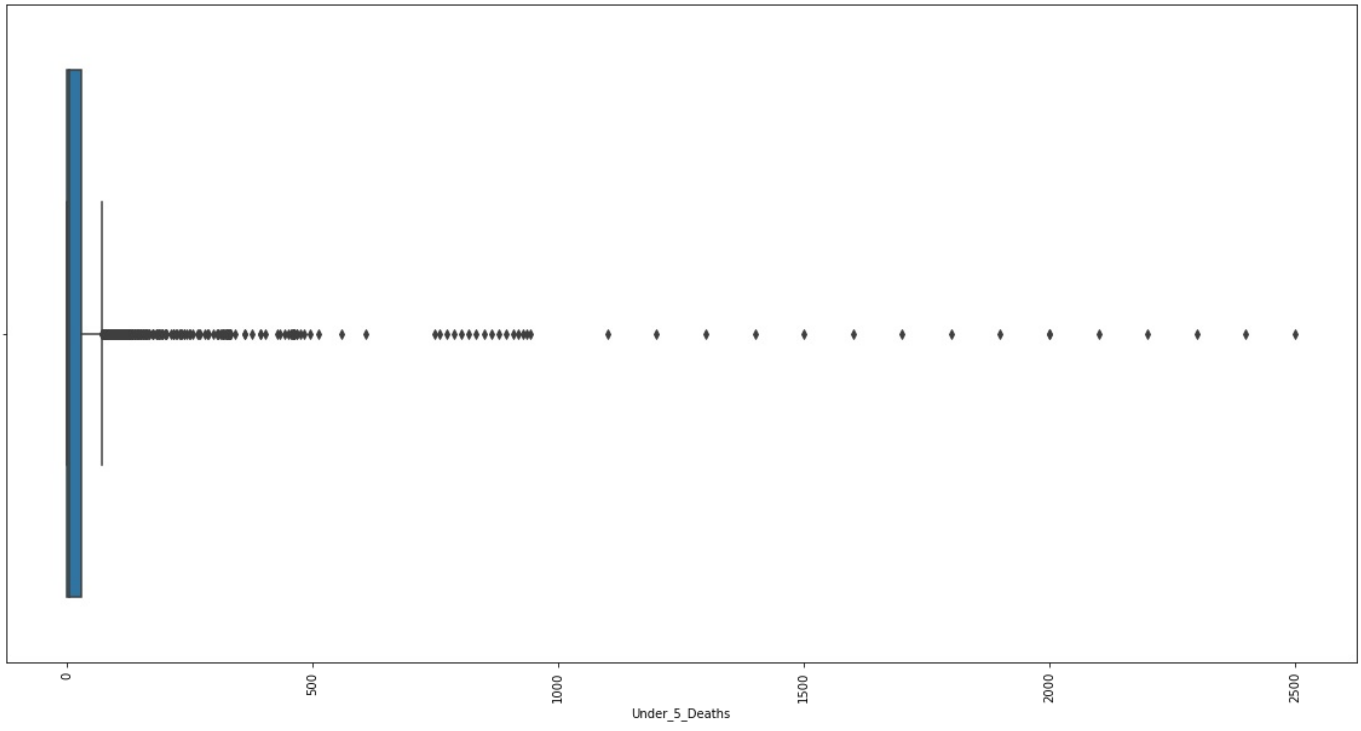

0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Alcohol

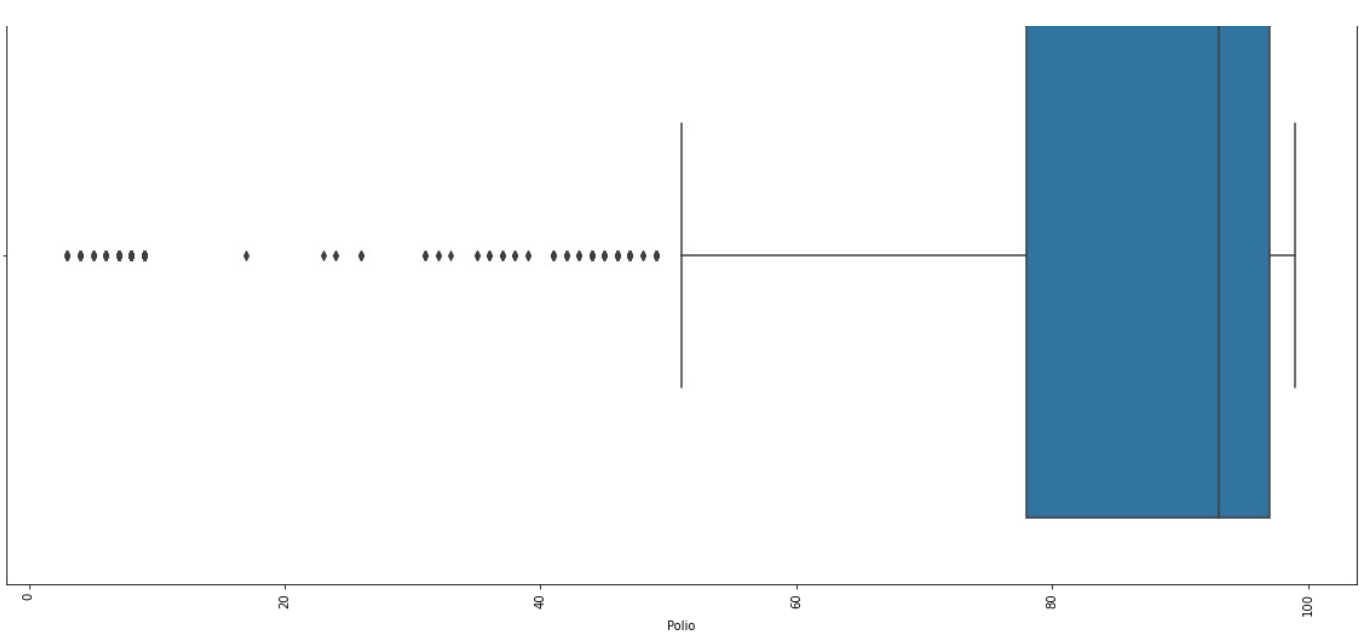

0
2500
5000
7500
10000
12500
15000
17500
20000
Percentage_Expenditure

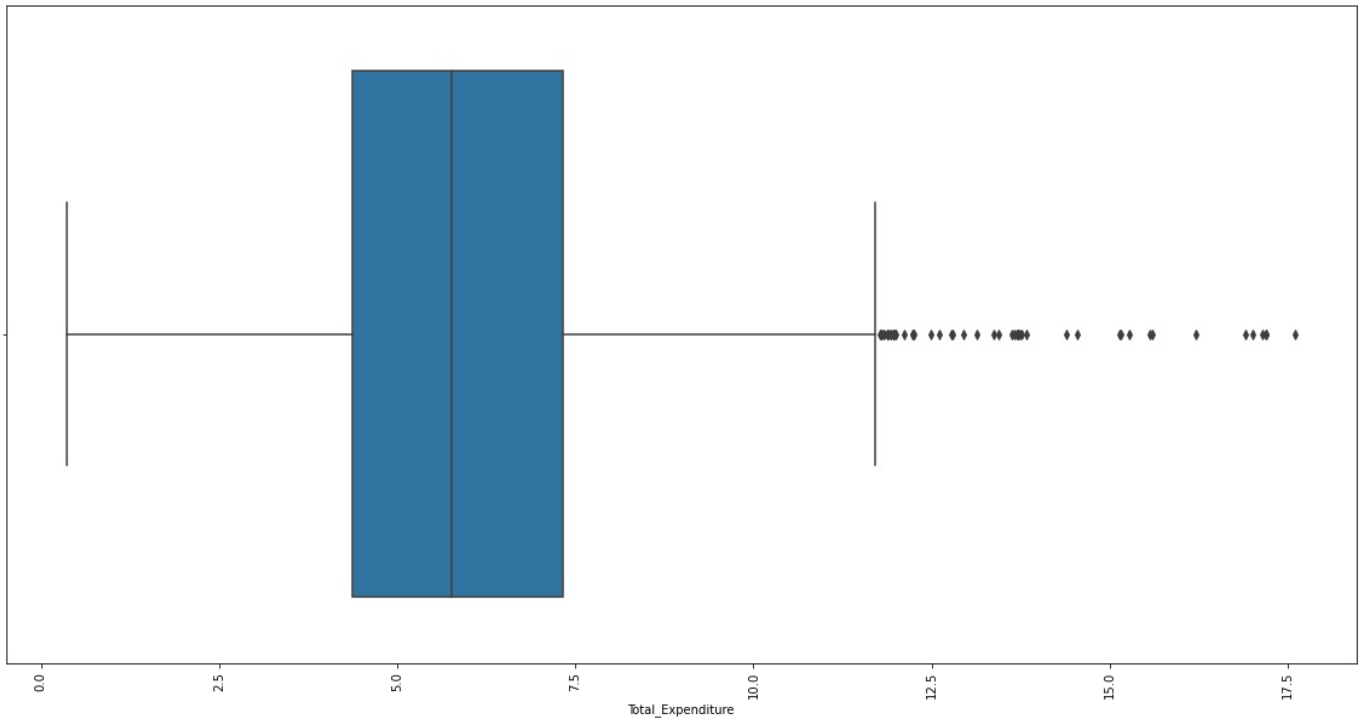

0
20
40
60
80
100
Hepatitis_B

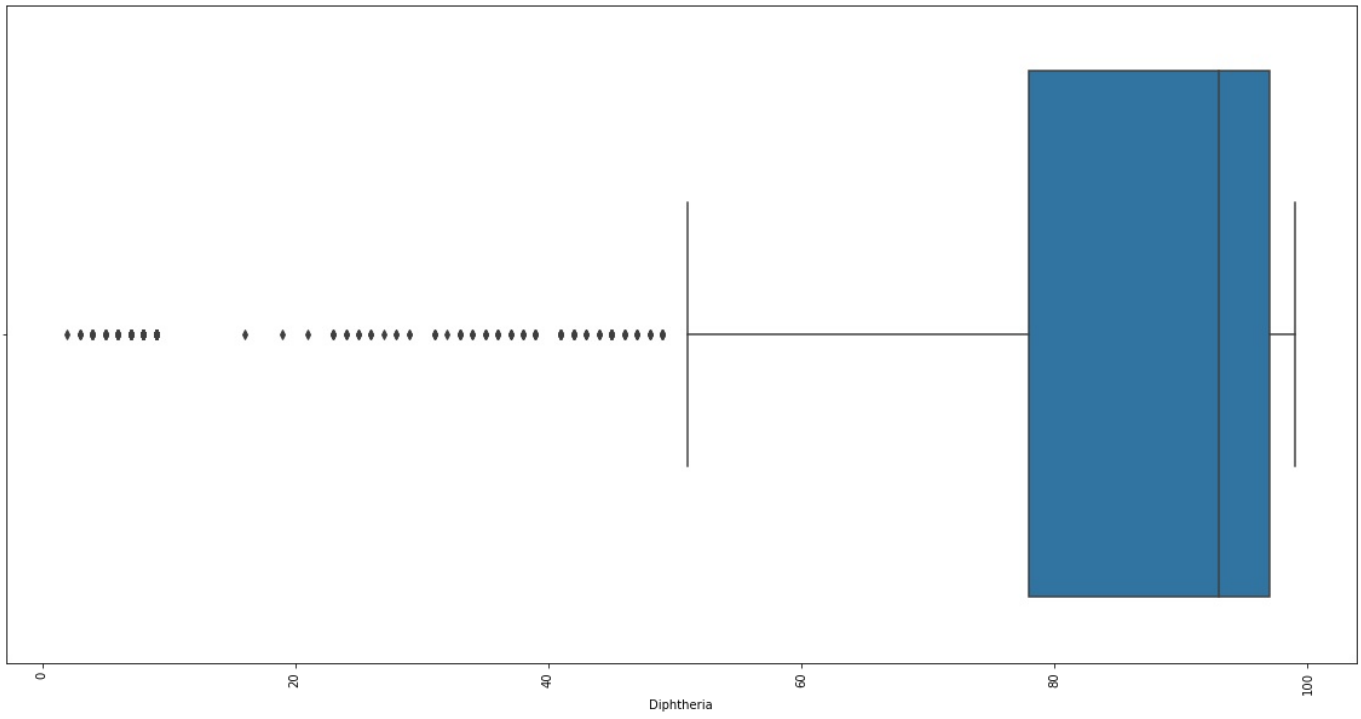

0
50000
100000
150000
200000
Measles

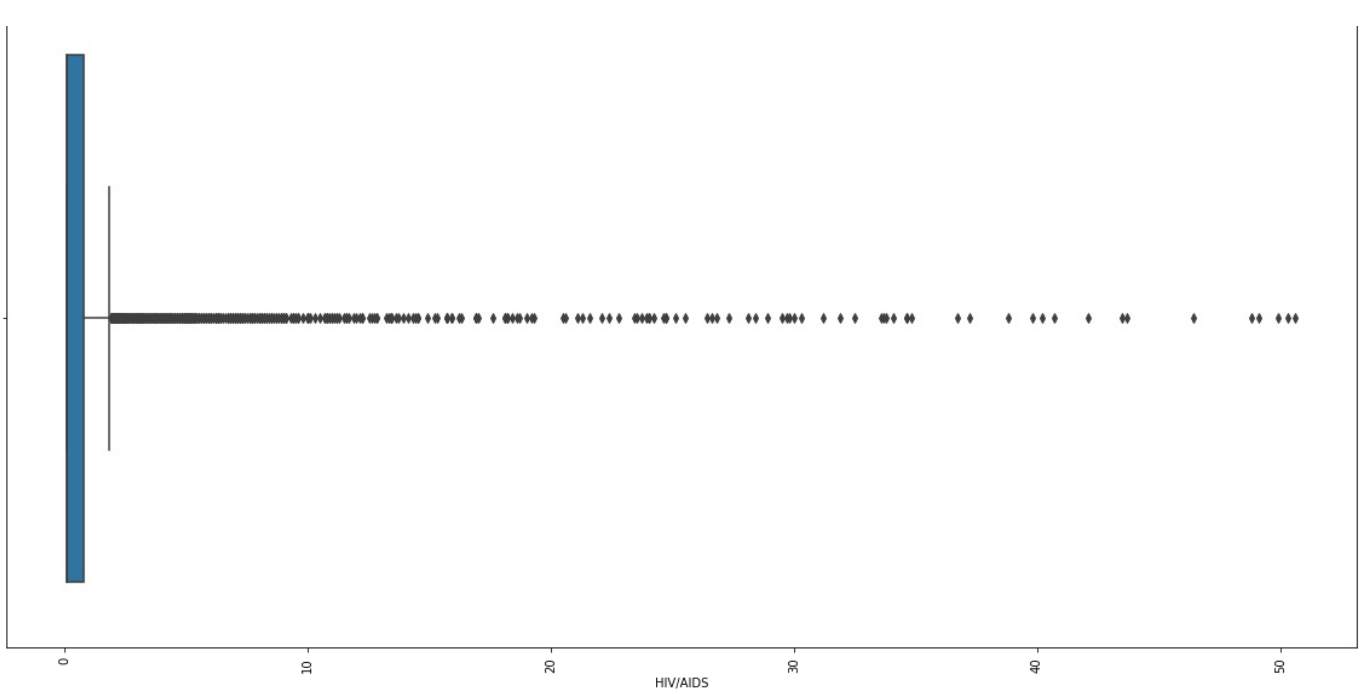

0
10
20
30
40
50
60
70
80
BMI

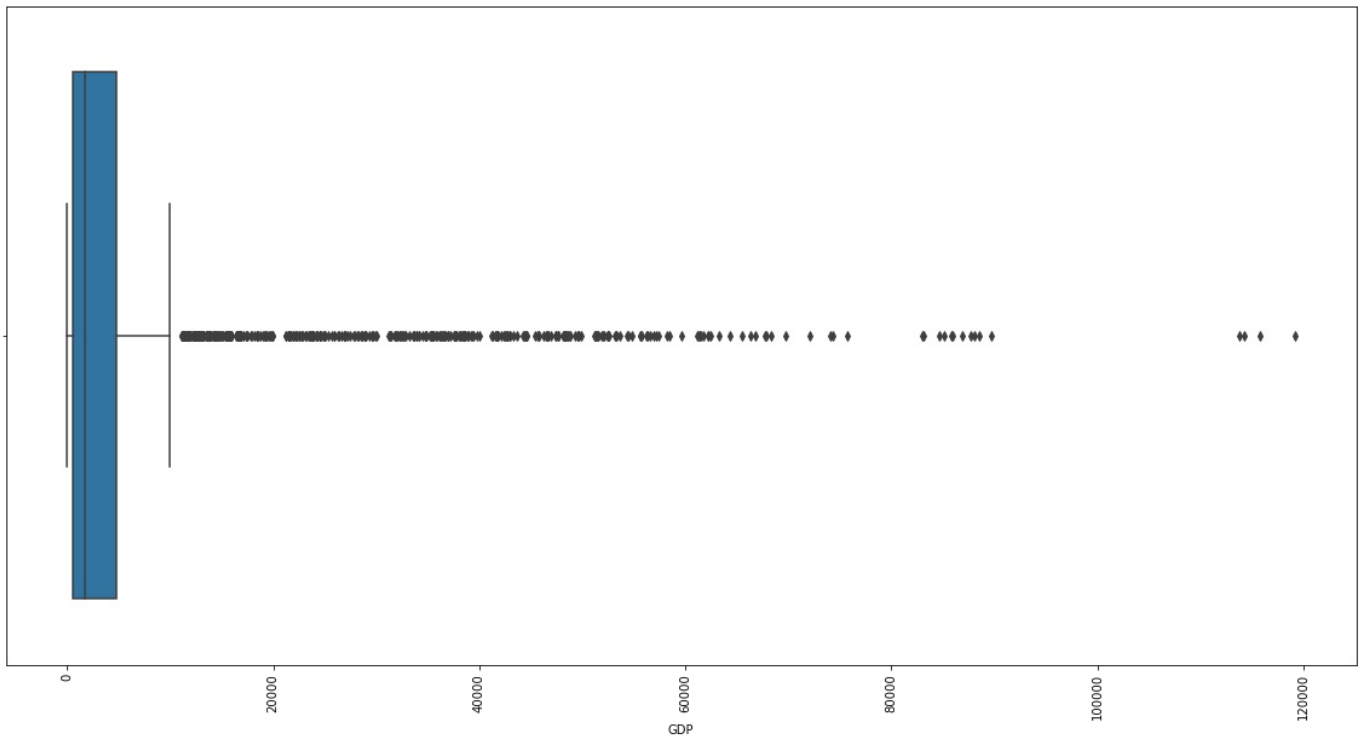

0
500
1000
1500
2000
2500
Under_5_Deaths

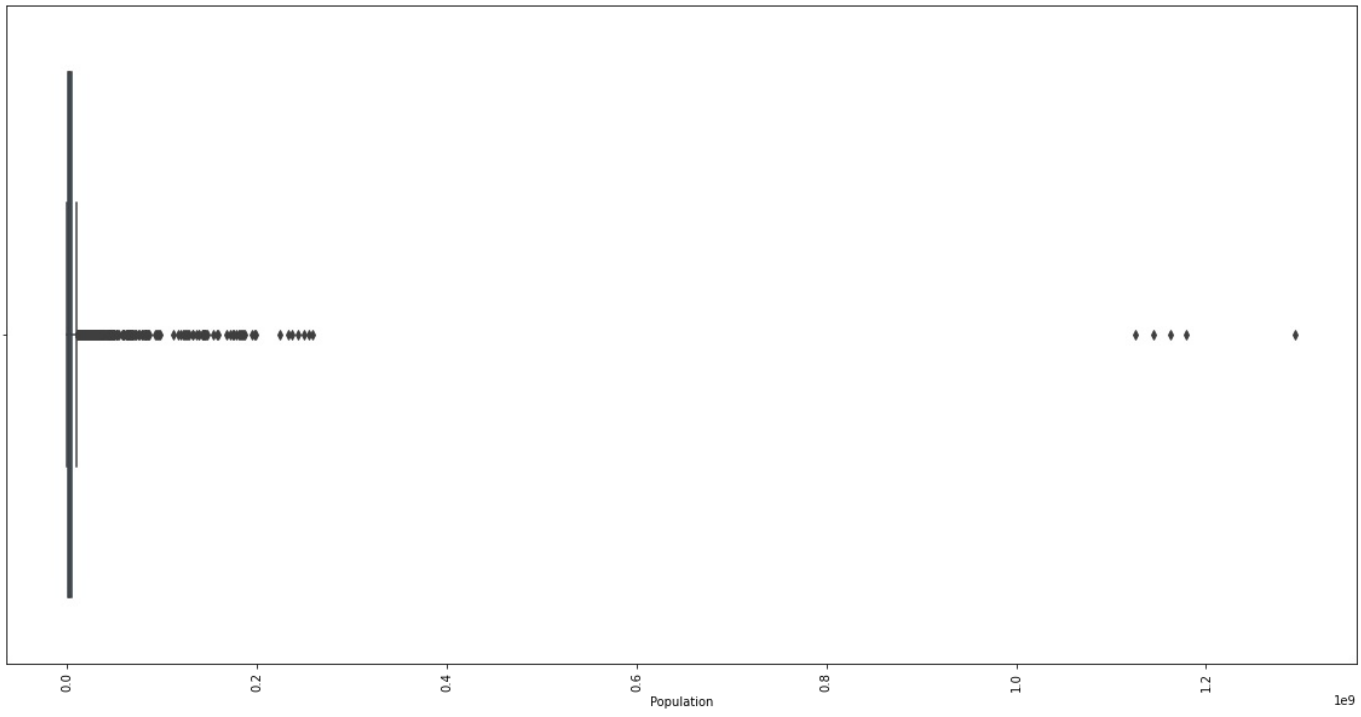

0
20
40
60
80
100
Polio

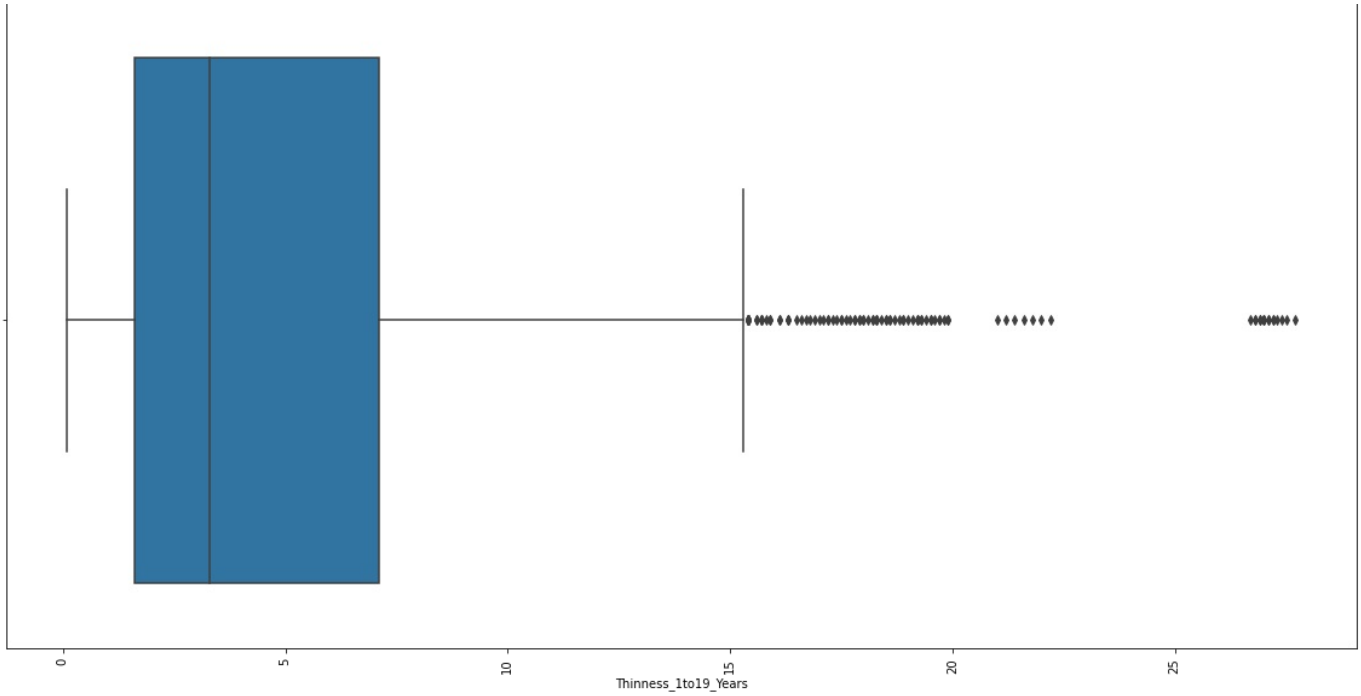

0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Total_Expenditure

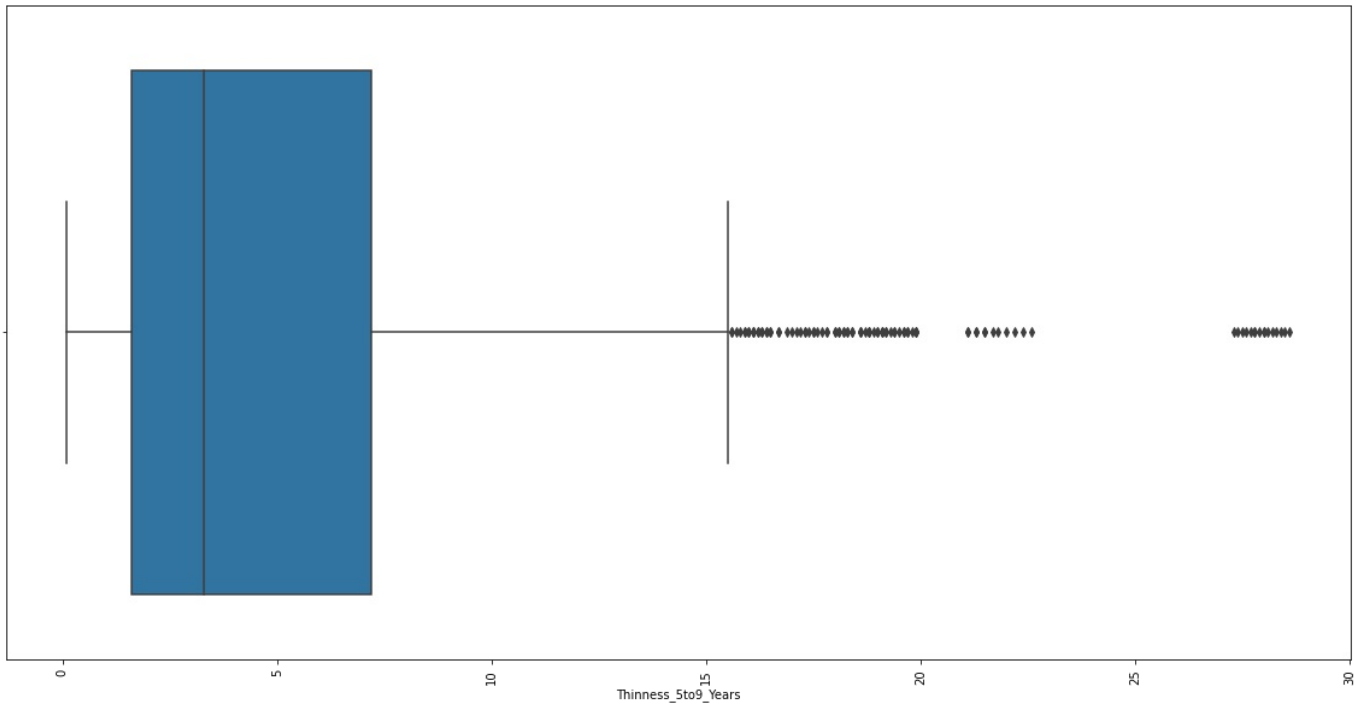

0
20
40
60
80
100
Diphtheria

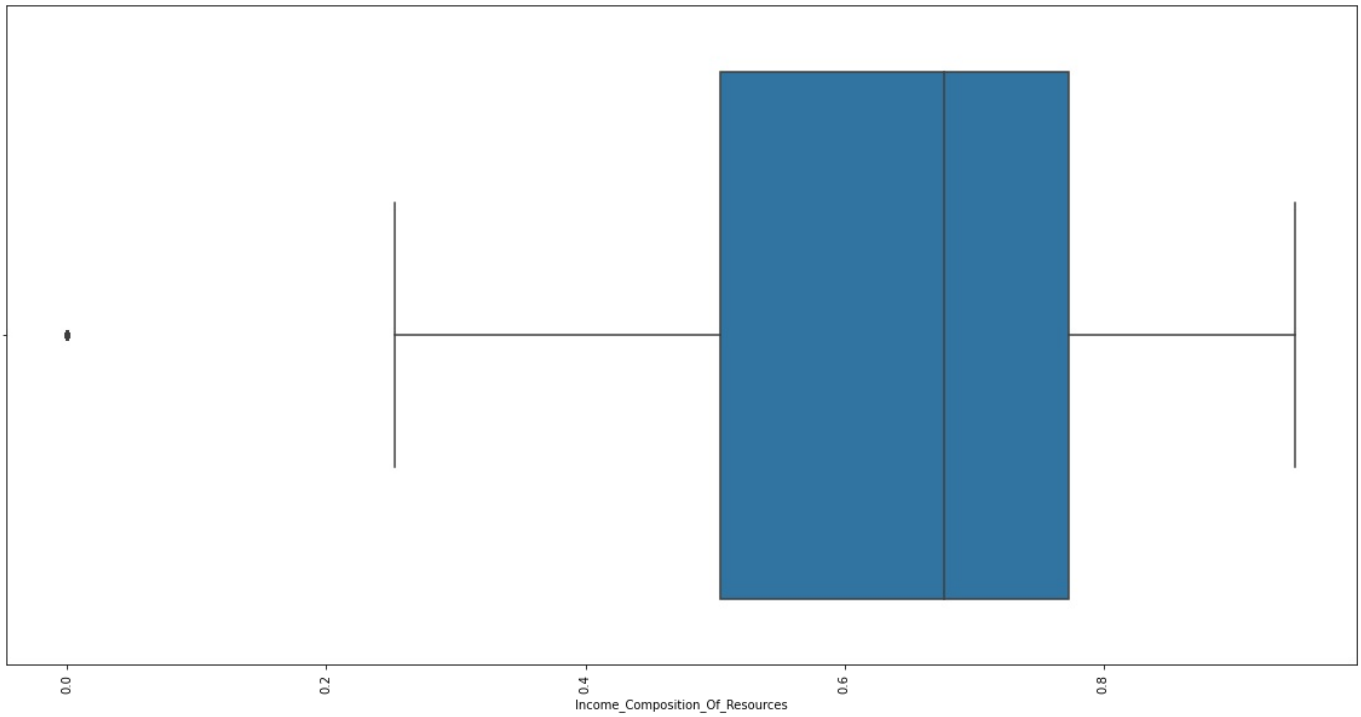

0
10
20
30
40
50
HIV/AIDS

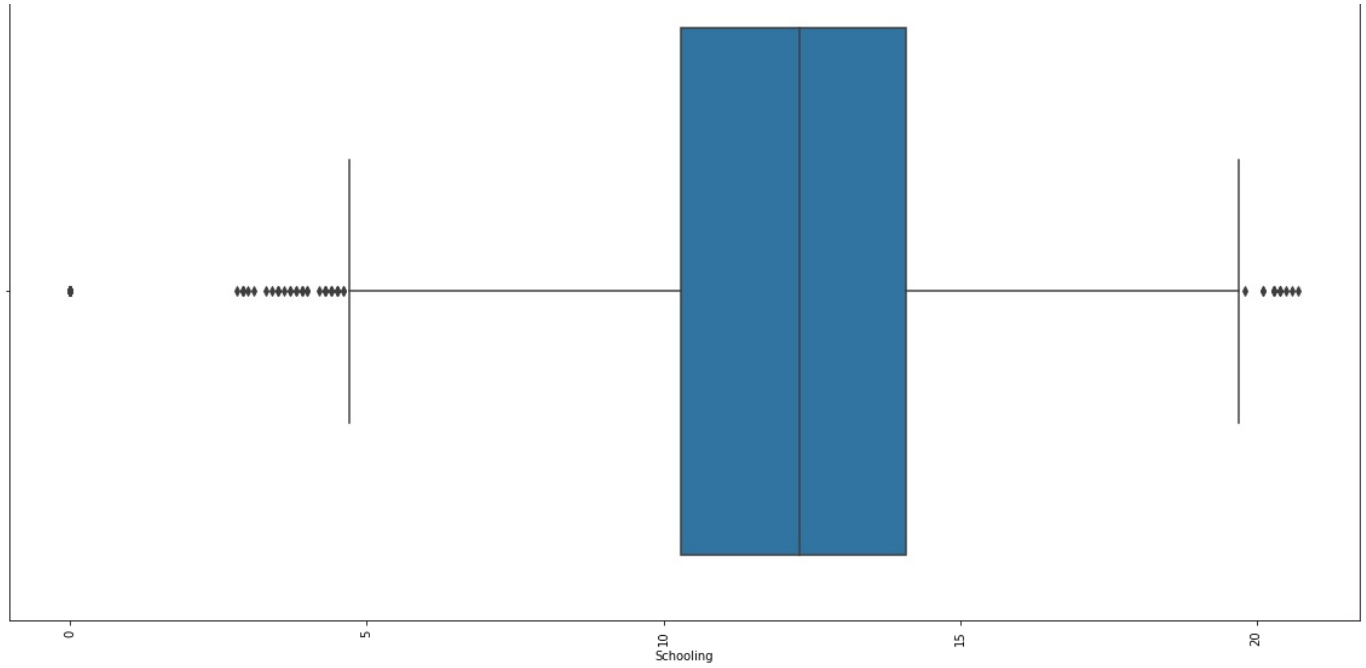

0
20000
40000
60000
80000
100000
120000
GDP


0.0
0.2
0.4
0.6
0.8
1.0
1.2
Population
1e9


0
5
10
15
20
25
Thinness_1to19_Years


0
5
10
15
20
25
30
Thinness_5to9_Years

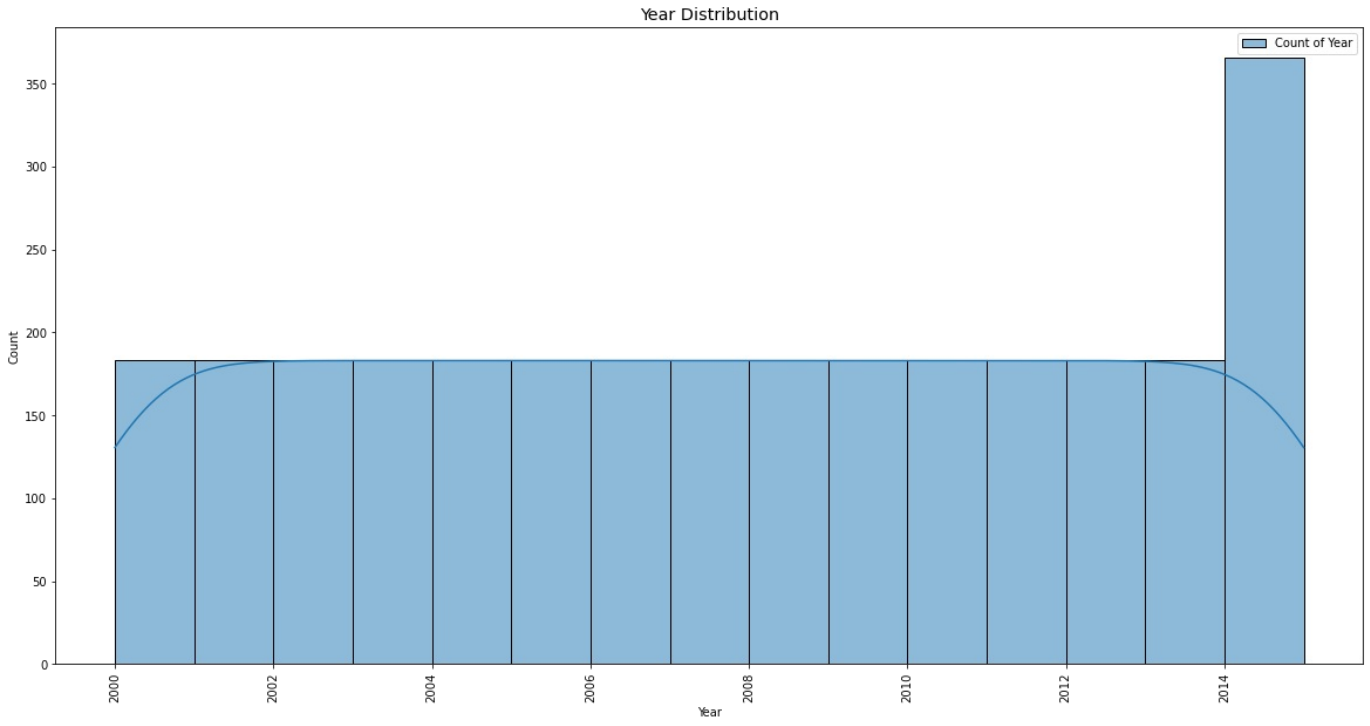

0.0
0.2
0.4
0.6
0.8
Income_Composition_Of_Resources

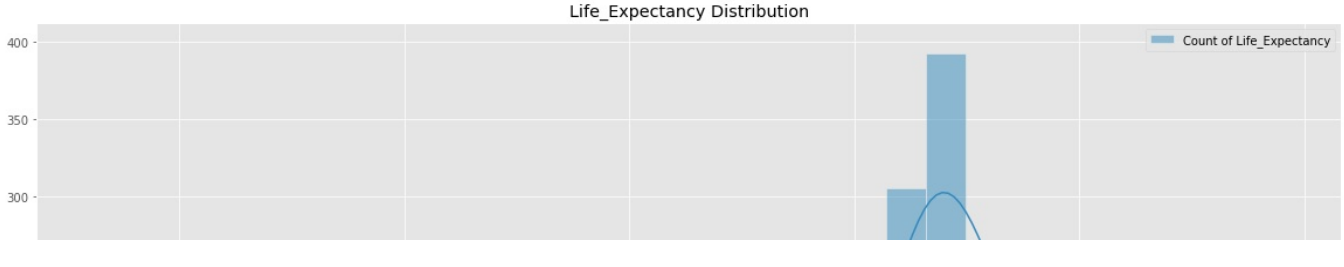


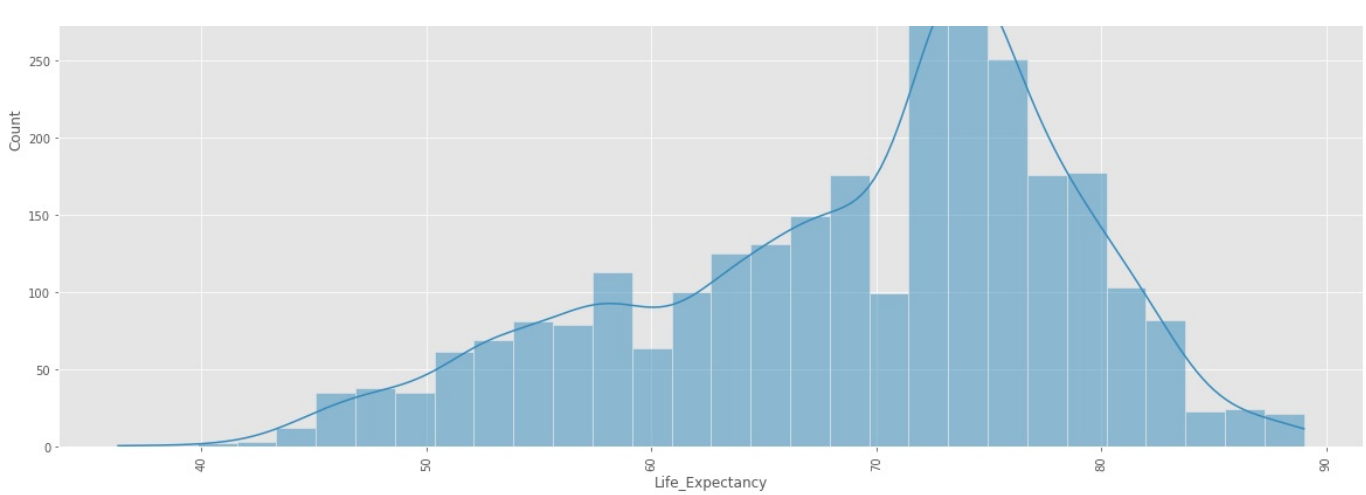
- Adult_Mortality, Infant_Deaths ,Percentage_Expenditure, Hepatitis_B, Measles,Under_5_Deaths,Polio,Total_Expenditure,Diphtheria,HIV/AIDS,GDP,PopulationThinnes_1to19_Years,Thinnes all these columns have extensive outliers

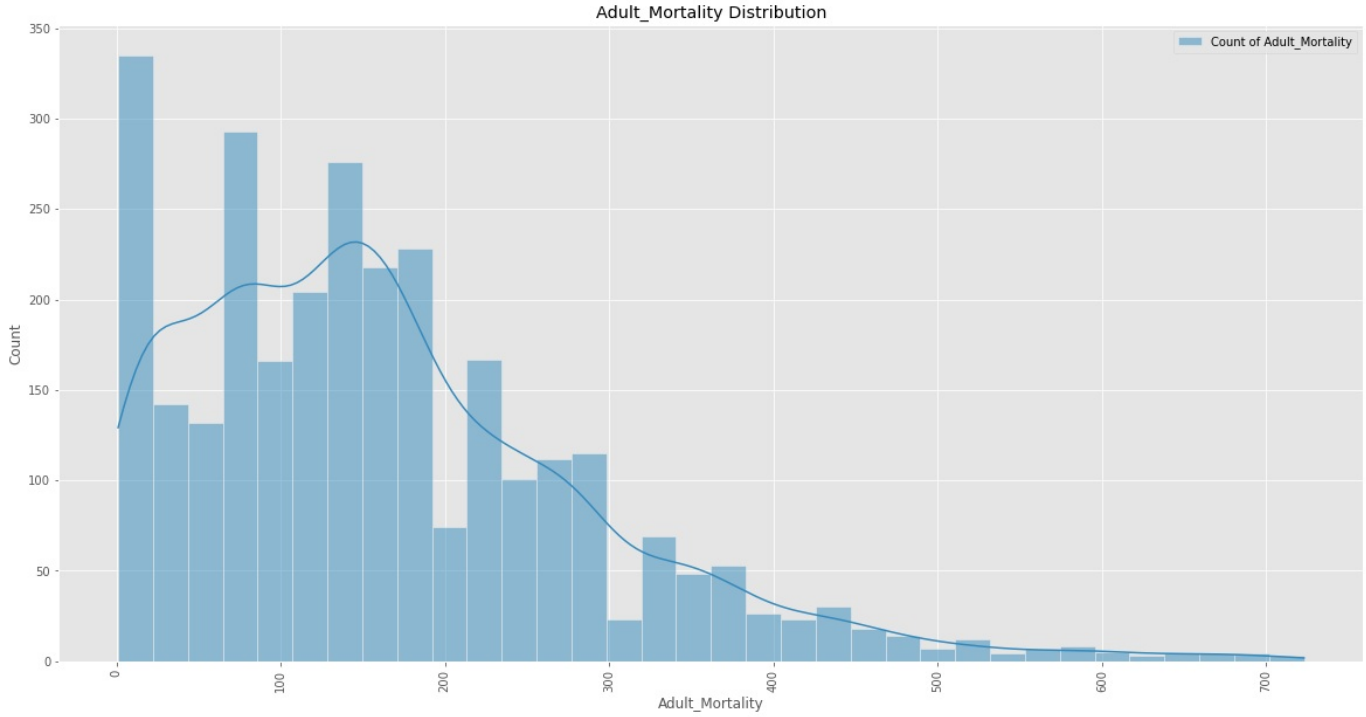
ii . HISTPLOT

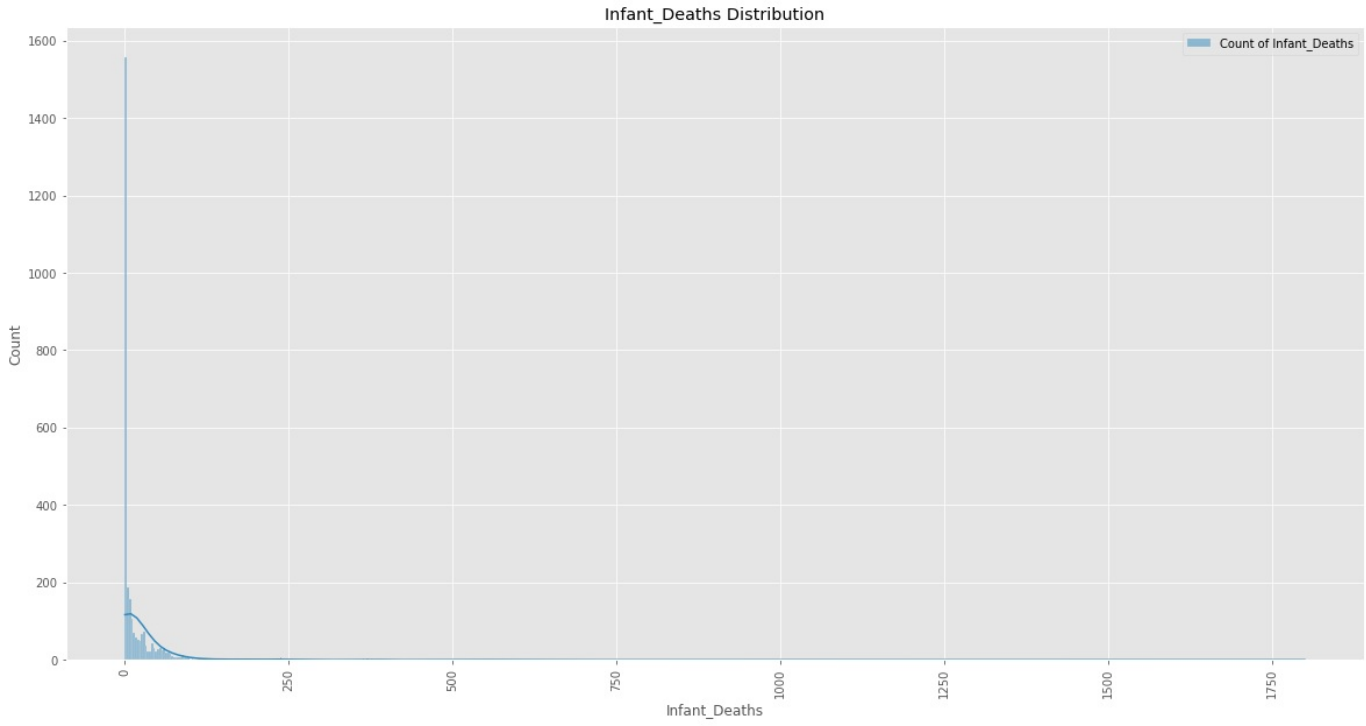
In [46]:
```
plot_univariate_graphs(life_expectancy_data,numeric_cols,'h')
```



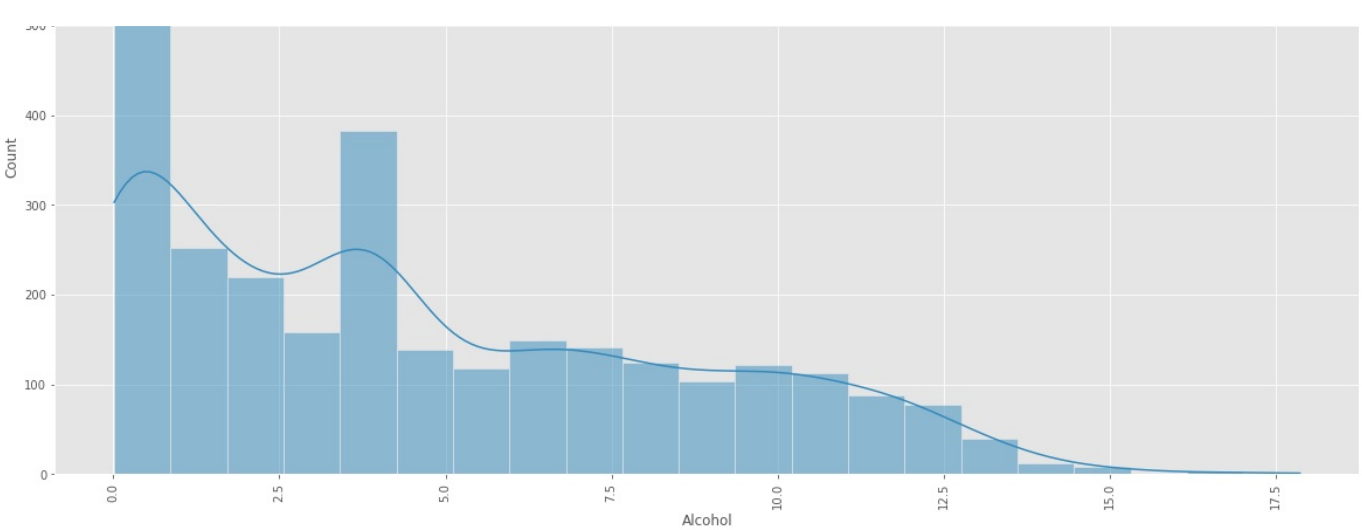


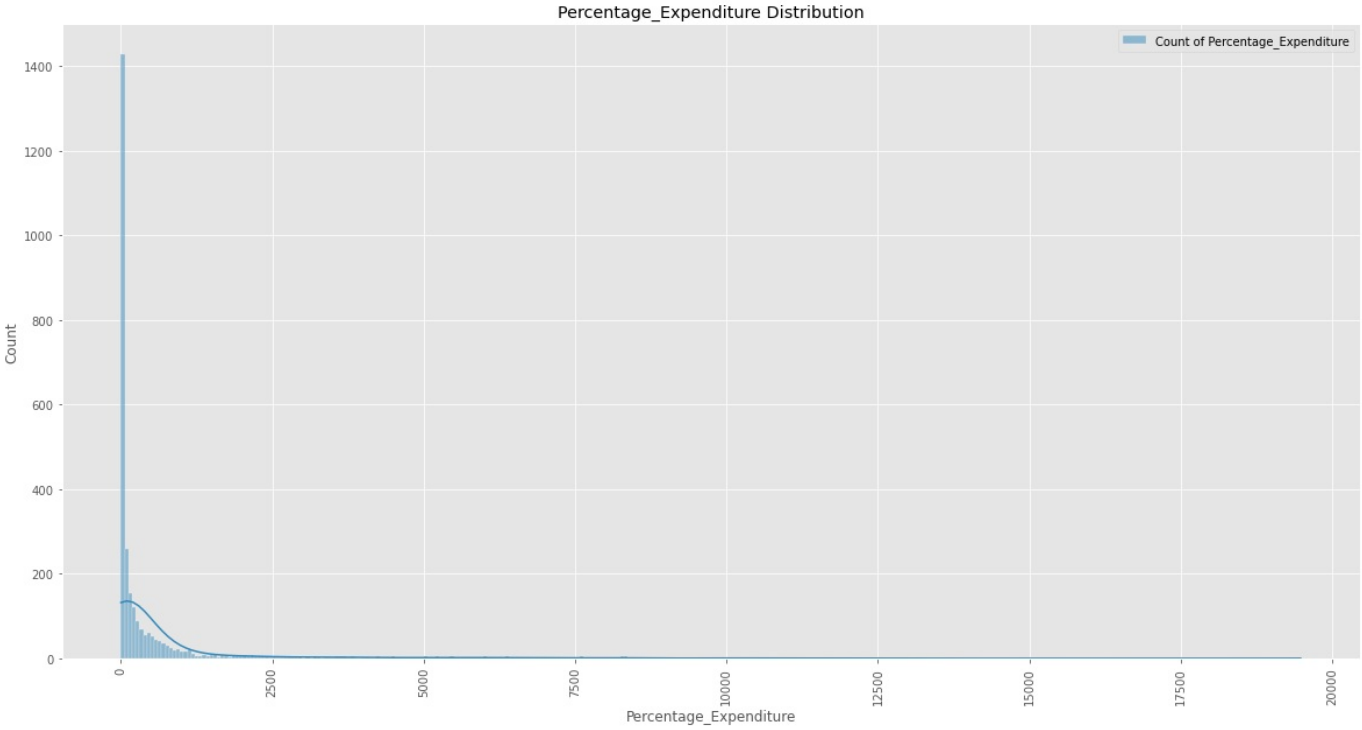

Count
Life_Expectancy

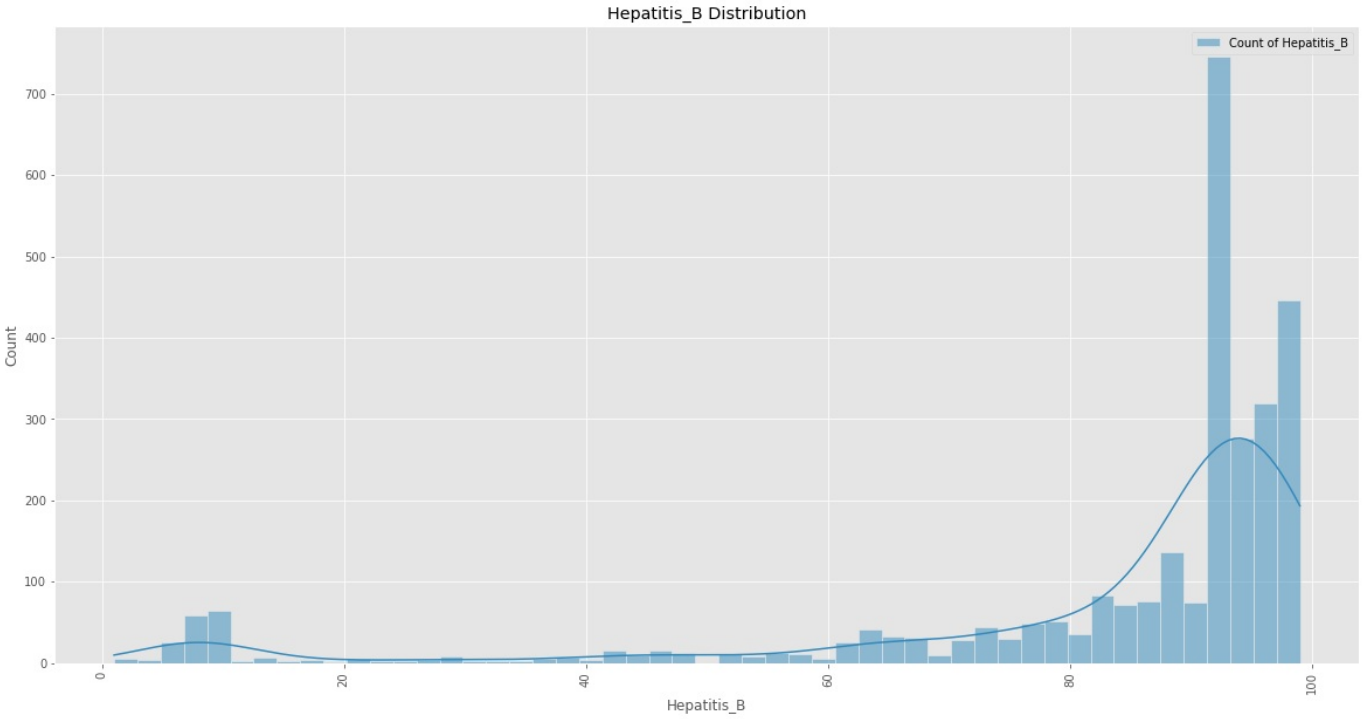

Adult_Mortality Distribution
Count of Adult_Mortality
Count
Adult_Mortality


Infant_Deaths Distribution
Count of Infant_Deaths
Count
Infant_Deaths

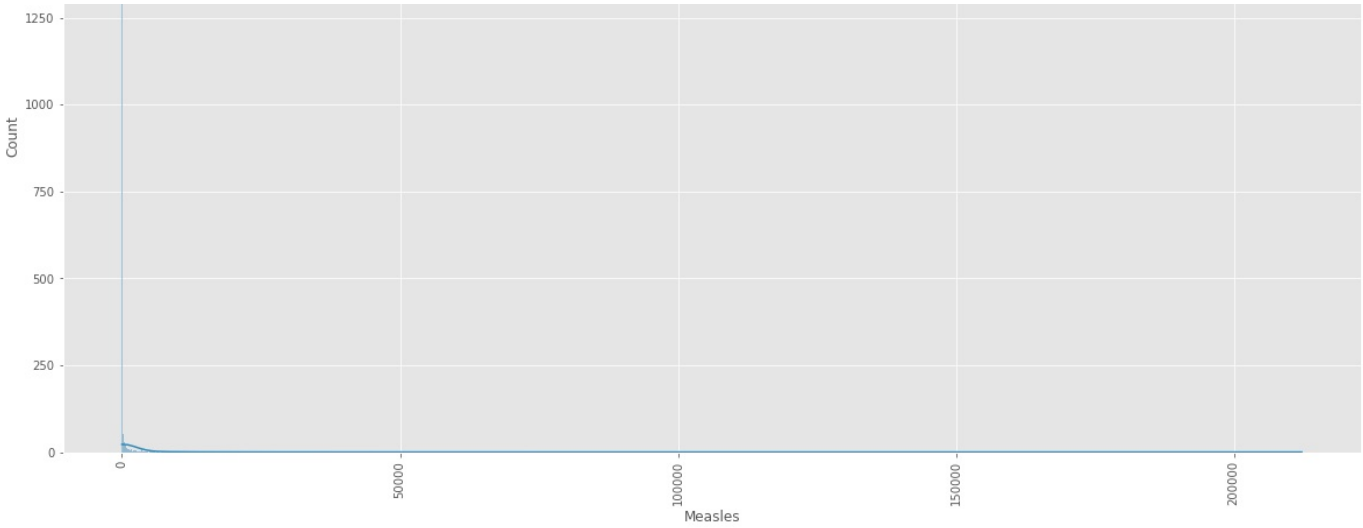

Alcohol Distribution
Count of Alcohol

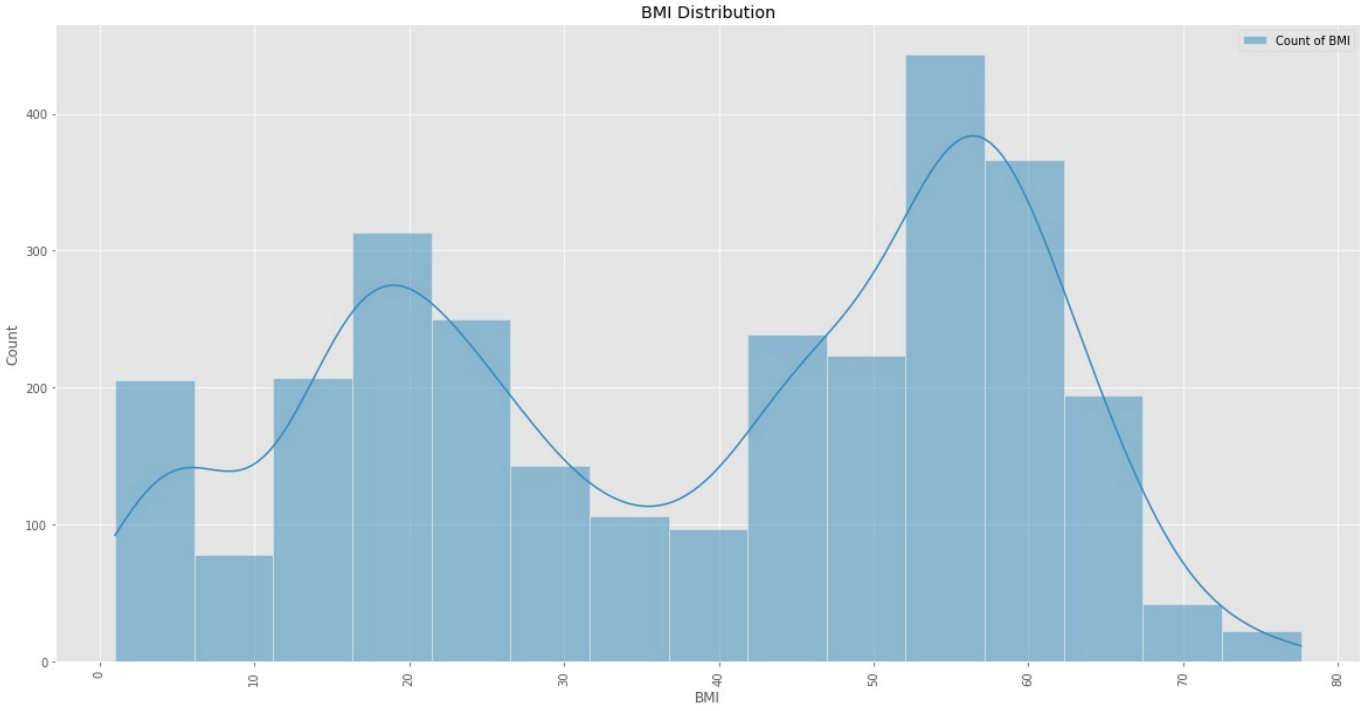

Count
Alcohol

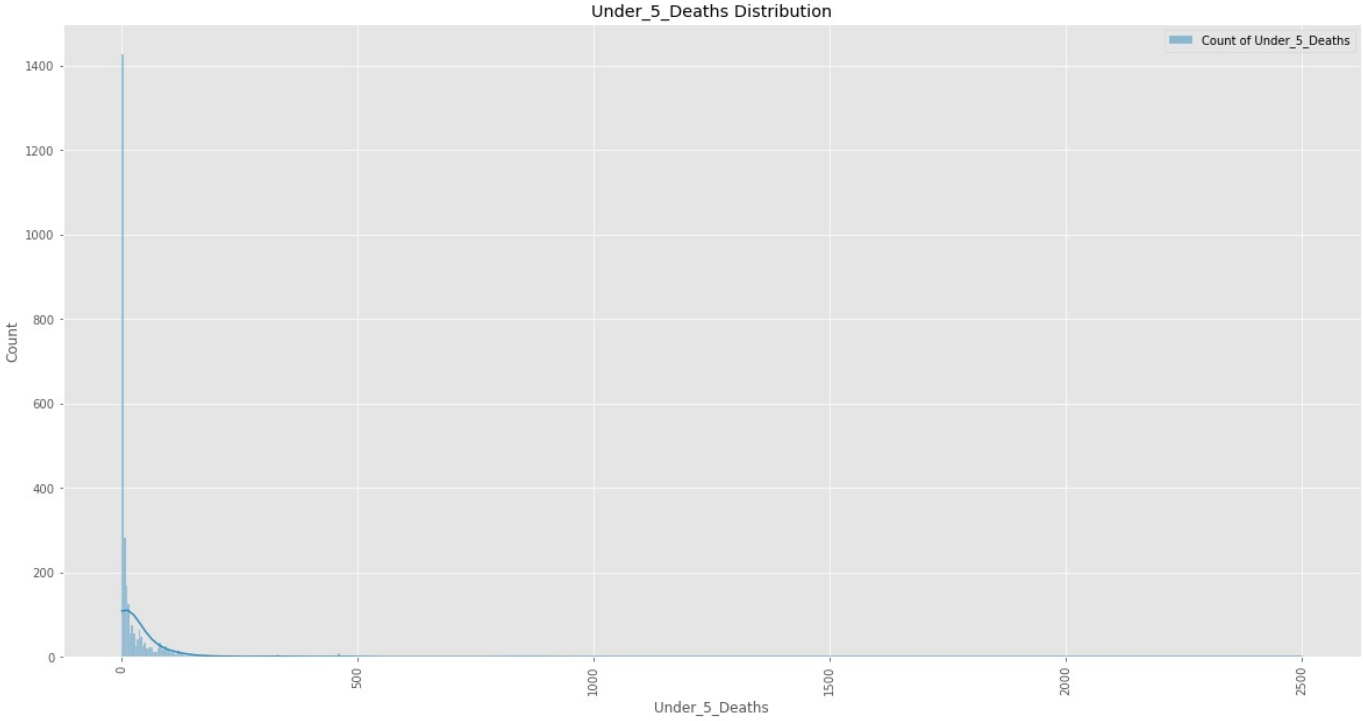

Percentage_Expenditure Distribution
Count of Percentage_Expenditure
Count
Percentage_Expenditure

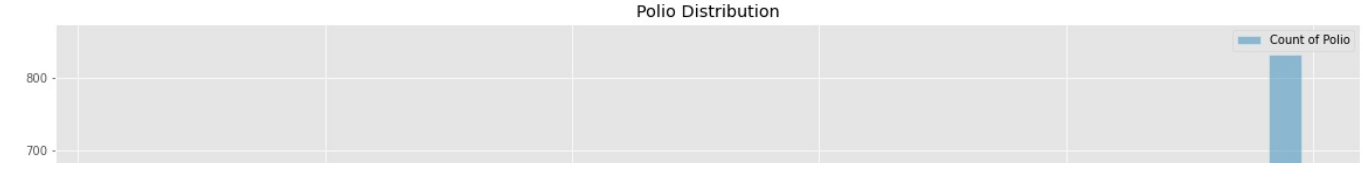

Hepatitis_B Distribution
Count of Hepatitis_B
Count
Hepatitis_B

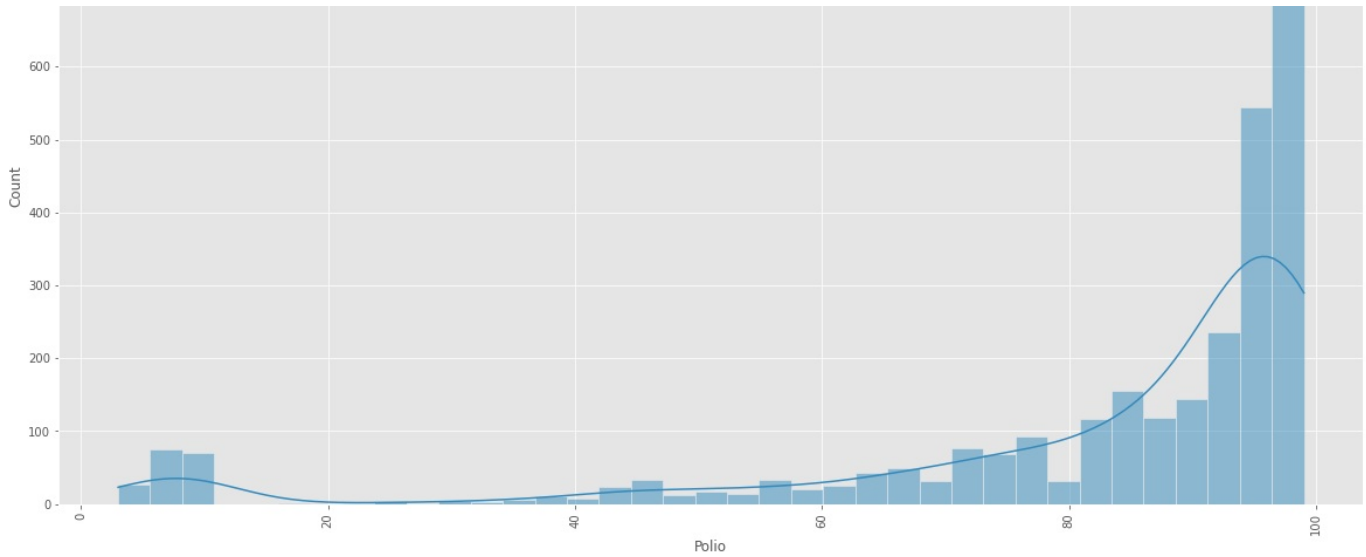

Measles Distribution
Count of Measles

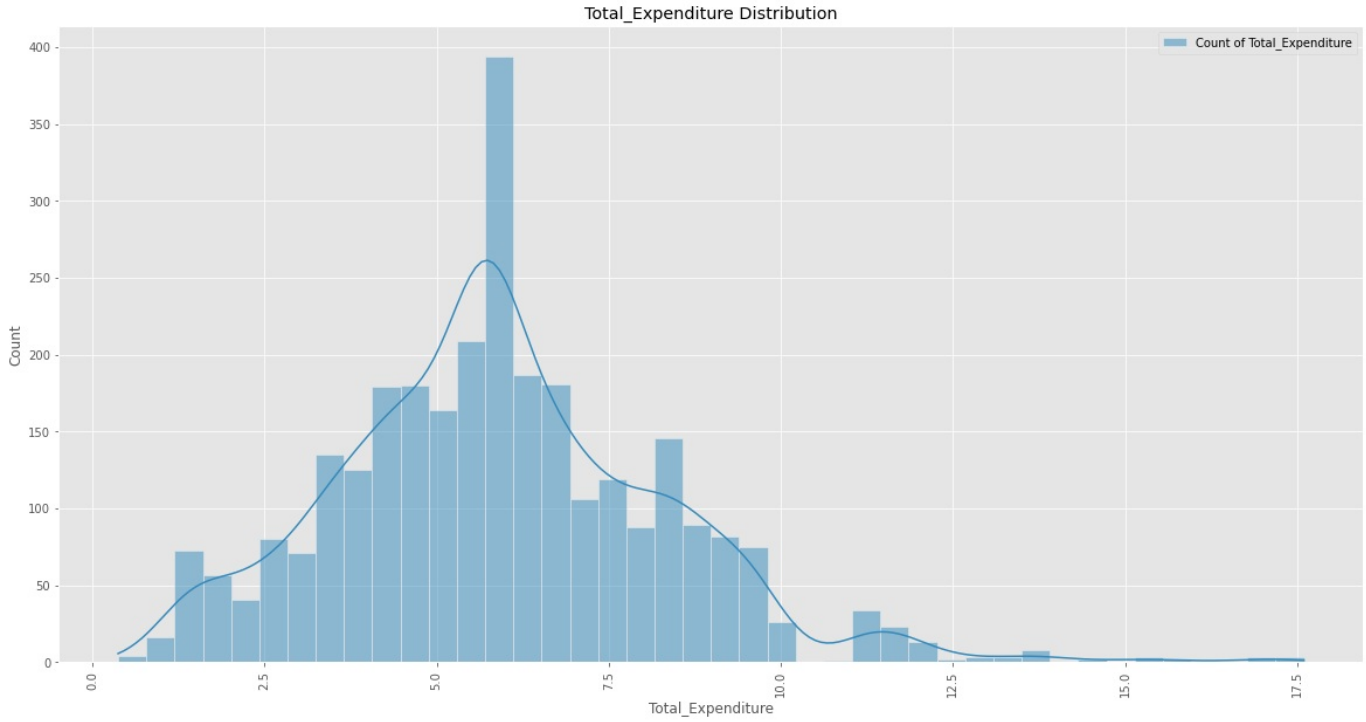

Count
Measles

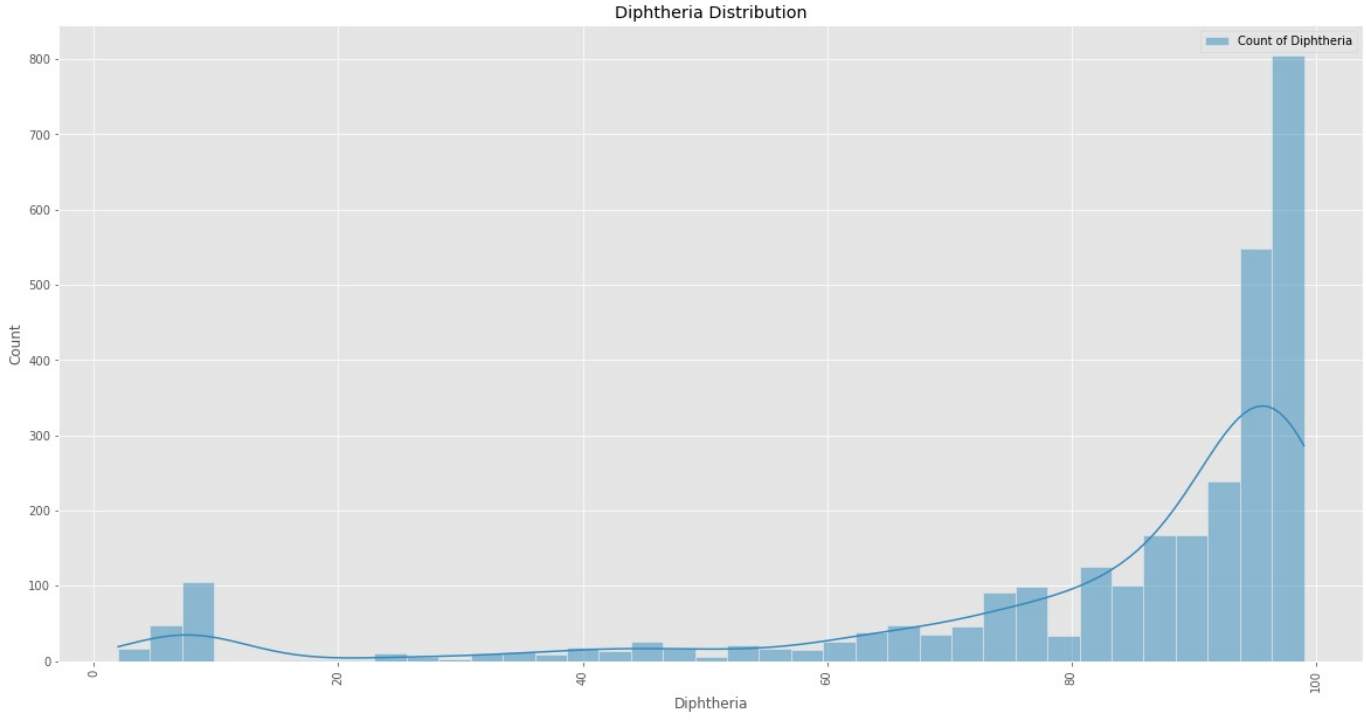

BMI Distribution
Count of BMI
Count
BMI

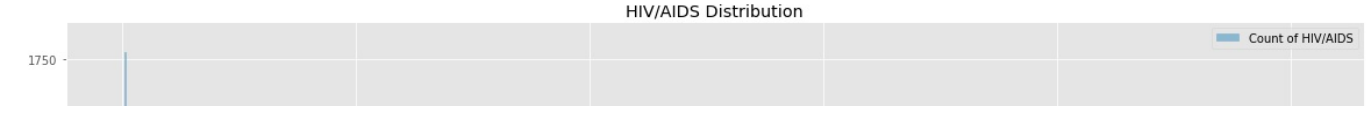

Under_5_Deaths Distribution
Count of Under_5_Deaths
Count
Under_5_Deaths

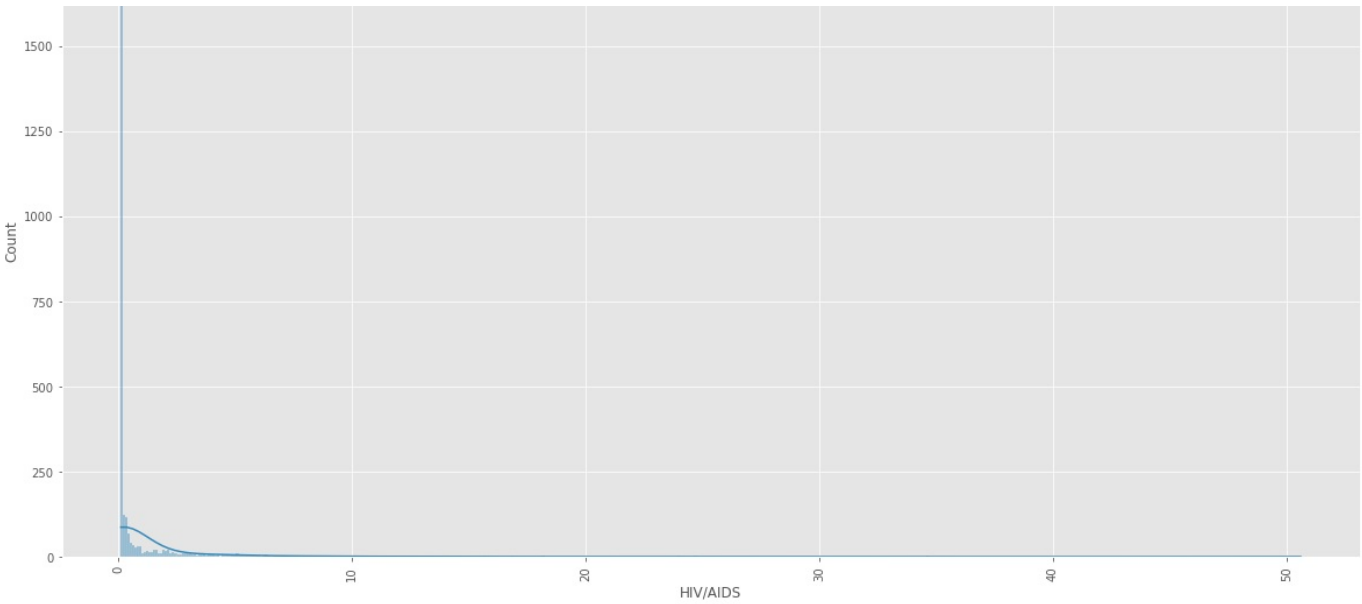

Polio Distribution
Count of Polio

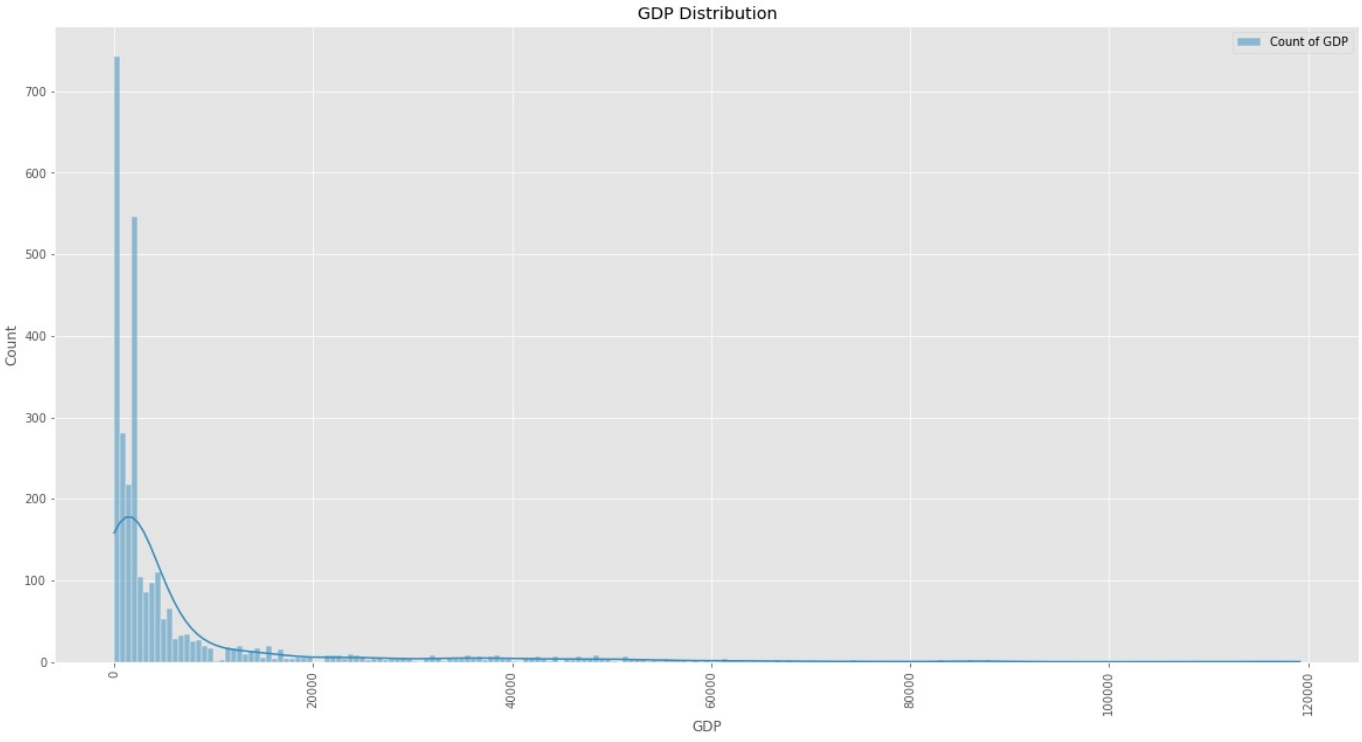

Count
Polio

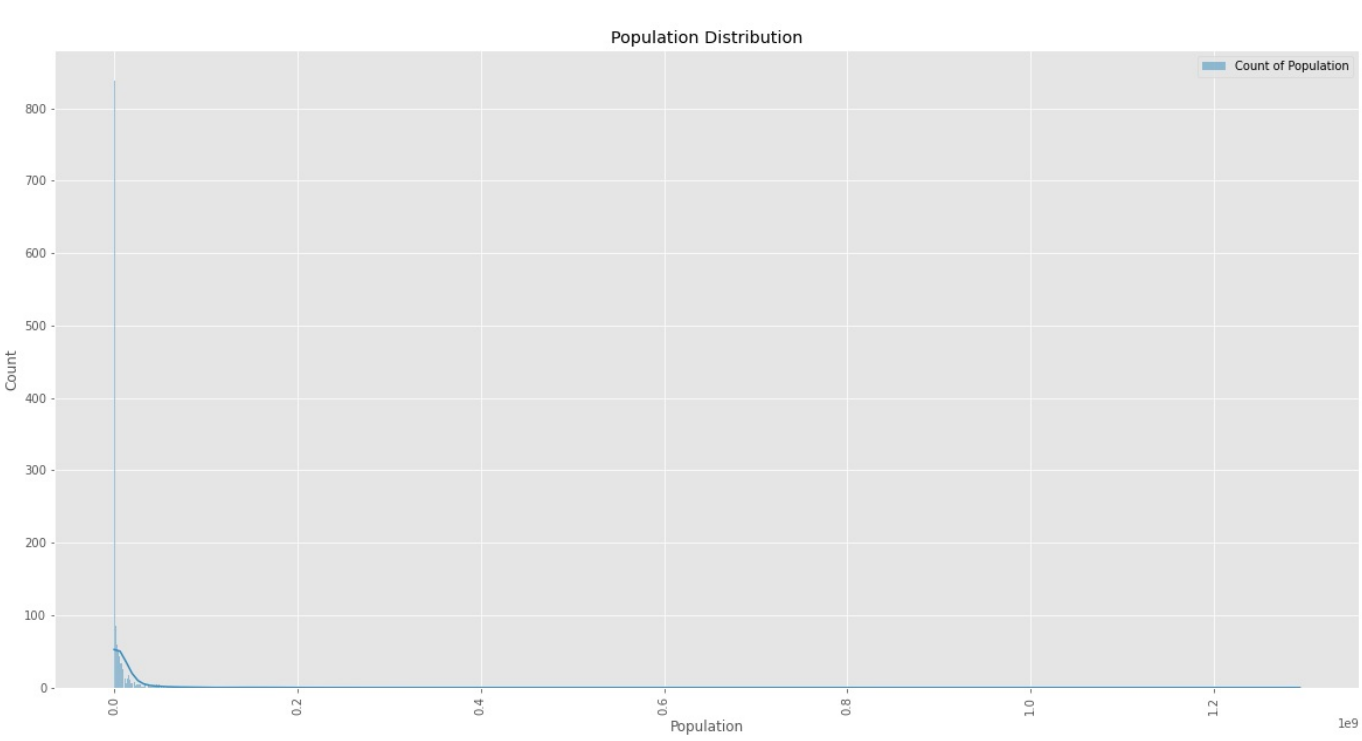

Total_Expenditure Distribution
Count of Total_Expenditure
Count
Total_Expenditure

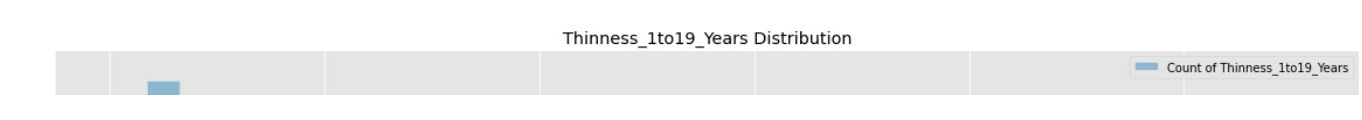

Diphtheria Distribution
Count of Diphtheria
Count
Diphtheria

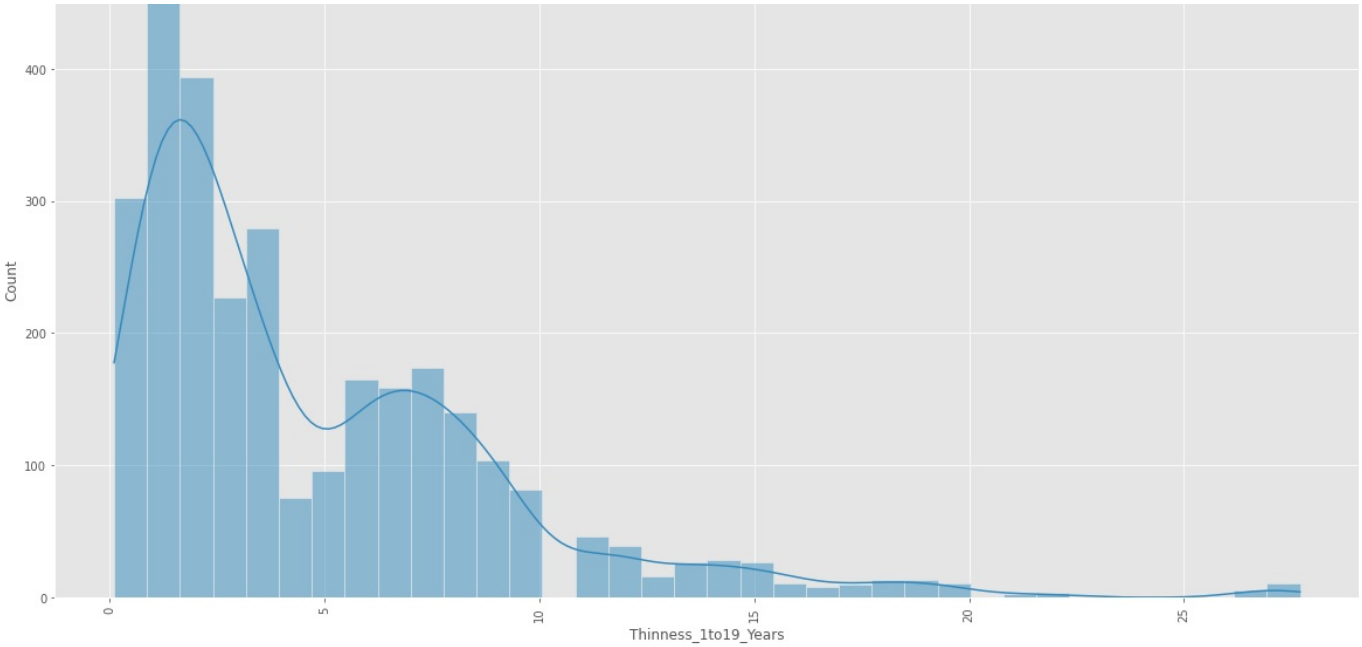

HIV/AIDS Distribution
Count of HIV/AIDS


Count
HIV/AIDS

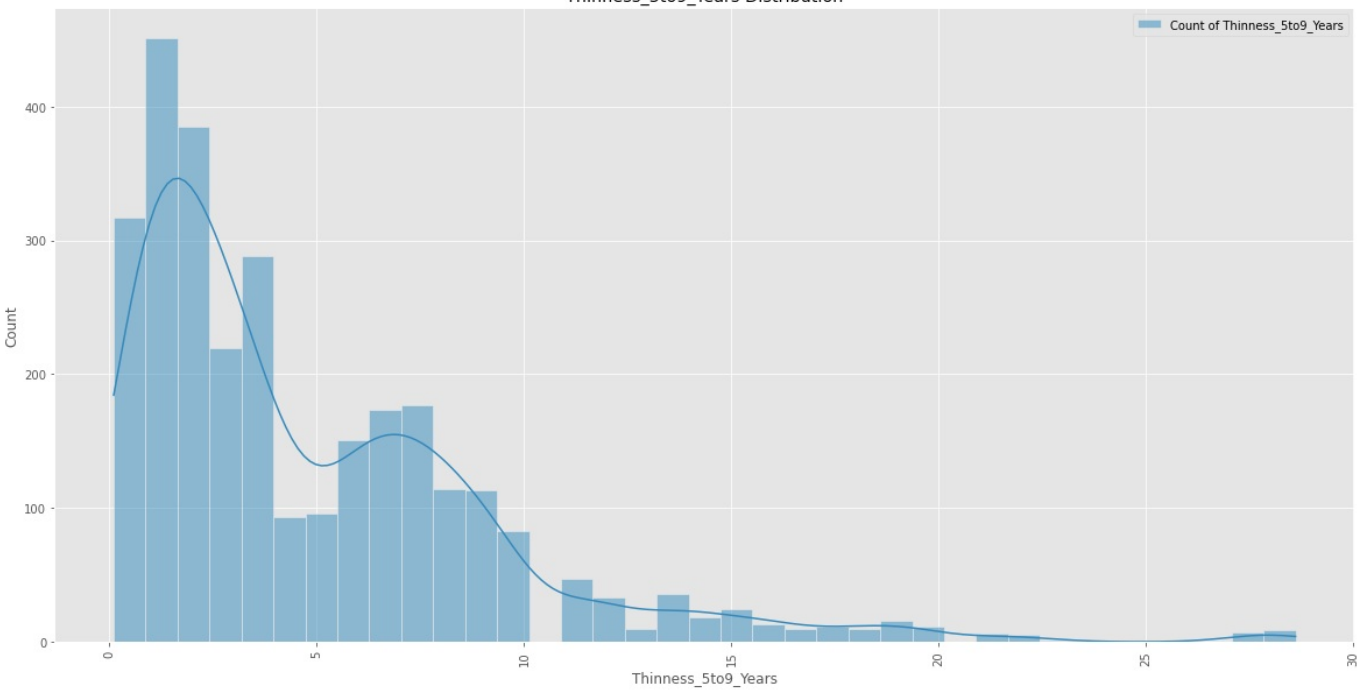

GDP Distribution
Count of GDP
Count
GDP


Population Distribution
Count of Population
Count
Population
1e9

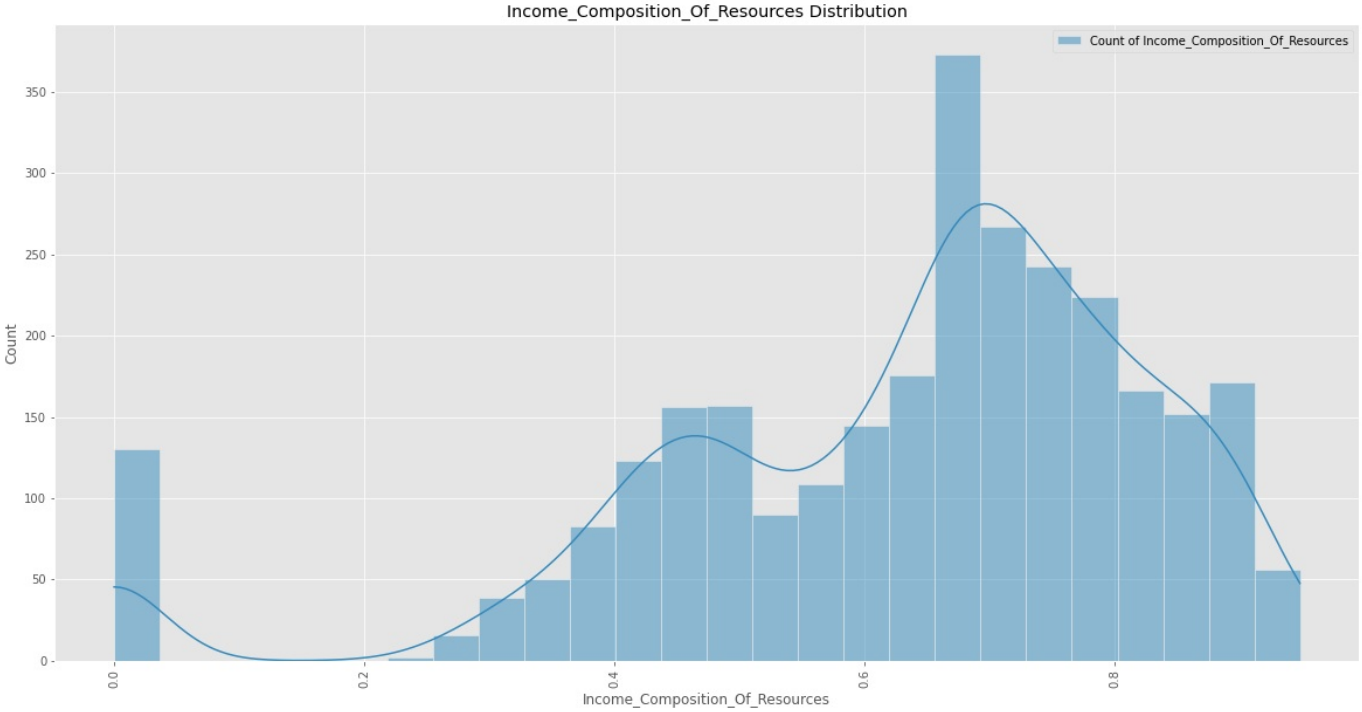

Thinness_1to19_Years Distribution
Count of Thinness_1to19_Years



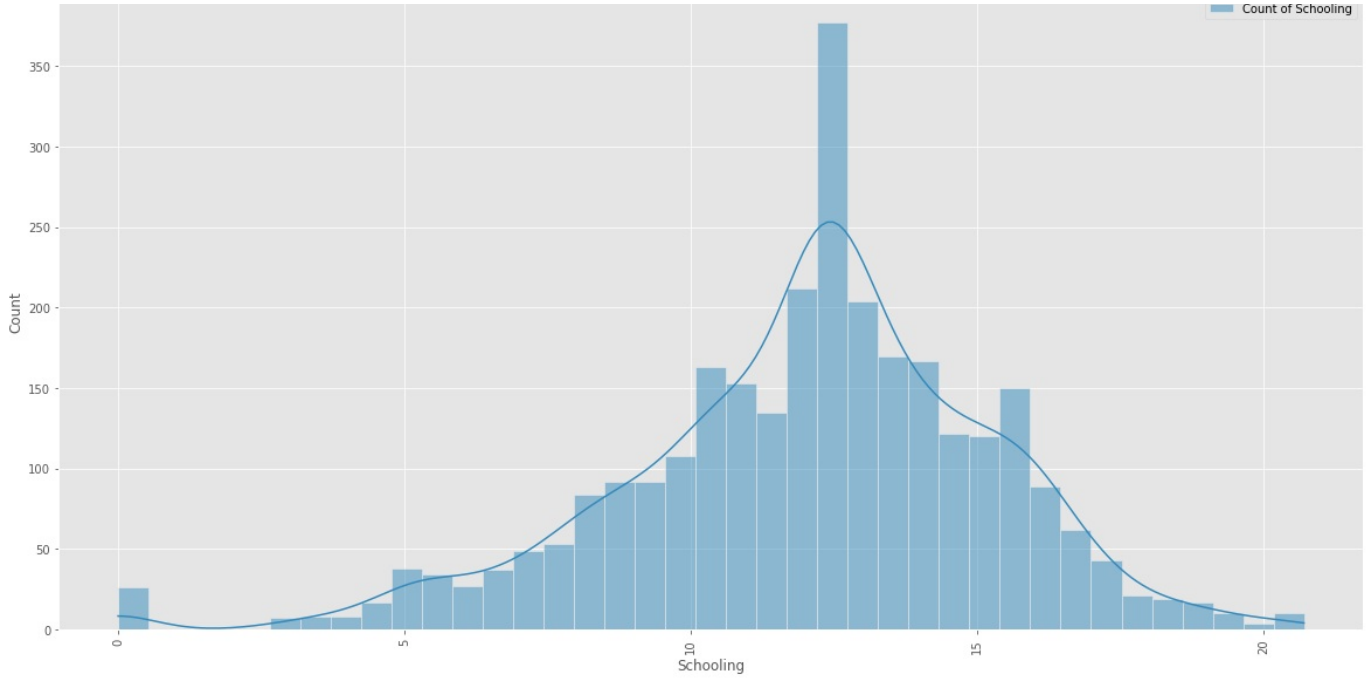
Thinness_5to9_Years Distribution



Income_Composition_Of_Resources Distribution



Schooling Distribution



- MOST OF THE COLUMNS HAVE SKEWNESS IN DATA

- PEOPLE LIVE MIN OF 36 AND MAXIMUM OF 89 AVERAGELY PEOPLE HAVE LIFE OF 70 (MAJORITY PEOPLE LIFE SPAN IS BETWEEN 70 - 75) VERY FEW LIVE BEYOND 80

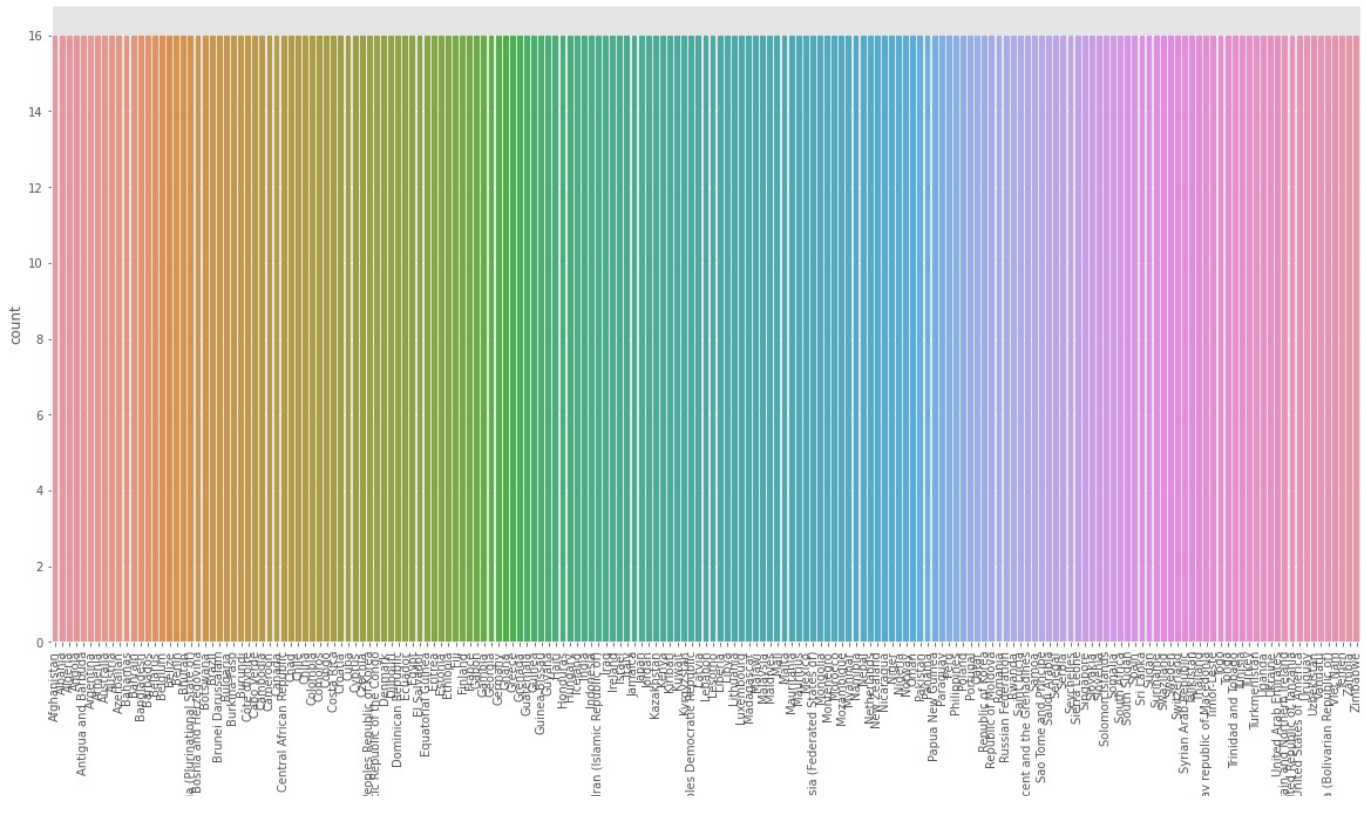
- ALCOHOL CONSUMPTION IS SEEN VERY COMMONLY IN PEOPLE

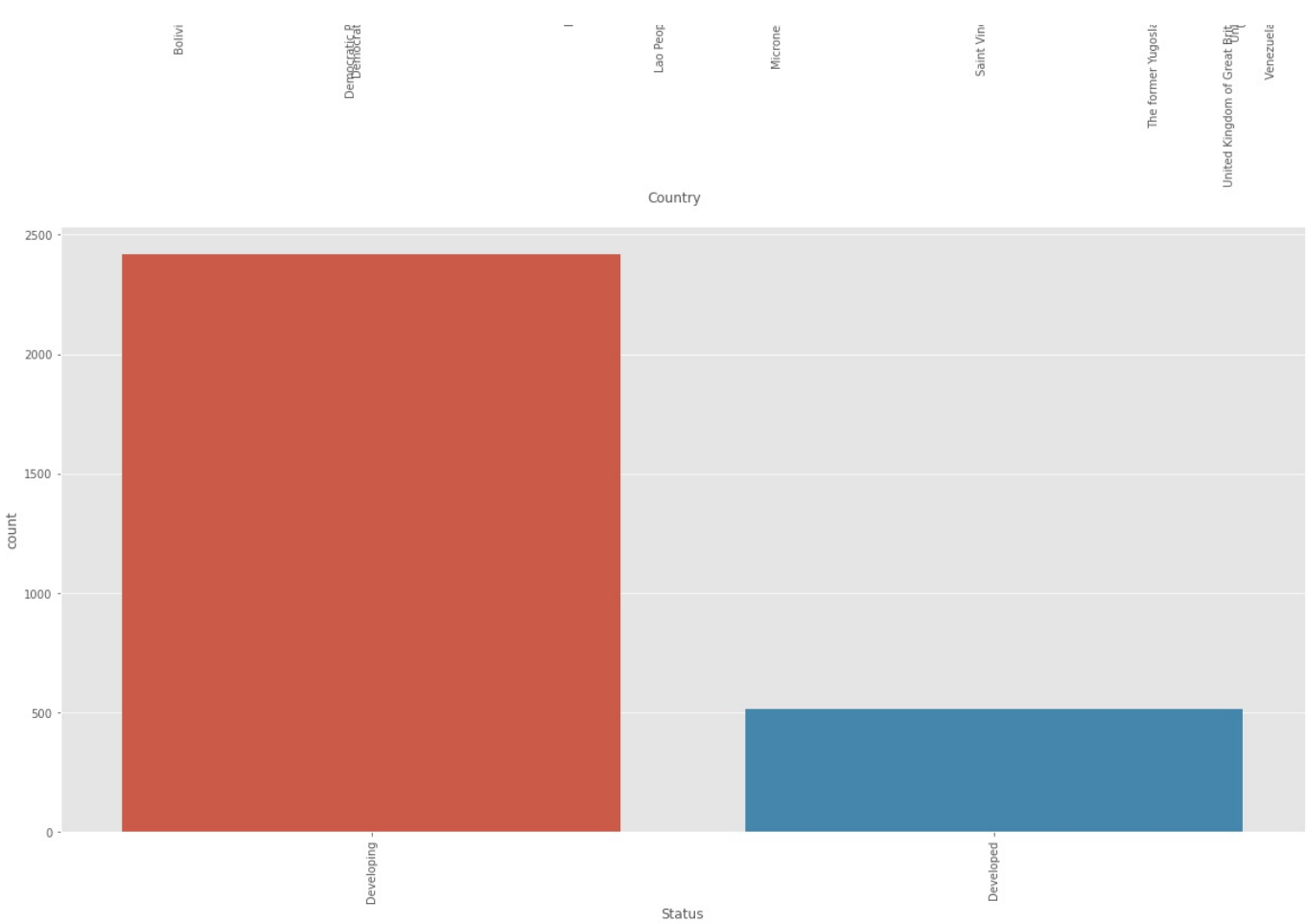
-

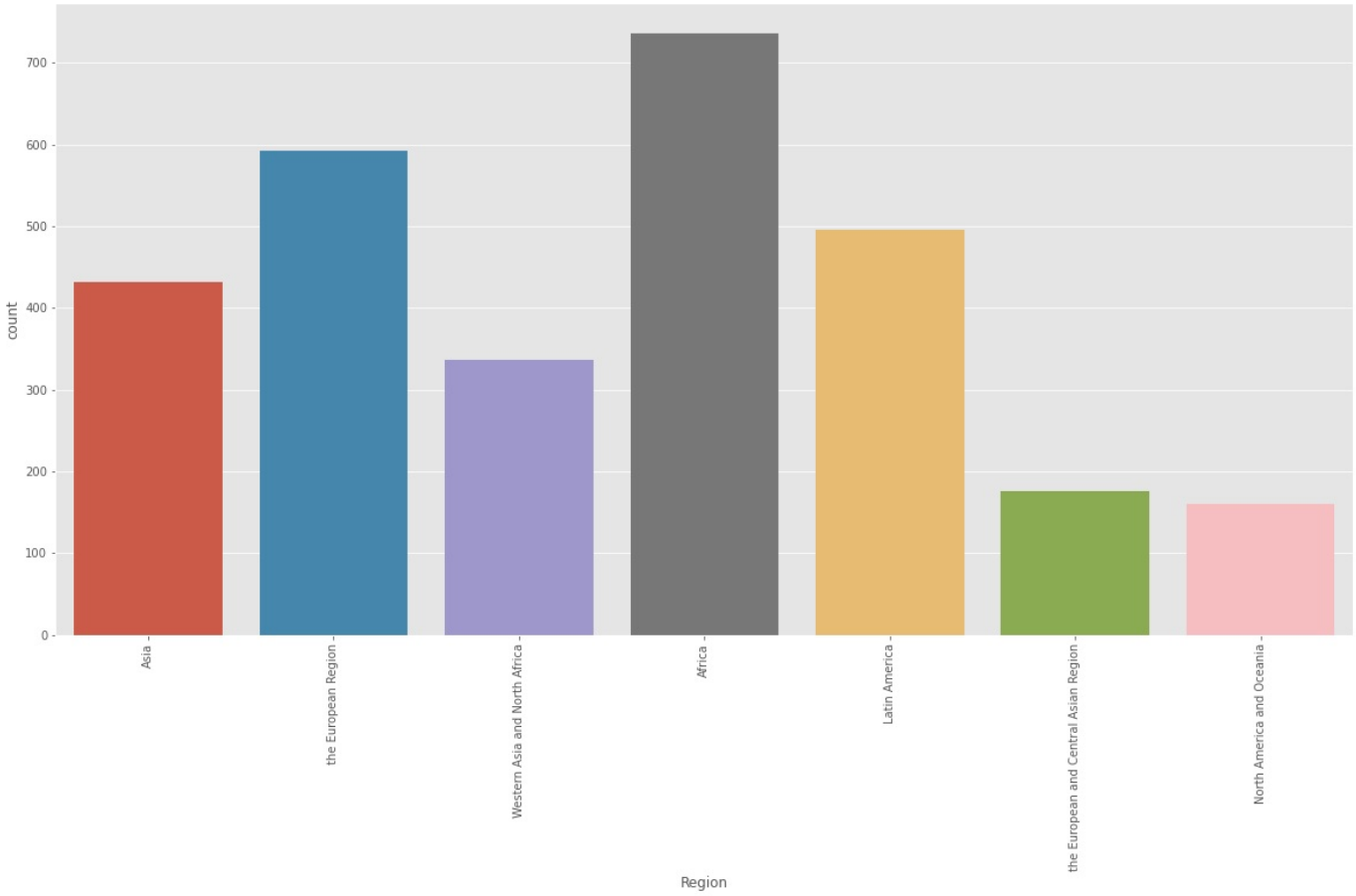
* CATEGORIC COLUMNS UNIVARIATE ANALYSIS

i. COUNTPLOT

In [47]:
```
plot_univariate_graphs(life_expectancy_data,factor_cols,'c')
```







- MORE DEVELOPING COUNTRIES ARE PRESENT THAN THAT OF DEVELOPED ONES

- AFRICA AND EUROPEAN REGIONS ARE IN MAJORITY

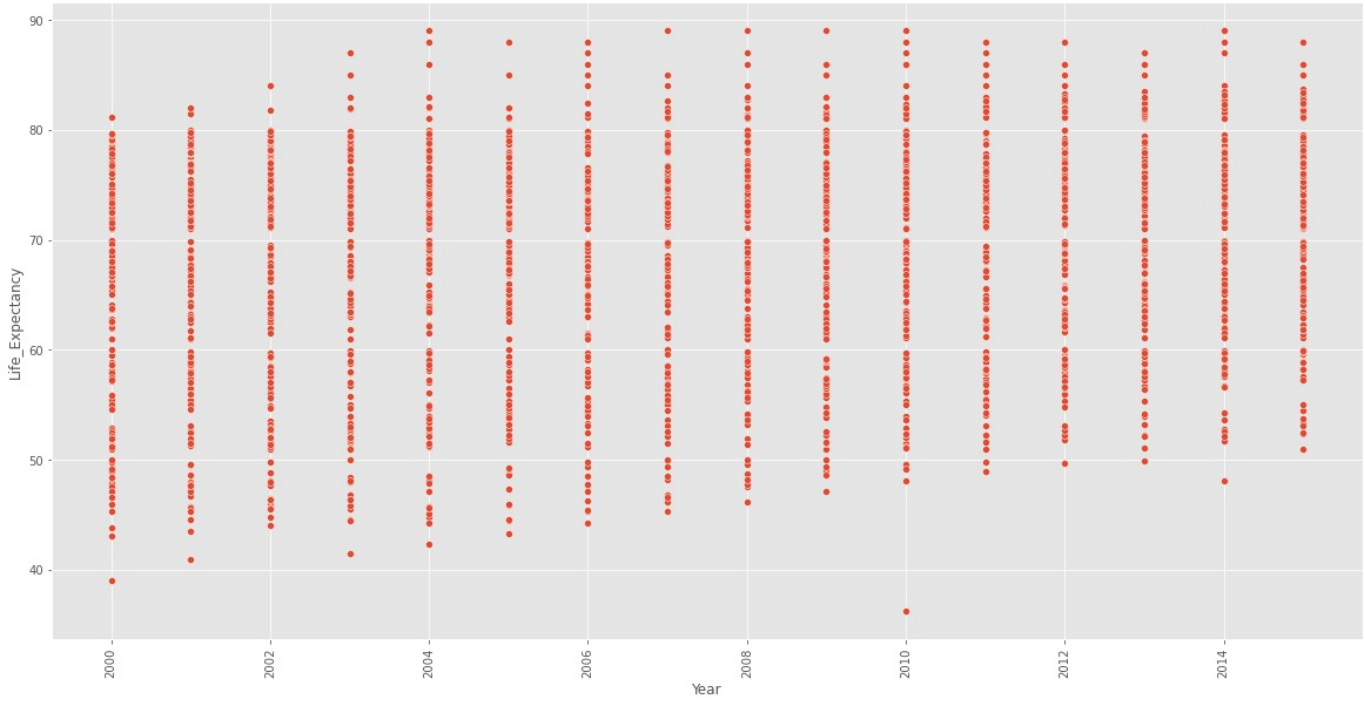
II. BIVARIATE ANALYSIS - ANALYSIS OF TWO VARIABLES

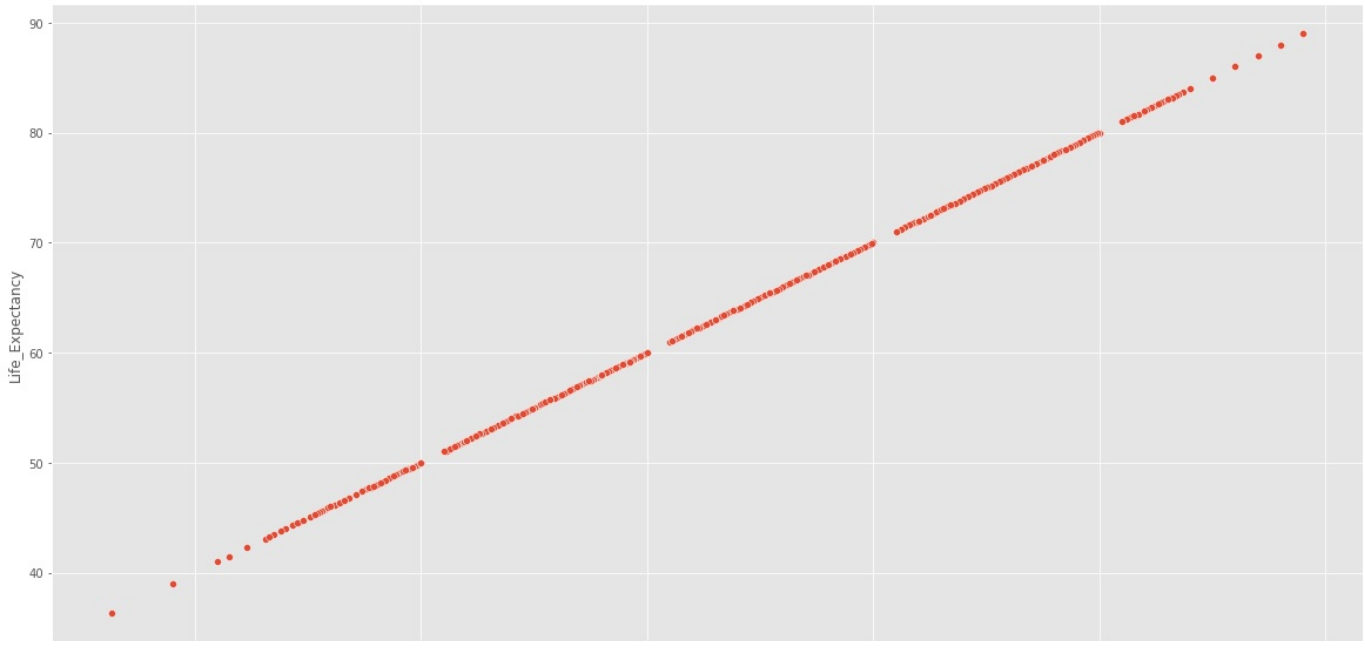
In [48]:
```
def plot_bivariate_graphs(data,cols,type_graph):
    for c in cols:
        plt.figure(figsize = (20,10))
        if type_graph == 's':
            sns.scatterplot(x=c,y='Life_Expectancy',data = data )
            plt.xticks(rotation=90)
        elif type_graph == 'bo':
            sns.boxplot(x=c,y='Life_Expectancy',data = data )
            plt.xticks(rotation=90)
        elif type_graph == 'ba':
            sns.barplot(x =c,y = 'Life_Expectancy',data = data)
            plt.xticks(rotation=90)
        elif type_graph == 'c':
            sns.countplot(x = data[c],hue='Region',data = data)
            plt.xticks(rotation=90)
        elif type_graph == 'h':
            sns.histplot(x=data[c],y='Life_Expectancy',hue = 'Region',data = data)
            plt.xticks(rotation=90)
        elif type_graph == 'l':
            sns.lineplot(x=data[c],y='Life_Expectancy',data = data)
            plt.xticks(rotation=90)
    plt.show()
```

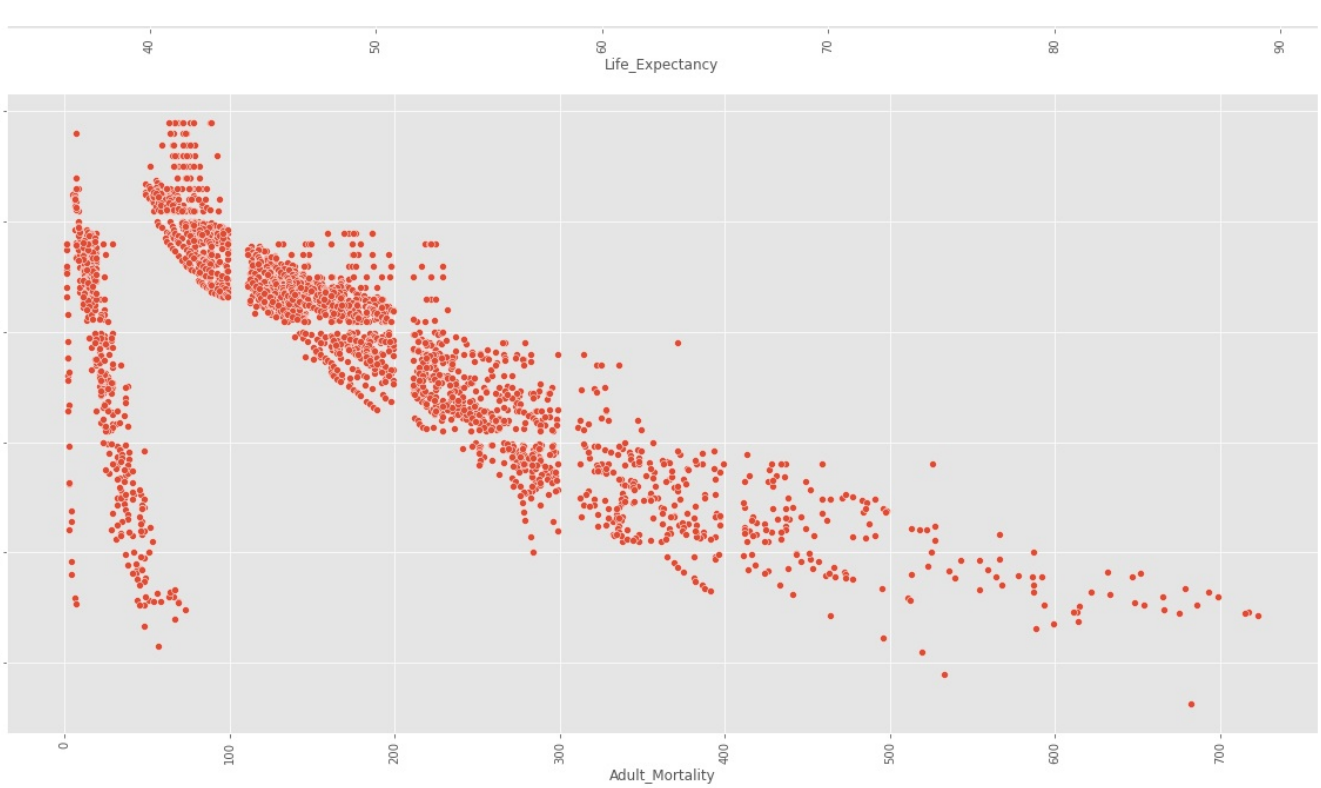
* BIVARIATE ANALYSIS (NUMERIC - NUMERIC)

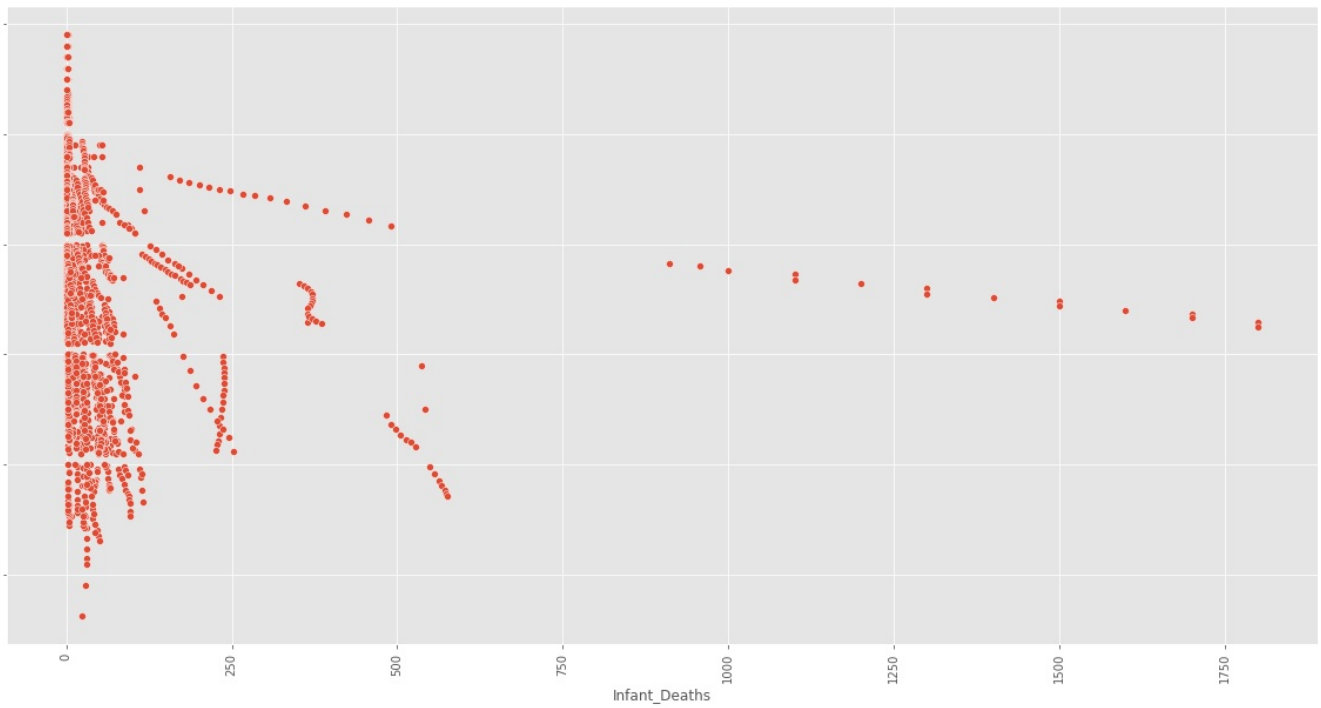
i.SCATTERPLOT

In [49]:
```
plot_bivariate_graphs(life_expectancy_data,numeric_cols,'s')
```

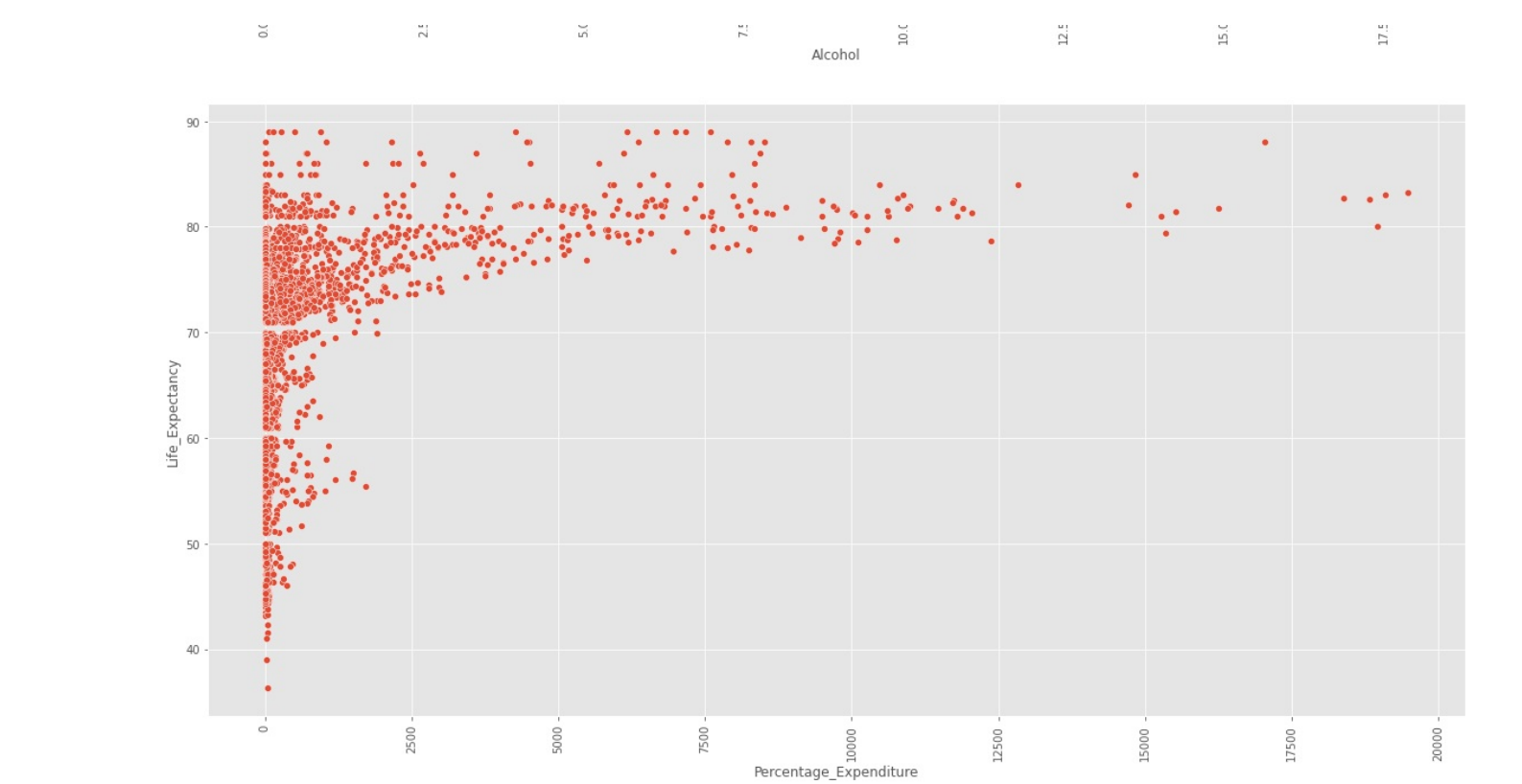





Life_Expectancy
Life_Expectancy
Adult_Mortality

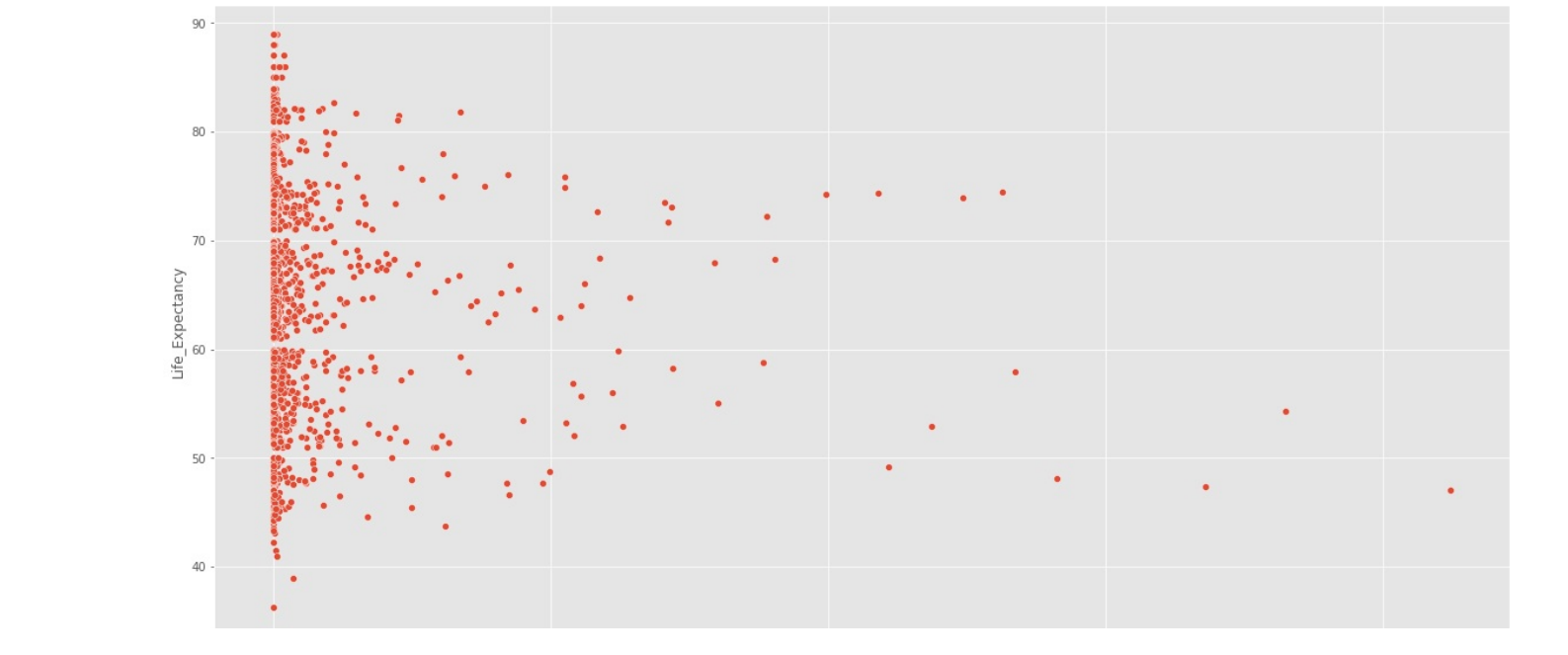

Life_Expectancy
Infant_Deaths


Life_Expectancy

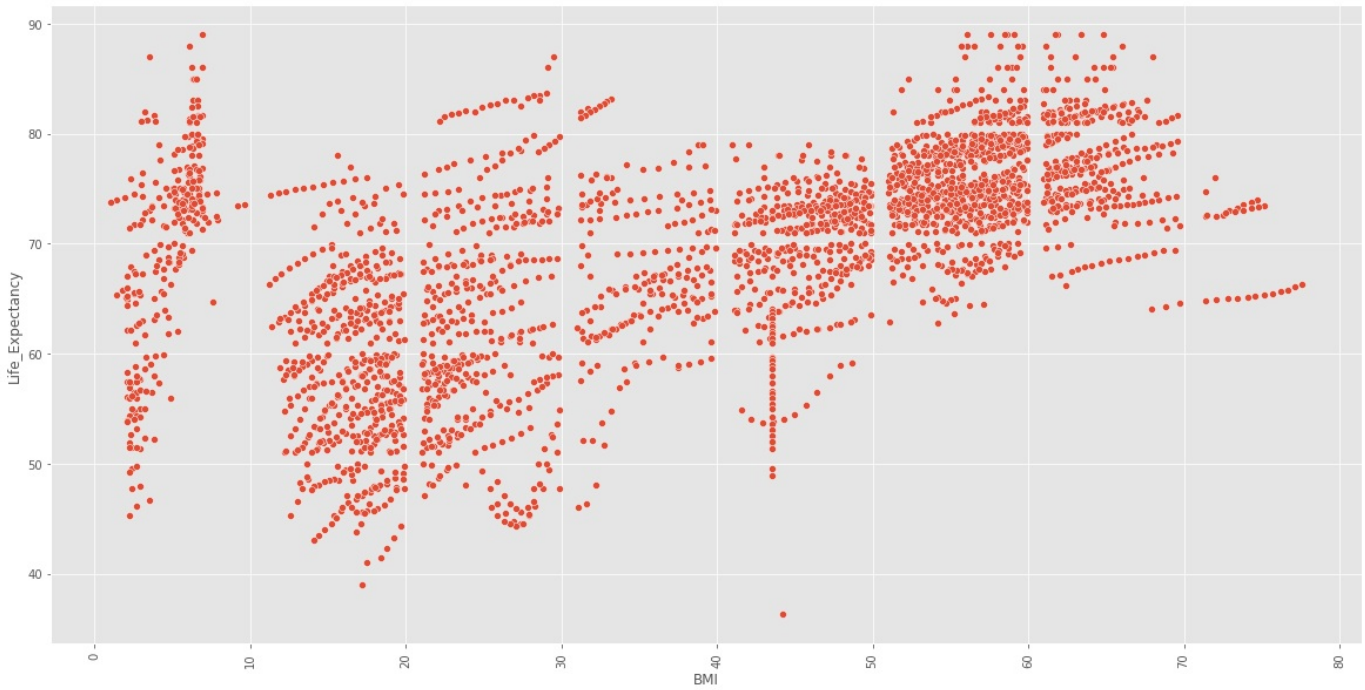

Alcohol

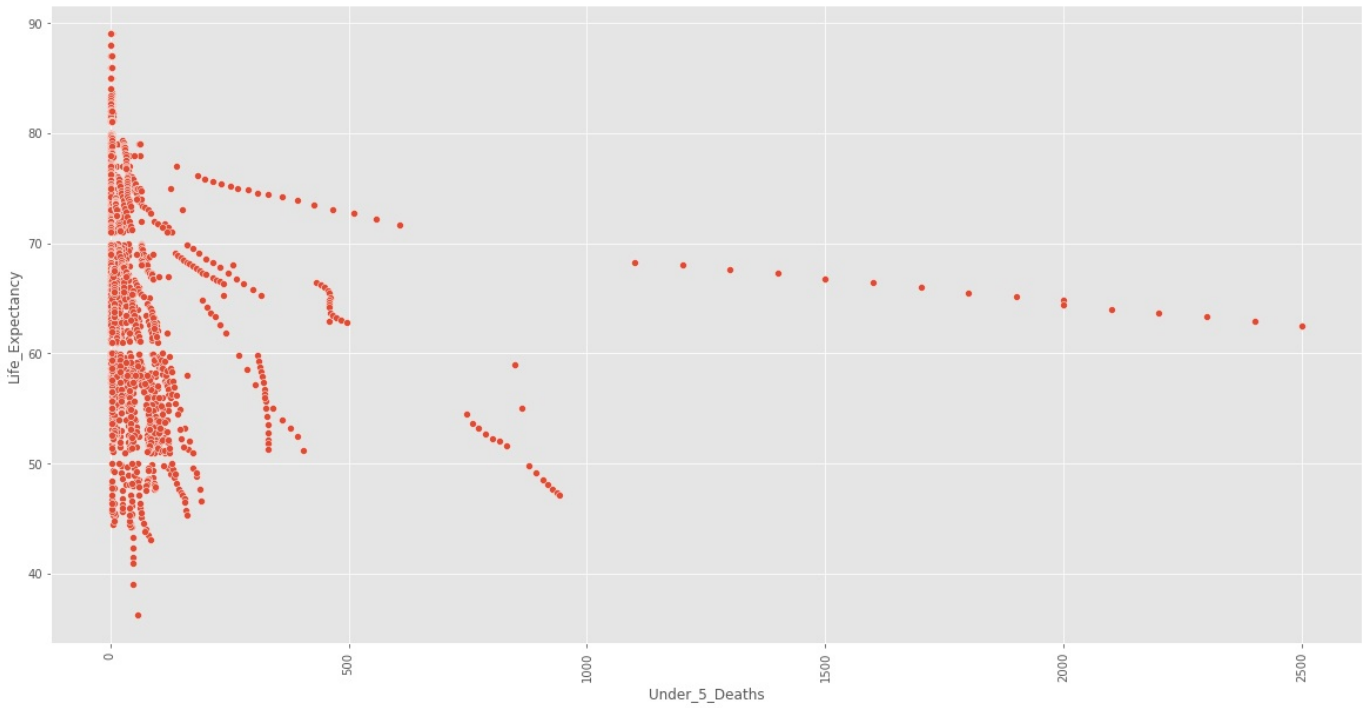
Life_Expectancy
Percentage_Expenditure

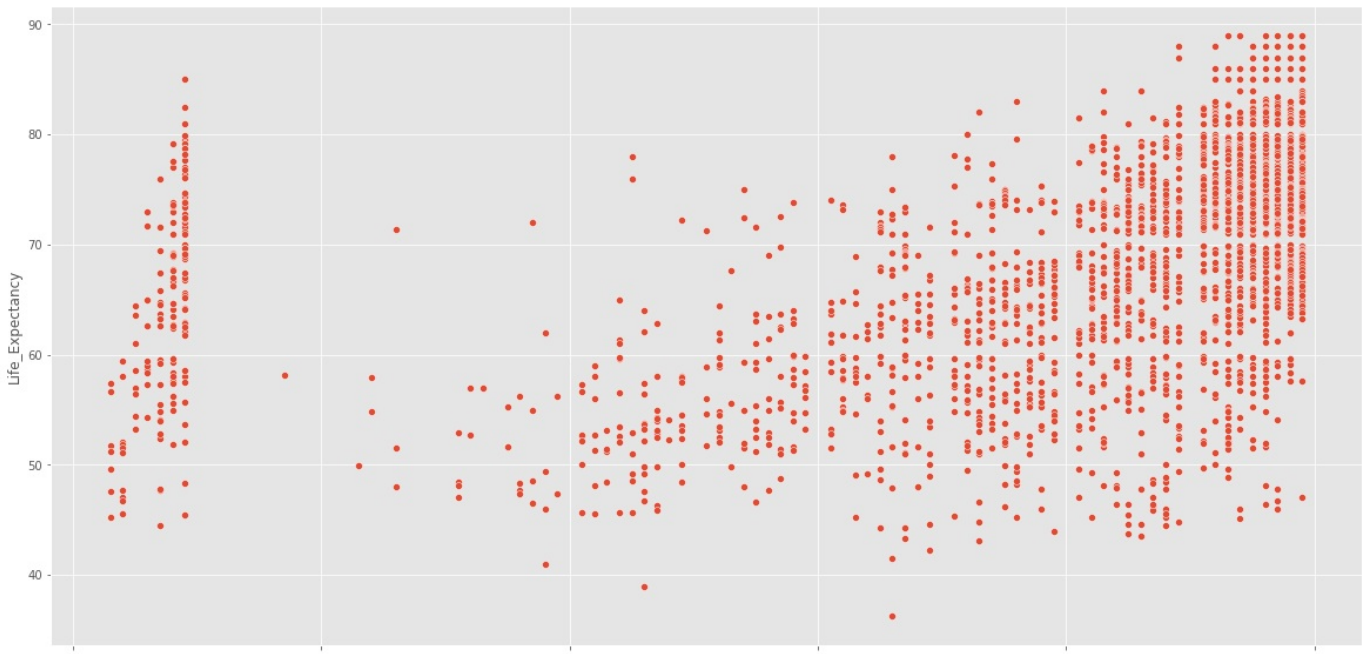

Life_Expectancy
Hepatitis_B


Life_Expectancy

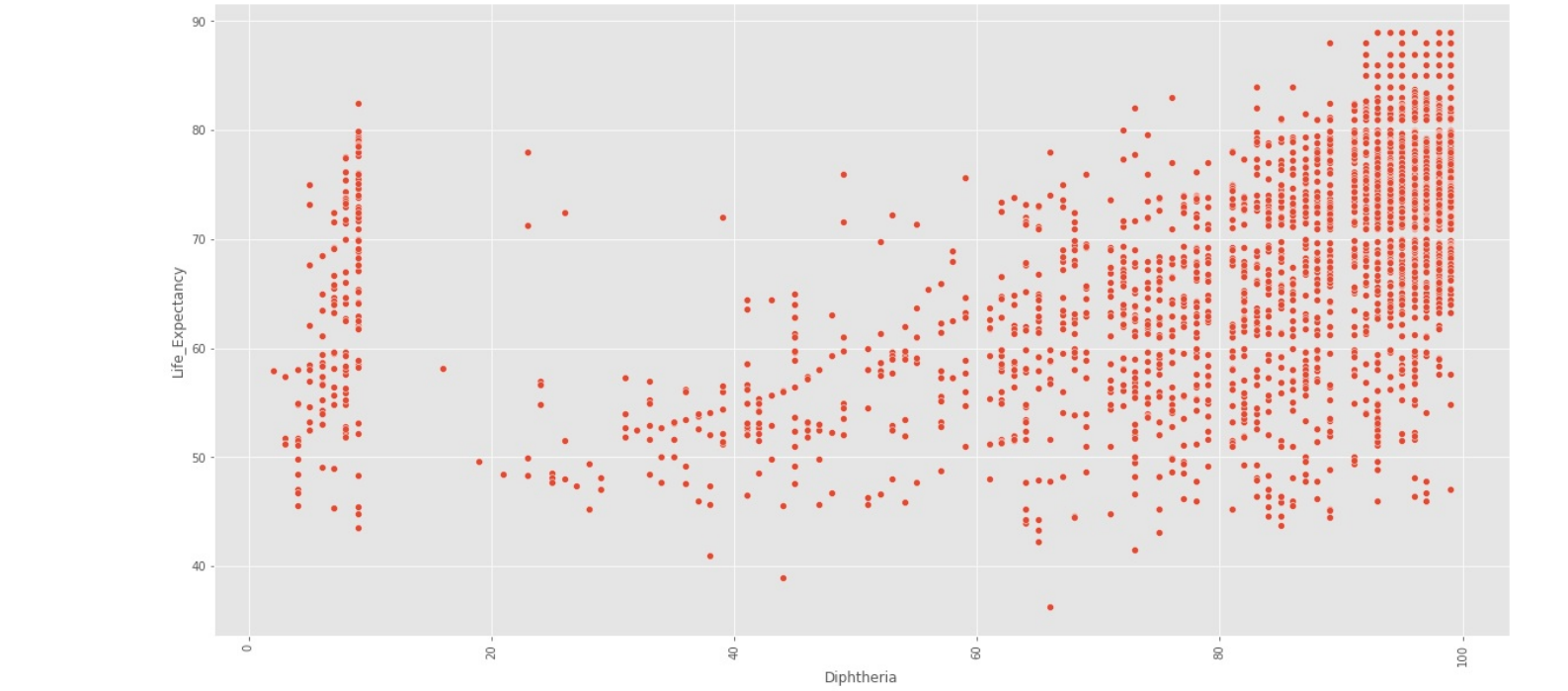

0
50000
100000
150000
200000
Measles

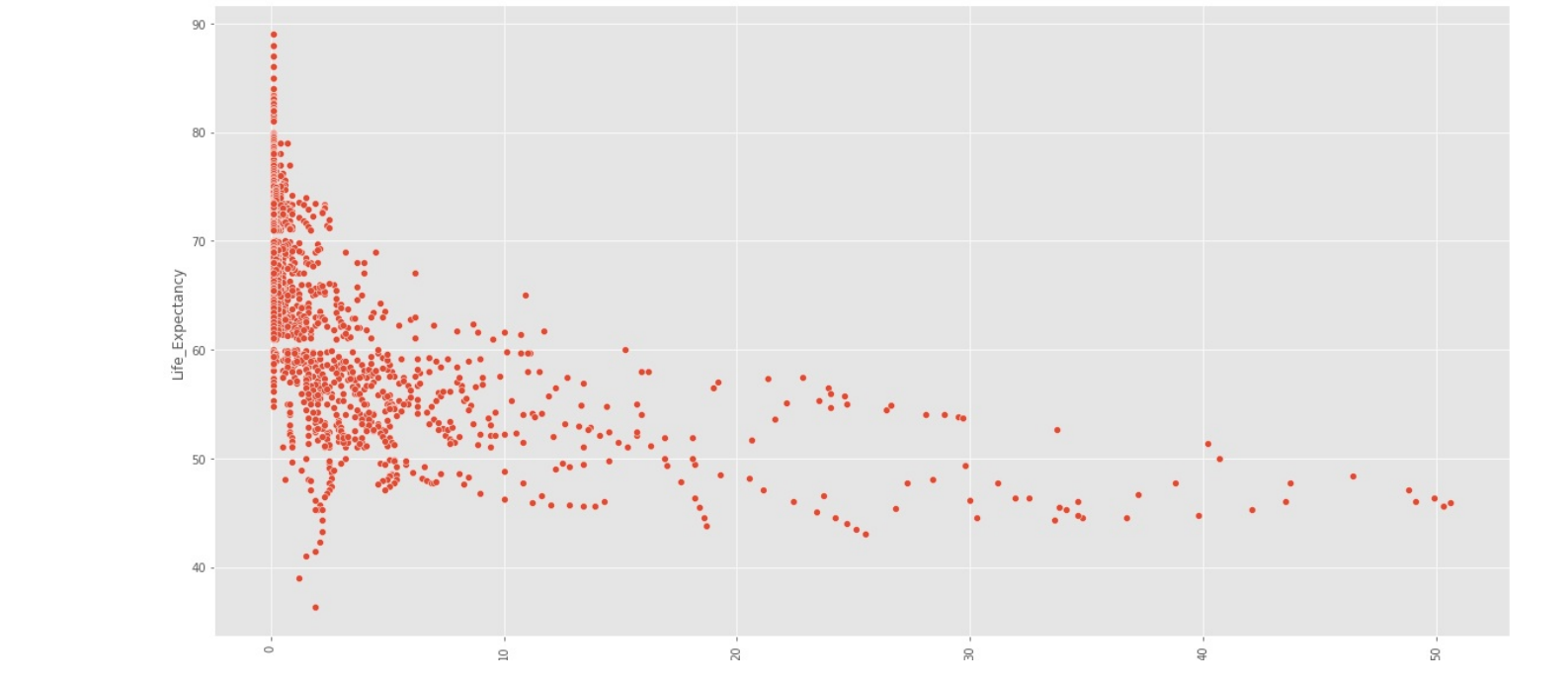

Life_Expectancy
90
80
70
60
50
40
0
10
20
30
40
50
60
70
80
BMI


Life_Expectancy
90
80
70
60
50
40
0
500
1000
1500
2000
2500
Under_5_Deaths

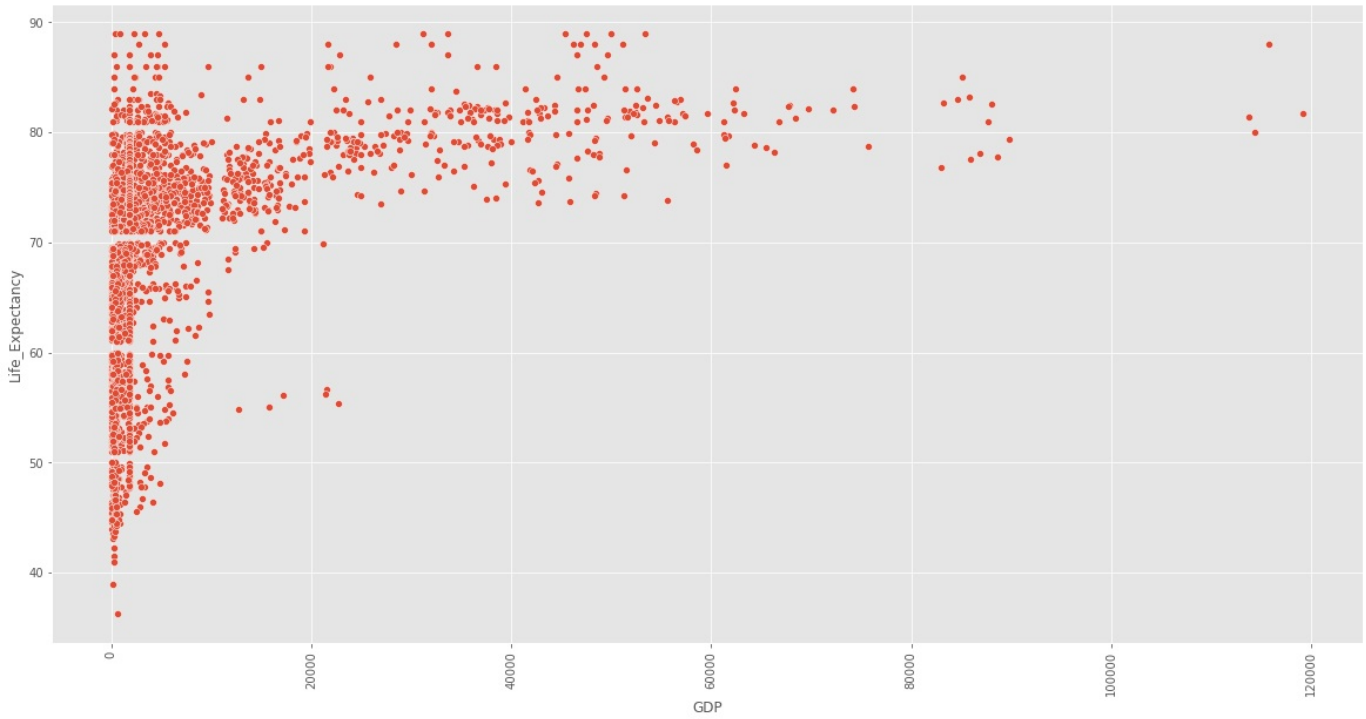

Life_Expectancy
90
80
70
60
50
40

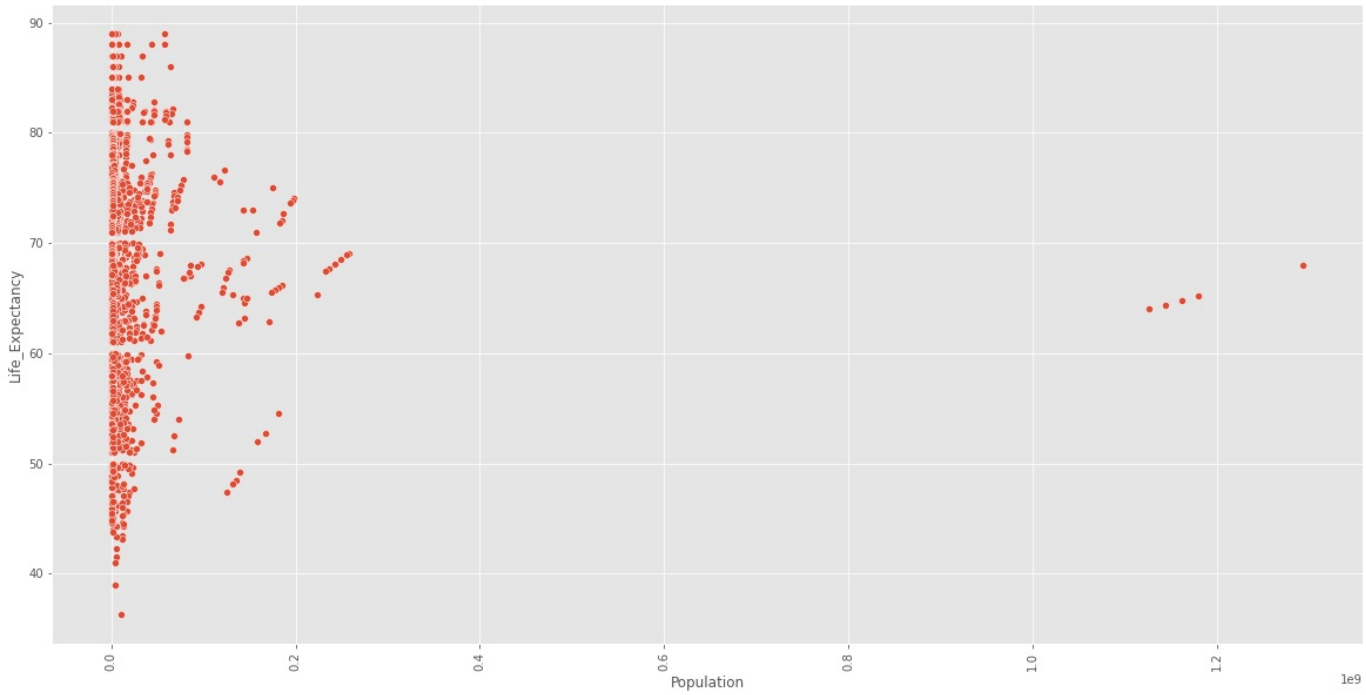

Polio
Life_Expectancy
Total_Expenditure

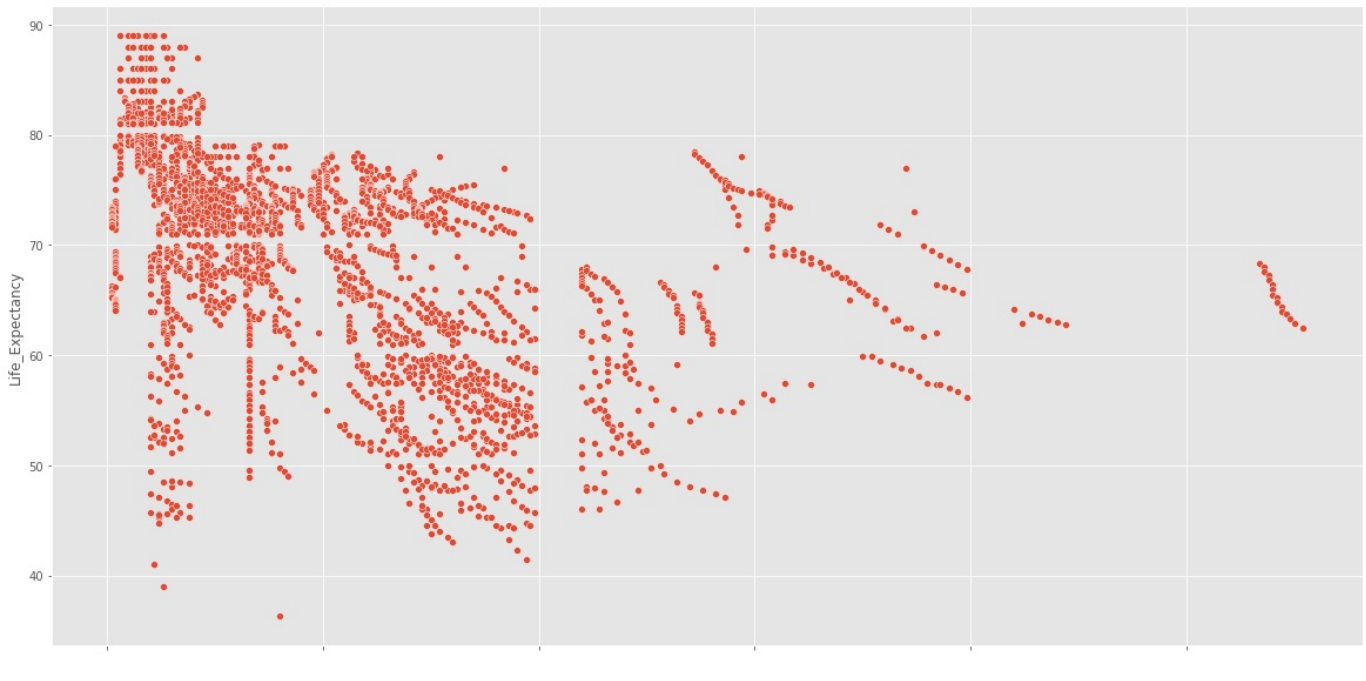

Life_Expectancy
Diphtheria

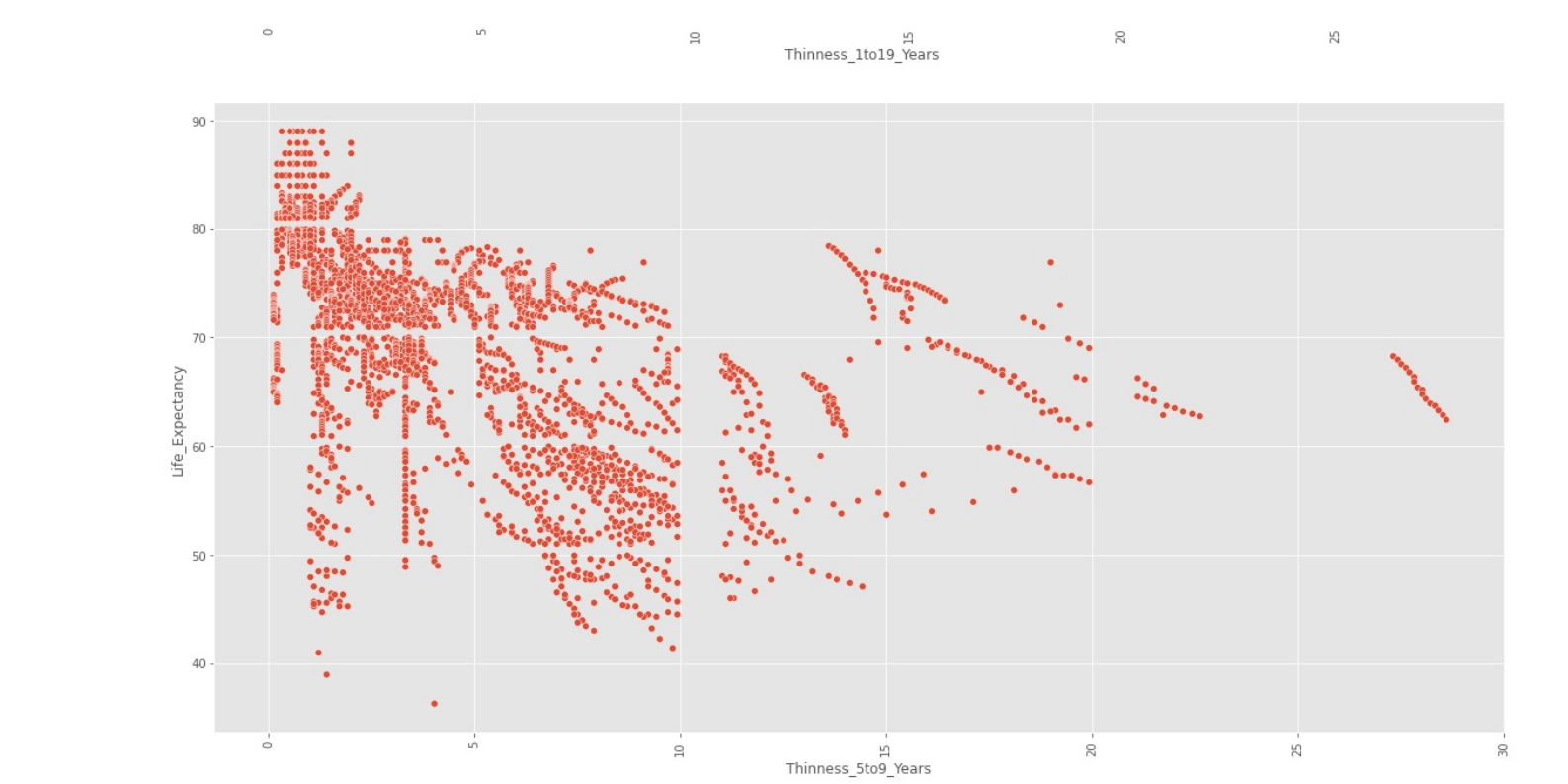

Life_Expectancy

HIV/AIDS

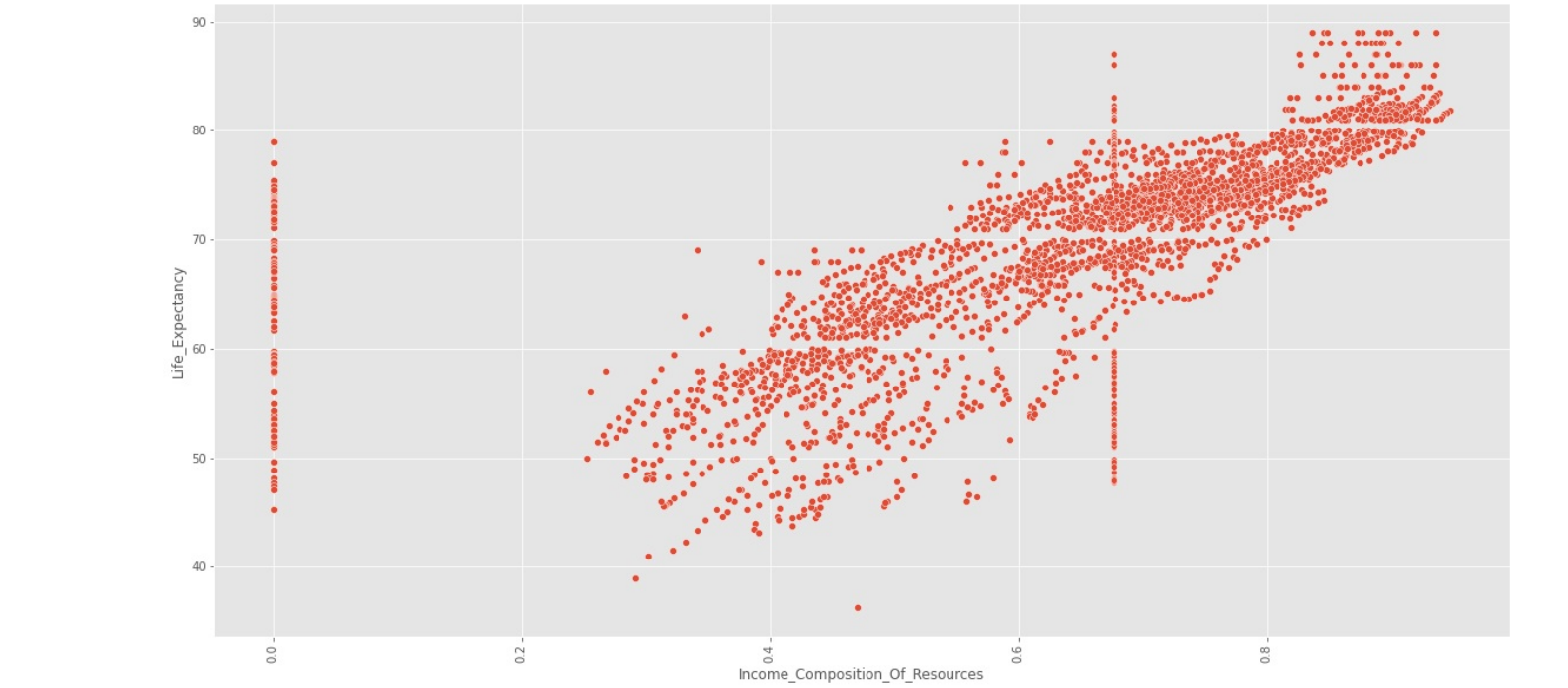

Life_Expectancy
90
80
70
60
50
40
0
20000
40000
60000
80000
100000
120000
GDP

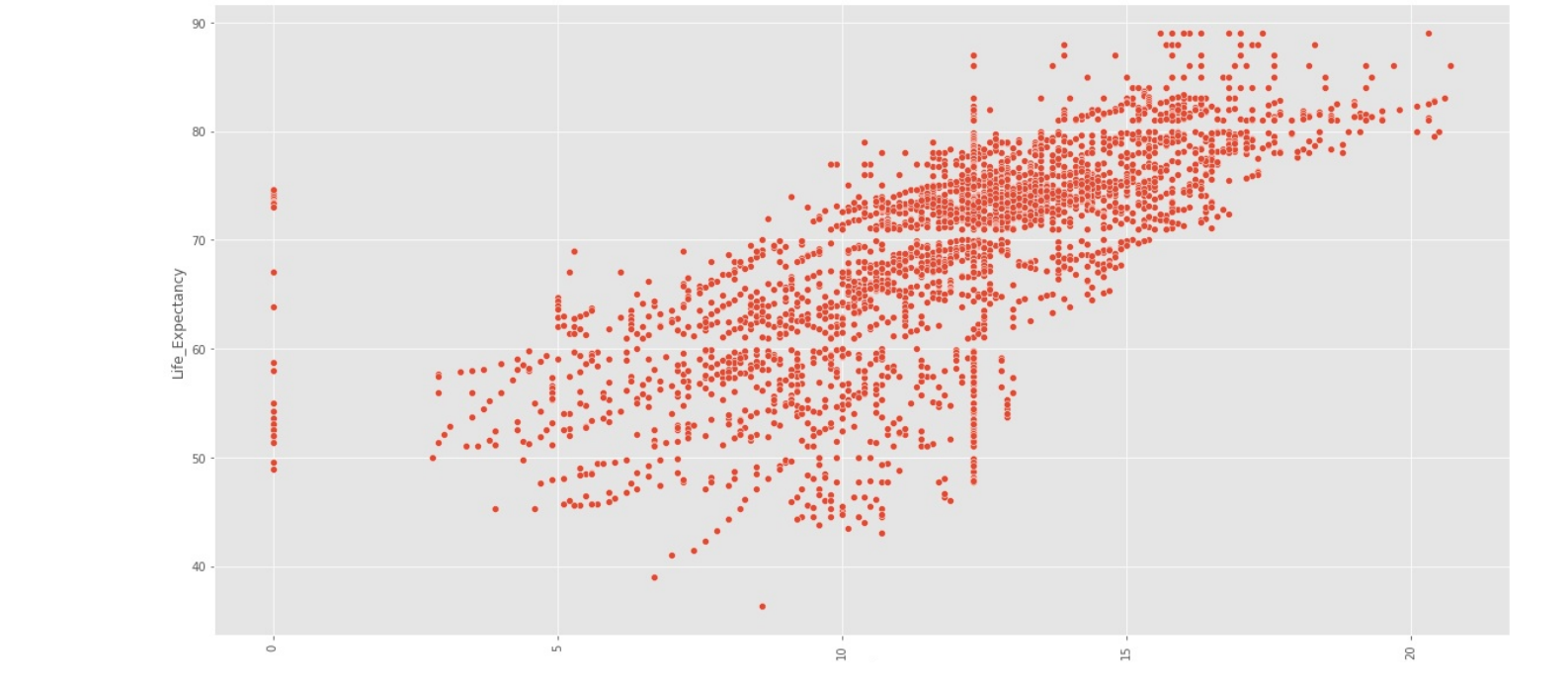

Life_Expectancy
90
80
70
60
50
40
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Population
1e9


Life_Expectancy
90
80
70
60
50
40


Thinness_1to19_Years

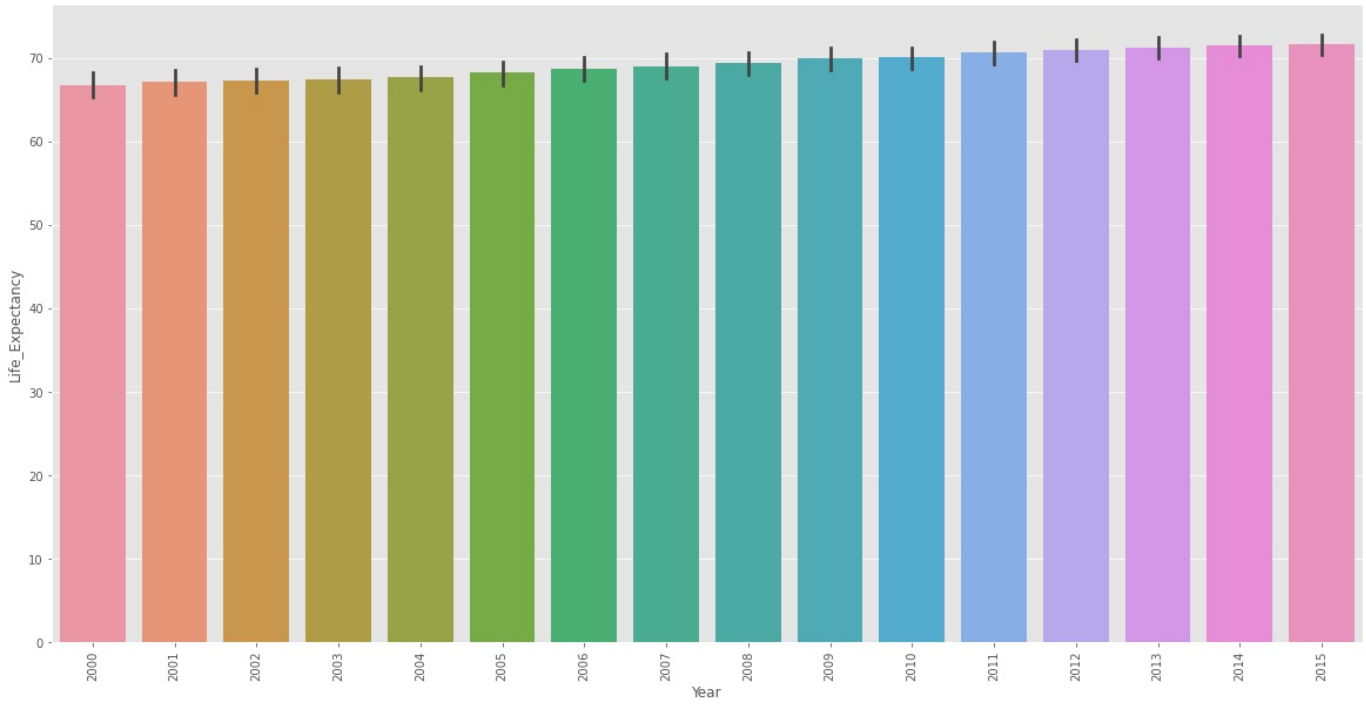
Life_Expectancy
Thinness_5to9_Years

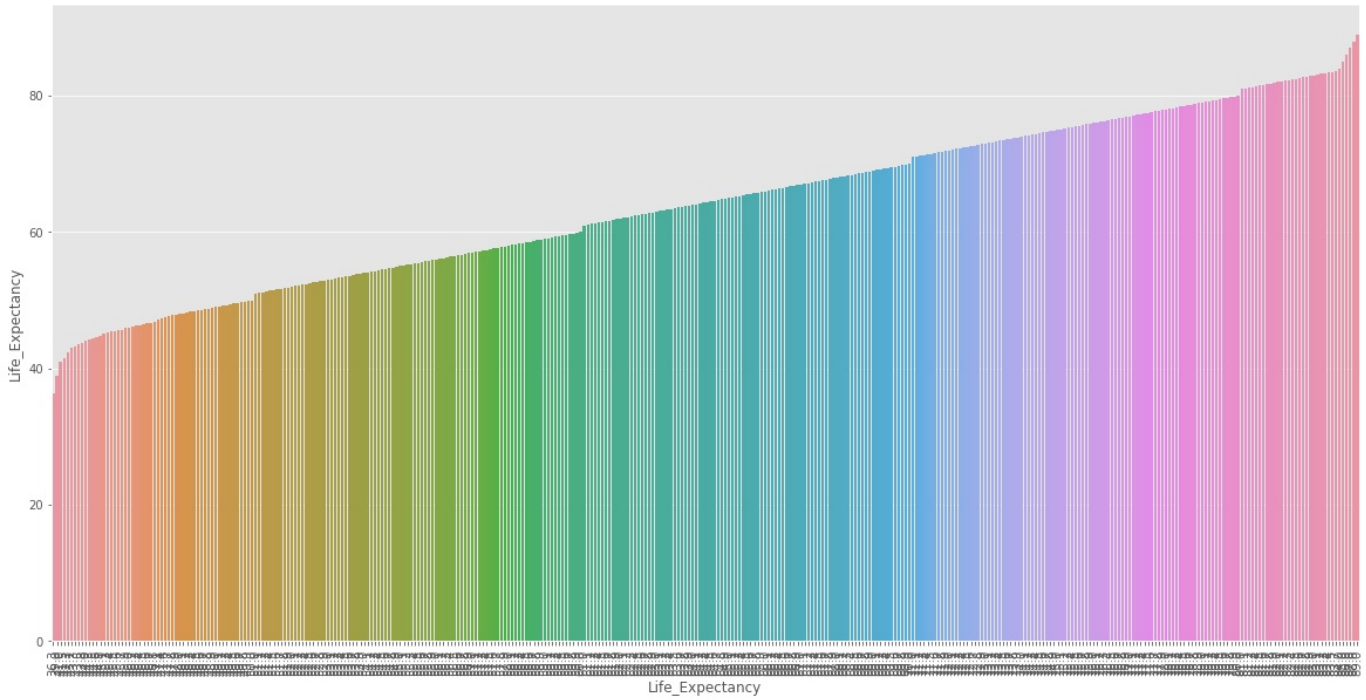

Life_Expectancy
Income_Composition_Of_Resources

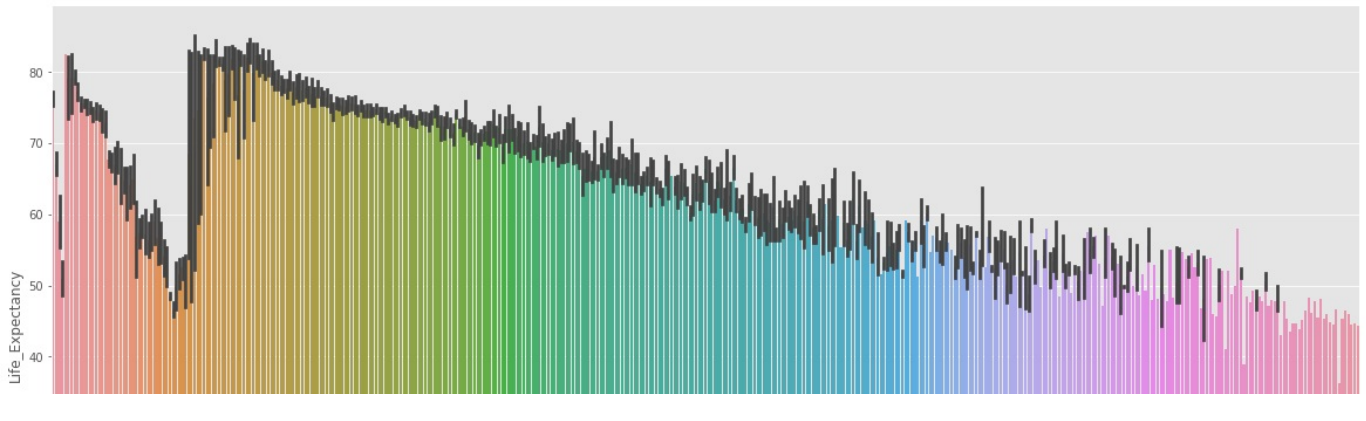

Life_Expectancy

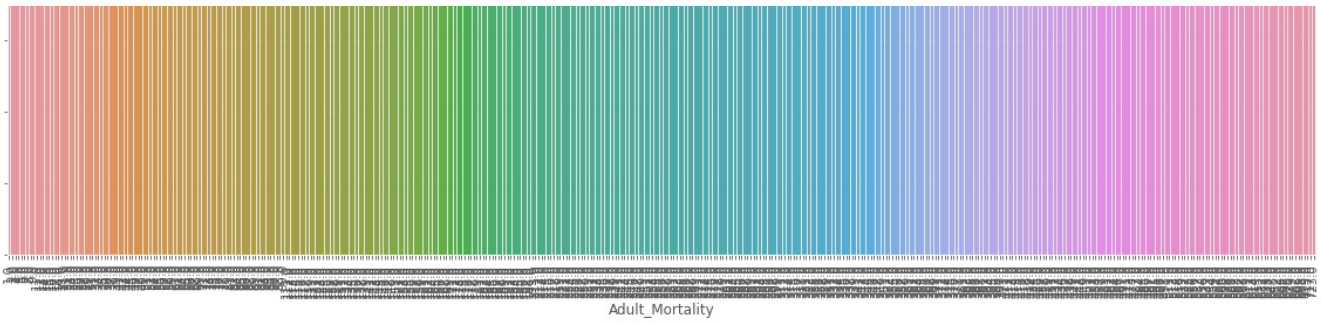
ii. BARPLOT

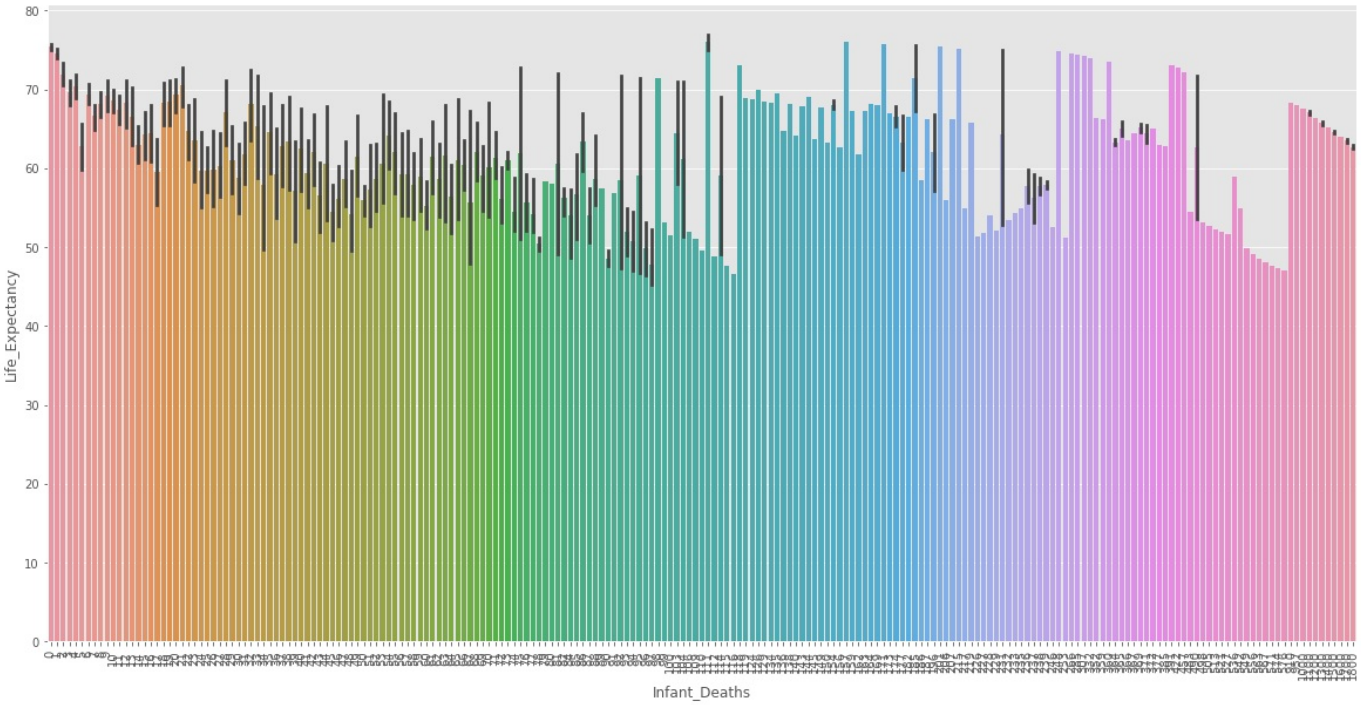
In [50]:
```
plot_bivariate_graphs(life_expectancy_data,numeric_cols,'ba')
```

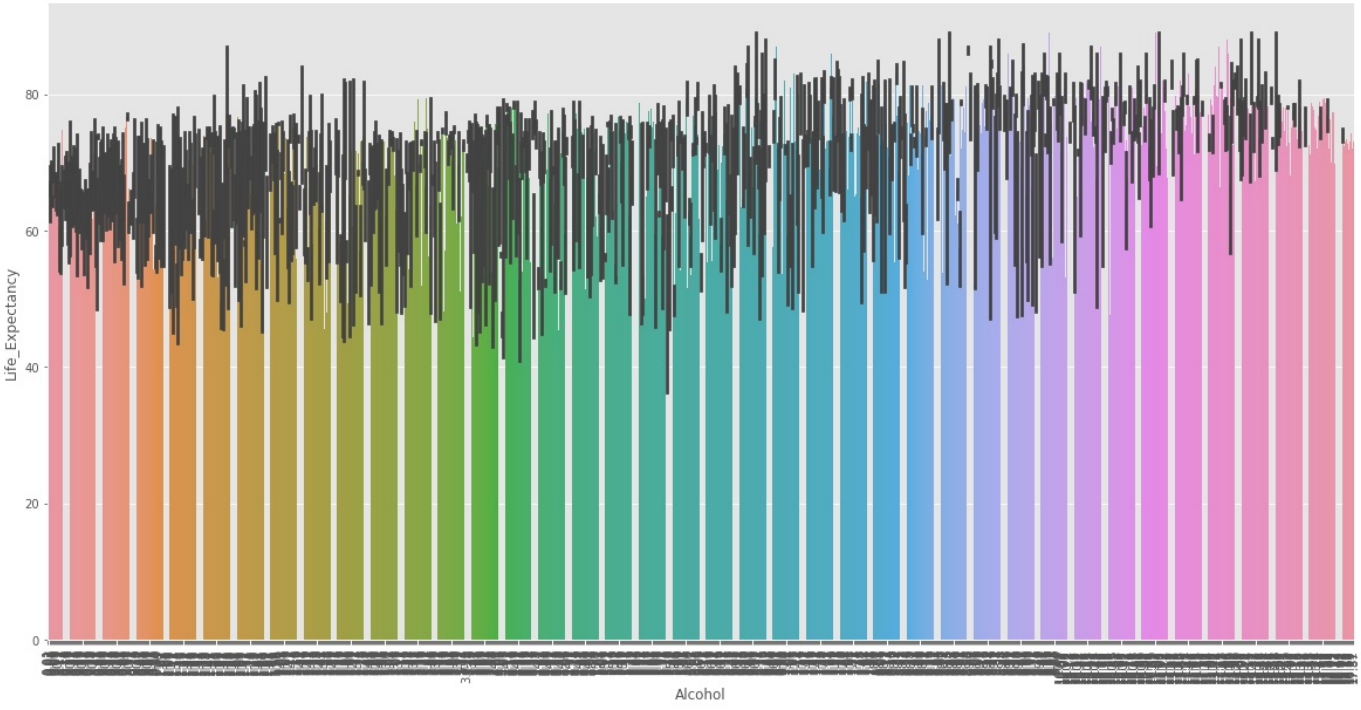


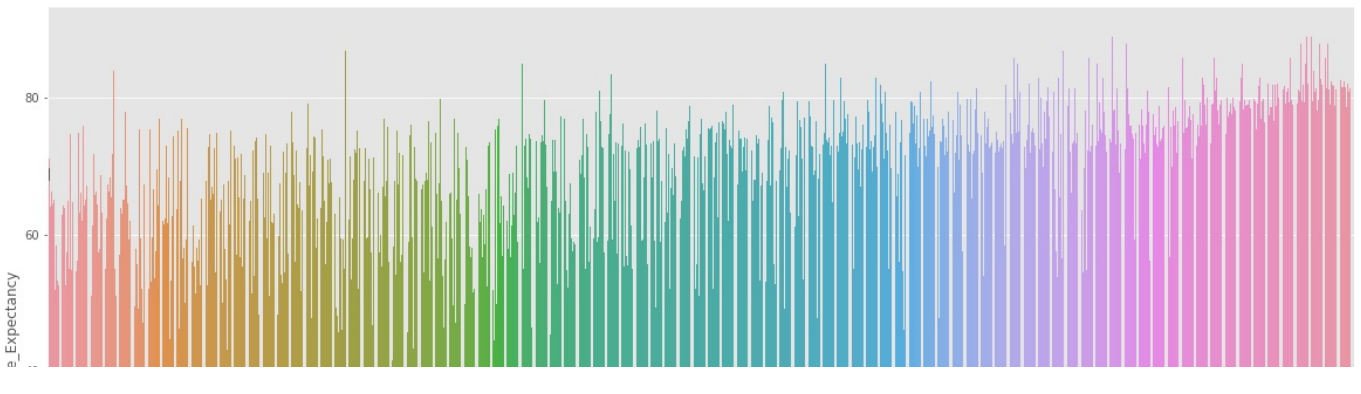


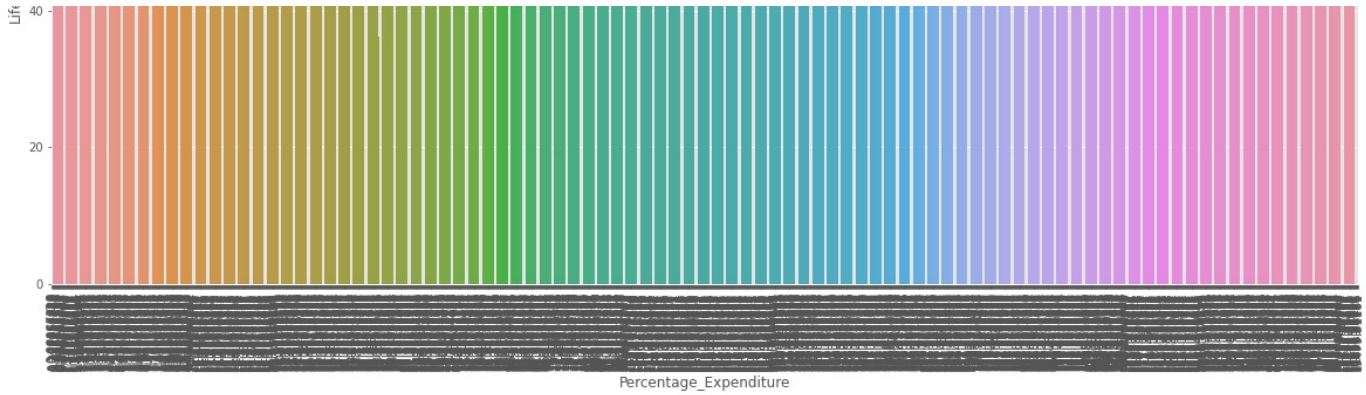


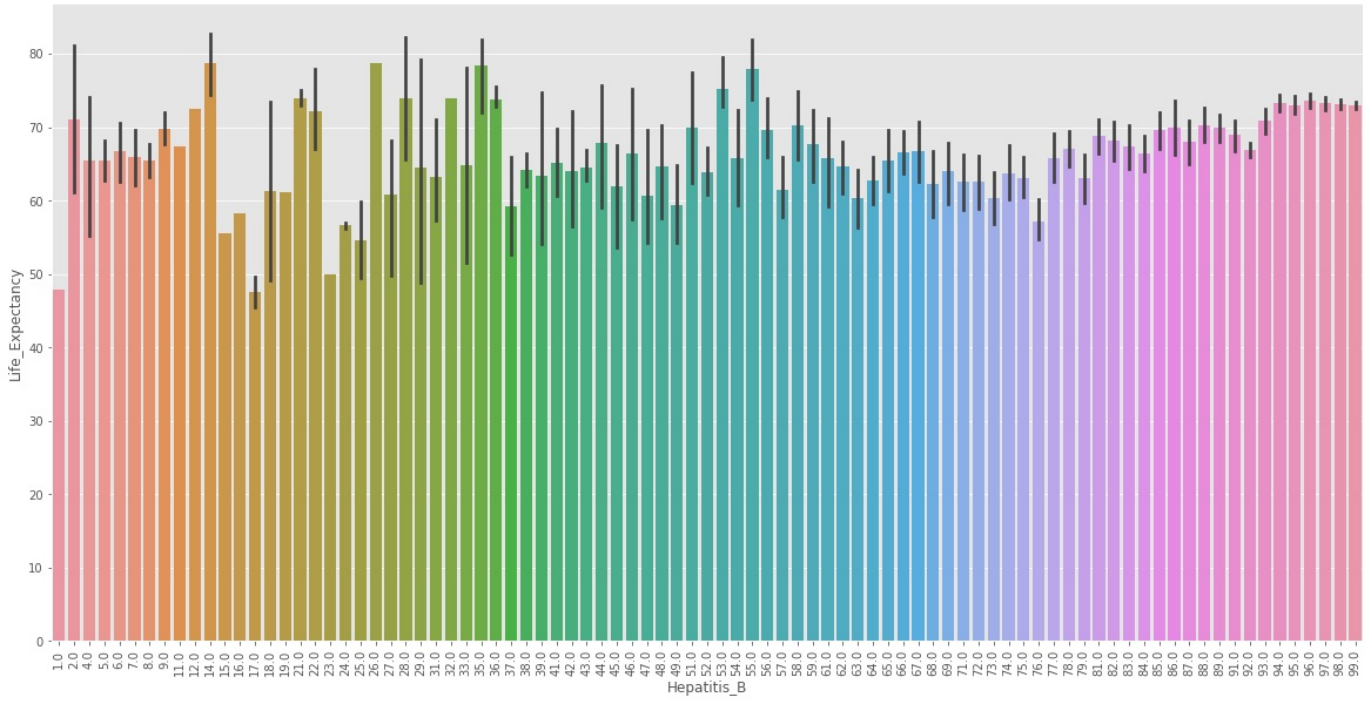

Adult_Mortality

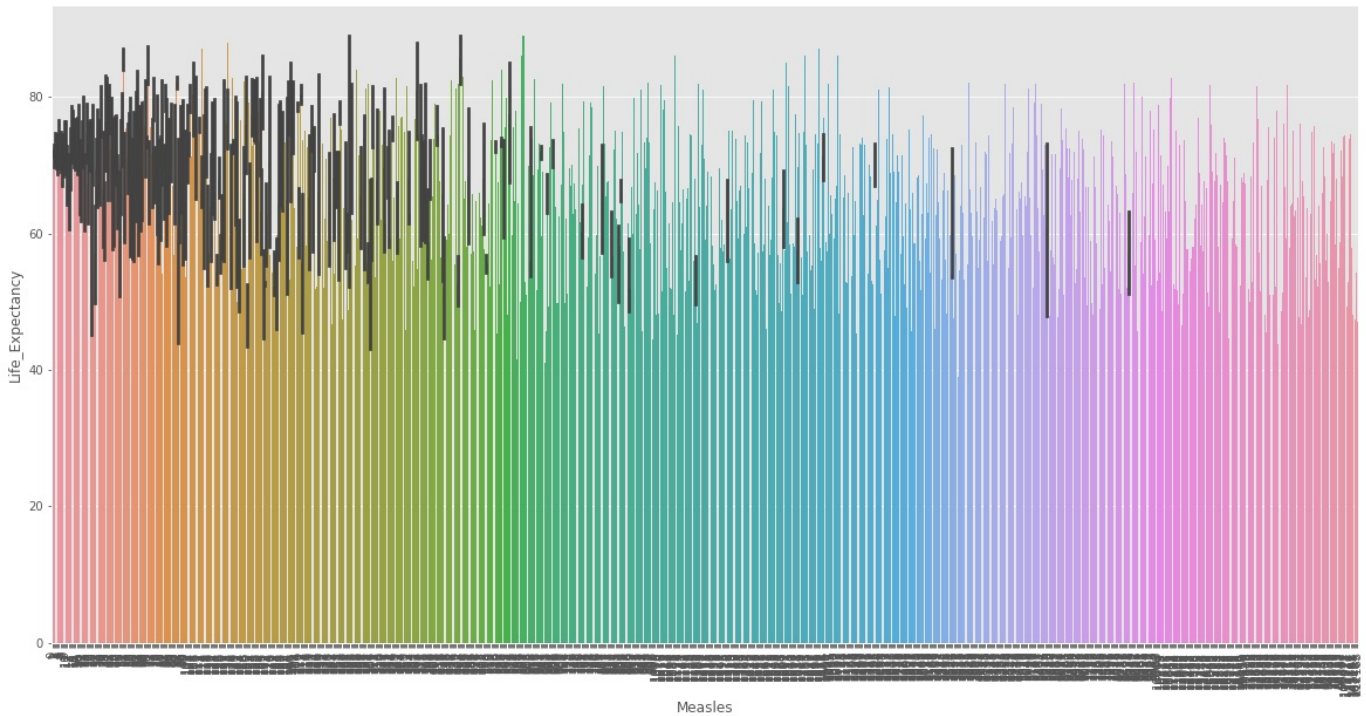

Life_Expectancy
Infant_Deaths

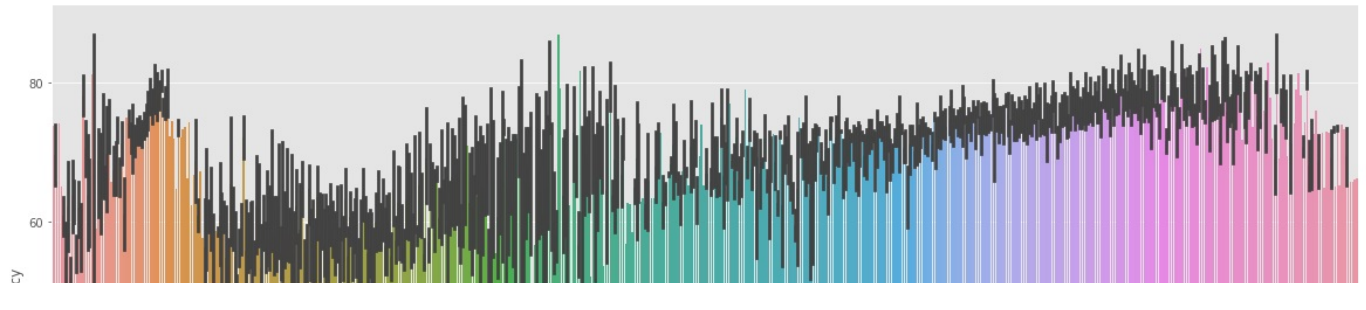

Life_Expectancy
Alcohol

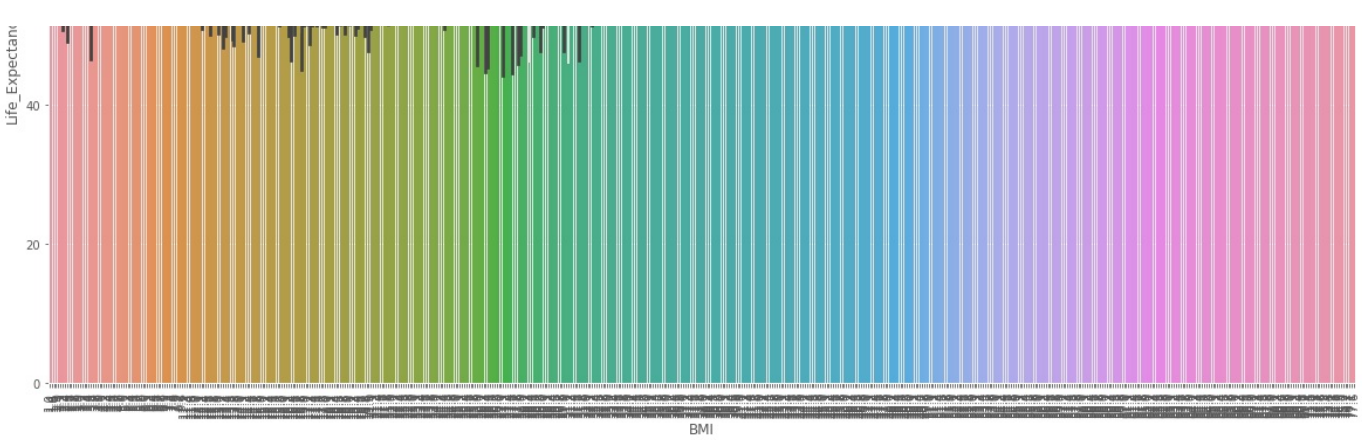

Life_Expectancy

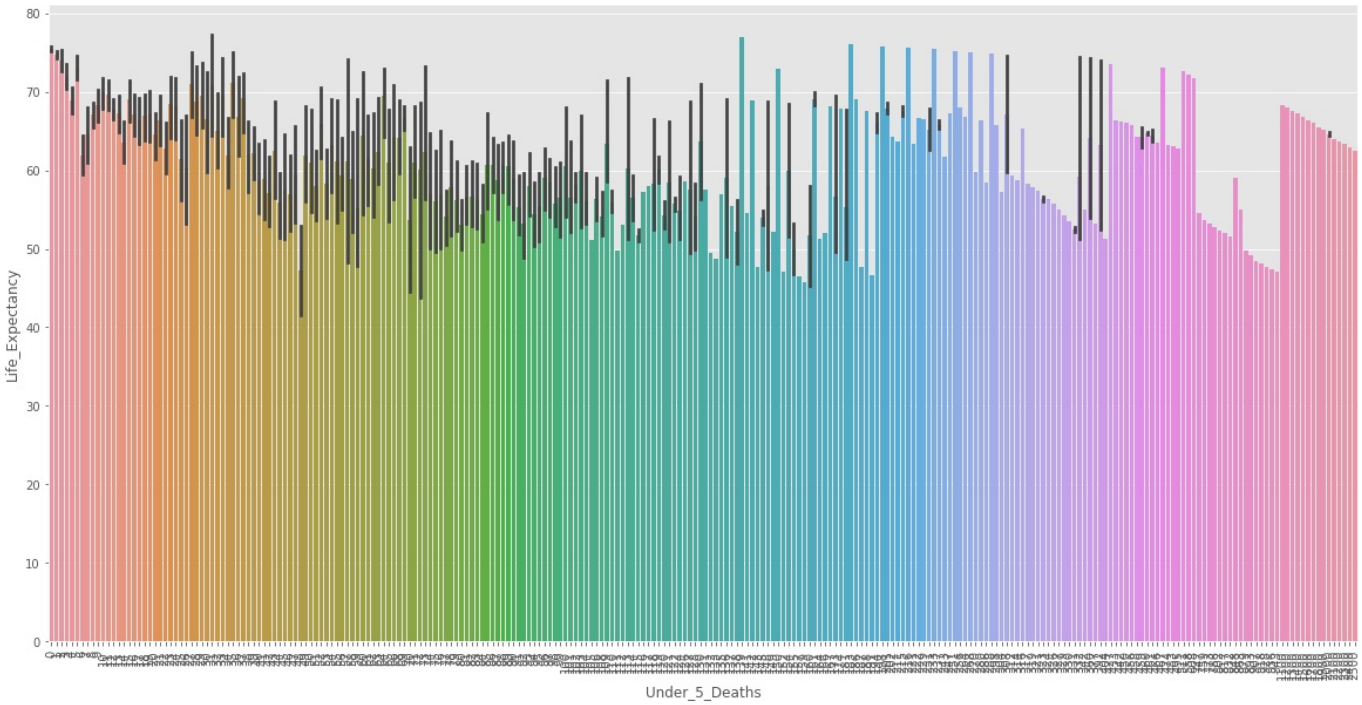

Percentage_Expenditure

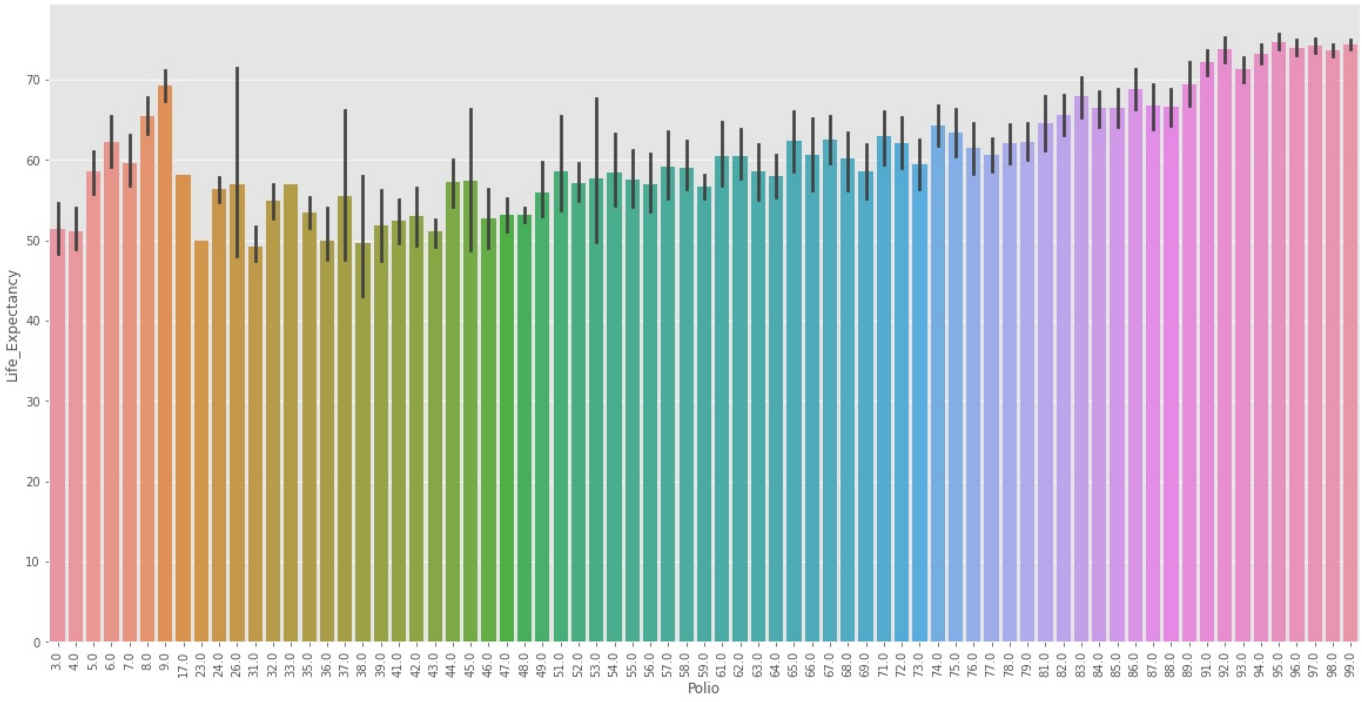

Life_Expectancy
Hepatitis_B

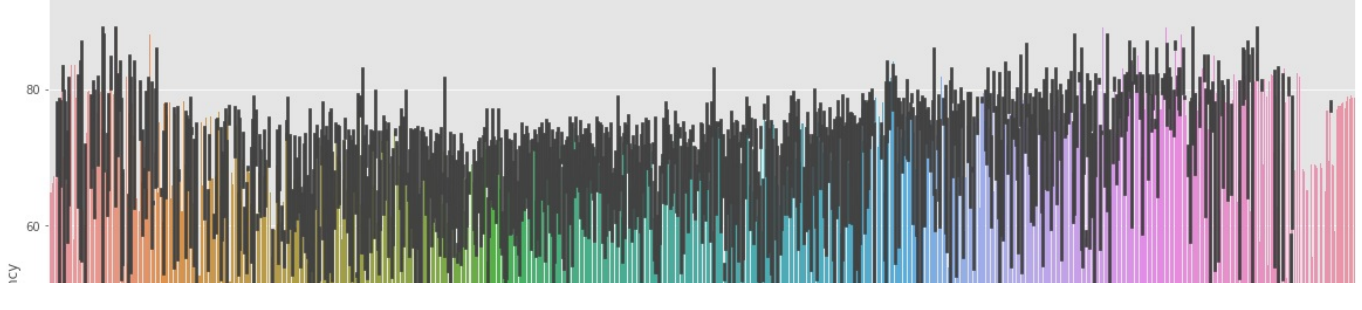

Life_Expectancy
Measles

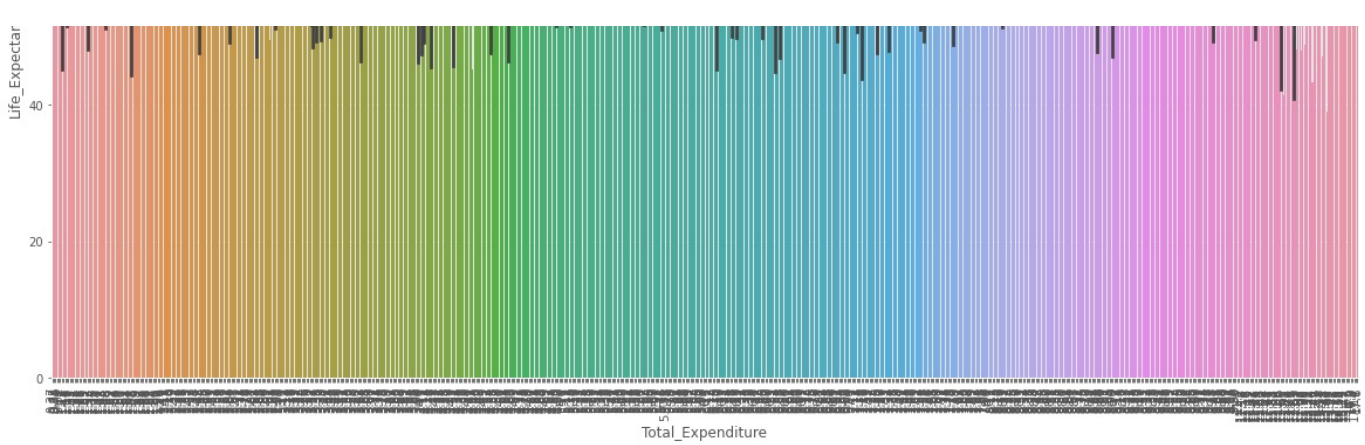


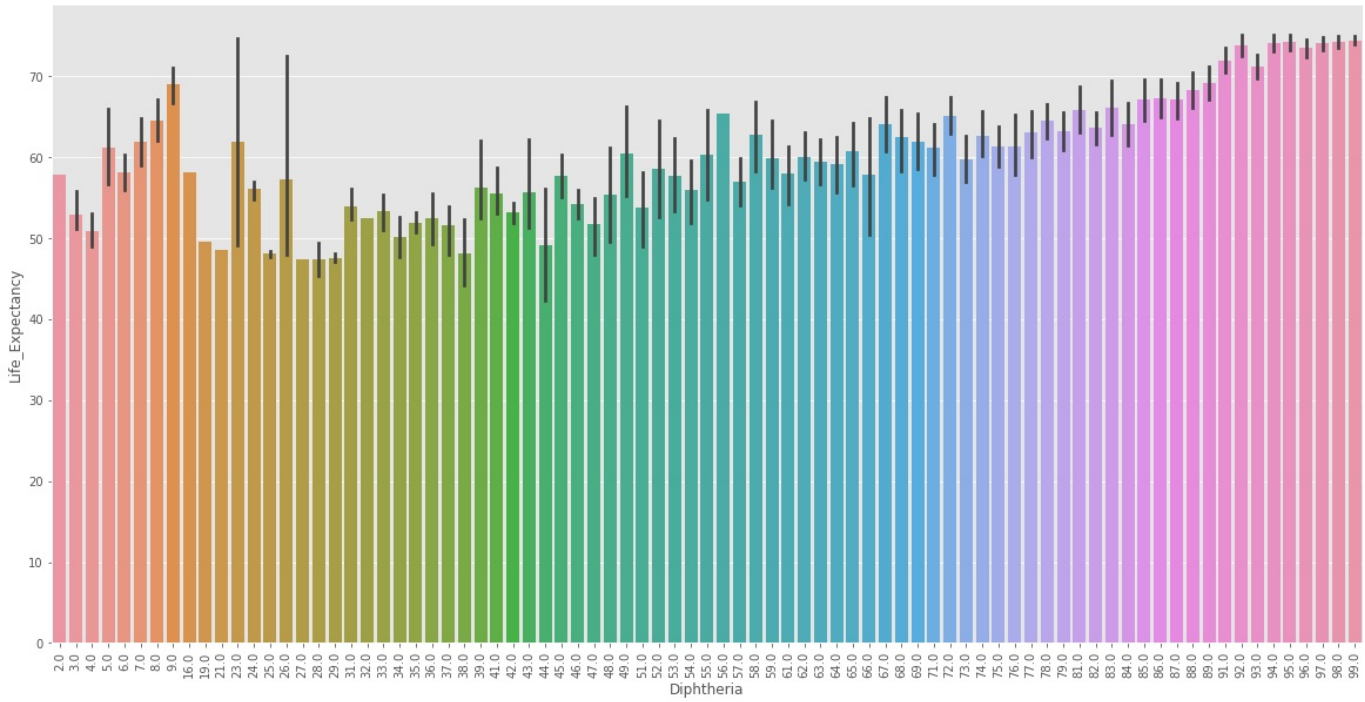

Life_Expectanc
BMI

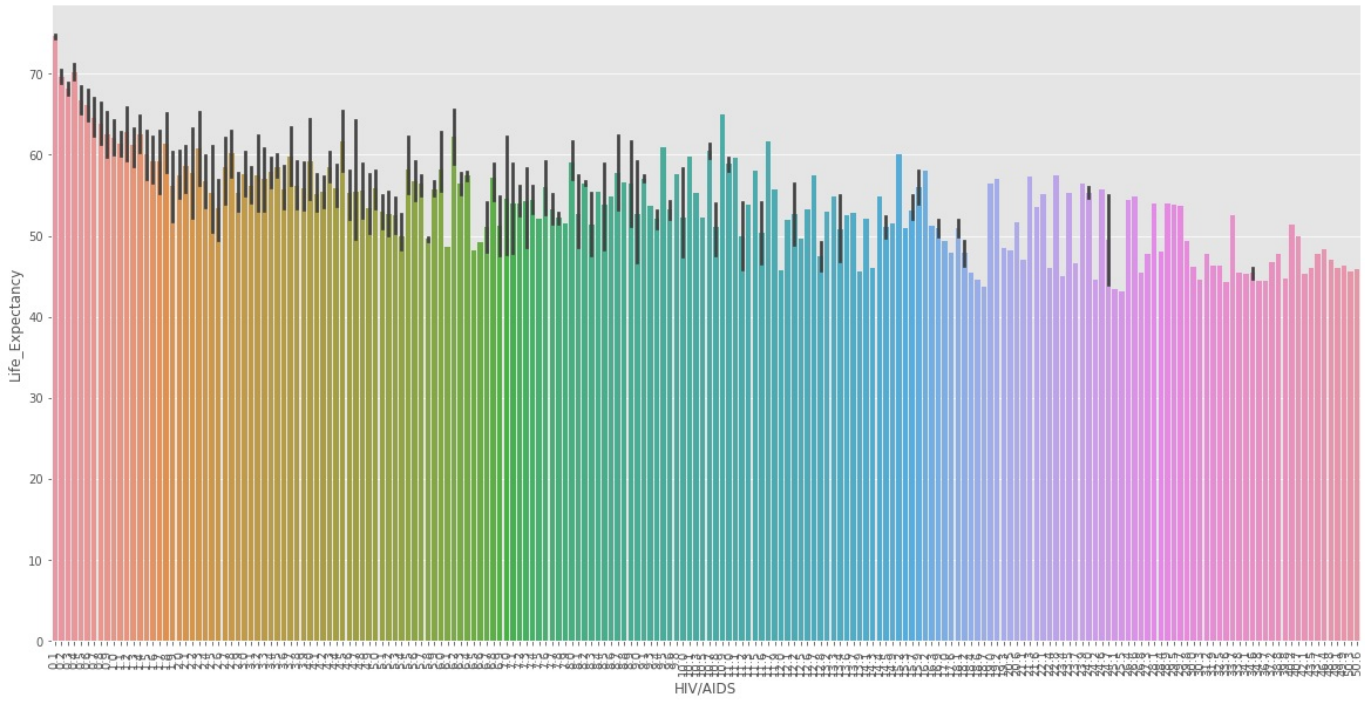

Life_Expectancy
Under_5_Deaths

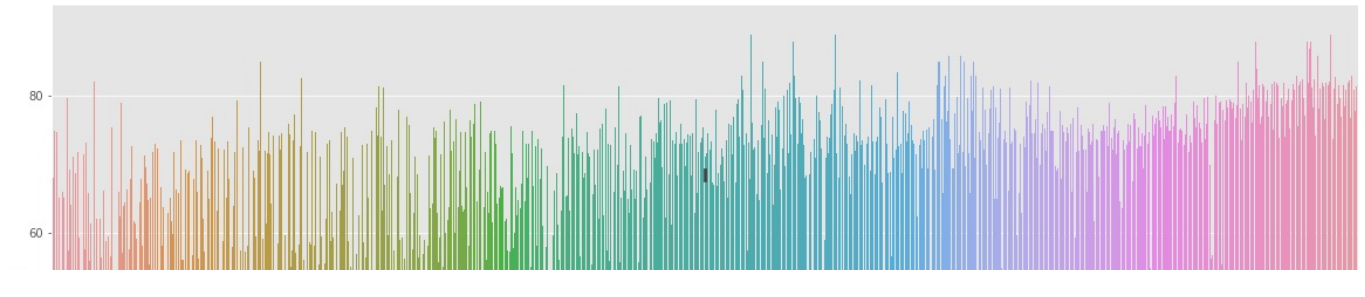

Life_Expectancy
Polio

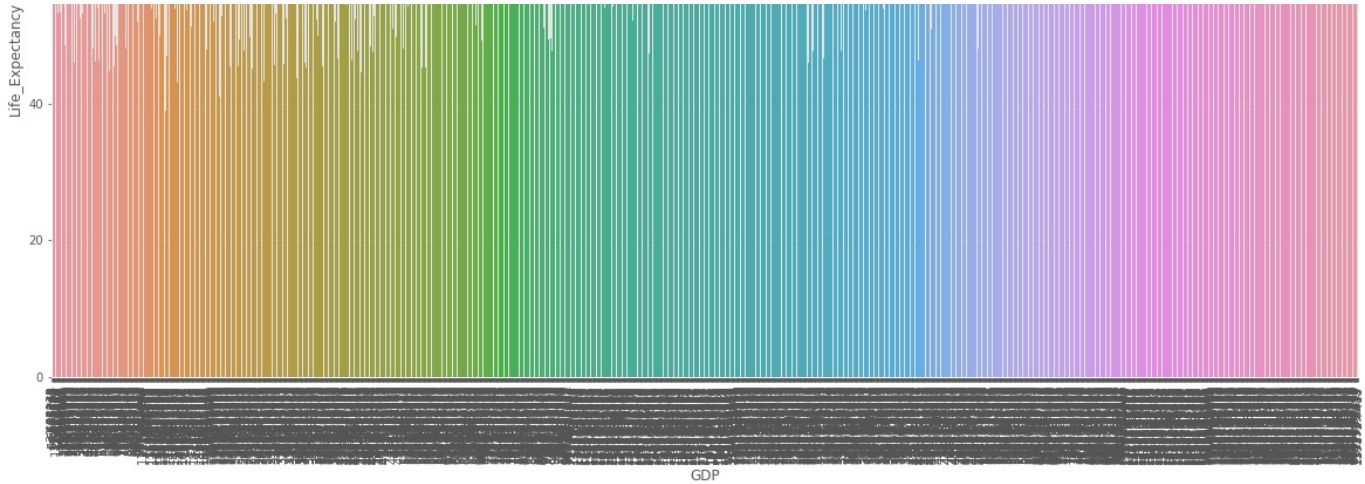


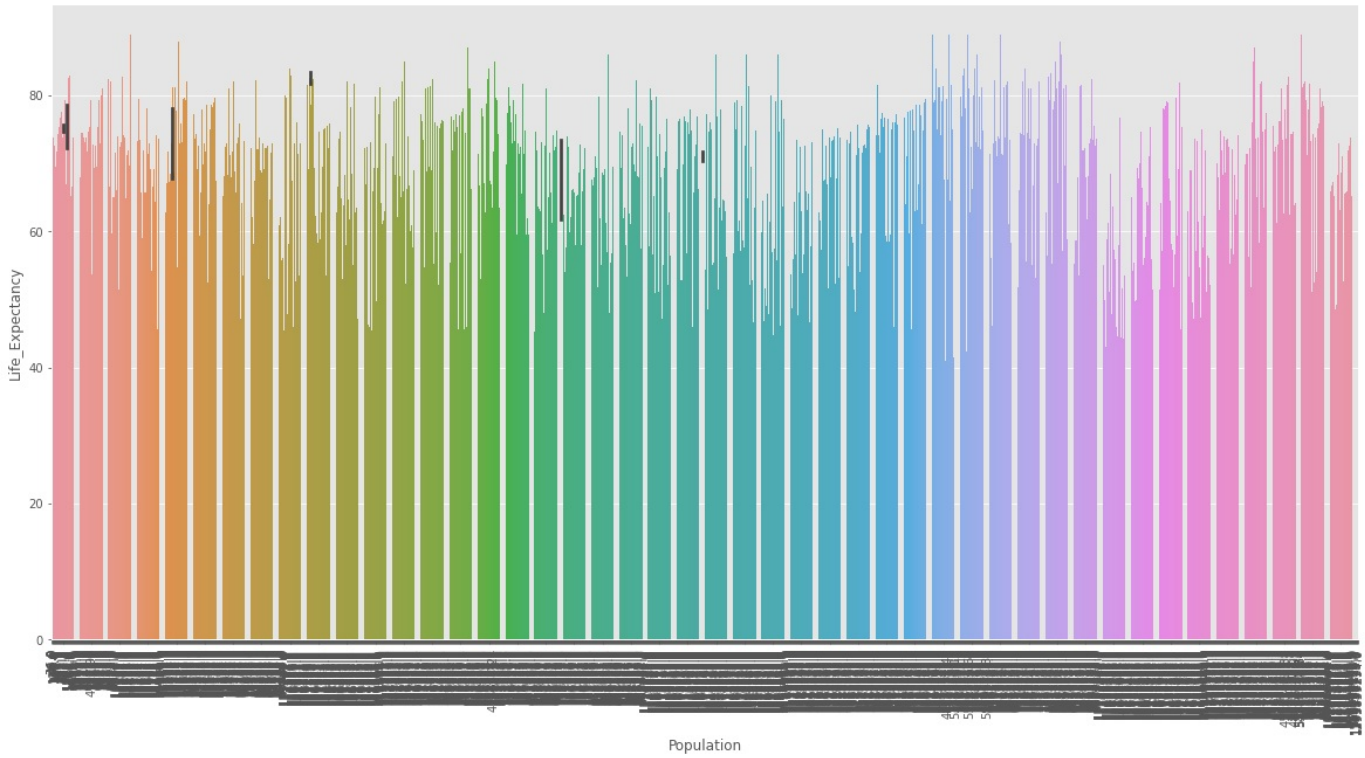

Life_Expectar
Total_Expenditure

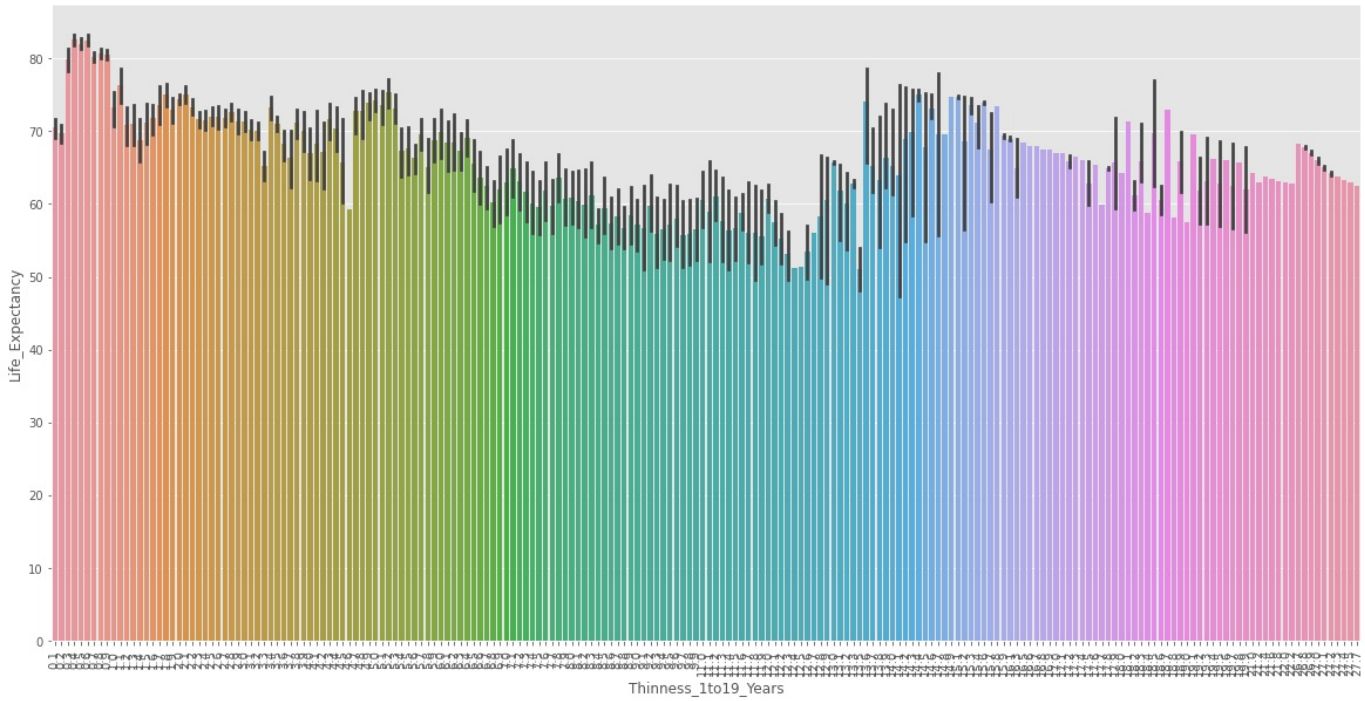

Life_Expectancy
Diphtheria

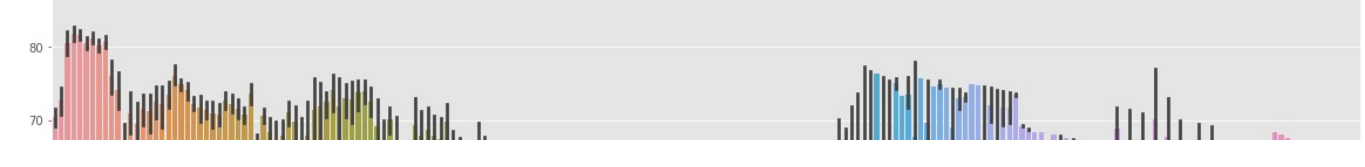

Life_Expectancy
HIV/AIDS

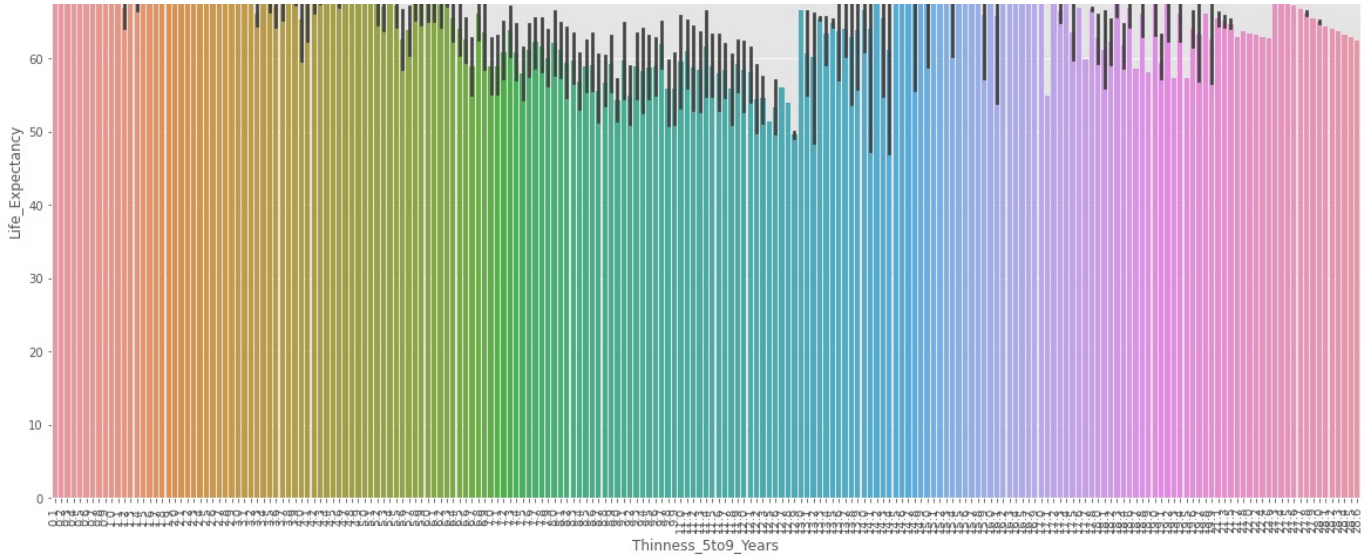


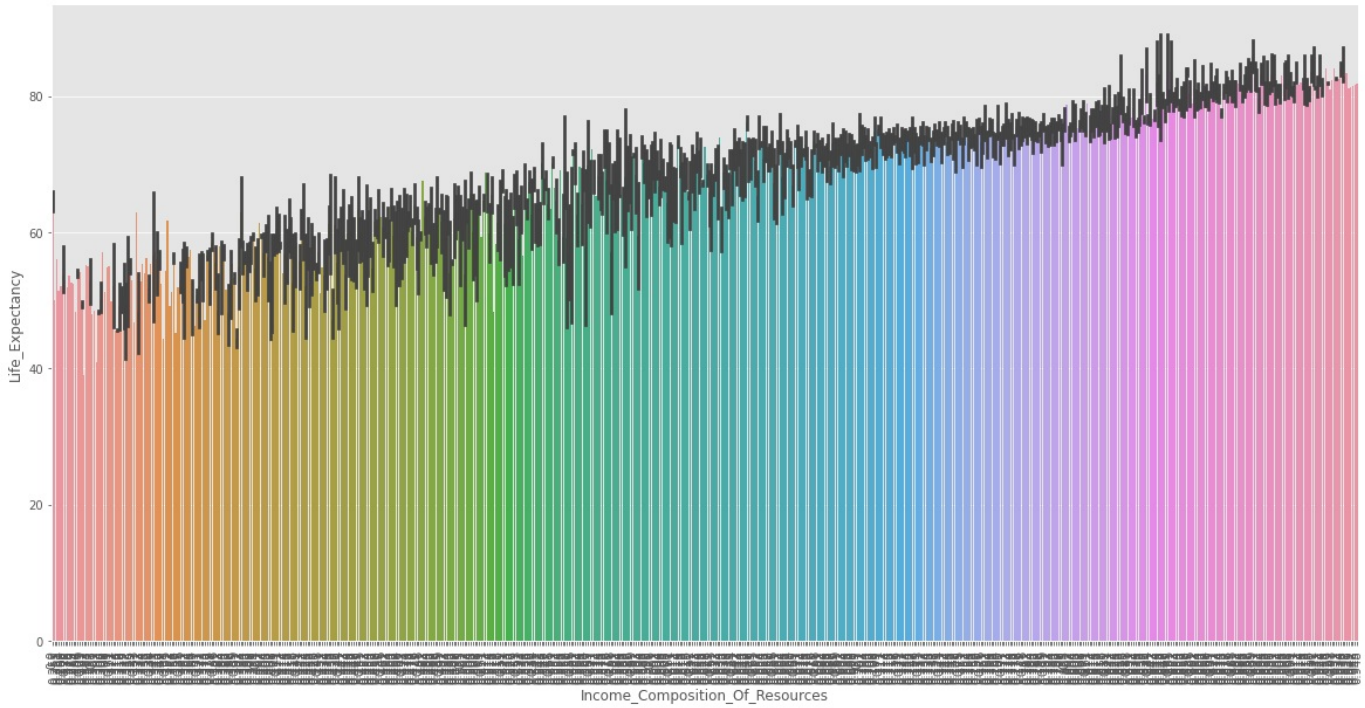

Life_Expectancy
0
20
40
GDP

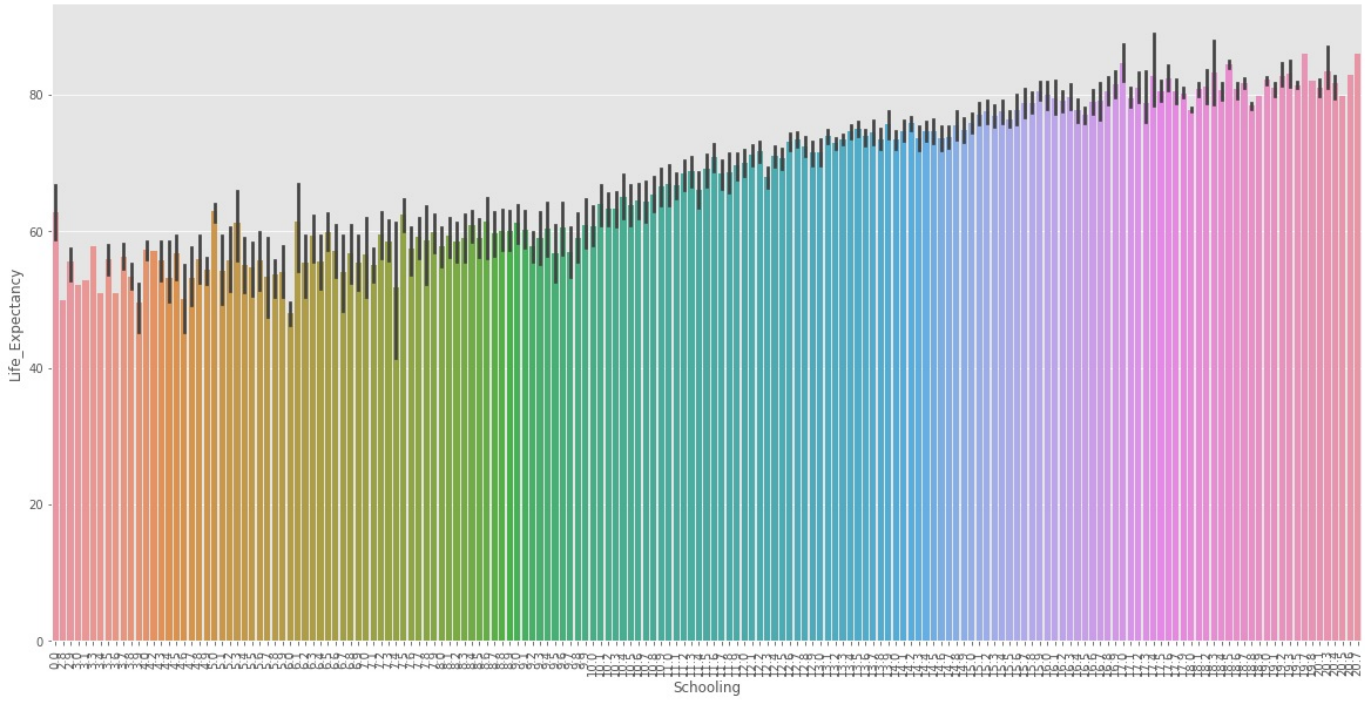

Life_Expectancy
0
20
40
60
80
Population


Life_Expectancy
0
10
20
30
40
50
60
70
80
Thinness_1to19_Years


70
80

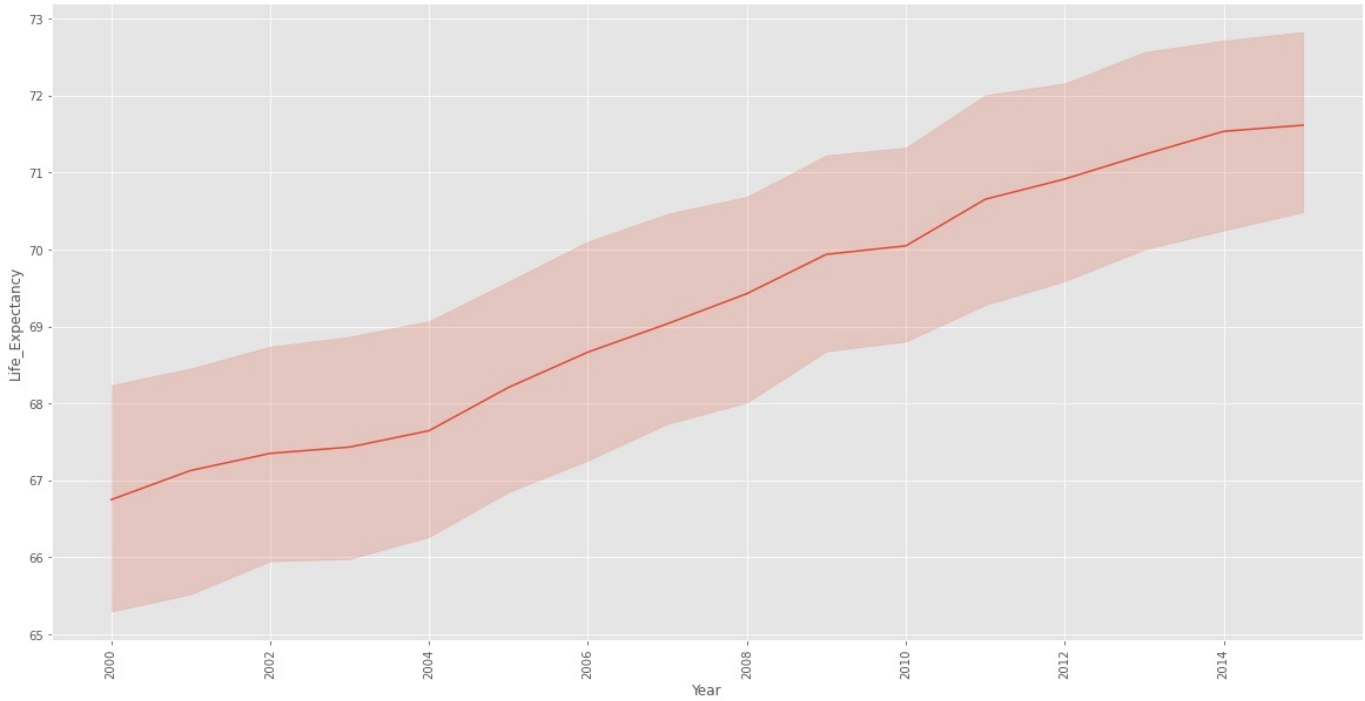


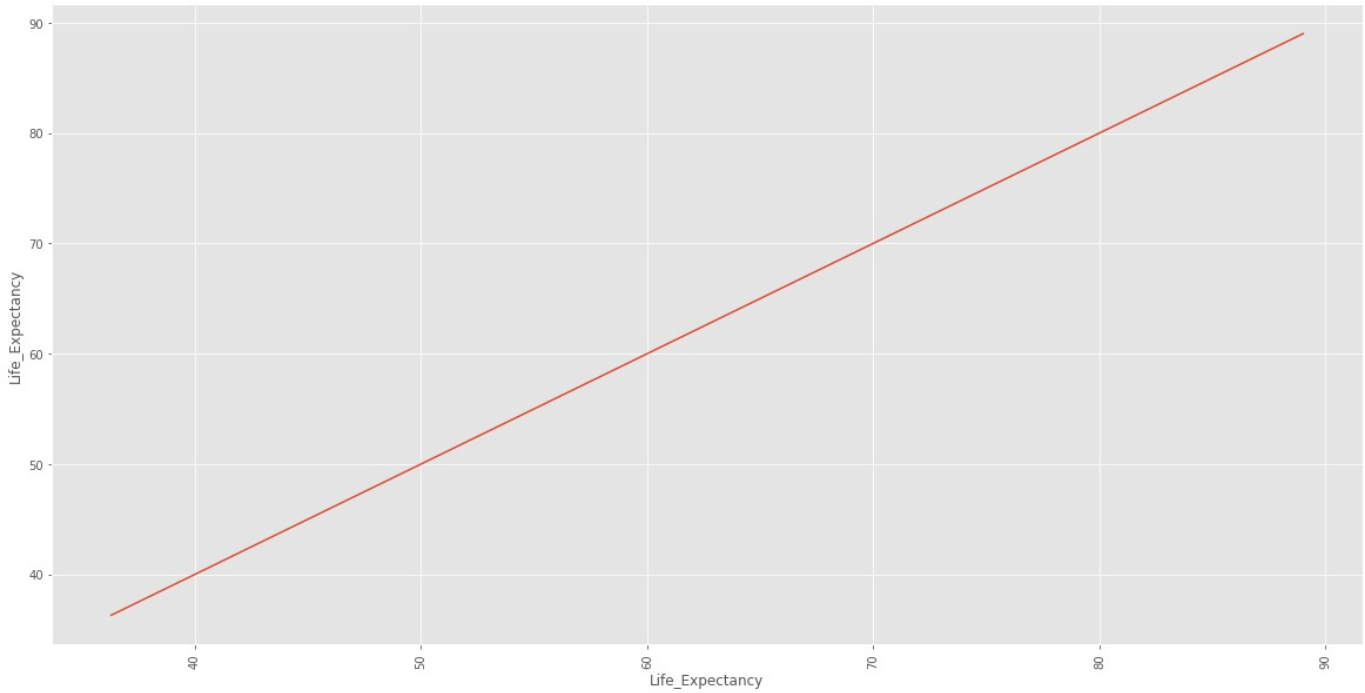


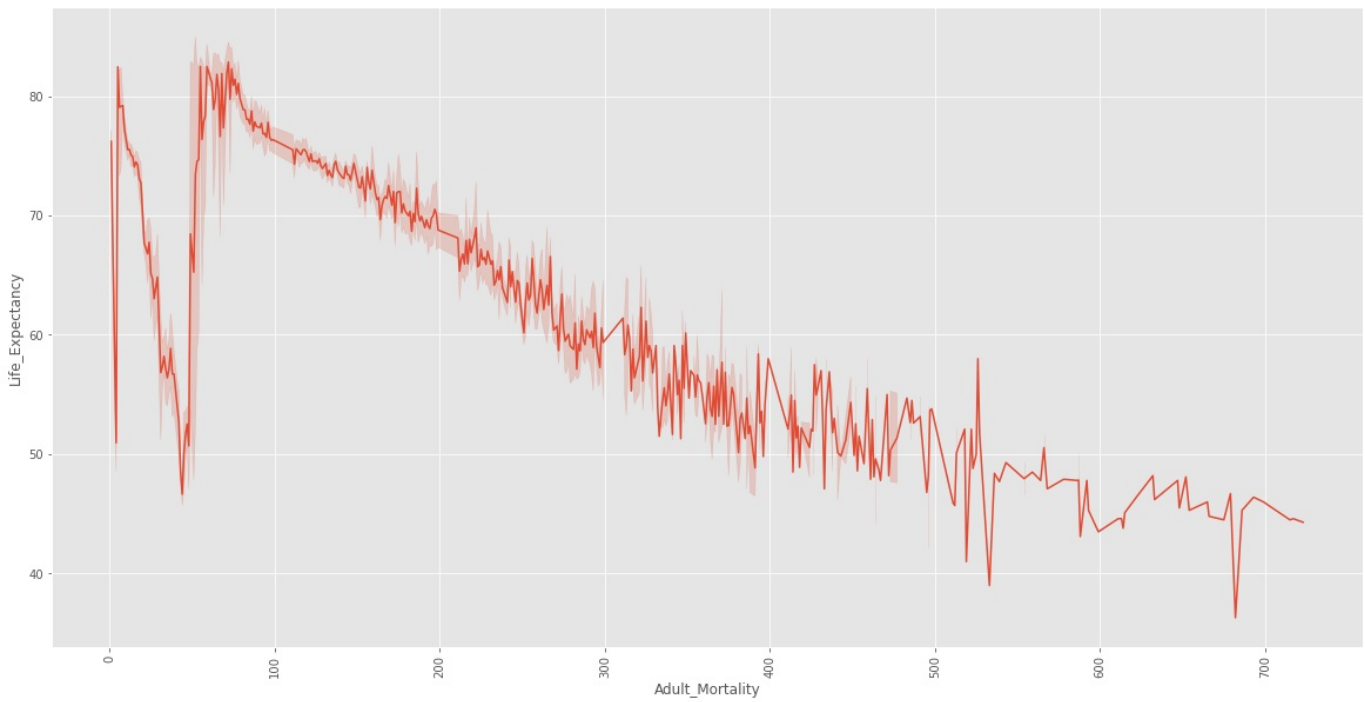


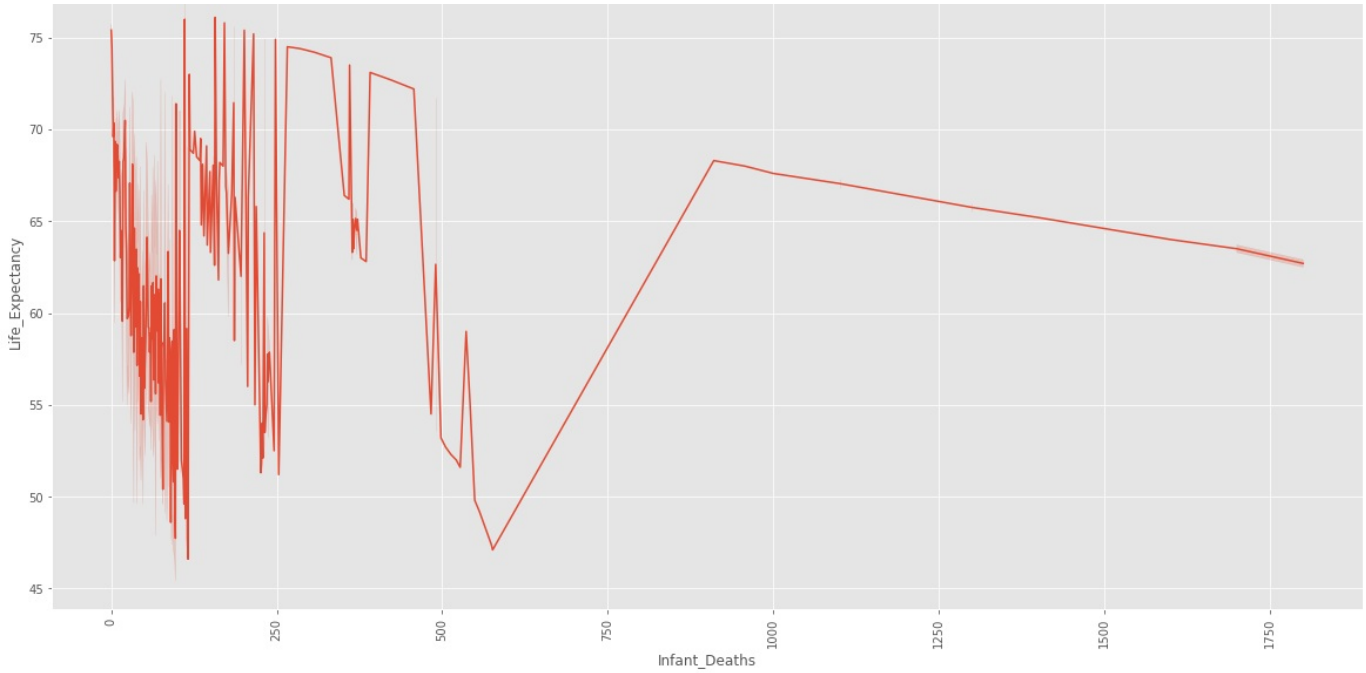
In [51]:

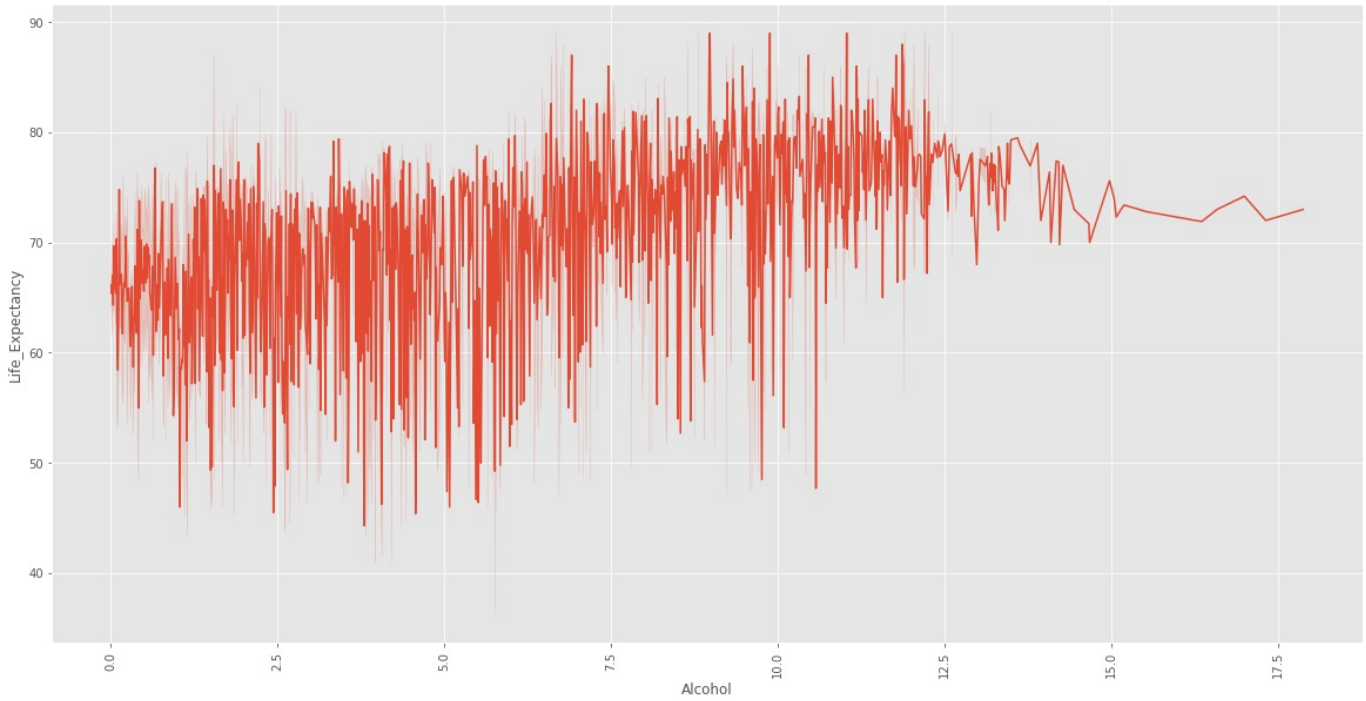
```
plot_bivariate_graphs(life_expectancy_data,numeric_cols,'l')
```

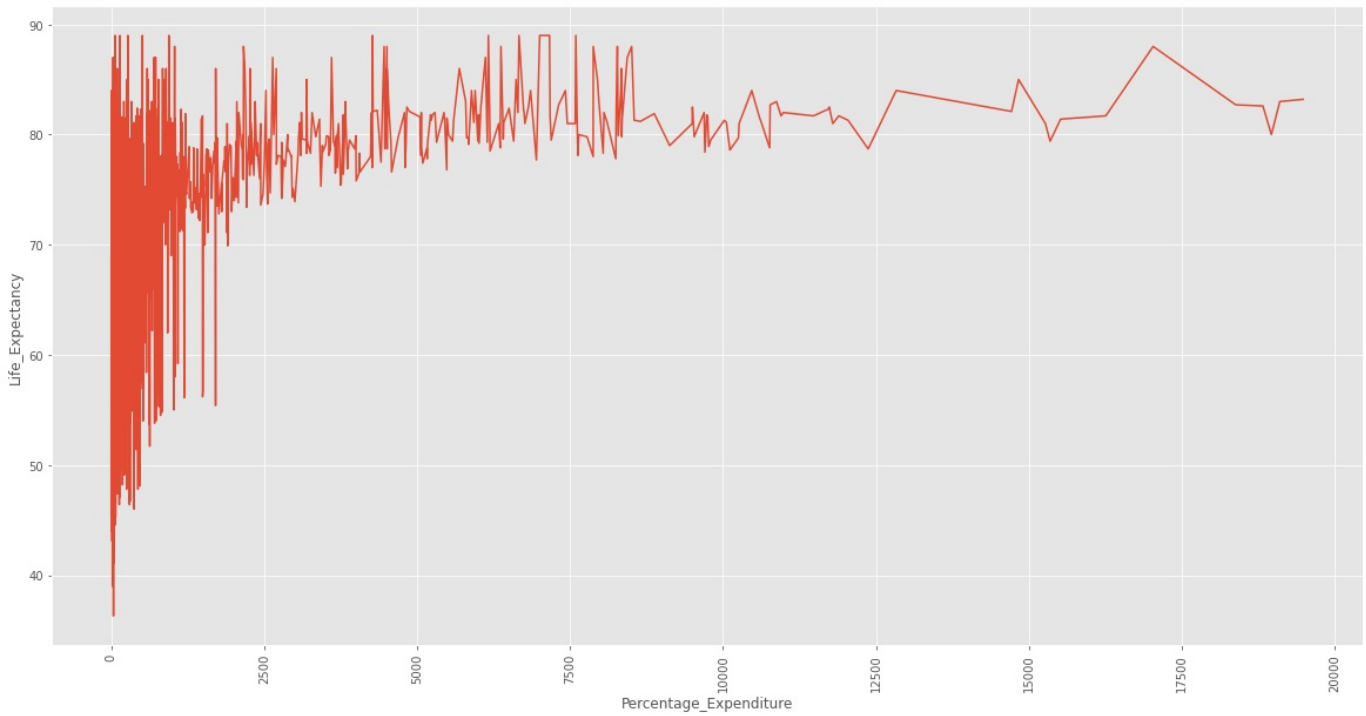

Life_Expectancy
Year

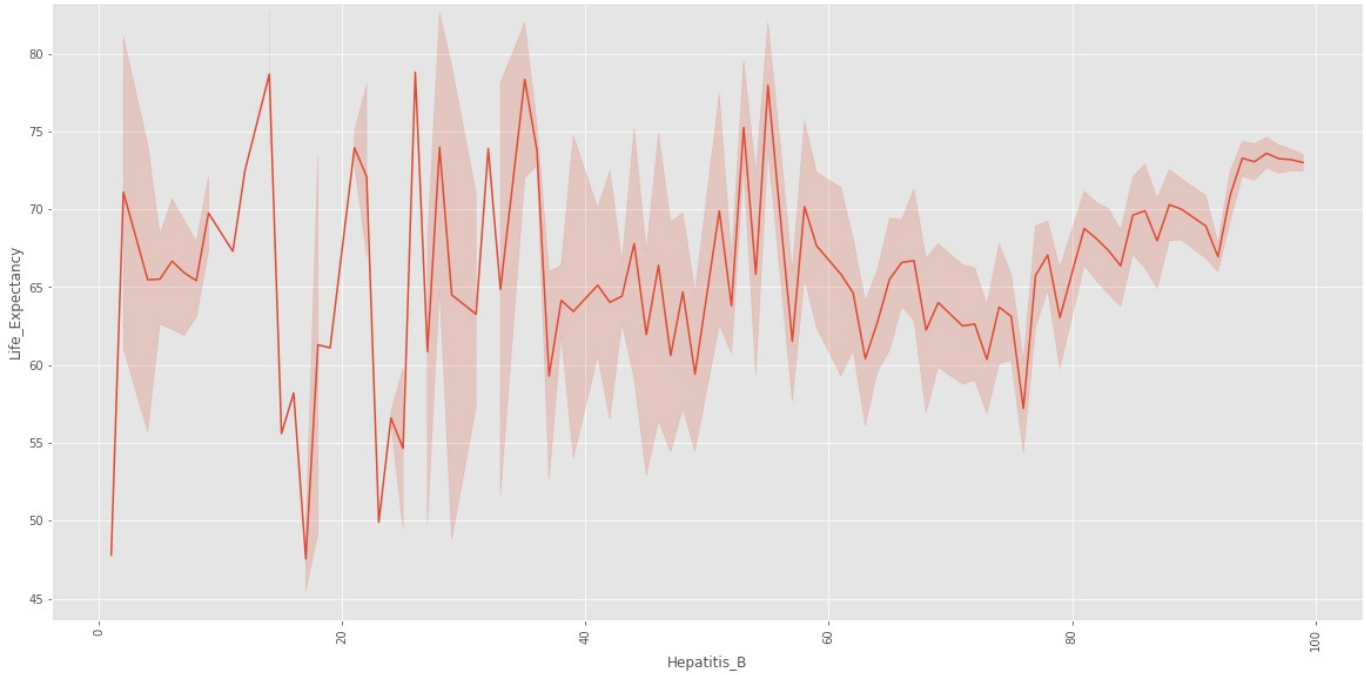

Life_Expectancy
Life_Expectancy

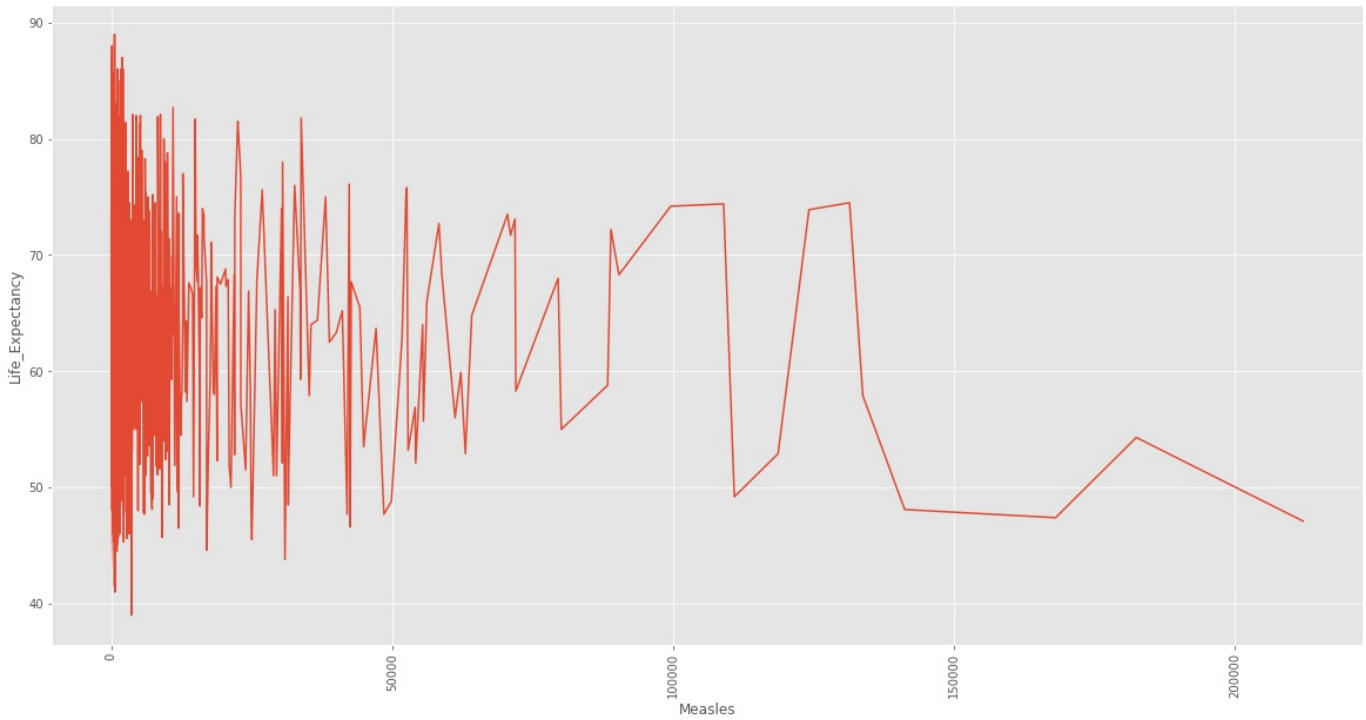

Life_Expectancy
Adult_Mortality

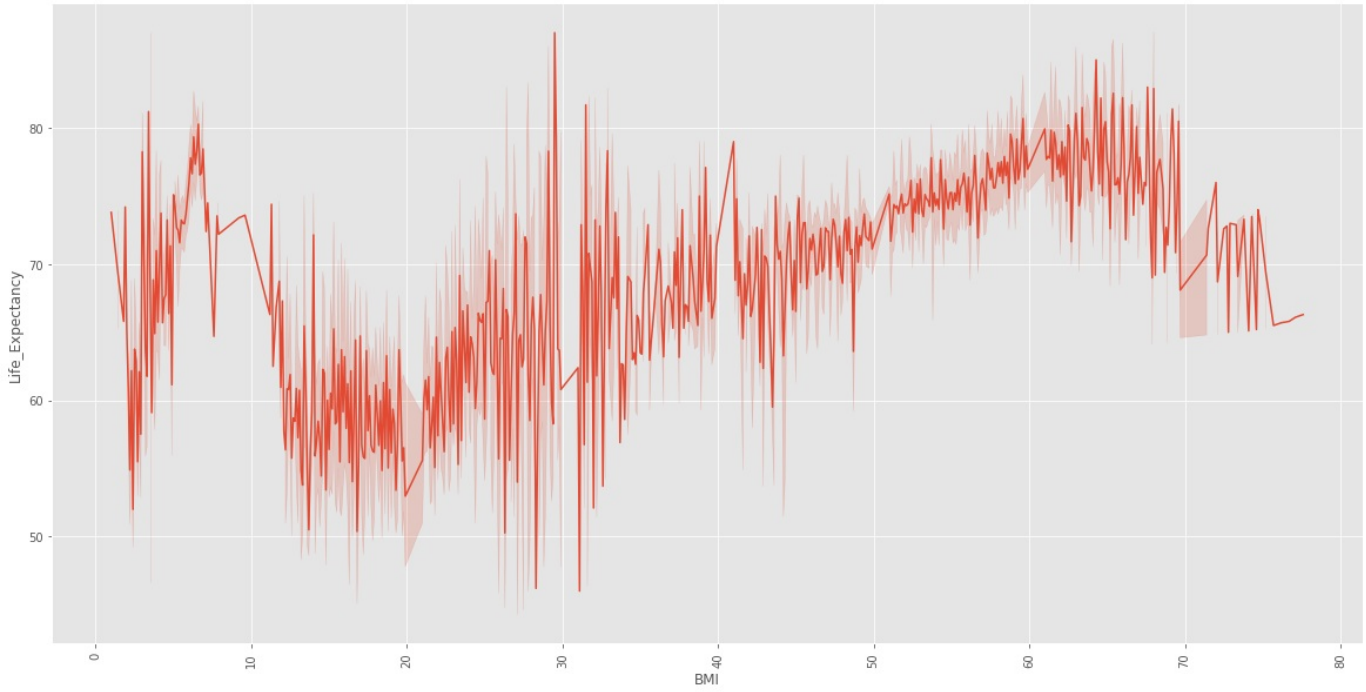

Life_Expectancy
75
70
65
60
55
50
45
0
250
500
750
1000
1250
1500
1750
Infant_Deaths

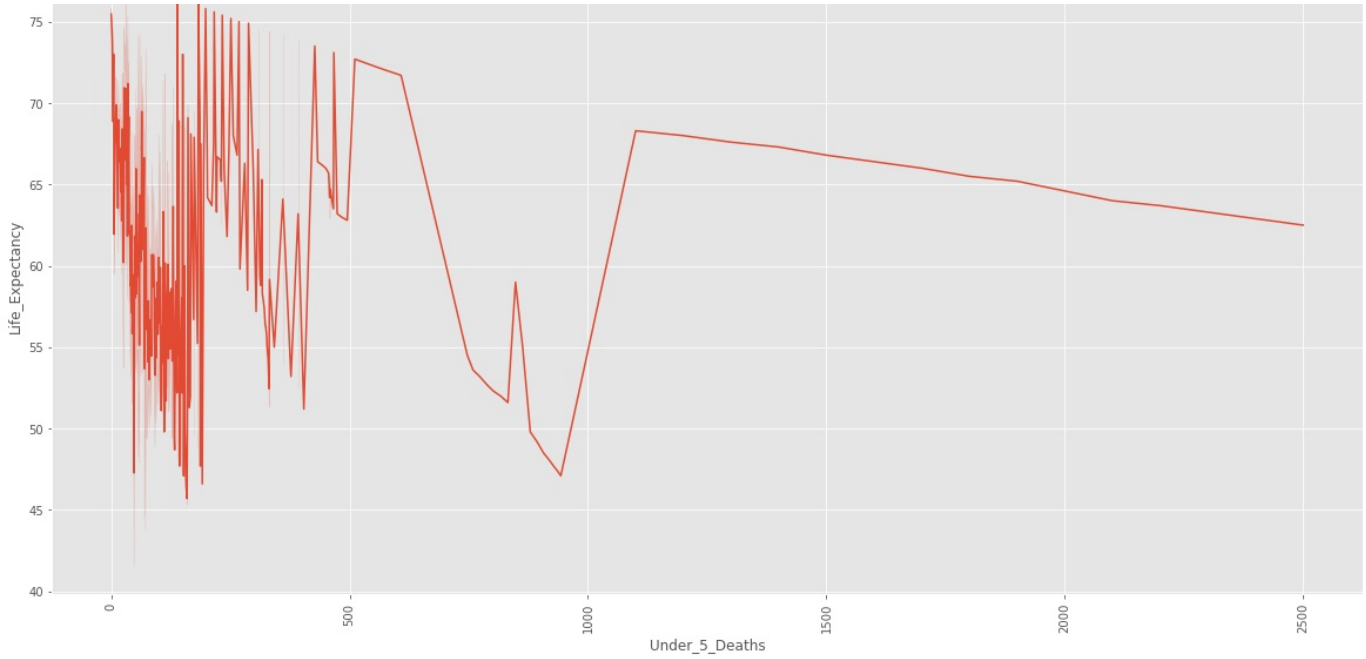

Life_Expectancy
90
80
70
60
50
40
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Alcohol

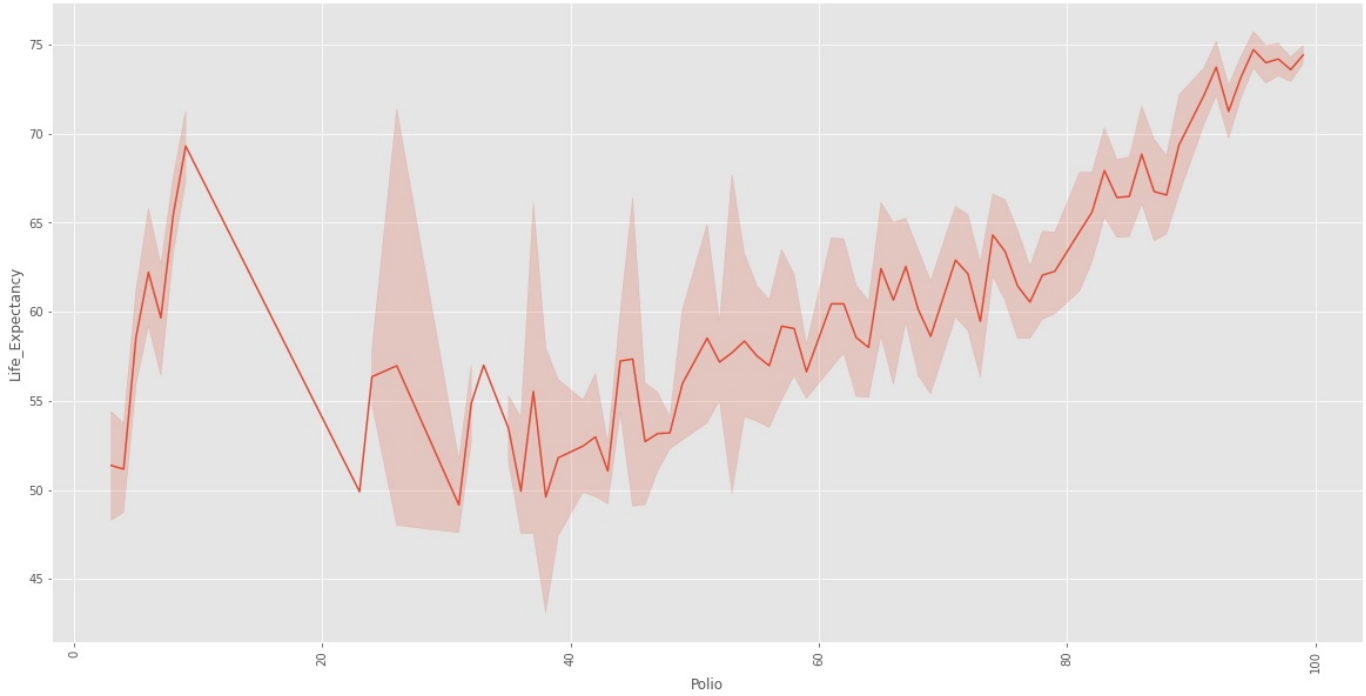

Life_Expectancy
90
80
70
60
50
40
0
2500
5000
7500
10000
12500
15000
17500
20000
Percentage_Expenditure

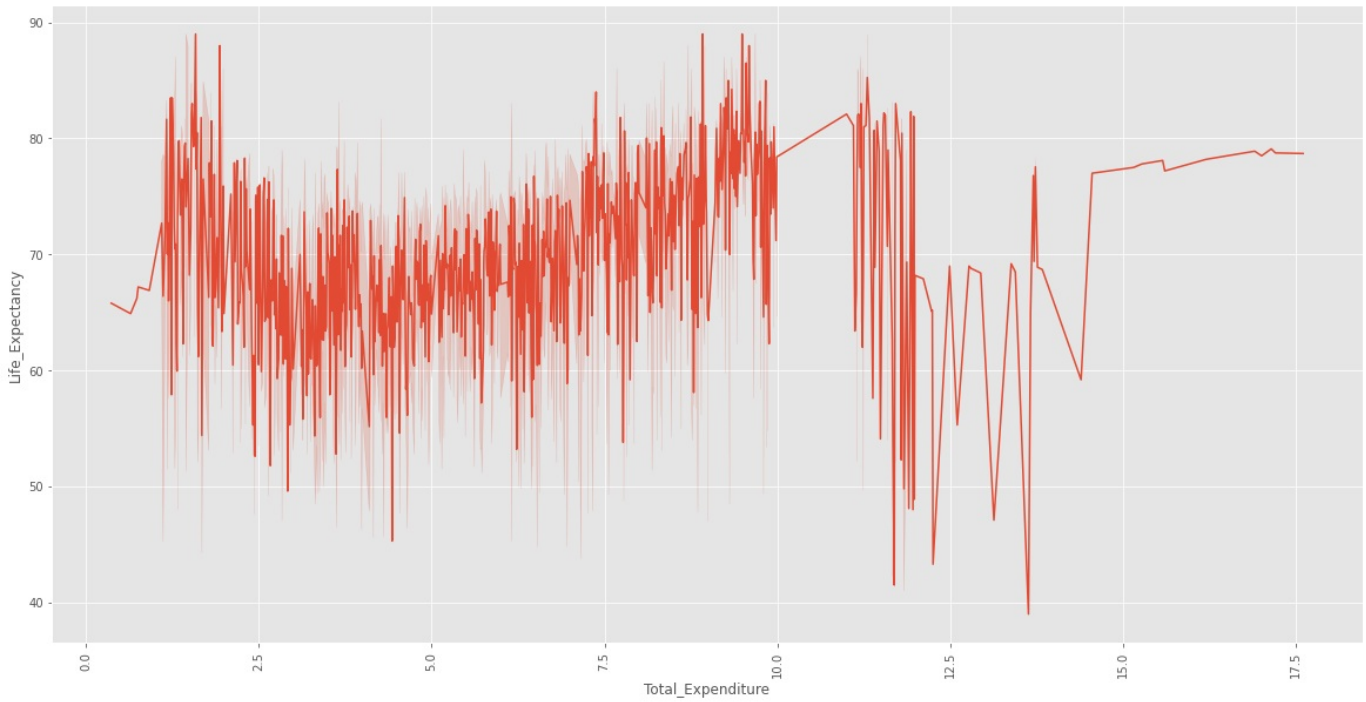

Life_Expectancy
80
75
70
65
60
55
50
45
0
20
40
60
80
100
Hepatitis_B


Life_Expectancy
90
80
70
60
50
40
0
50000
100000
150000
200000
Measles

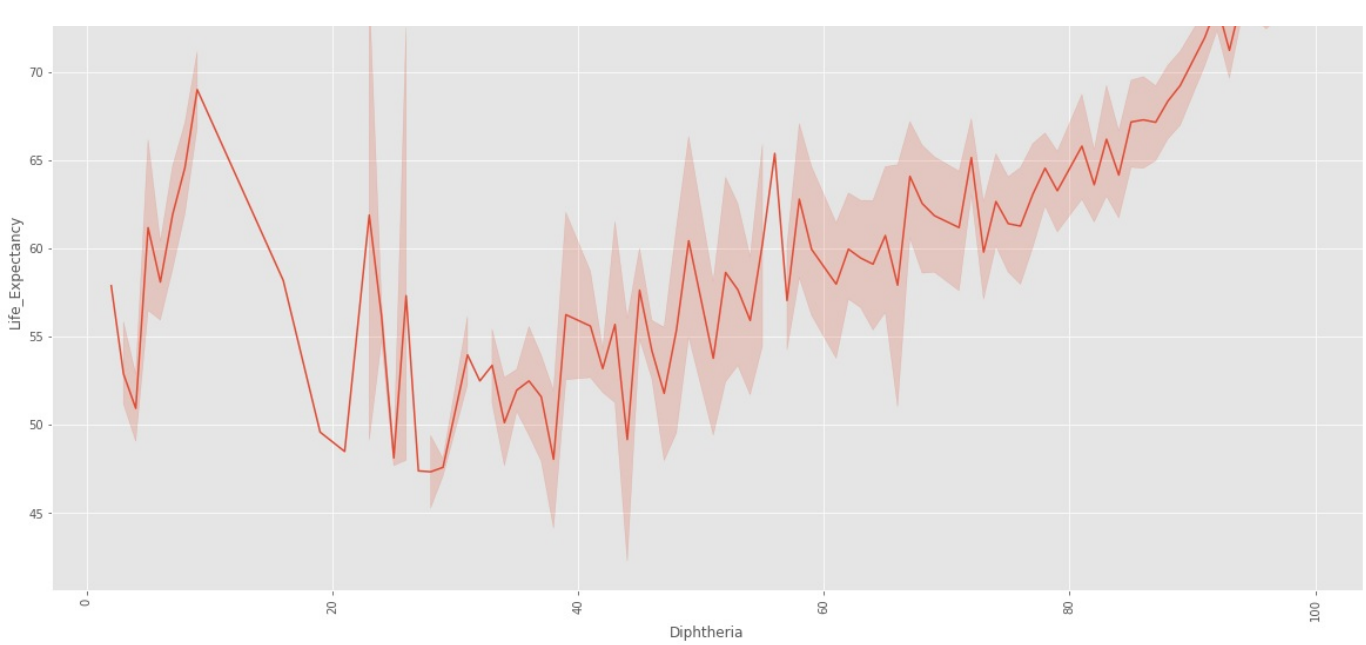

Life_Expectancy
80
70
60
50
0
10
20
30
40
50
60
70
80
BMI

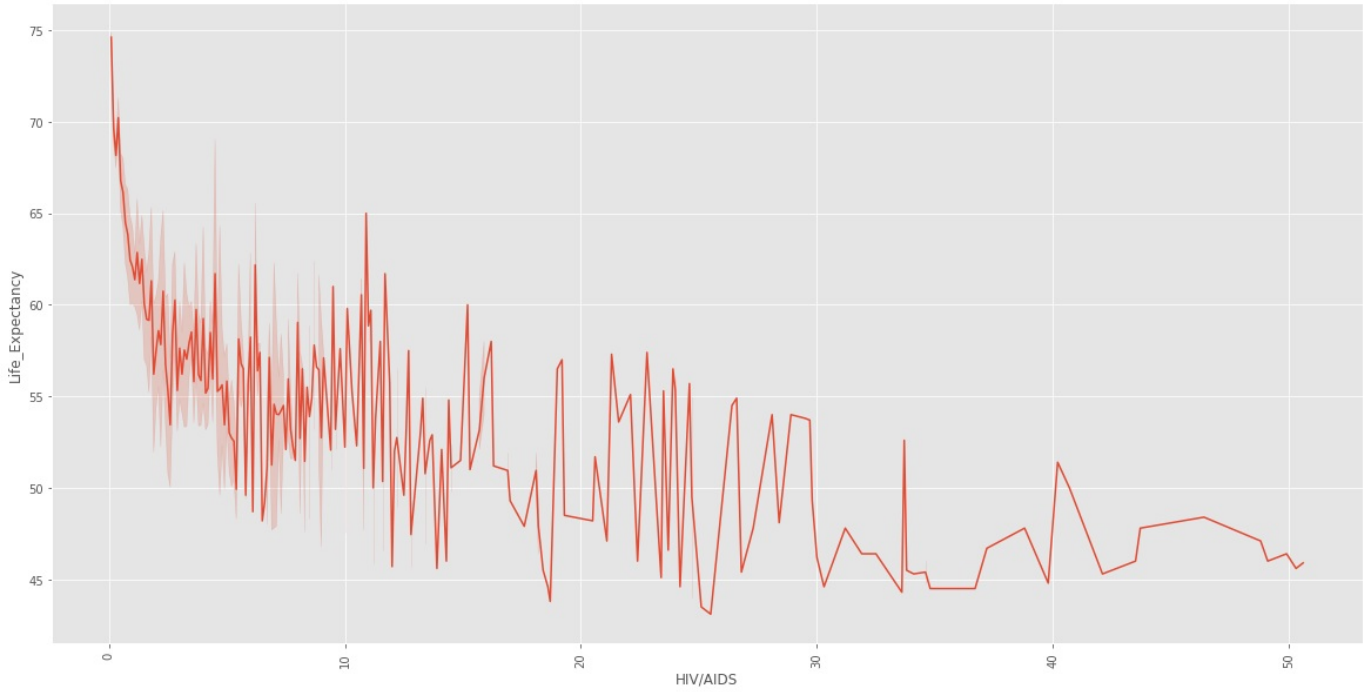

Life_Expectancy
75
70
65
60
55
50
45
40
0
500
1000
1500
2000
2500
Under_5_Deaths

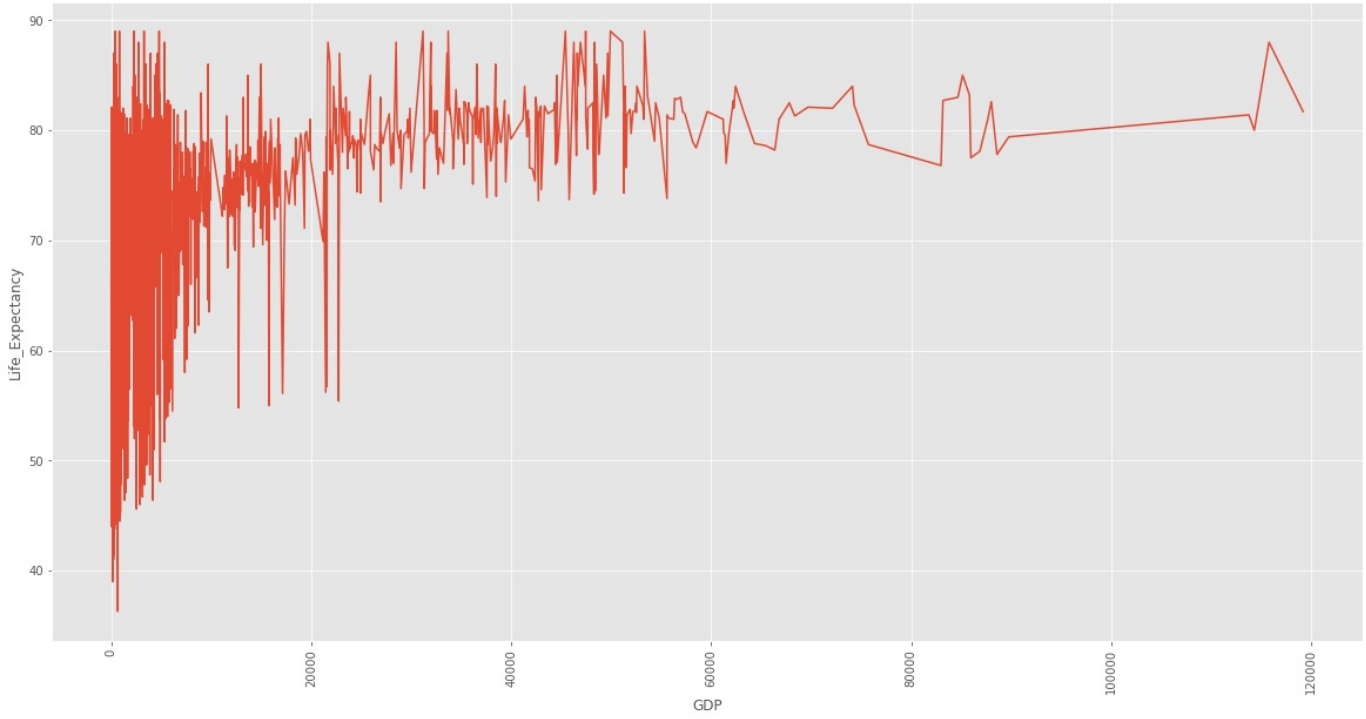

Life_Expectancy
75
70
65
60
55
50
45
0
20
40
60
80
100
Polio

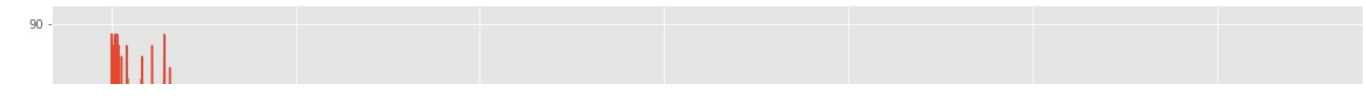

Life_Expectancy
90
80
70
60
50
40
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Total_Expenditure

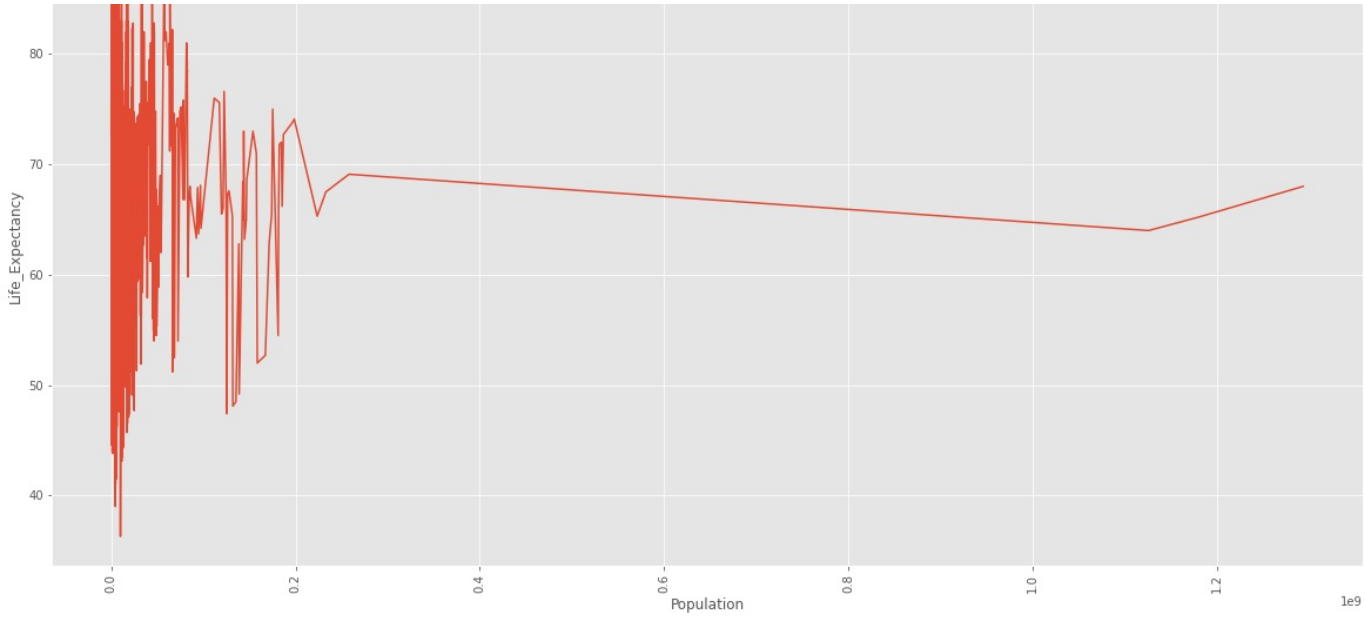

75

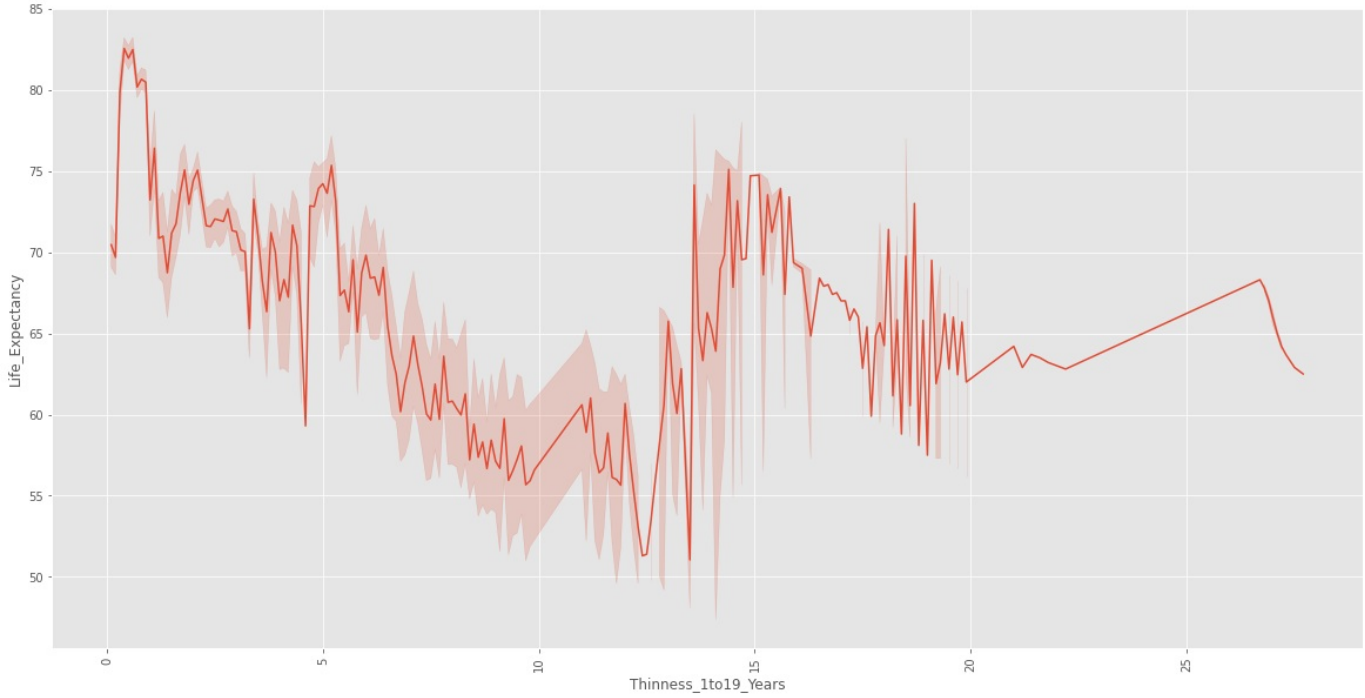

Life_Expectancy
70
65
60
55
50
45
0
20
40
60
80
100
Diphtheria

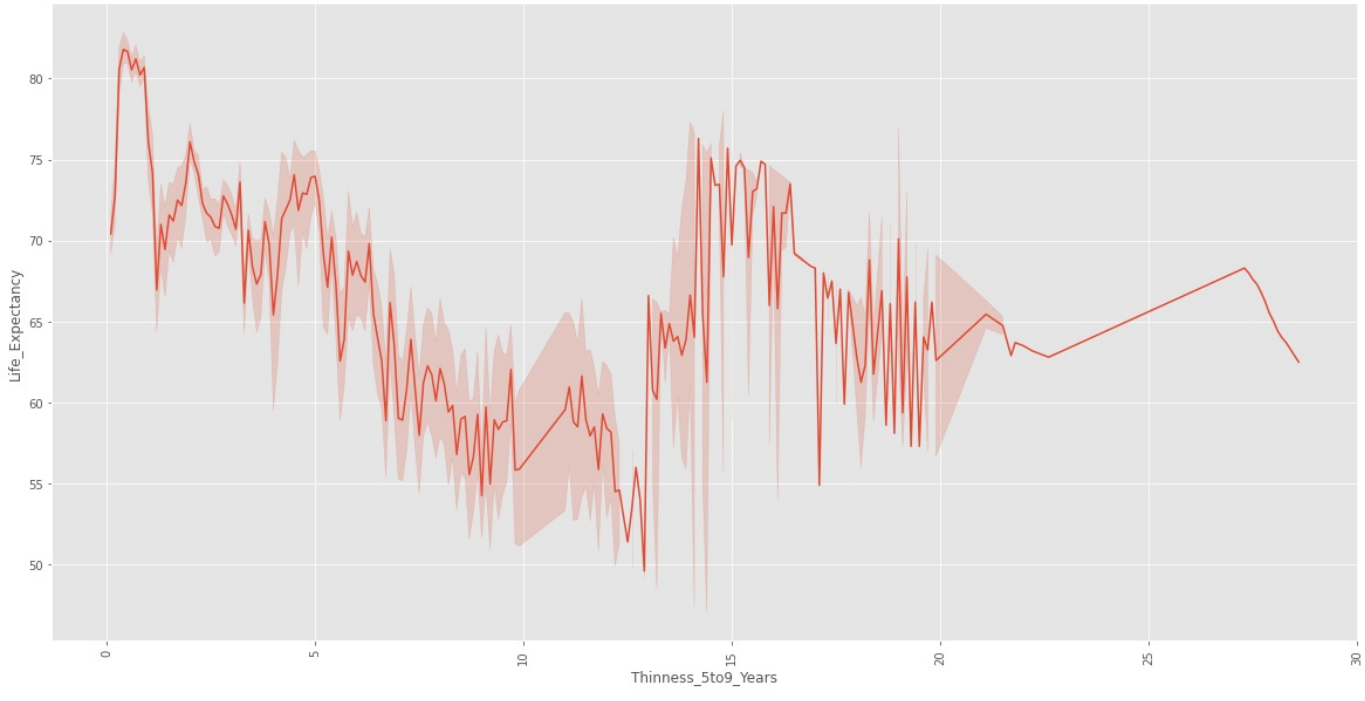

Life_Expectancy
75
70
65
60
55
50
45
0
10
20
30
40
50
HIV/AIDS

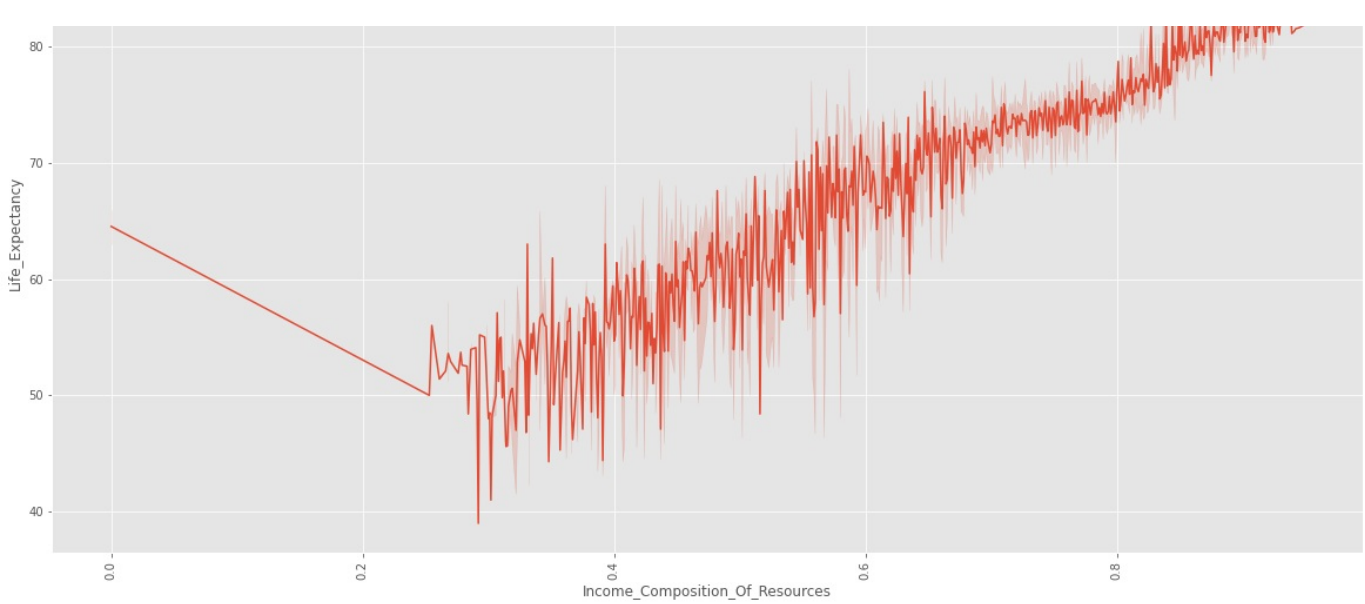

Life_Expectancy
90
80
70
60
50
40
0
20000
40000
60000
80000
100000
120000
GDP

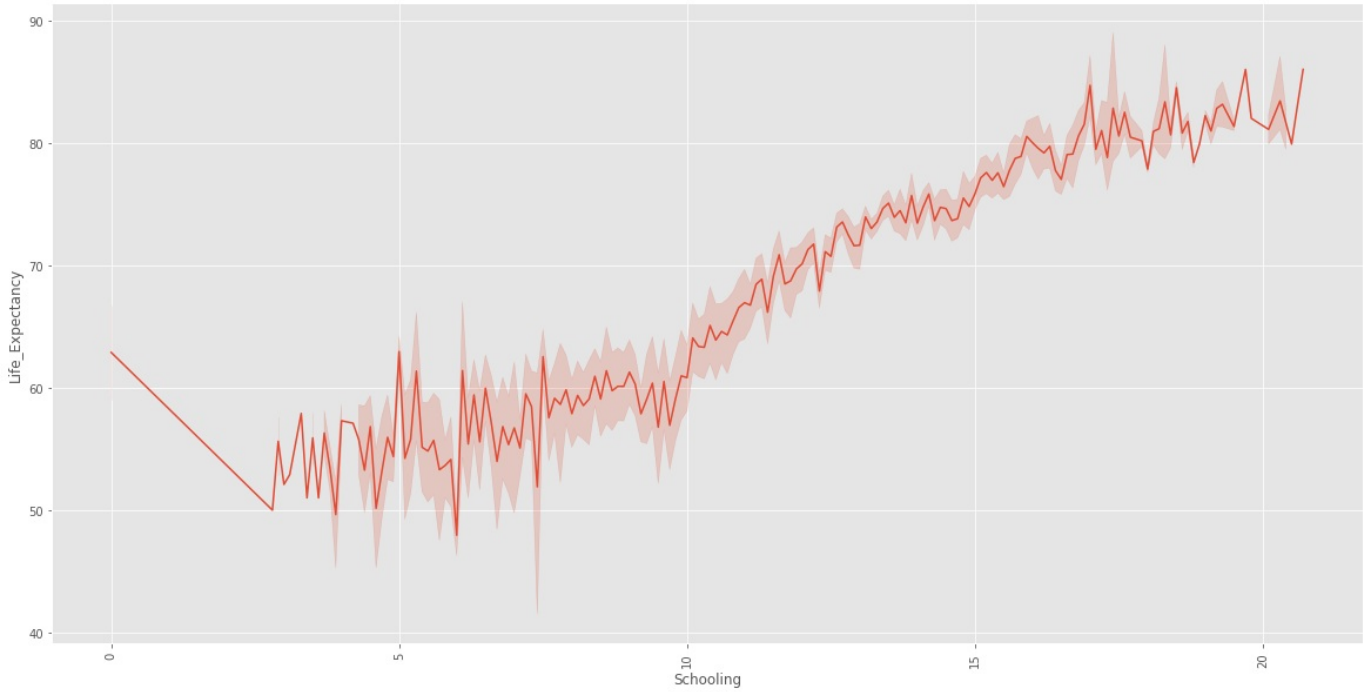

90


Life_Expectancy
Population
1e9


Life_Expectancy
Thinness_1to19_Years


Life_Expectancy
Thinness_5to9_Years

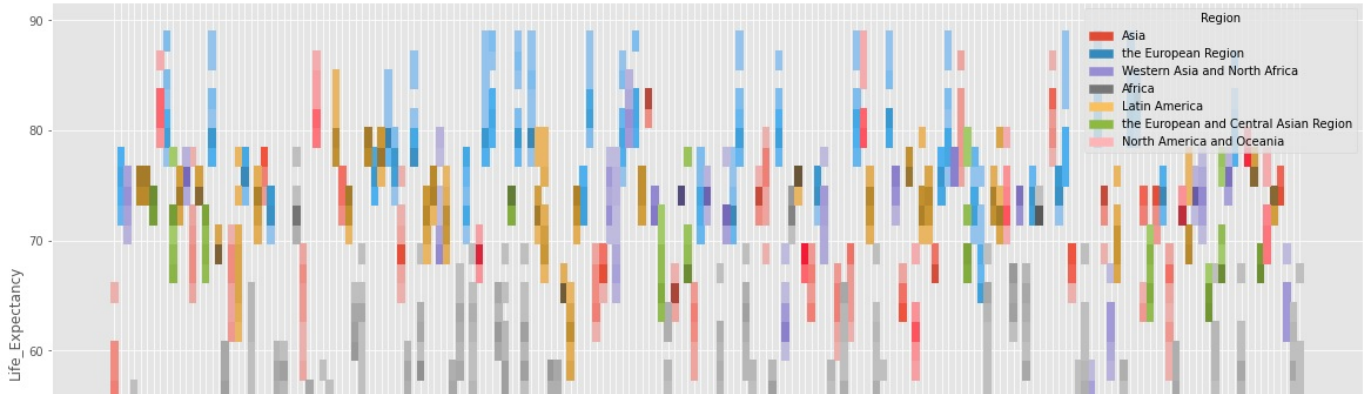


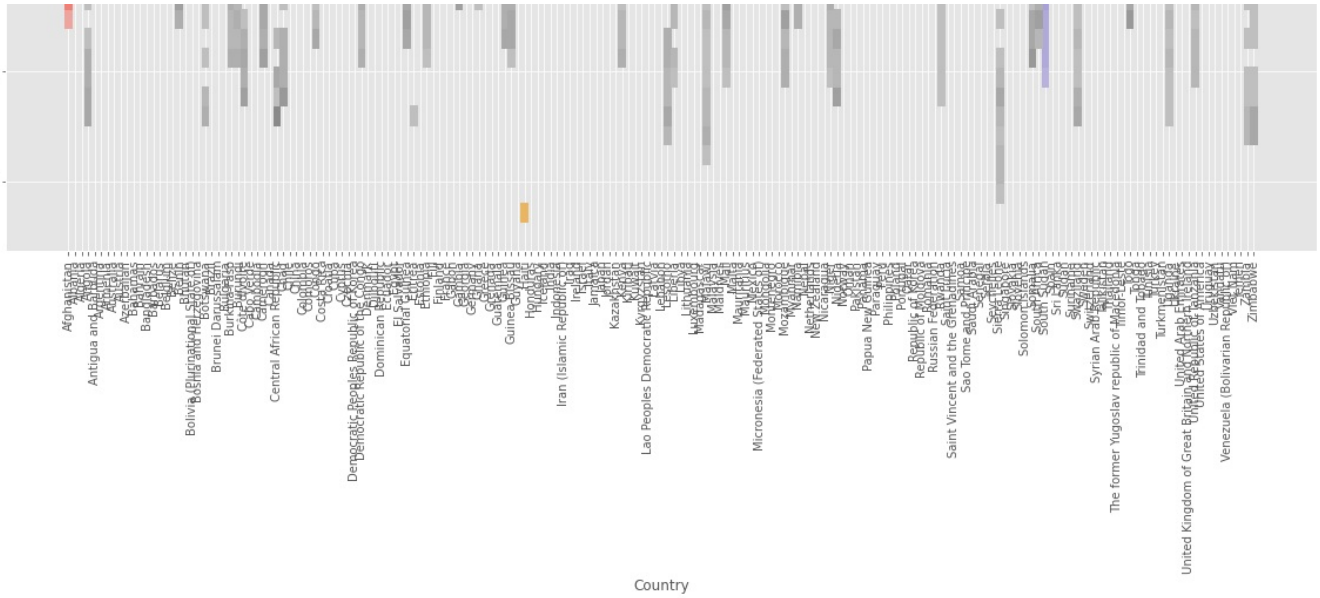


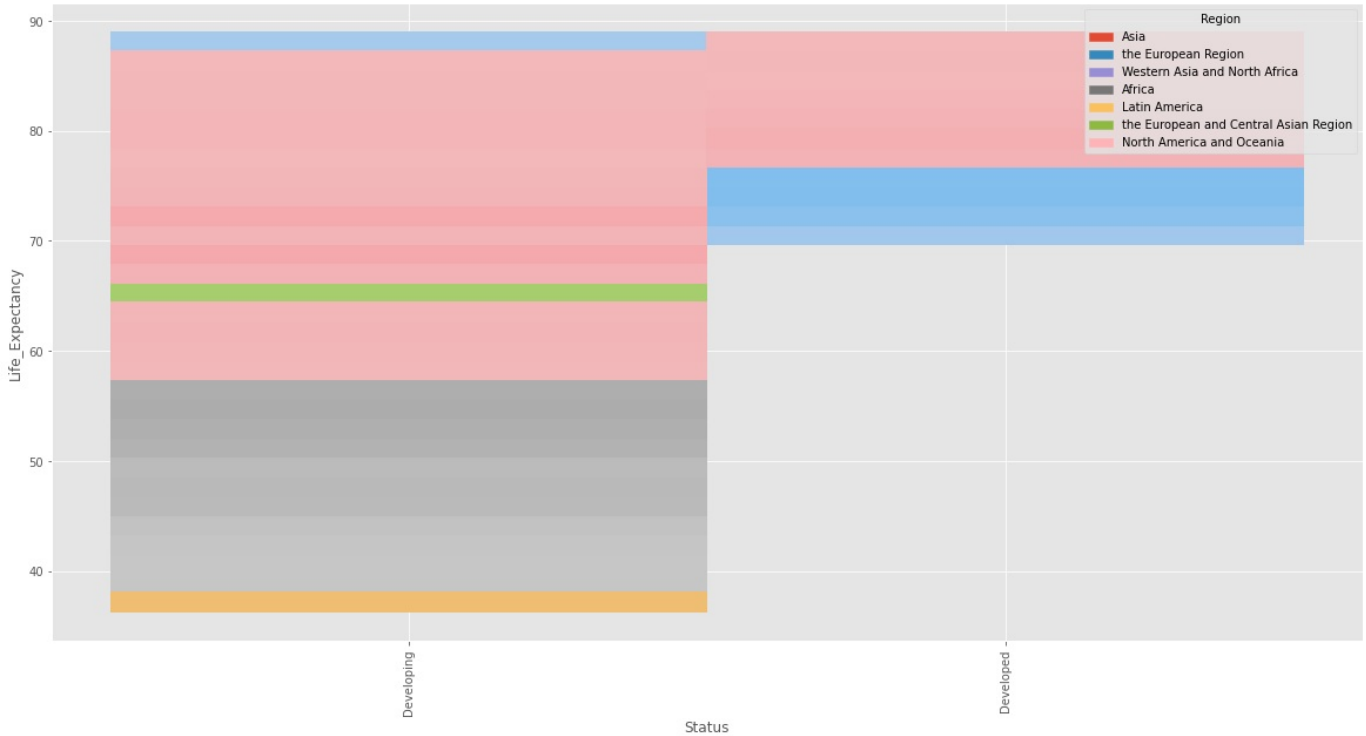
* BIVARIATE ANALYSIS - (NUMERIC - CATEGORIC)

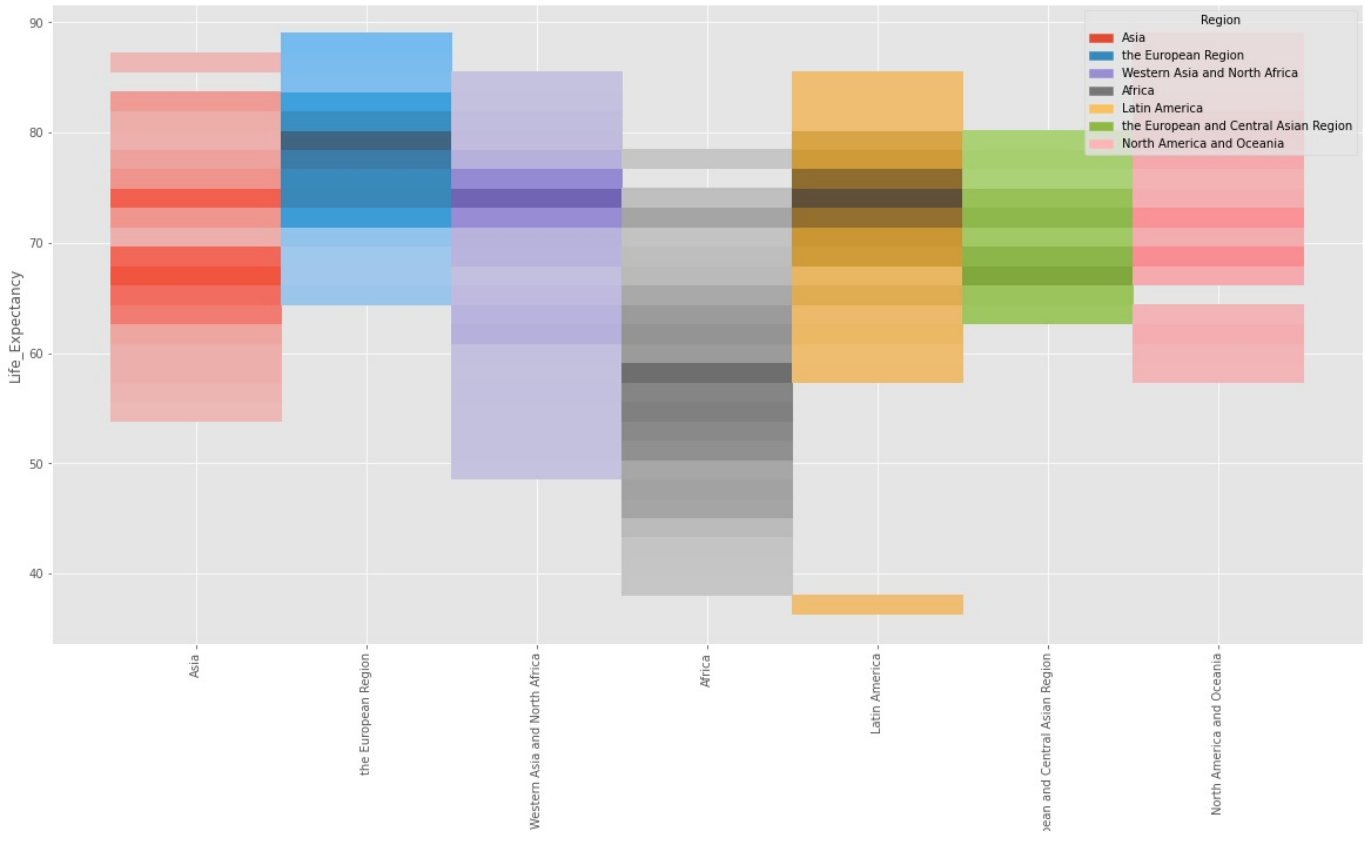
i.HISTOGRAM WITH HUE

In [52]:
```
plot_bivariate_graphs(life_expectancy_data,factor_cols,'h')
```



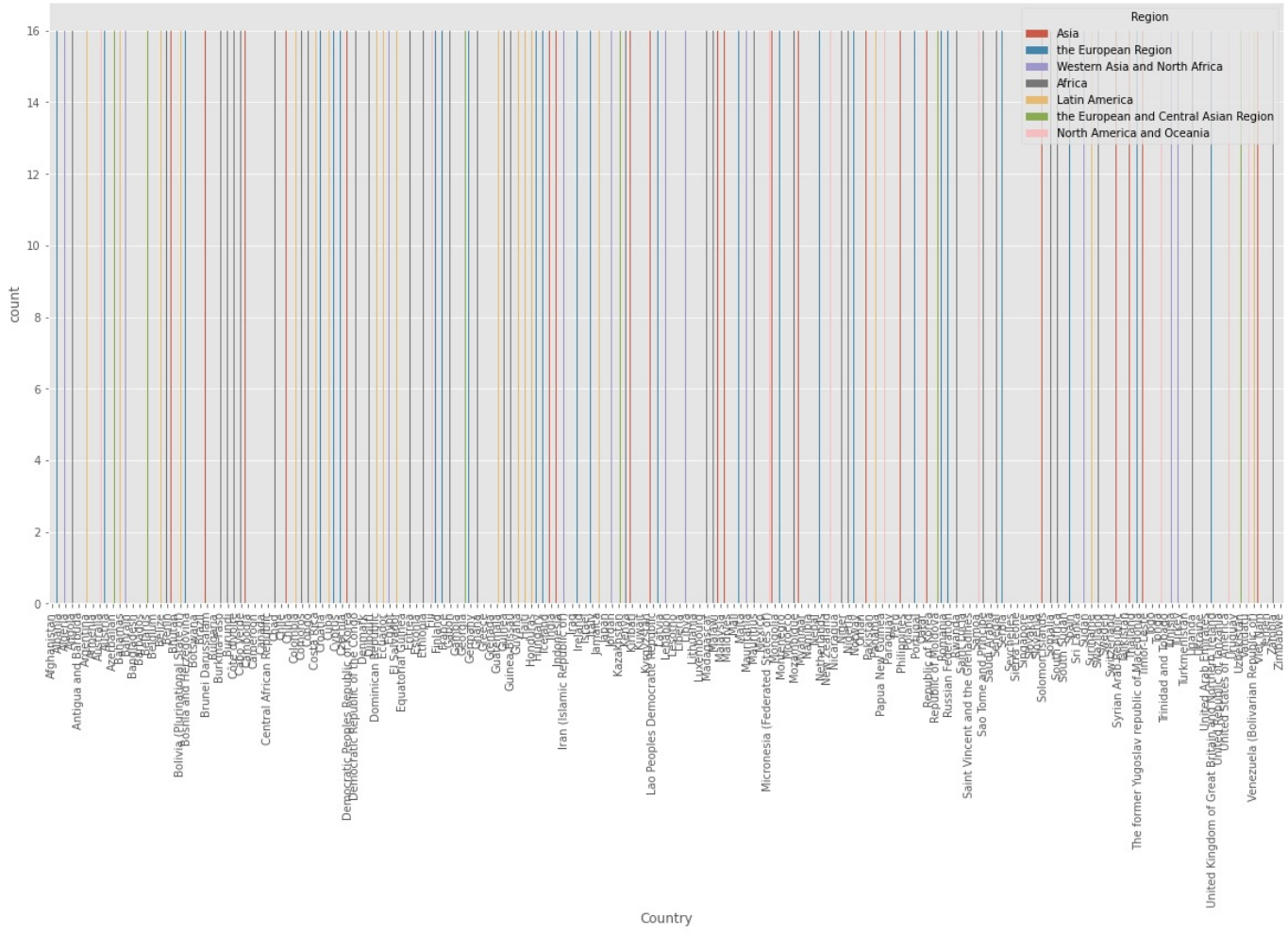

50
40
Country

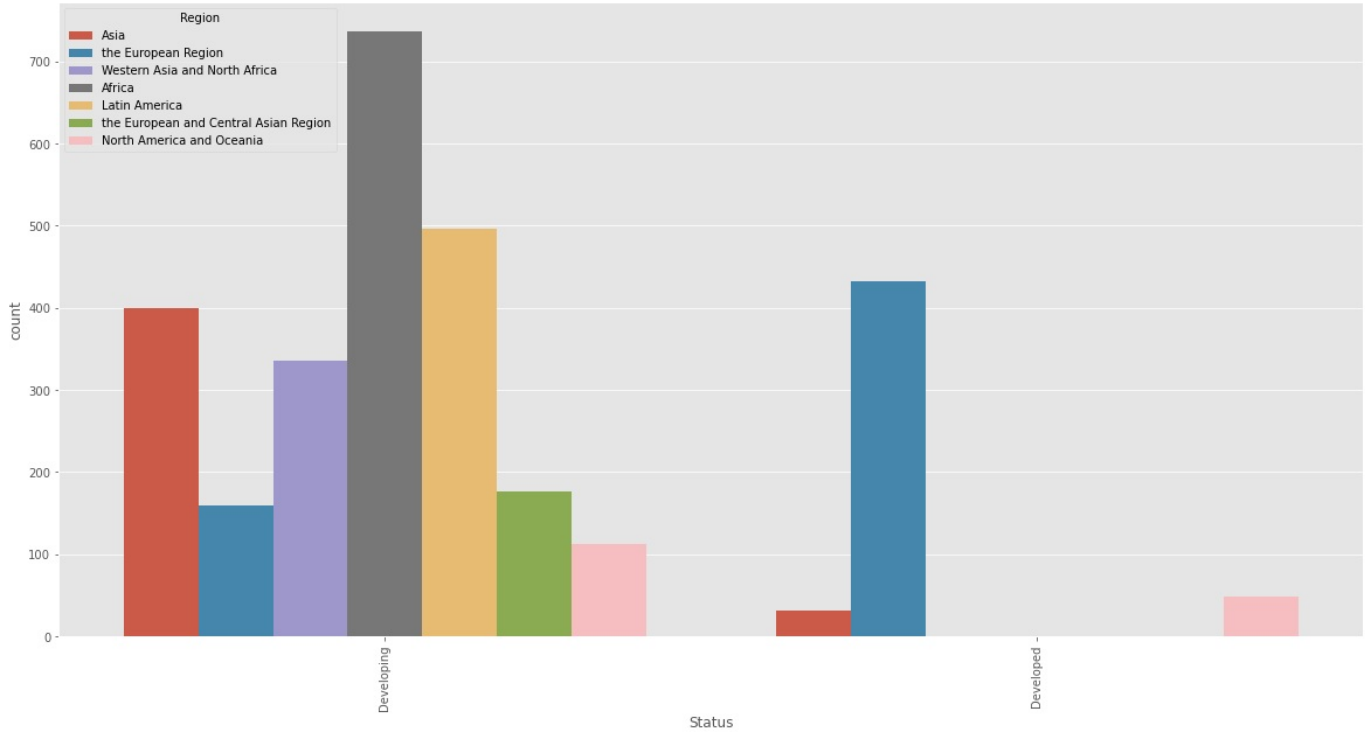

Region
Asia
the European Region
Western Asia and North Africa
Africa
Latin America
the European and Central Asian Region
North America and Oceania
90
80
70
60
50
40
Life_Expectancy
Developing
Developed
Status

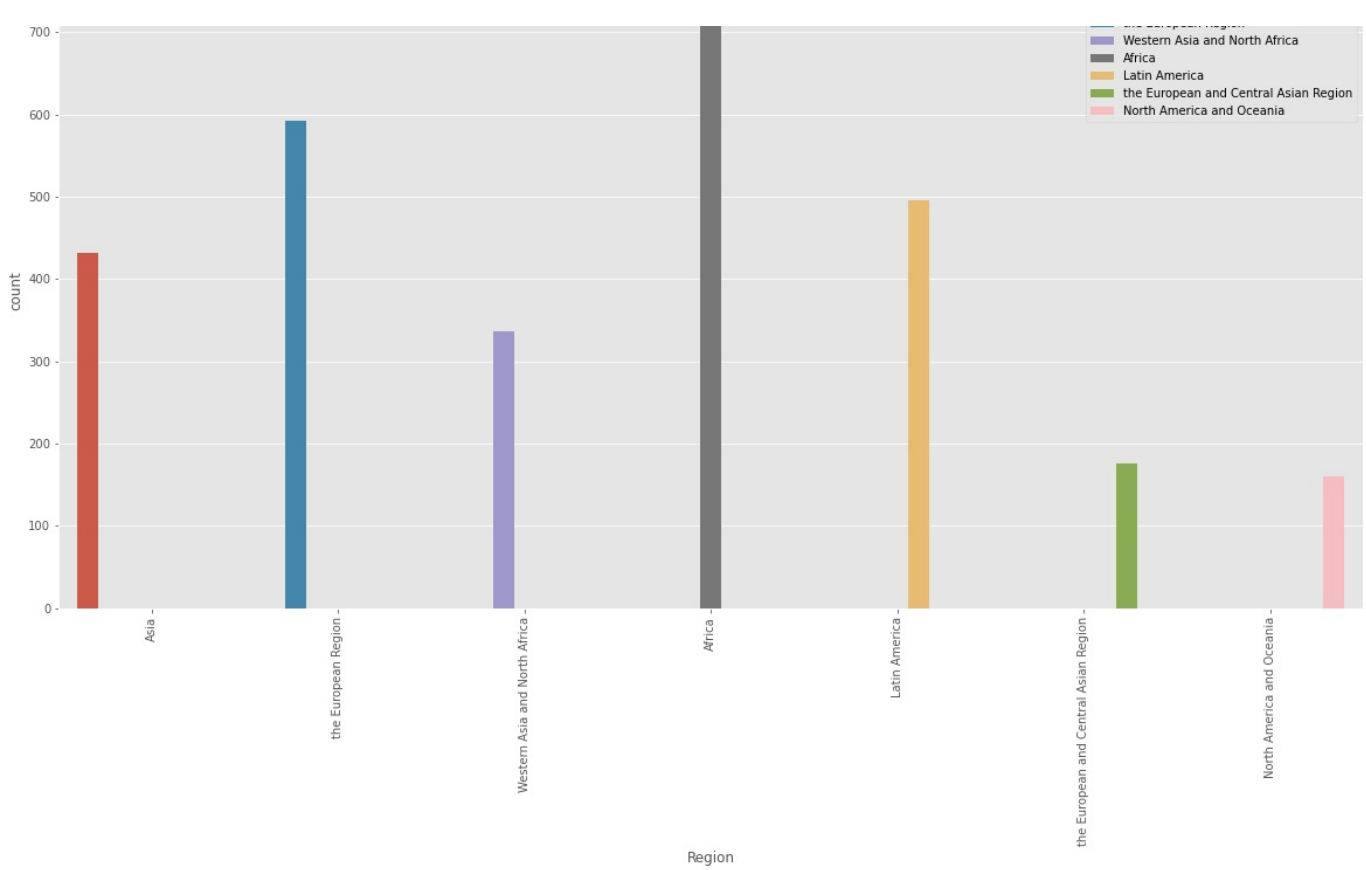

Region
Asia
the European Region
Western Asia and North Africa
Africa
Latin America
the European and Central Asian Region
North America and Oceania
90
80
70
60
50
40
Life_Expectancy
Asia
the European Region
Western Asia and North Africa
Africa
Latin America
North America and Oceania

ii. COUNTPLOT WITH HUE

In [53]:
```
plot_bivariate_graphs(life_expectancy_data,factor_cols,'c')
```

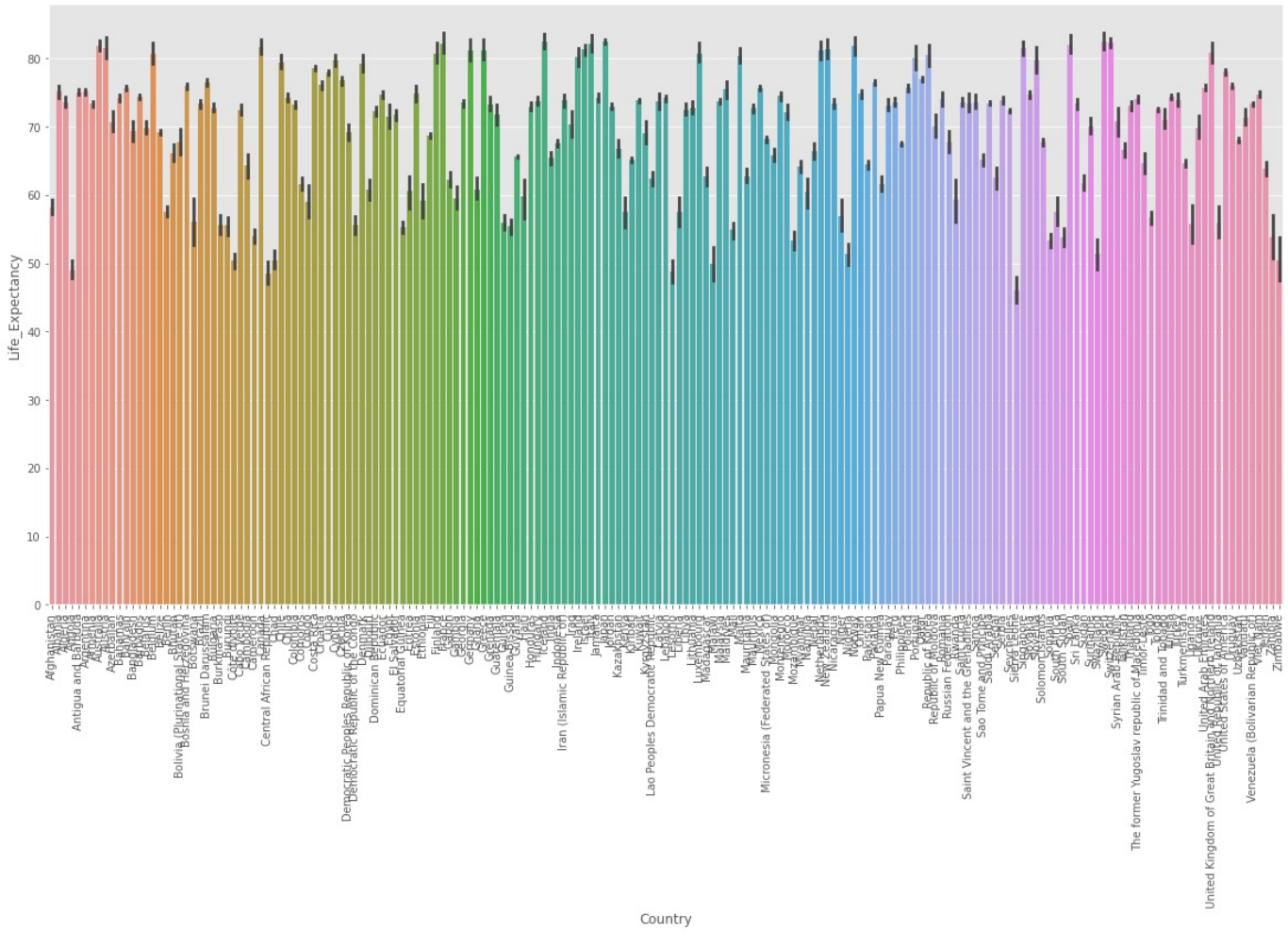


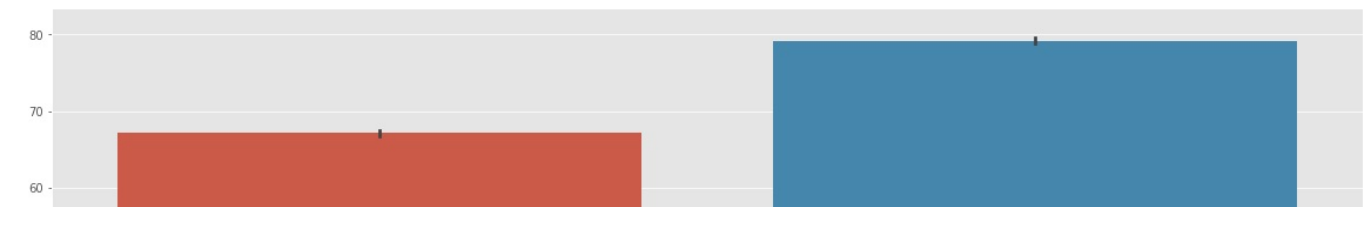


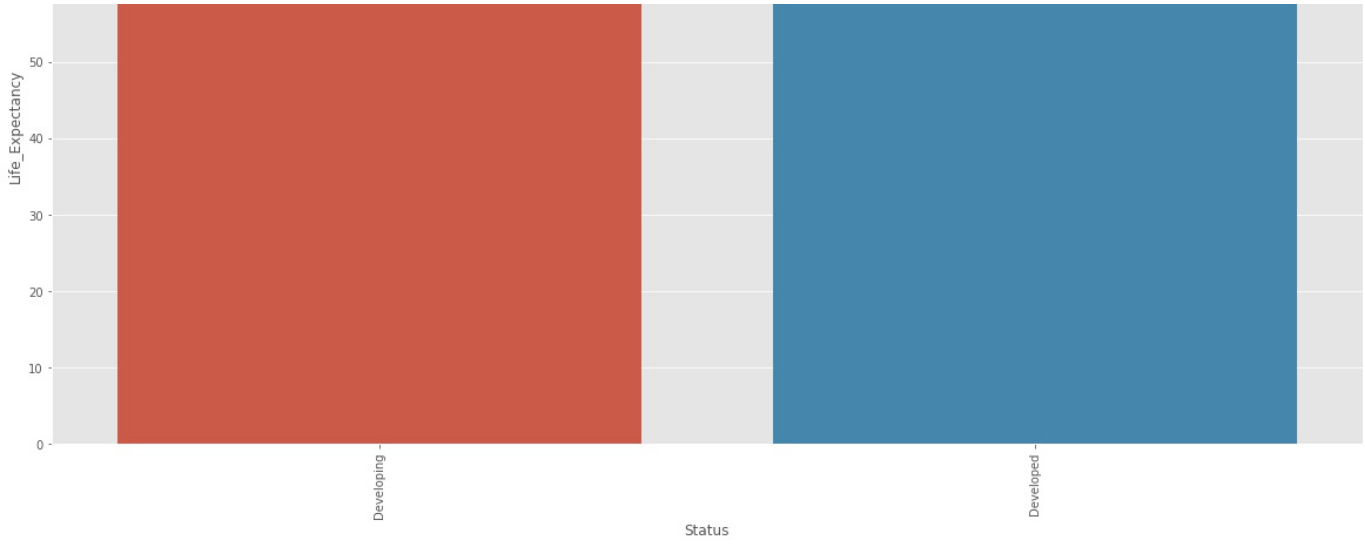


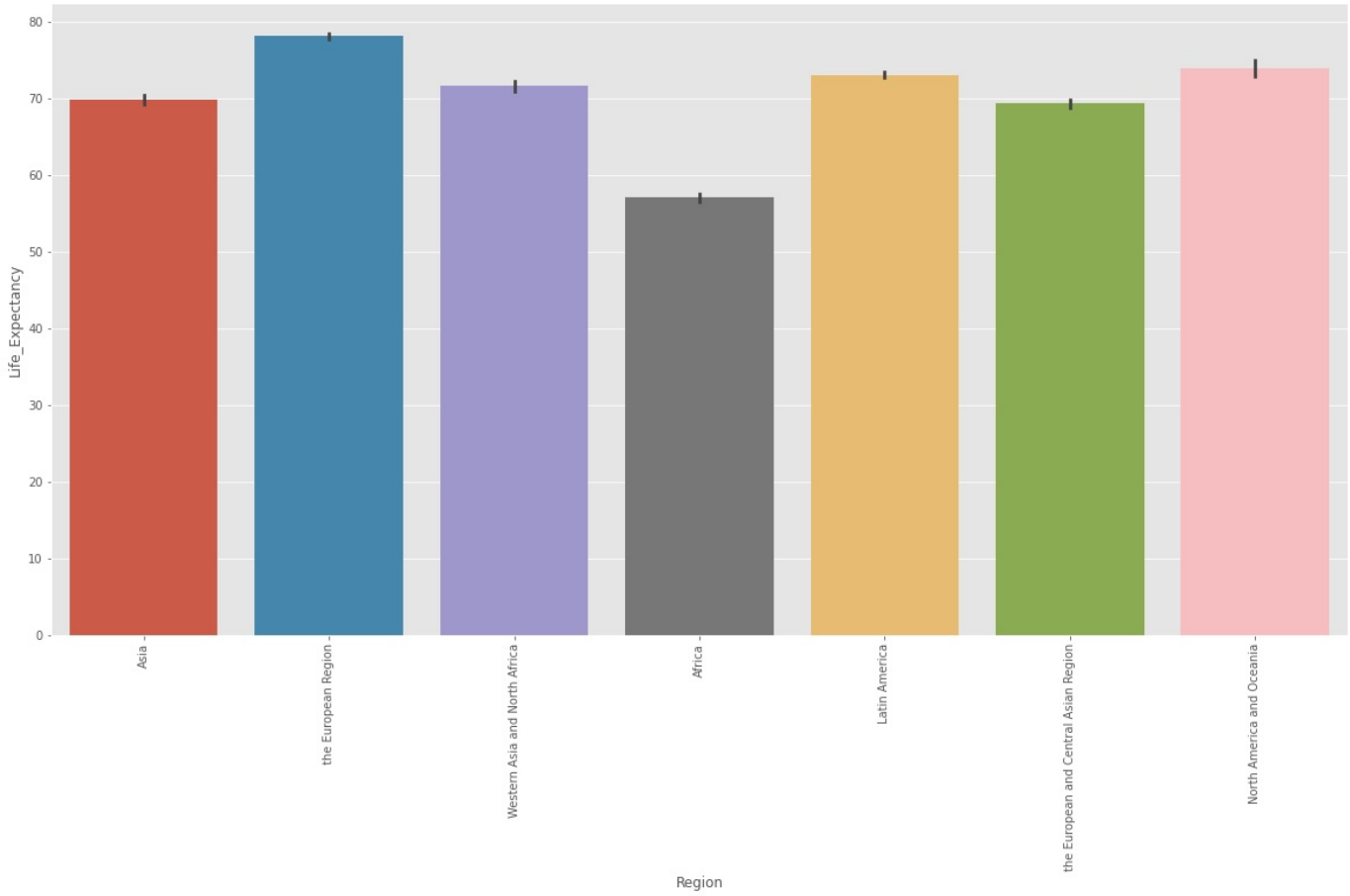
In [54]:

```
plot_bivariate_graphs(life_expectancy_data,factor_cols,'ba')
```









INSIGHTS FROM EDA

- AS WE CAN CLEARLY SEE THAT THE DEVELOPED COUNTRIES (North America and Ocenia and the European region) HAD HIGH LIFE EXPECTANCY AS COMPARED TO DEVELOPING COUNTRIES.

- AS YEARS INCREASE THE TECHNOLOGIES AND MEDICATIONS HAVE HELPED PEOPLE LIVE A BETTER AND MORE SPAN OF LIFE THAN EARLIER (LIFE EXPECTANCY VALUE IS DECREASING FROM 2010 TILL 2014)

- INFANT DEATH RATES DO NOT SHOW MUCH RELATION WITH LIFE EXPECTANCY

- ADULT MORTALITY HAS A SIGNIFICANT RELATIONSHIP WITH LIFE EXPECTANCY

- PROPER SCHOOLING LEADS TO HEALTHY HABBITS AND DISCIPLINE IN LIFE HENCE SCHOOLING IS SEEN TO HAVE HIGH CORRELATION WITH LIFE EXPECTANCY

- PERCENTAGE EXPENDITURE ON HEALTH MORE RESULTING INTO MORE LIFE EXPECTANCY

- AFRICAN REGION HAVE LESS LIFE EXPECTANCY OBSERVED AND EUROPEAN REGION HAS MORE LIFE EXPECTANCY

- ALCOHOL INTAKE MORE IN MIDDLE AGE GROUP ITS CONSUMPTION DOES NOT AFFECT COUNTRIES AVERAGE LIFE EXPECTANCY

- HIV/AIDS, BMI, DIPTHERIA, THINNESS_1_19_YEARS, THINNESS_5_9_YEARS, POLIO, GDP and ALCOHOL HAVE MEDIUM CORRELATION BETWEEN LIFE EXPECTANCY

- COUNTRIES WITH AND AVERAGE BMI BETWEEN 50 AND 70 CITIZENS HAVE LONG LIFE

- PERCENTAGE_EXPENDITURE, HEPATITIS_B, TOTAL_EXPENDITURE, UNDER_5_DEATHS, INFANT_DEATHS, YEAR, AND MEASLES HAVE LOW CORRELATION BETWEEN LIFE EXPECTANCY

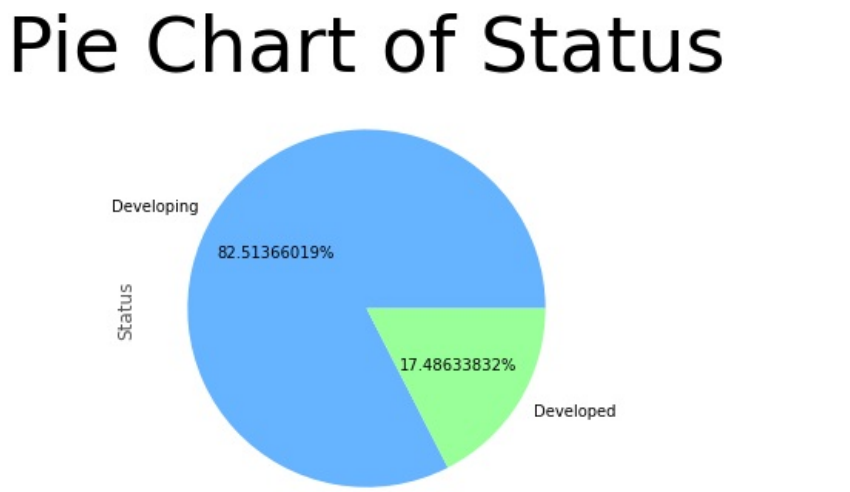
- GDP IMPROVES AVERAGE LIFE EXPECTANCY IN DEVELOPING COUNTRIES

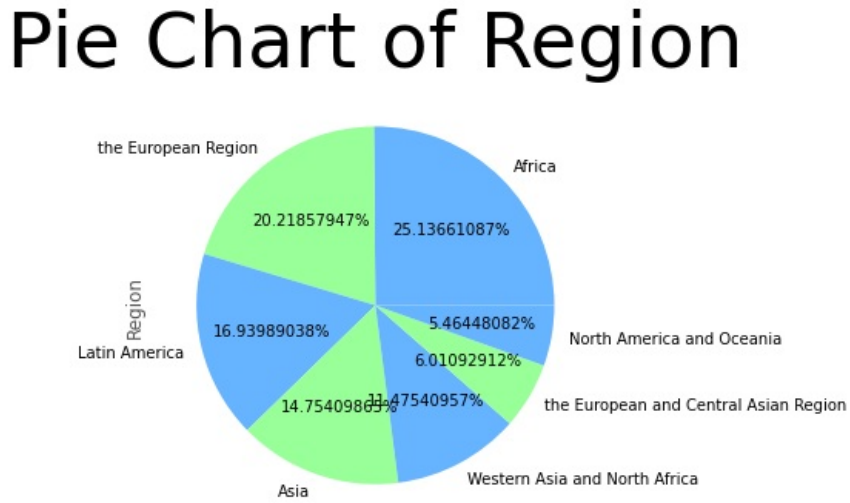
In [55]:
```
factor_cols
```

Out[55]:
```
array(['Country', 'Status', 'Region'], dtype=object)
```

In [56]:
```
for c in factor_cols:
    plt.figure(figsize=(8,5))
    if c!= "Country":
        life_expectancy_data[c].value_counts().plot.pie(autopct='%1.8f%%', colors = ['#66b3ff','#99ff99'])
        plt.title("Pie Chart of "+c , fontdict={'fontsize': 50})
        plt.tight_layout()
```

```
<Figure size 576x360 with 0 Axes>
```

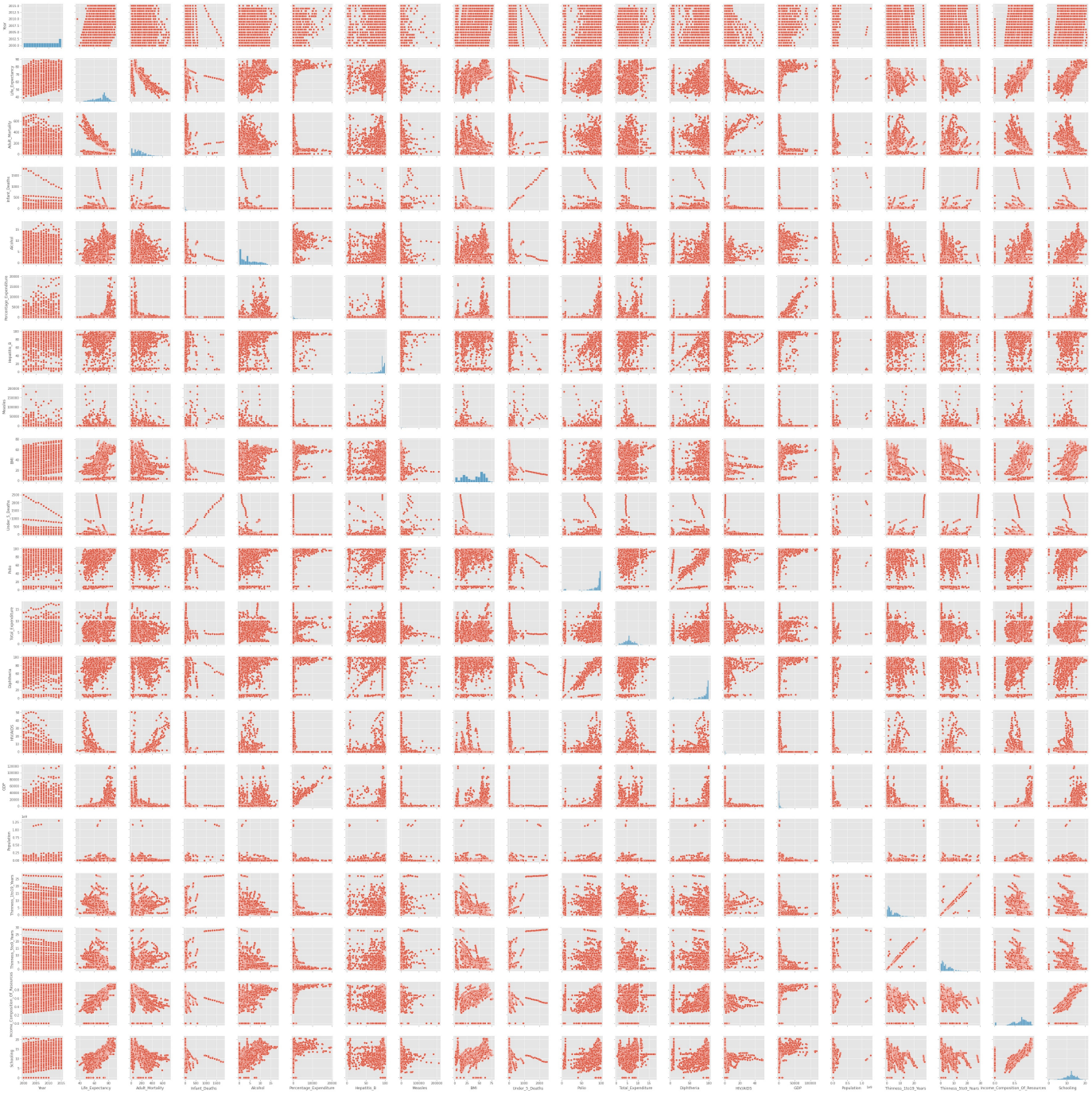




In [57]:
```
plt.figure(figsize=(20,20))
sns.pairplot(life_expectancy_data)
```

Out[57]:
```
<seaborn.axisgrid.PairGrid at 0x26bfdbaeeb0>
```

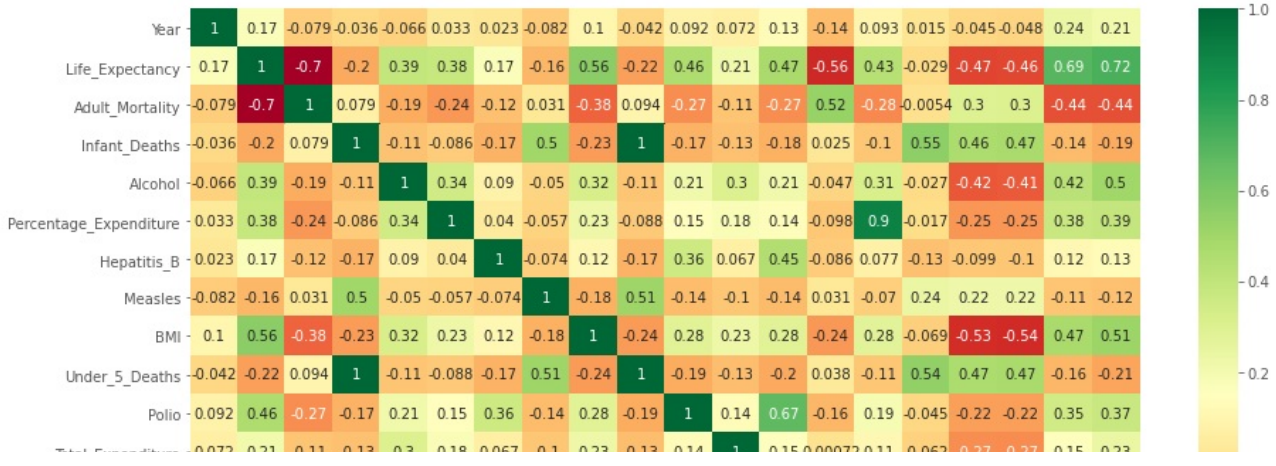
```
<Figure size 1440x1440 with 0 Axes>
```



CHECK MULTICOLINARITY

In [58]:
```
plt.figure(figsize=(15,10))
sns.heatmap(life_expectancy_data.corr(),annot = True,cmap='RdYlGn')
```

Out[58]: <AxesSubplot:>



INSIGHTS OF HEATMAP (CORRELATION METRICS)

- OUR TARGET COLUMN Life_Expectancy IS HIGHLY CORRELATED WITH Schooling,Income_Composition_Of_ResourcesAND NEGTIVELY HIGHLY CORRELATED WITH Adult_Mortality

- ALSO THERE EXIST MUTUAL HIGH CORRELATION AMONG GDP AND Percentage_Expenditure,Thinness_1to19_Years AND Thinness_5to9_Years,Income_Composition_Of_Resources AND Schooling,Polio AND Diphtheria

- LOOKING TO THIS HIGH CORRELATION WE CAN DELETE THE FOLLOWING COLUMNS 'Diphtheria', 'GDP', 'Thinness_5to9_Years','Schooling'

In [59]:
```
cor_matrix = life_expectancy_data.corr().abs()
print(cor_matrix)
```

```
                                     Year  Life_Expectancy  Adult_Mortality  \
Year                             1.000000         0.170033         0.079052
Life_Expectancy                  0.170033         1.000000         0.696359
Adult_Mortality                  0.079052         0.696359         1.000000
Infant_Deaths                    0.036464         0.196557         0.078756
Alcohol                          0.066014         0.390633         0.191066
Percentage_Expenditure           0.032723         0.381864         0.242860
Hepatitis_B                      0.022671         0.171255         0.123971
Measles                          0.081840         0.157586         0.031176
BMI                              0.104054         0.558778         0.380498
Under_5_Deaths                   0.041980         0.222529         0.094146
Polio                            0.091727         0.459458         0.270597
Total_Expenditure                0.071597         0.209590         0.112165
Diphtheria                       0.131852         0.473268         0.270877
HIV/AIDS                         0.138789         0.556556         0.523821
GDP                              0.093170         0.430992         0.281715
Population                       0.014749         0.028846         0.005389
Thinness_1to19_Years             0.044805         0.467859         0.296076
Thinness_5to9_Years              0.047856         0.462316         0.301578
Income_Composition_Of_Resources  0.235866         0.688591         0.436268
Schooling                        0.207311         0.717314         0.435926

                                 Infant_Deaths   Alcohol  \
Year                                  0.036464  0.066014
Life_Expectancy                       0.196557  0.390633
Adult_Mortality                       0.078756  0.191066
Infant_Deaths                         1.000000  0.113901
Alcohol                               0.113901  1.000000
Percentage_Expenditure                0.085906  0.344304
Hepatitis_B                           0.168393  0.089531
Measles                               0.501038  0.050466
BMI                                   0.226953  0.321078
Under_5_Deaths                        0.996628  0.110781
Polio                                 0.171273  0.212066
Total_Expenditure                     0.126480  0.302206
Diphtheria                            0.175747  0.212735
HIV/AIDS                              0.024955  0.047270
GDP                                   0.103176  0.312739
```

```
Population                         0.551608  0.027376
Thinness_1to19_Years               0.464762  0.416948
Thinness_5to9_Years                0.470397  0.405897
Income_Composition_Of_Resources    0.141329  0.419971
Schooling                          0.192421  0.499656

                                Percentage_Expenditure  Hepatitis_B  \
Year                                          0.032723     0.022671
Life_Expectancy                               0.381864     0.171255
Adult_Mortality                               0.242860     0.123971
Infant_Deaths                                 0.085906     0.168393
Alcohol                                       0.344304     0.089531
Percentage_Expenditure                        1.000000     0.039805
Hepatitis_B                                   0.039805     1.000000
Measles                                       0.056831     0.073544
BMI                                           0.229887     0.116741
Under_5_Deaths                                0.088152     0.171539
Polio                                         0.146546     0.362537
Total_Expenditure                             0.177297     0.066914
Diphtheria                                    0.142897     0.447373
HIV/AIDS                                      0.098230     0.086197
GDP                                           0.901802     0.076937
Population                                    0.017067     0.125716
Thinness_1to19_Years                          0.250729     0.099491
Thinness_5to9_Years                           0.252261     0.102500
Income_Composition_Of_Resources               0.375234     0.118166
Schooling                                     0.387937     0.134263

                                 Measles       BMI  Under_5_Deaths     Polio  \
Year                            0.081840  0.104054        0.041980  0.091727
Life_Expectancy                 0.157586  0.558778        0.222529  0.459458
Adult_Mortality                 0.031176  0.380498        0.094146  0.270597
Infant_Deaths                   0.501038  0.226953        0.996628  0.171273
Alcohol                         0.050466  0.321078        0.110781  0.212066
Percentage_Expenditure          0.056831  0.229887        0.088152  0.146546
Hepatitis_B                     0.073544  0.116741        0.171539  0.362537
Measles                         1.000000  0.176133        0.507718  0.136966
BMI                             0.176133  1.000000        0.237242  0.284244
Under_5_Deaths                  0.507718  0.237242        1.000000  0.189284
Polio                           0.136966  0.284244        0.189284  1.000000
Total_Expenditure               0.104306  0.227088        0.128170  0.137454
Diphtheria                      0.142680  0.283177        0.196224  0.672557
HIV/AIDS                        0.030673  0.243355        0.037783  0.159484
GDP                             0.069531  0.277081        0.106448  0.190584
Population                      0.237096  0.069296        0.539198  0.044924
Thinness_1to19_Years            0.224606  0.531257        0.466640  0.219364
Thinness_5to9_Years             0.220866  0.538229        0.471109  0.220140
Income_Composition_Of_Resources 0.110884  0.473444        0.159022  0.345770
Schooling                       0.121817  0.510540        0.207801  0.374263

                                Total_Expenditure  Diphtheria  HIV/AIDS  \
Year                                     0.071597    0.131852  0.138789
Life_Expectancy                          0.209590    0.473268  0.556556
Adult_Mortality                          0.112165    0.270877  0.523821
Infant_Deaths                            0.126480    0.175747  0.024955
Alcohol                                  0.302206    0.212735  0.047270
Percentage_Expenditure                   0.177297    0.142897  0.098230
Hepatitis_B                              0.066914    0.447373  0.086197
Measles                                  0.104306    0.142680  0.030673
BMI                                      0.227088    0.283177  0.243355
Under_5_Deaths                           0.128170    0.196224  0.037783
Polio                                    0.137454    0.672557  0.159484
Total_Expenditure                        1.000000    0.152959  0.000721
Diphtheria                               0.152959    1.000000  0.164768
HIV/AIDS                                 0.000721    0.164768  1.000000
GDP                                      0.113693    0.182592  0.123033
Population                               0.061530    0.033205  0.017285
Thinness_1to19_Years                     0.266595    0.227360  0.203110
Thinness_5to9_Years                      0.273160    0.220633  0.206328
Income_Composition_Of_Resources          0.154639    0.359984  0.247559
Schooling                                0.234157    0.377568  0.220577

                                     GDP  Population  Thinness_1to19_Years  \
Year                            0.093170    0.014749              0.044805
Life_Expectancy                 0.430992    0.028846              0.467859
Adult_Mortality                 0.281715    0.005389              0.296076
Infant_Deaths                   0.103176    0.551608              0.464762
Alcohol                         0.312739    0.027376              0.416948
Percentage_Expenditure          0.901802    0.017067              0.250729
Hepatitis_B                     0.076937    0.125716              0.099491
Measles                         0.069531    0.237096              0.224606
BMI                             0.277081    0.069296              0.531257
Under_5_Deaths                  0.106448    0.539198              0.466640
```

```
Polio                            0.190584    0.044924             0.219364
Total_Expenditure                0.113693    0.061530             0.266595
Diphtheria                       0.182592    0.033205             0.227360
HIV/AIDS                         0.123033    0.017285             0.203110
GDP                              1.000000    0.025480             0.264649
Population                       0.025480    1.000000             0.236263
Thinness_1to19_Years             0.264649    0.236263             1.000000
Thinness_5to9_Years              0.268260    0.235127             0.939035
Income_Composition_Of_Resources  0.436577    0.017047             0.401183
Schooling                        0.434110    0.037340             0.442320

                                 Thinness_5to9_Years  \
Year                                        0.047856
Life_Expectancy                             0.462316
Adult_Mortality                             0.301578
Infant_Deaths                               0.470397
Alcohol                                     0.405897
Percentage_Expenditure                      0.252261
Hepatitis_B                                 0.102500
Measles                                     0.220866
BMI                                         0.538229
Under_5_Deaths                              0.471109
Polio                                       0.220140
Total_Expenditure                           0.273160
Diphtheria                                  0.220633
HIV/AIDS                                    0.206328
GDP                                         0.268260
Population                                  0.235127
Thinness_1to19_Years                        0.939035
Thinness_5to9_Years                         1.000000
Income_Composition_Of_Resources             0.390684
Schooling                                   0.432035

                                 Income_Composition_Of_Resources  Schooling
Year                                                    0.235866   0.207311
Life_Expectancy                                         0.688591   0.717314
Adult_Mortality                                         0.436268   0.435926
Infant_Deaths                                           0.141329   0.192421
Alcohol                                                 0.419971   0.499656
Percentage_Expenditure                                  0.375234   0.387937
Hepatitis_B                                             0.118166   0.134263
Measles                                                 0.110884   0.121817
BMI                                                     0.473444   0.510540
Under_5_Deaths                                          0.159022   0.207801
Polio                                                   0.345770   0.374263
Total_Expenditure                                       0.154639   0.234157
Diphtheria                                              0.359984   0.377568
HIV/AIDS                                                0.247559   0.220577
GDP                                                     0.436577   0.434110
Population                                              0.017047   0.037340
Thinness_1to19_Years                                    0.401183   0.442320
Thinness_5to9_Years                                     0.390684   0.432035
Income_Composition_Of_Resources                         1.000000   0.799817
Schooling                                               0.799817   1.000000
```

In [60]:

```
upper_tri = cor_matrix.where(np.triu(np.ones(cor_matrix.shape),k=1).astype(np.bool))
print(upper_tri)
```

```
                                 Year  Life_Expectancy  Adult_Mortality  \
Year                              NaN         0.170033         0.079052
Life_Expectancy                   NaN              NaN         0.696359
Adult_Mortality                   NaN              NaN              NaN
Infant_Deaths                     NaN              NaN              NaN
Alcohol                           NaN              NaN              NaN
Percentage_Expenditure            NaN              NaN              NaN
Hepatitis_B                       NaN              NaN              NaN
Measles                           NaN              NaN              NaN
BMI                               NaN              NaN              NaN
Under_5_Deaths                    NaN              NaN              NaN
Polio                             NaN              NaN              NaN
Total_Expenditure                 NaN              NaN              NaN
Diphtheria                        NaN              NaN              NaN
HIV/AIDS                          NaN              NaN              NaN
GDP                               NaN              NaN              NaN
Population                        NaN              NaN              NaN
Thinness_1to19_Years              NaN              NaN              NaN
Thinness_5to9_Years               NaN              NaN              NaN
Income_Composition_Of_Resources   NaN              NaN              NaN
Schooling                         NaN              NaN              NaN
```

	Infant_Deaths	Alcohol
Year	0.036464	0.066014
Life_Expectancy	0.196557	0.390633
Adult_Mortality	0.078756	0.191066
Infant_Deaths	NaN	0.113901
Alcohol	NaN	NaN
Percentage_Expenditure	NaN	NaN
Hepatitis_B	NaN	NaN
Measles	NaN	NaN
BMI	NaN	NaN
Under_5_Deaths	NaN	NaN
Polio	NaN	NaN
Total_Expenditure	NaN	NaN
Diphtheria	NaN	NaN
HIV/AIDS	NaN	NaN
GDP	NaN	NaN
Population	NaN	NaN
Thinness_1to19_Years	NaN	NaN
Thinness_5to9_Years	NaN	NaN
Income_Composition_Of_Resources	NaN	NaN
Schooling	NaN	NaN

	Percentage_Expenditure	Hepatitis_B
Year	0.032723	0.022671
Life_Expectancy	0.381864	0.171255
Adult_Mortality	0.242860	0.123971
Infant_Deaths	0.085906	0.168393
Alcohol	0.344304	0.089531
Percentage_Expenditure	NaN	0.039805
Hepatitis_B	NaN	NaN
Measles	NaN	NaN
BMI	NaN	NaN
Under_5_Deaths	NaN	NaN
Polio	NaN	NaN
Total_Expenditure	NaN	NaN
Diphtheria	NaN	NaN
HIV/AIDS	NaN	NaN
GDP	NaN	NaN
Population	NaN	NaN
Thinness_1to19_Years	NaN	NaN
Thinness_5to9_Years	NaN	NaN
Income_Composition_Of_Resources	NaN	NaN
Schooling	NaN	NaN

	Measles	BMI	Under_5_Deaths	Polio
Year	0.081840	0.104054	0.041980	0.091727
Life_Expectancy	0.157586	0.558778	0.222529	0.459458
Adult_Mortality	0.031176	0.380498	0.094146	0.270597
Infant_Deaths	0.501038	0.226953	0.996628	0.171273
Alcohol	0.050466	0.321078	0.110781	0.212066
Percentage_Expenditure	0.056831	0.229887	0.088152	0.146546
Hepatitis_B	0.073544	0.116741	0.171539	0.362537
Measles	NaN	0.176133	0.507718	0.136966
BMI	NaN	NaN	0.237242	0.284244
Under_5_Deaths	NaN	NaN	NaN	0.189284
Polio	NaN	NaN	NaN	NaN
Total_Expenditure	NaN	NaN	NaN	NaN
Diphtheria	NaN	NaN	NaN	NaN
HIV/AIDS	NaN	NaN	NaN	NaN
GDP	NaN	NaN	NaN	NaN
Population	NaN	NaN	NaN	NaN
Thinness_1to19_Years	NaN	NaN	NaN	NaN
Thinness_5to9_Years	NaN	NaN	NaN	NaN
Income_Composition_Of_Resources	NaN	NaN	NaN	NaN
Schooling	NaN	NaN	NaN	NaN

	Total_Expenditure	Diphtheria	HIV/AIDS
Year	0.071597	0.131852	0.138789
Life_Expectancy	0.209590	0.473268	0.556556
Adult_Mortality	0.112165	0.270877	0.523821
Infant_Deaths	0.126480	0.175747	0.024955
Alcohol	0.302206	0.212735	0.047270
Percentage_Expenditure	0.177297	0.142897	0.098230
Hepatitis_B	0.066914	0.447373	0.086197
Measles	0.104306	0.142680	0.030673
BMI	0.227088	0.283177	0.243355
Under_5_Deaths	0.128170	0.196224	0.037783
Polio	0.137454	0.672557	0.159484
Total_Expenditure	NaN	0.152959	0.000721
Diphtheria	NaN	NaN	0.164768
HIV/AIDS	NaN	NaN	NaN
GDP	NaN	NaN	NaN
Population	NaN	NaN	NaN

```
Thinness_1to19_Years                          NaN         NaN       NaN
Thinness_5to9_Years                           NaN         NaN       NaN
Income_Composition_Of_Resources               NaN         NaN       NaN
Schooling                                     NaN         NaN       NaN

                                     GDP  Population  Thinness_1to19_Years  \
Year                            0.093170    0.014749              0.044805
Life_Expectancy                 0.430992    0.028846              0.467859
Adult_Mortality                 0.281715    0.005389              0.296076
Infant_Deaths                   0.103176    0.551608              0.464762
Alcohol                         0.312739    0.027376              0.416948
Percentage_Expenditure          0.901802    0.017067              0.250729
Hepatitis_B                     0.076937    0.125716              0.099491
Measles                         0.069531    0.237096              0.224606
BMI                             0.277081    0.069296              0.531257
Under_5_Deaths                  0.106448    0.539198              0.466640
Polio                           0.190584    0.044924              0.219364
Total_Expenditure               0.113693    0.061530              0.266595
Diphtheria                      0.182592    0.033205              0.227360
HIV/AIDS                        0.123033    0.017285              0.203110
GDP                                  NaN    0.025480              0.264649
Population                           NaN         NaN              0.236263
Thinness_1to19_Years                 NaN         NaN                   NaN
Thinness_5to9_Years                  NaN         NaN                   NaN
Income_Composition_Of_Resources      NaN         NaN                   NaN
Schooling                            NaN         NaN                   NaN

                                Thinness_5to9_Years  \
Year                                       0.047856
Life_Expectancy                            0.462316
Adult_Mortality                            0.301578
Infant_Deaths                              0.470397
Alcohol                                    0.405897
Percentage_Expenditure                     0.252261
Hepatitis_B                                0.102500
Measles                                    0.220866
BMI                                        0.538229
Under_5_Deaths                             0.471109
Polio                                      0.220140
Total_Expenditure                          0.273160
Diphtheria                                 0.220633
HIV/AIDS                                   0.206328
GDP                                        0.268260
Population                                 0.235127
Thinness_1to19_Years                       0.939035
Thinness_5to9_Years                             NaN
Income_Composition_Of_Resources                 NaN
Schooling                                       NaN

                                Income_Composition_Of_Resources  Schooling
Year                                                   0.235866   0.207311
Life_Expectancy                                        0.688591   0.717314
Adult_Mortality                                        0.436268   0.435926
Infant_Deaths                                          0.141329   0.192421
Alcohol                                                0.419971   0.499656
Percentage_Expenditure                                 0.375234   0.387937
Hepatitis_B                                            0.118166   0.134263
Measles                                                0.110884   0.121817
BMI                                                    0.473444   0.510540
Under_5_Deaths                                         0.159022   0.207801
Polio                                                  0.345770   0.374263
Total_Expenditure                                      0.154639   0.234157
Diphtheria                                             0.359984   0.377568
HIV/AIDS                                               0.247559   0.220577
GDP                                                    0.436577   0.434110
Population                                             0.017047   0.037340
Thinness_1to19_Years                                   0.401183   0.442320
Thinness_5to9_Years                                    0.390684   0.432035
Income_Composition_Of_Resources                             NaN   0.799817
Schooling                                                   NaN        NaN
```

In [61]:

```
to_drop = [column for column in upper_tri.columns if any(upper_tri[column] > 0.60)]
print(to_drop)
```

```
['Adult_Mortality', 'Under_5_Deaths', 'Diphtheria', 'GDP', 'Thinness_5to9_Years', 'Income_Composition_Of_Resource
s', 'Schooling']
```

In [62]:

```
to_drop = ['Polio','Percentage_Expenditure','Thinness_5to9_Years']
```

DROPPED COLUMNS WITH MULTICOLINEARITY DATASET

In [63]:
```
life_expectancy_data.drop(columns=to_drop,inplace=True)
```

In [64]:
```
life_expectancy_data.columns
```

Out[64]:
```
Index(['Country', 'Year', 'Status', 'Life_Expectancy', 'Adult_Mortality',
       'Infant_Deaths', 'Alcohol', 'Hepatitis_B', 'Measles', 'BMI',
       'Under_5_Deaths', 'Total_Expenditure', 'Diphtheria', 'HIV/AIDS', 'GDP',
       'Population', 'Thinness_1to19_Years', 'Income_Composition_Of_Resources',
       'Schooling', 'Region'],
      dtype='object')
```

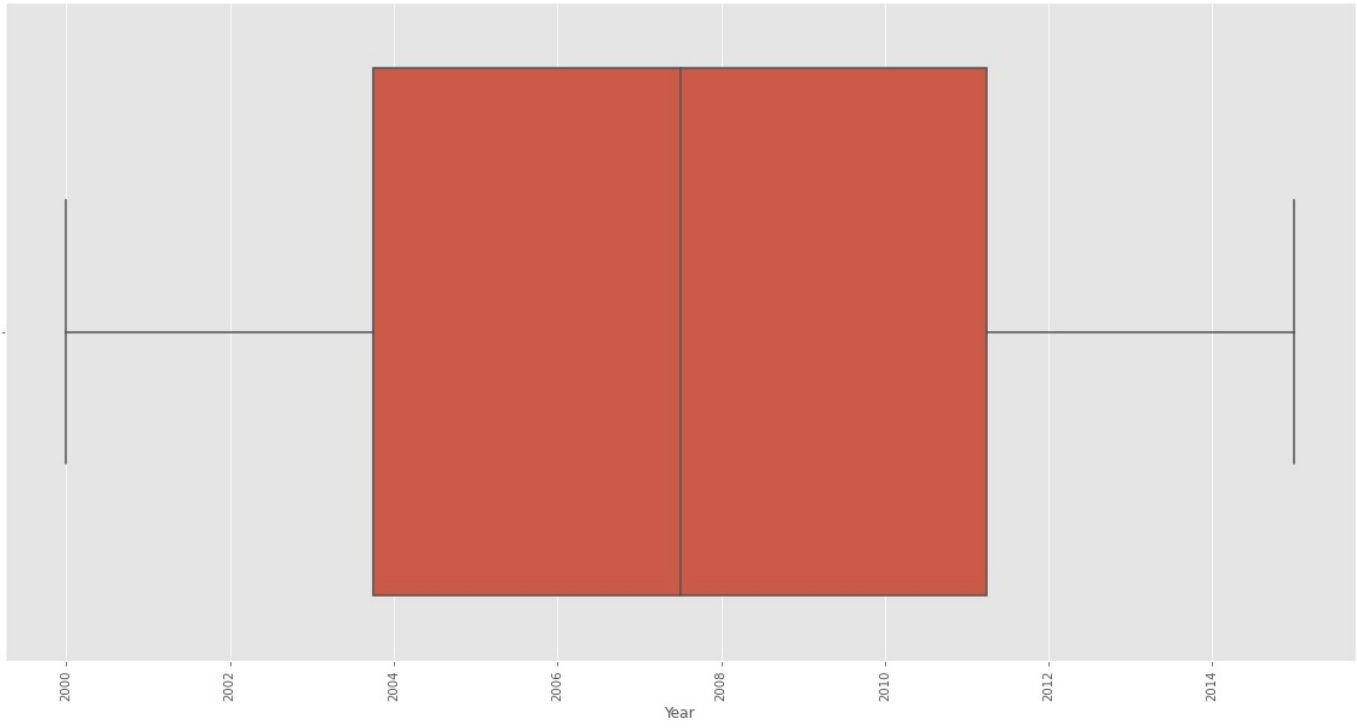
HANDLING OUTLIERS

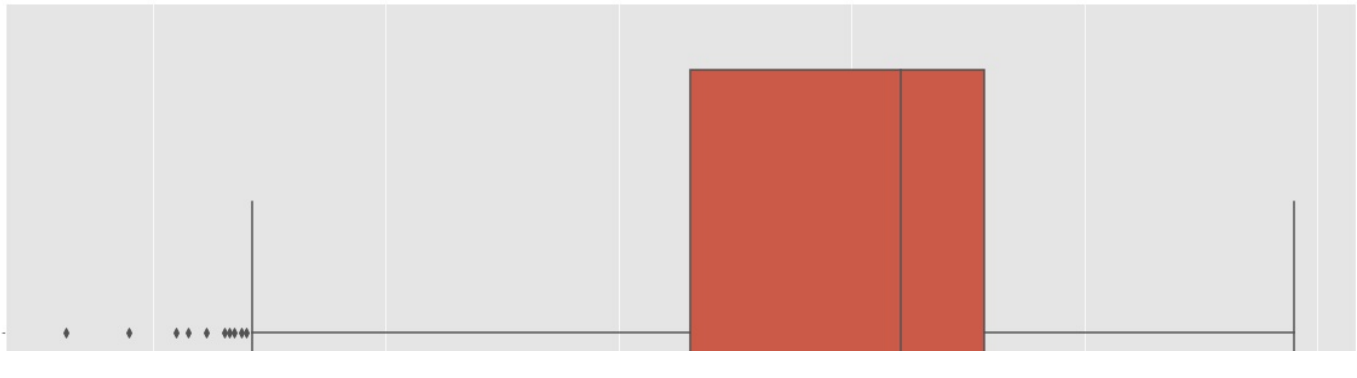
In [65]:
```
numeric_cols,factor_cols=split_cols(life_expectancy_data)
```

In [66]:
```
numeric_cols
```

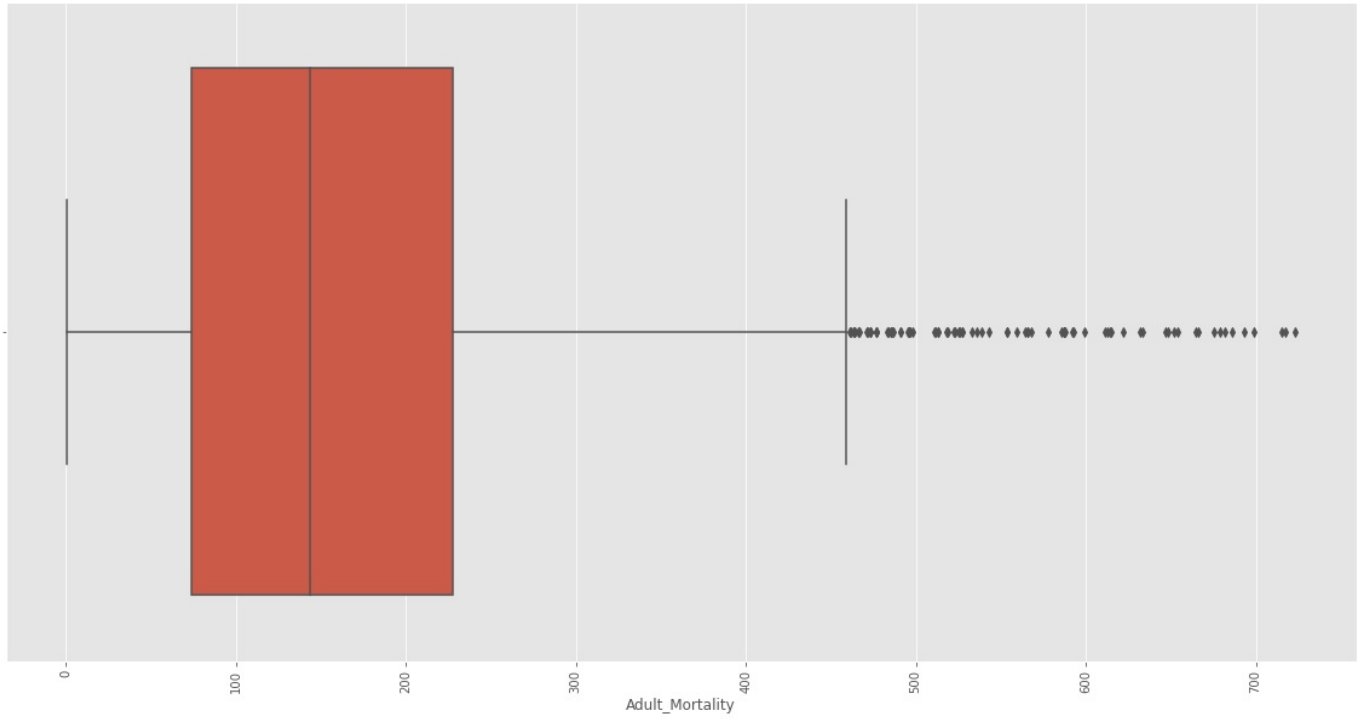
Out[66]:
```
array(['Year', 'Life_Expectancy', 'Adult_Mortality', 'Infant_Deaths',
       'Alcohol', 'Hepatitis_B', 'Measles', 'BMI', 'Under_5_Deaths',
       'Total_Expenditure', 'Diphtheria', 'HIV/AIDS', 'GDP', 'Population',
       'Thinness_1to19_Years', 'Income_Composition_Of_Resources',
       'Schooling'], dtype=object)
```

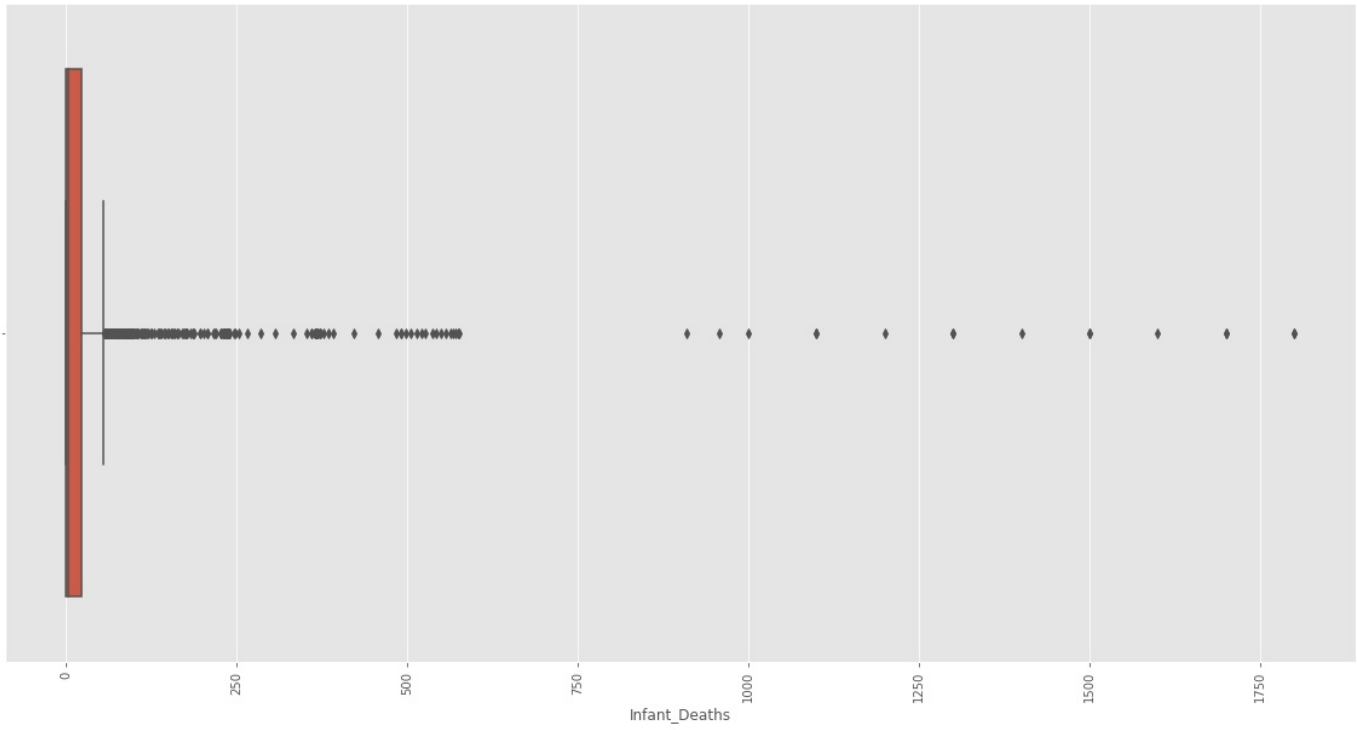
In [67]:
```
plot_univariate_graphs(life_expectancy_data,numeric_cols,'bo')
```

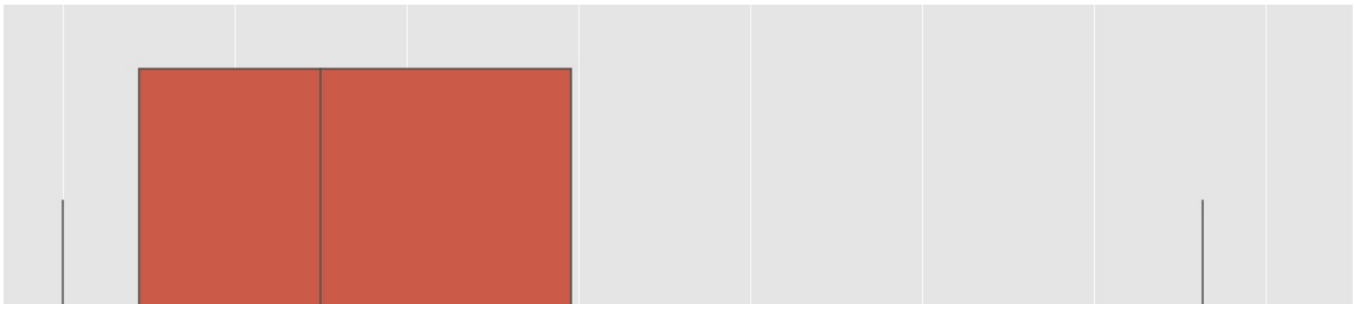


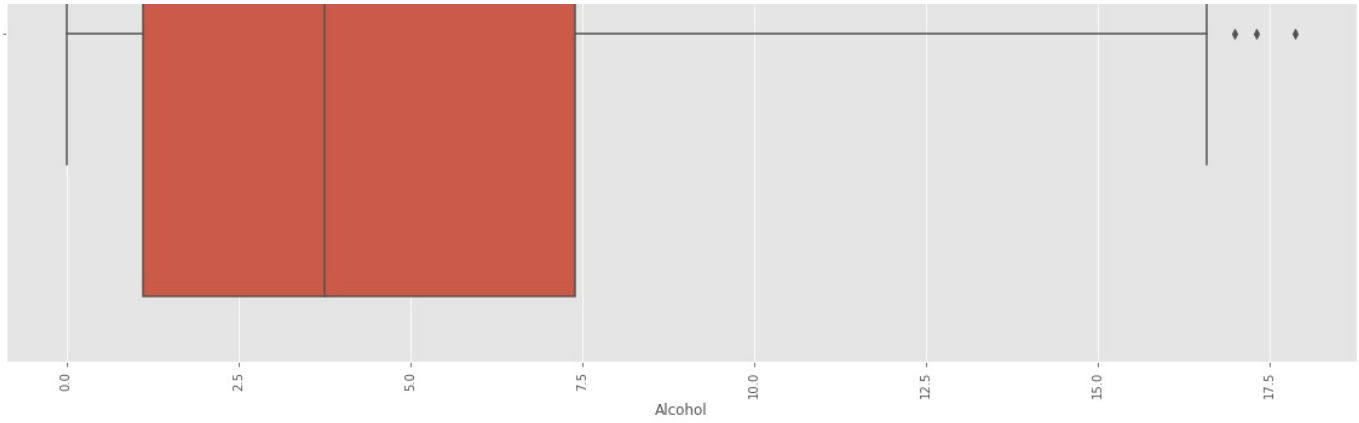


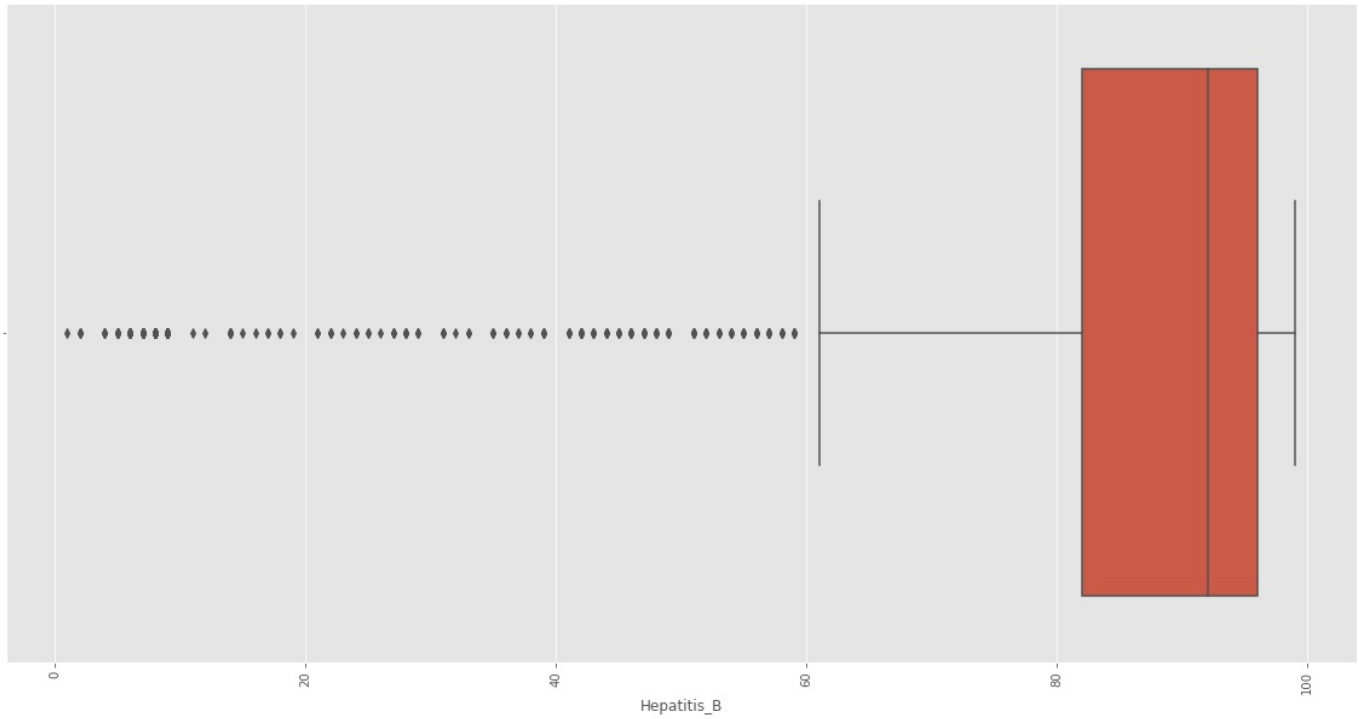

40
50
60
70
80
90
Life_Expectancy

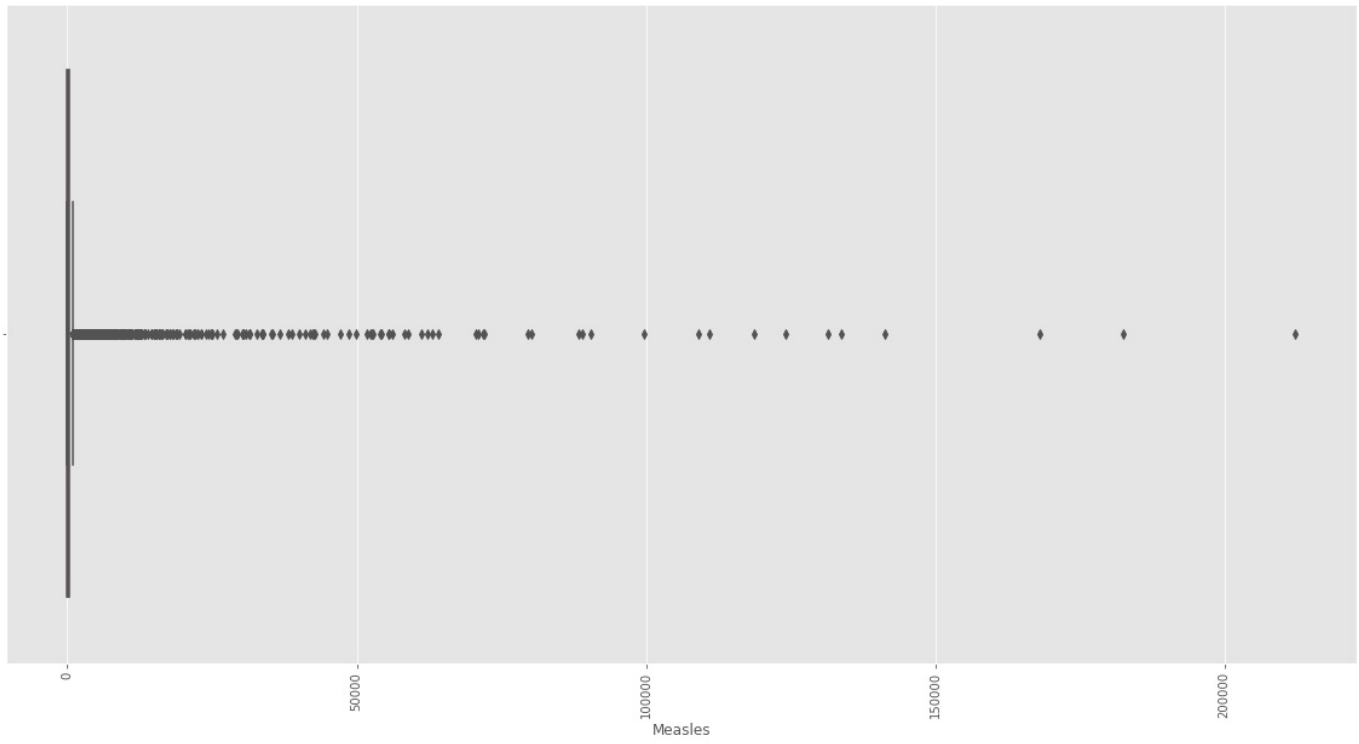

0
100
200
300
400
500
600
700
Adult_Mortality

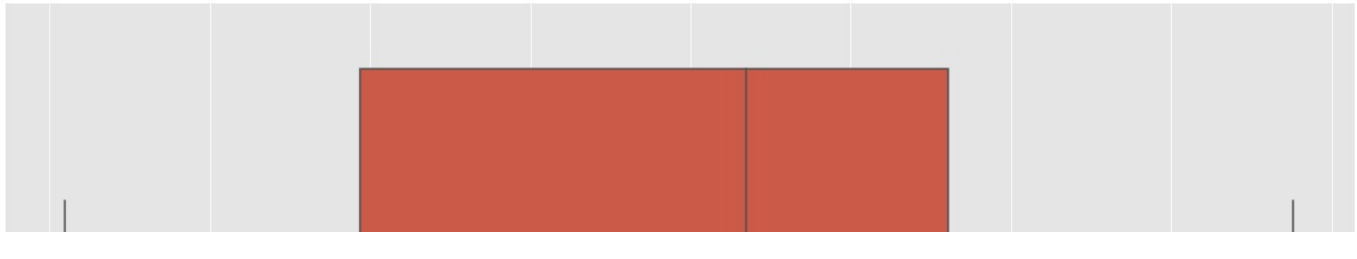

0
250
500
750
1000
1250
1500
1750
Infant_Deaths

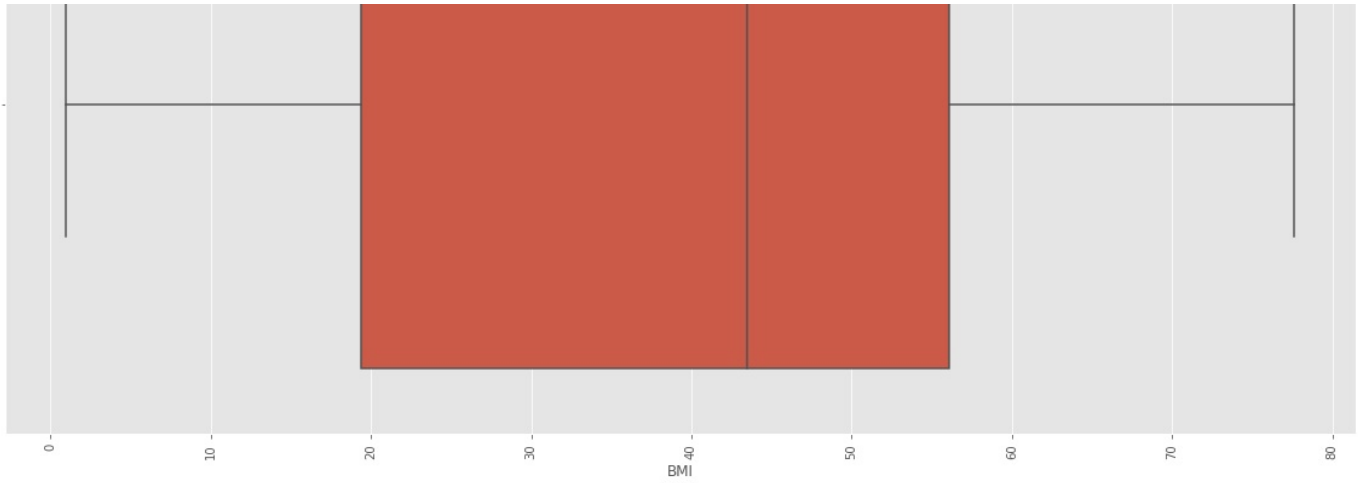


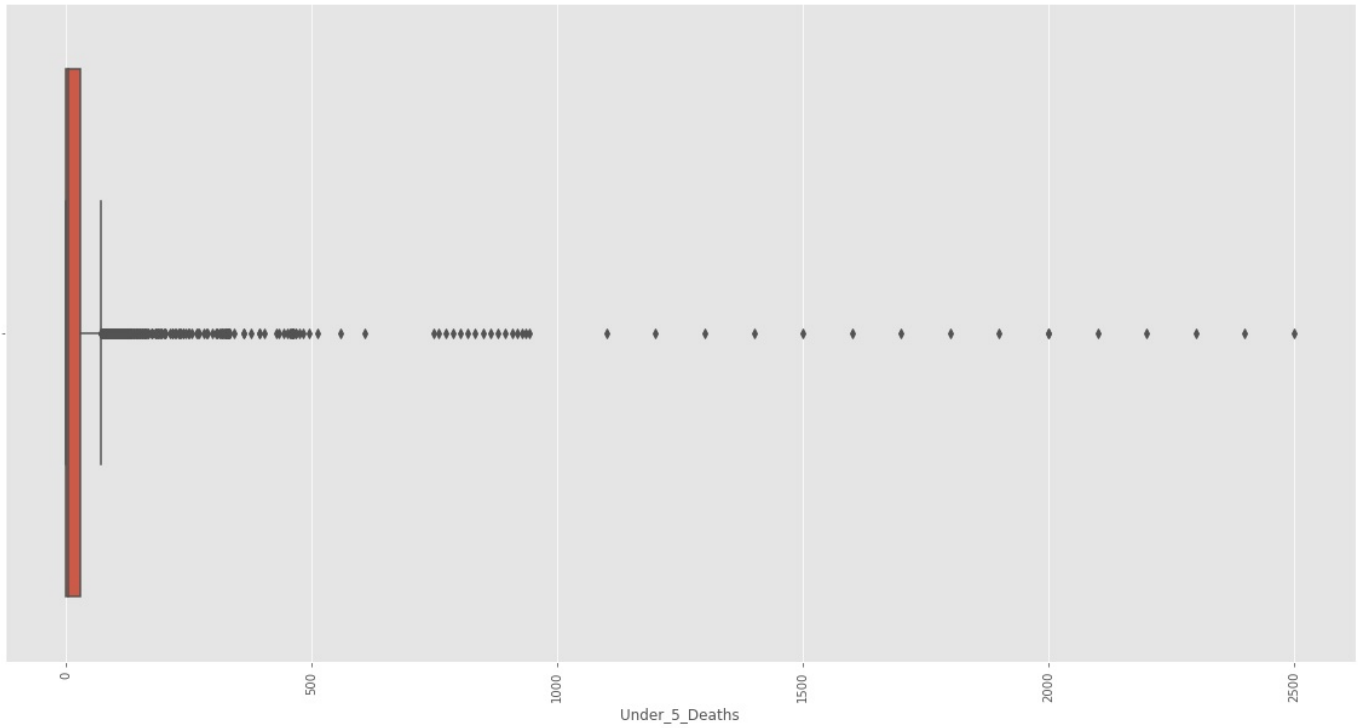

0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Alcohol

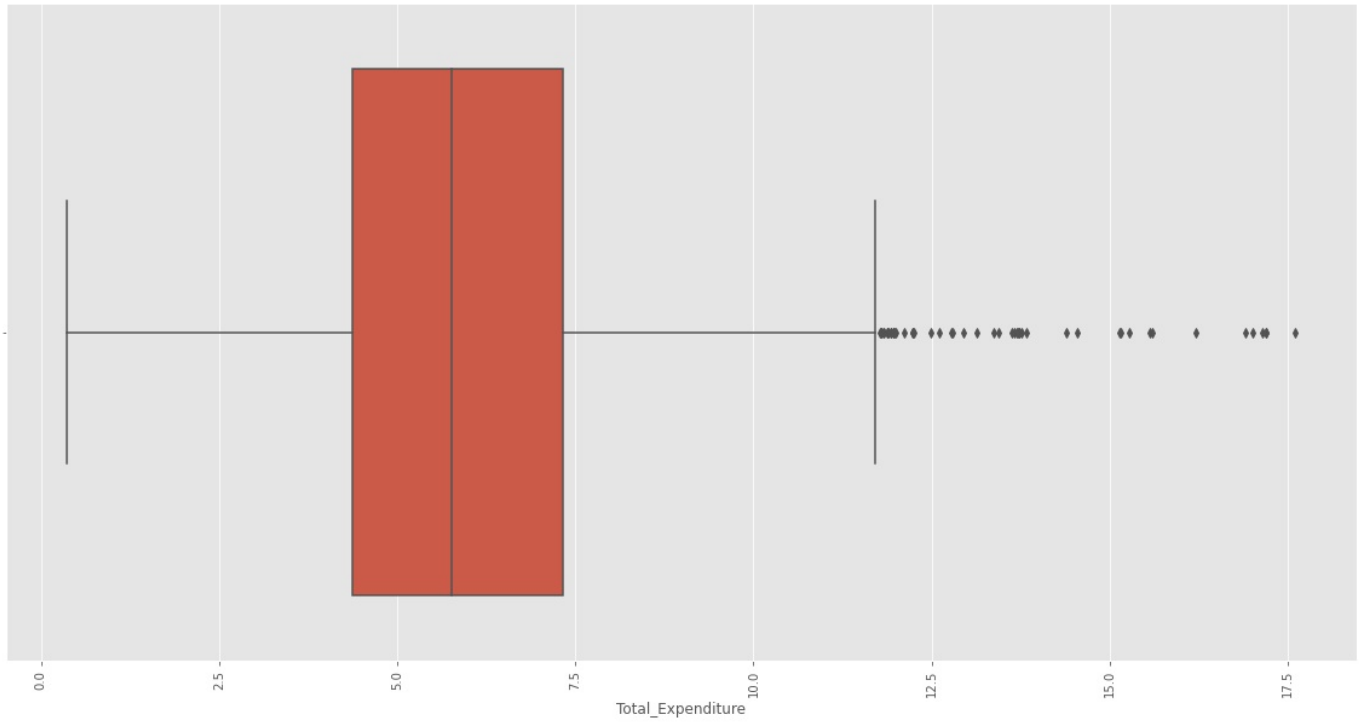

0
20
40
60
80
100
Hepatitis_B

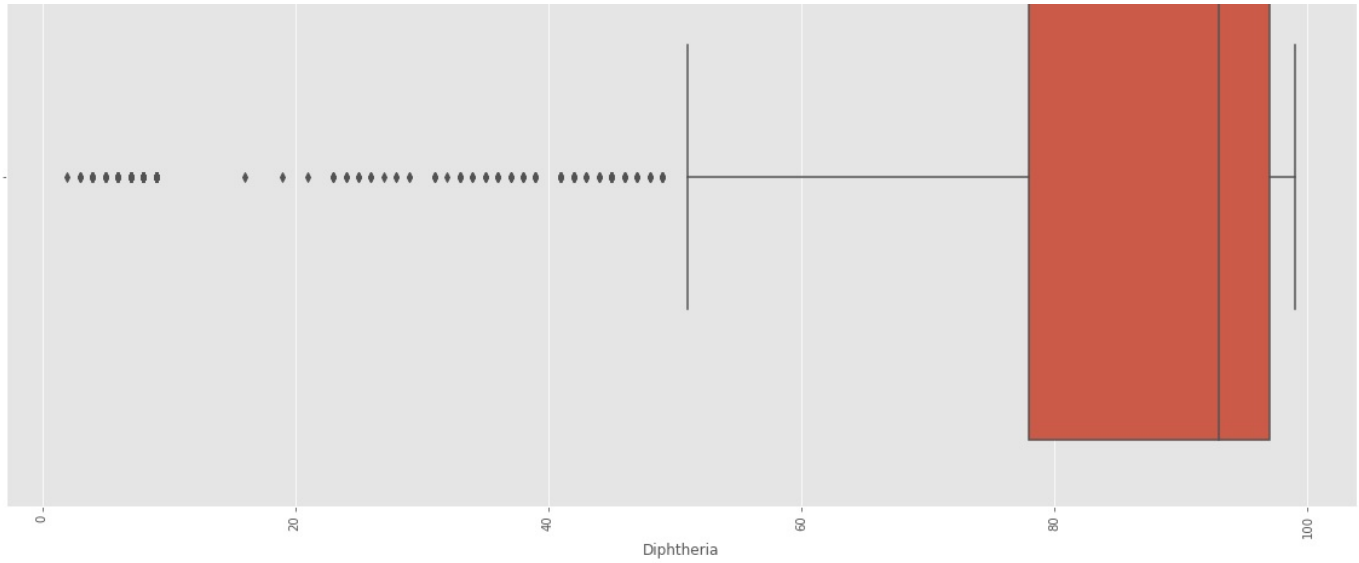

0
50000
100000
150000
200000
Measles

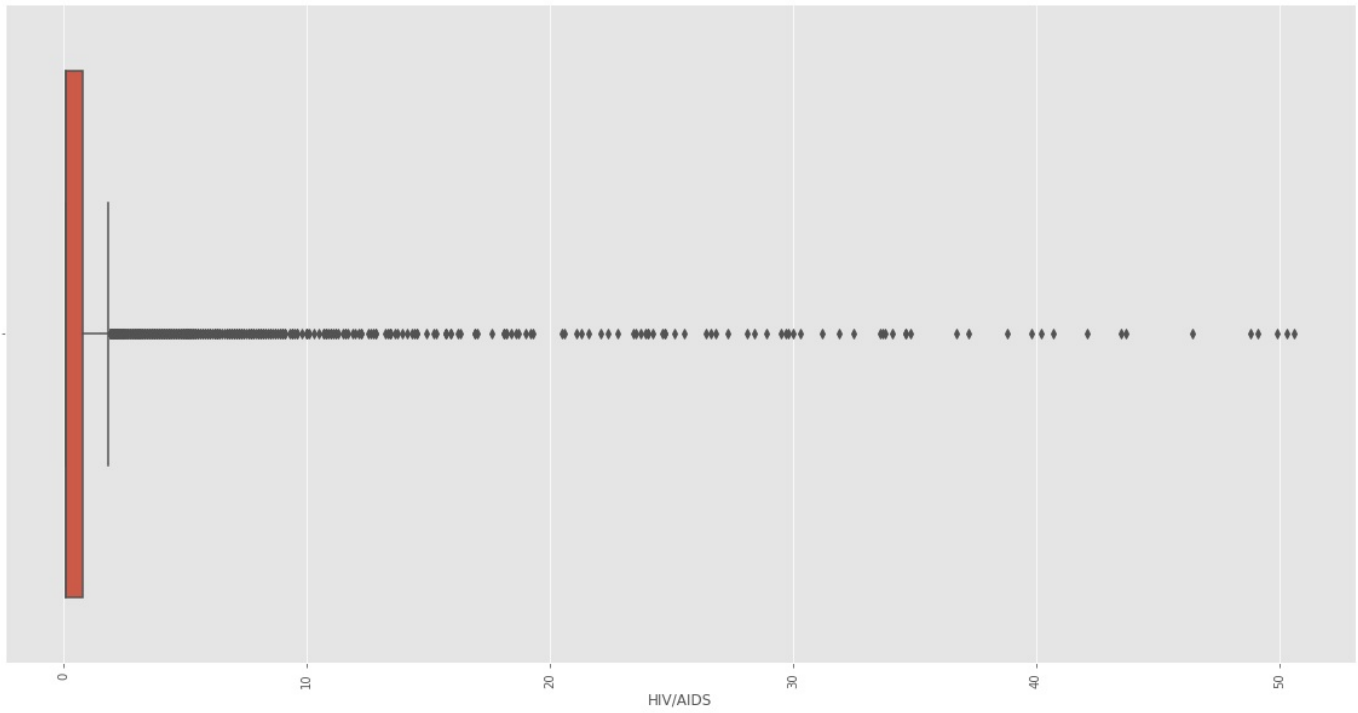


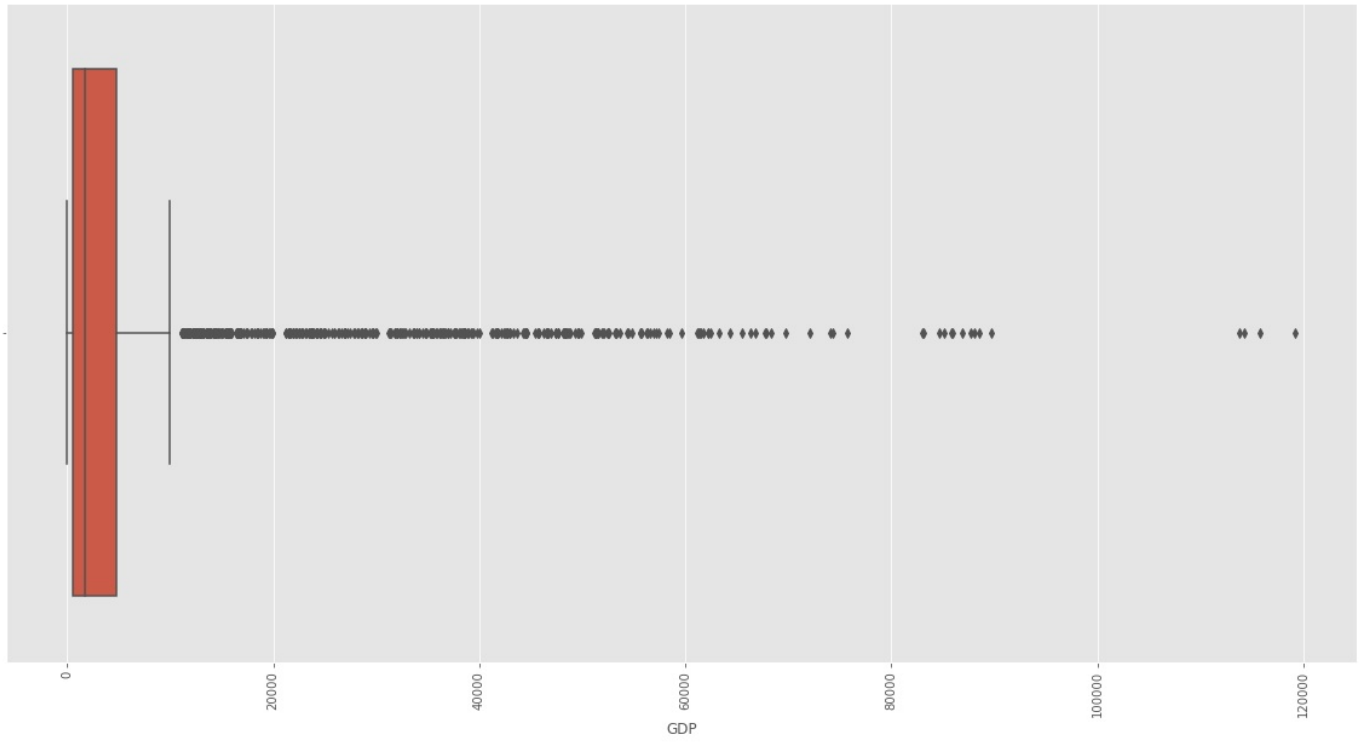

0
10
20
30
40
50
60
70
80
BMI

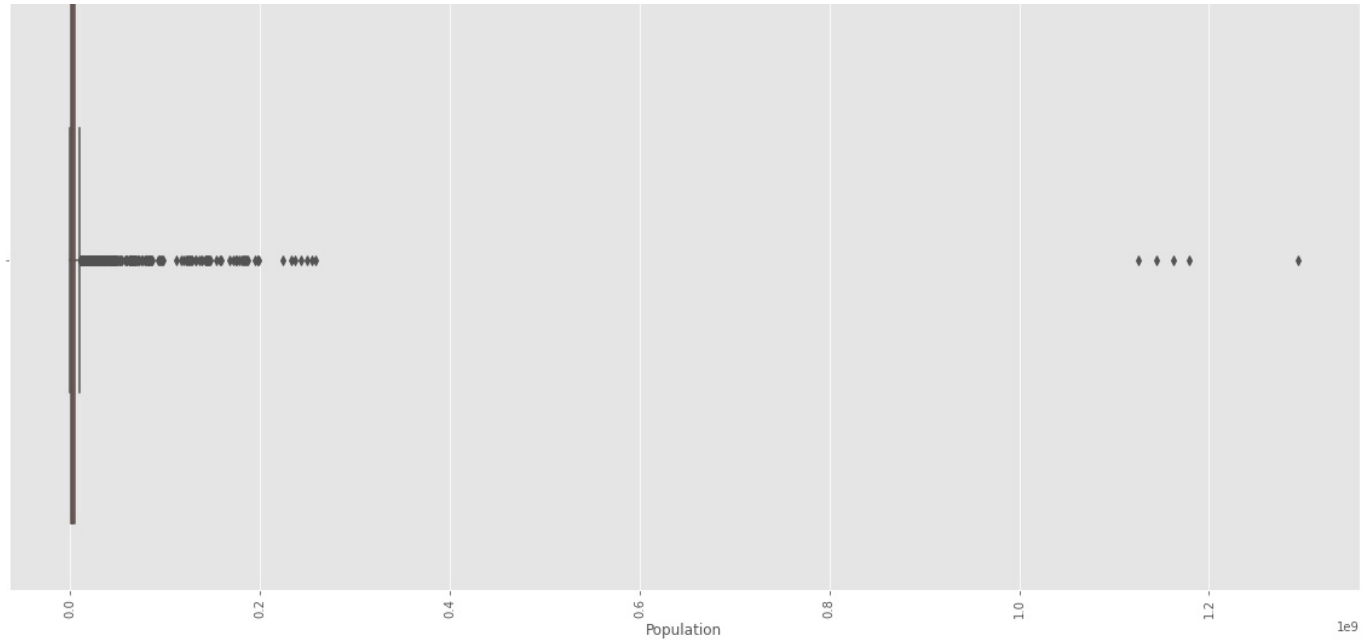

0
500
1000
1500
2000
2500
Under_5_Deaths

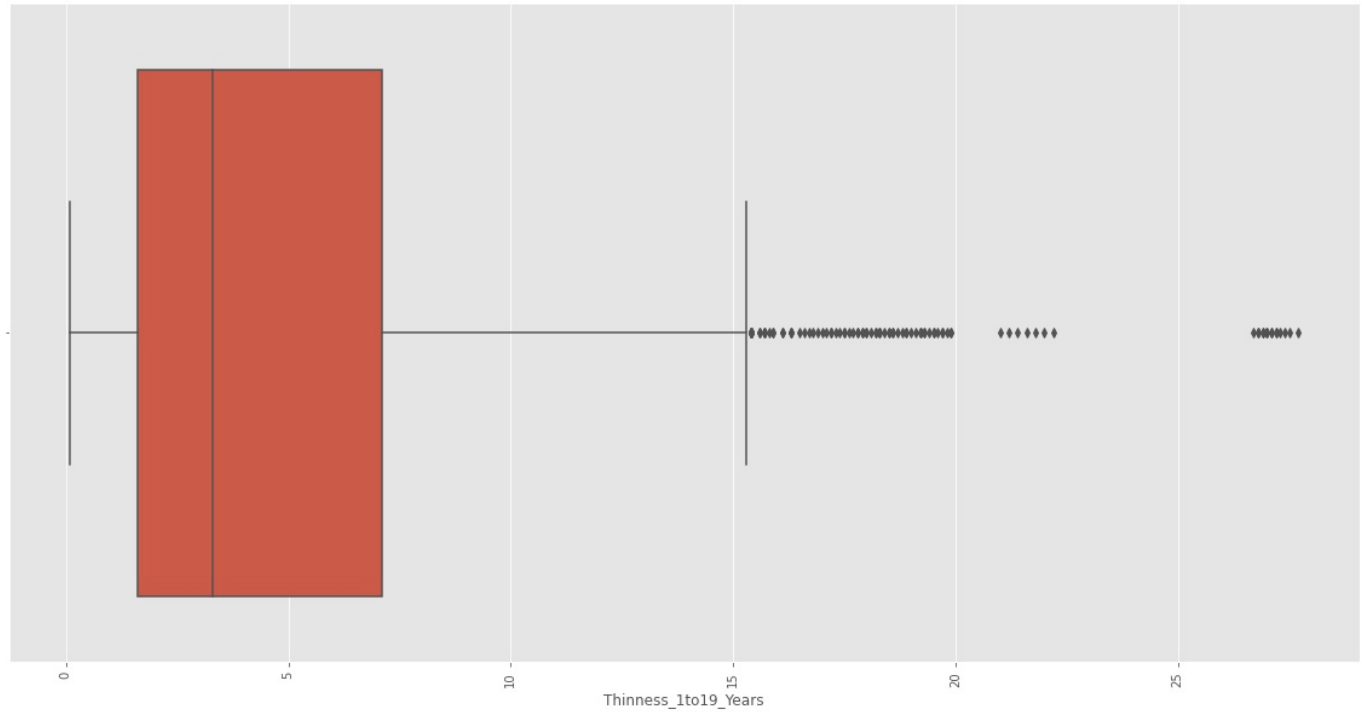

0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Total_Expenditure

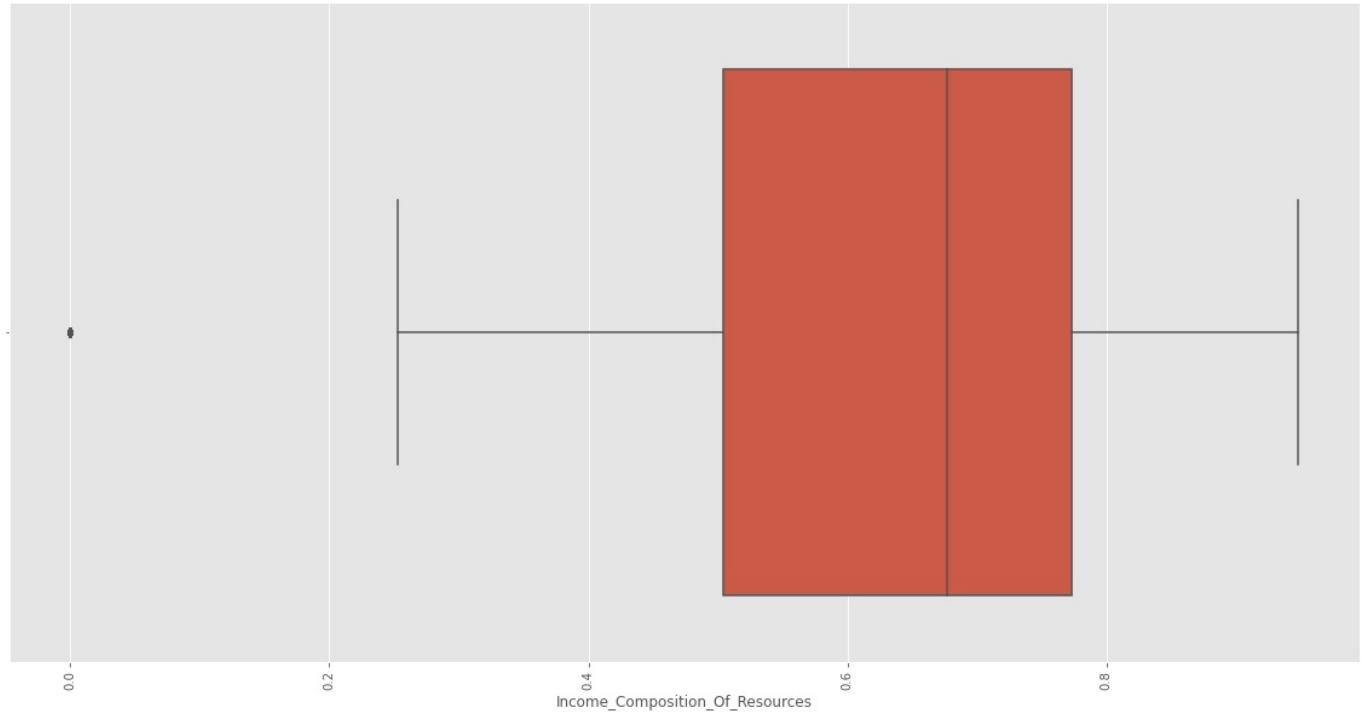

0
20
40
60
80
100
Diphtheria

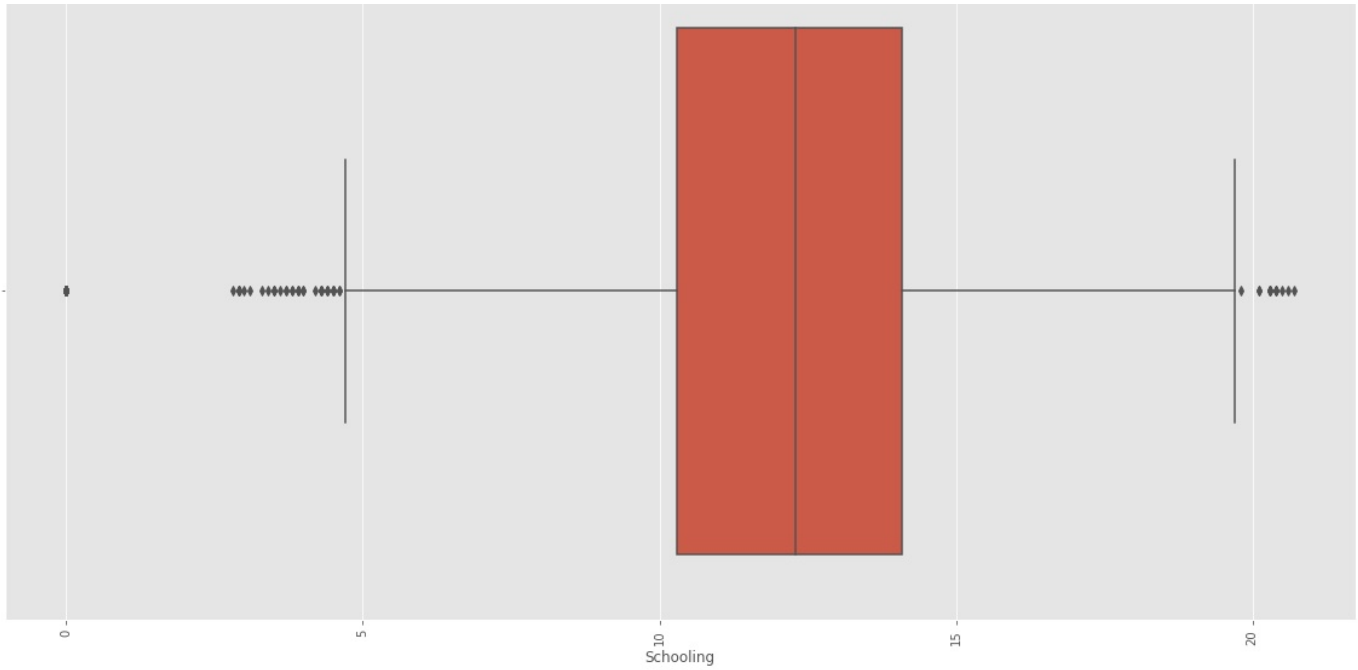

0
10
20
30
40
50
HIV/AIDS


0
20000
40000
60000
80000
100000
120000
GDP


0.0
0.2
0.4
0.6
0.8
1.0
1.2
Population
1e9


0
5
10
15
20
25
Thinness_1to19_Years

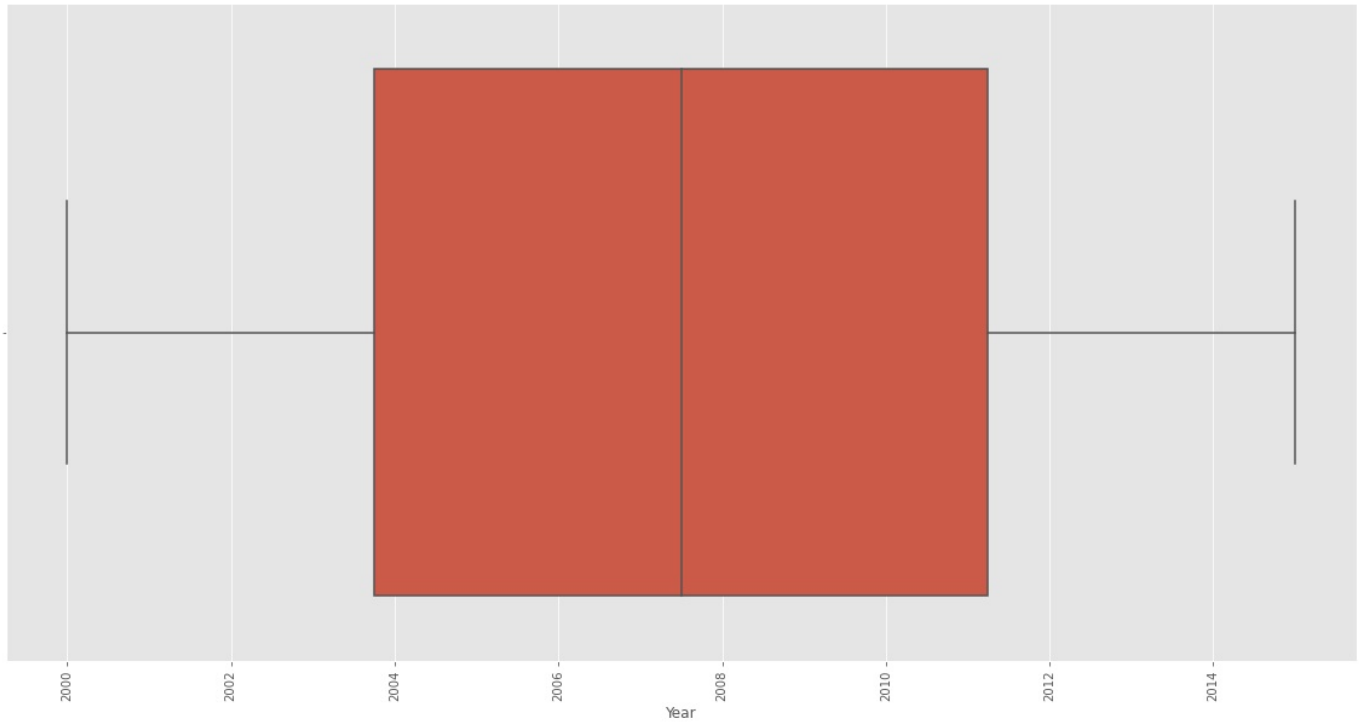

0.0
0.2
0.4
0.6
0.8
Income_Composition_Of_Resources

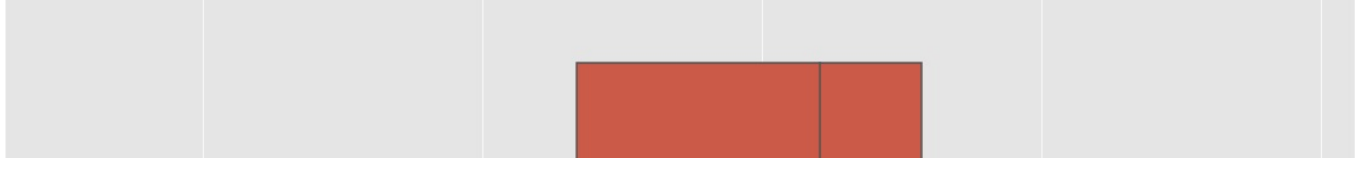


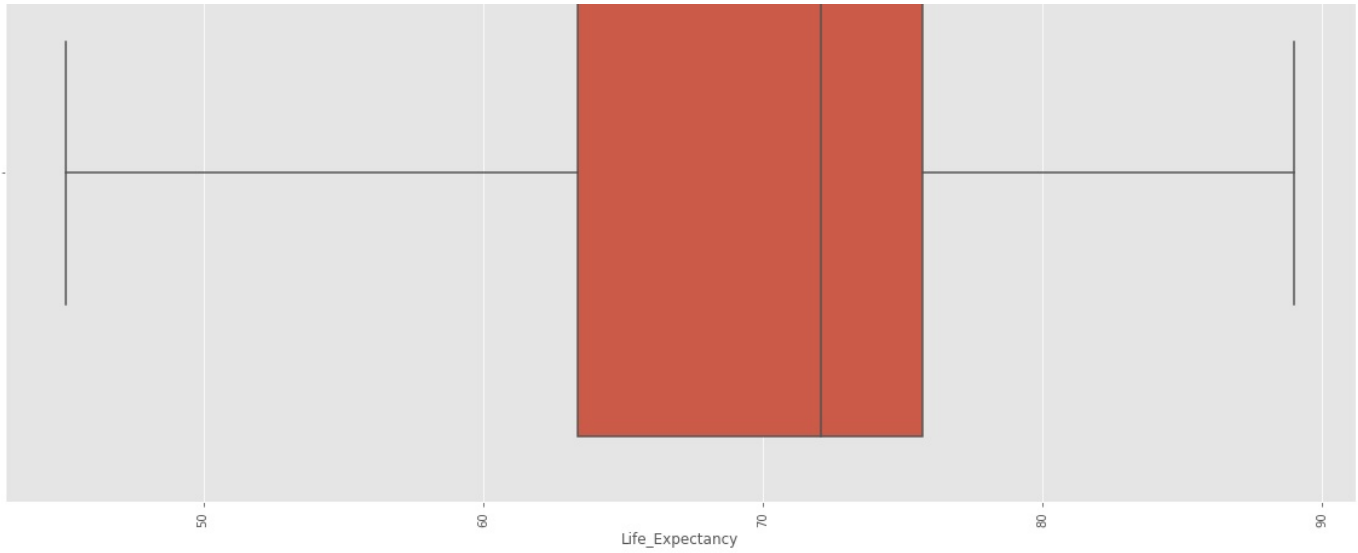
In [68]:
```
def outlier_removal_columnwise(data,col):
    for d in data[col]:
        q1=data[col].quantile(0.25)
        q3=data[col].quantile(0.75)
        iqr=q3-q1
        lower_limit=q1-1.5*iqr
        upper_limit=q3+1.5*iqr
        if (d > upper_limit or d < lower_limit):
            data[col]=data[col].replace(d,data[col].median())
    return data[col]
```

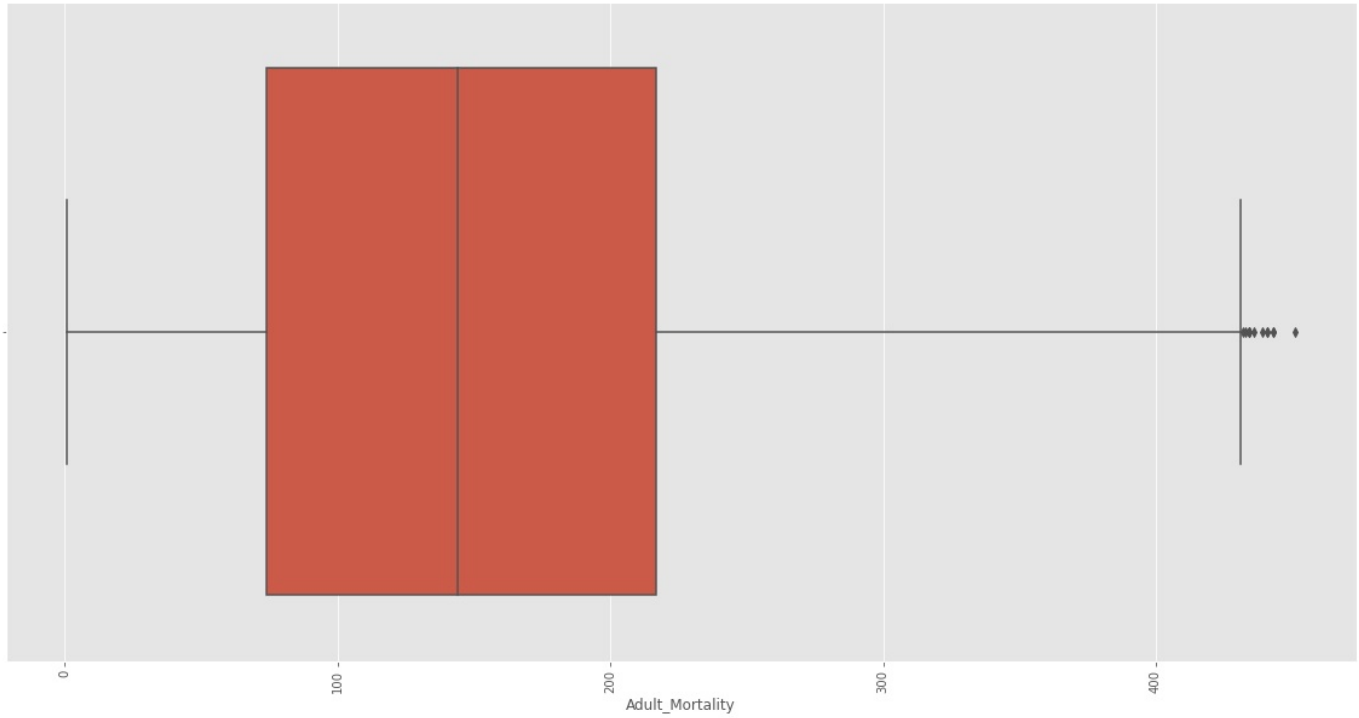
In [69]:
```
for c in numeric_cols:
    life_expectancy_data[c]=outlier_removal_columnwise(life_expectancy_data,c)
```

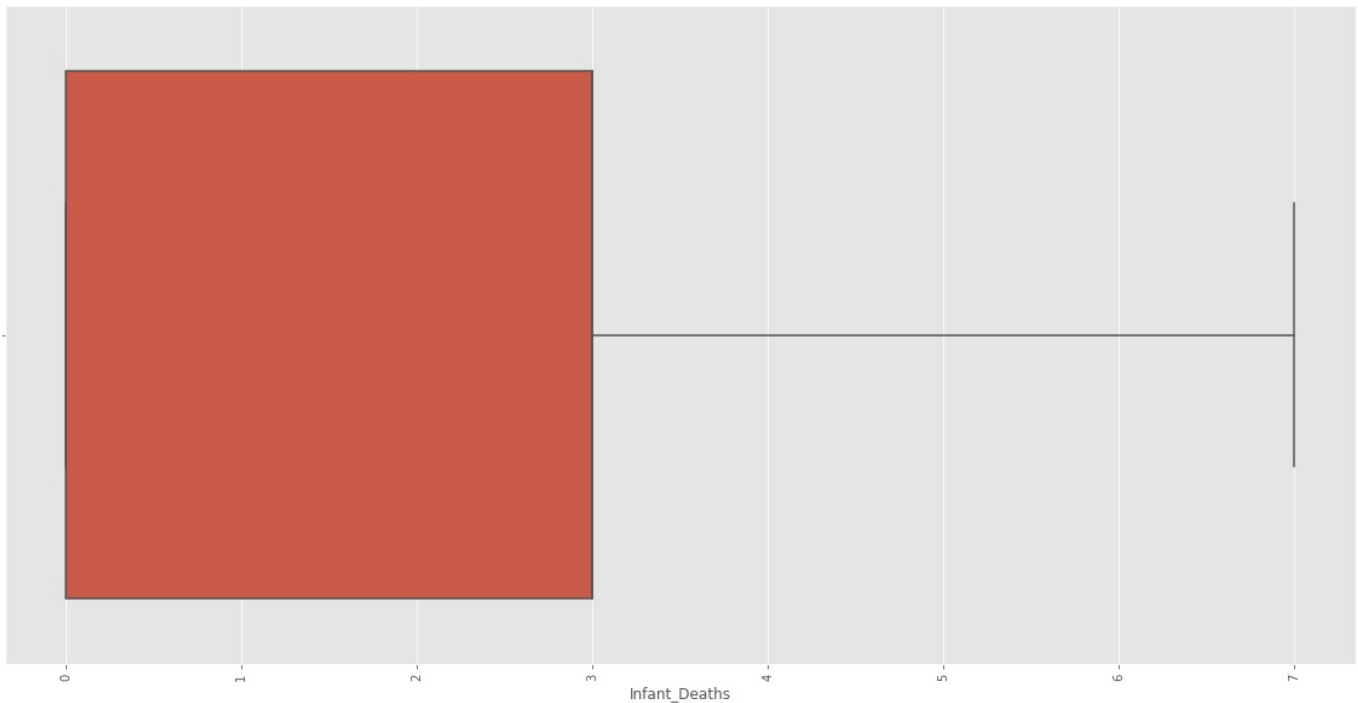
In [70]:
```
plot_univariate_graphs(life_expectancy_data,numeric_cols,'bo')
```

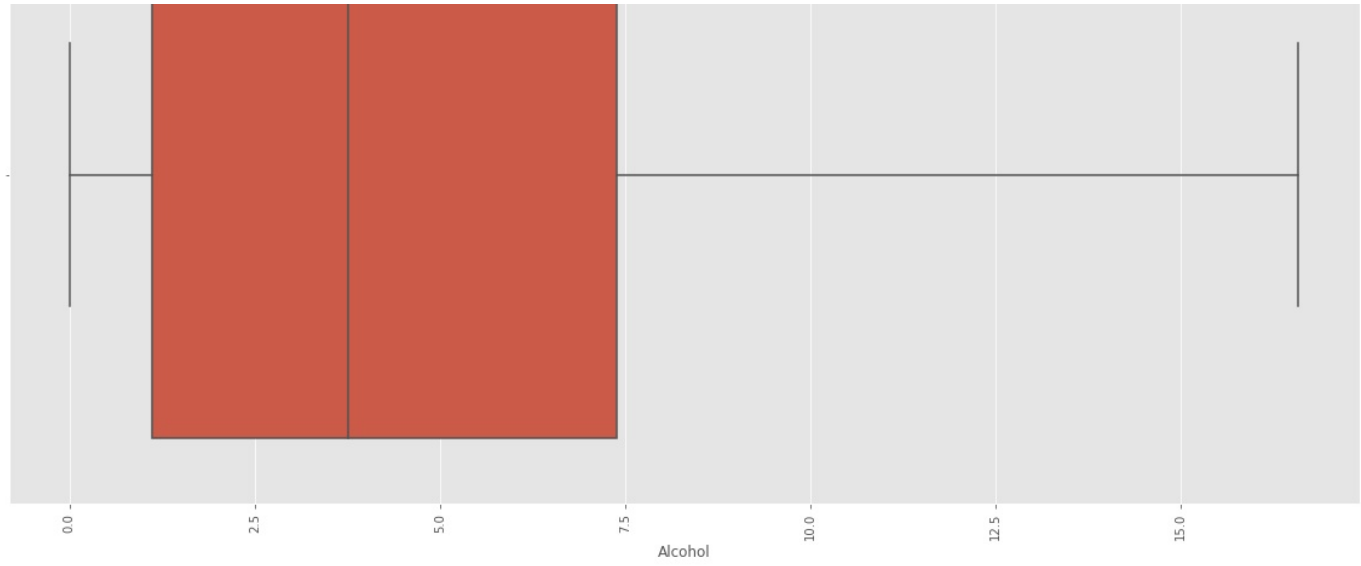


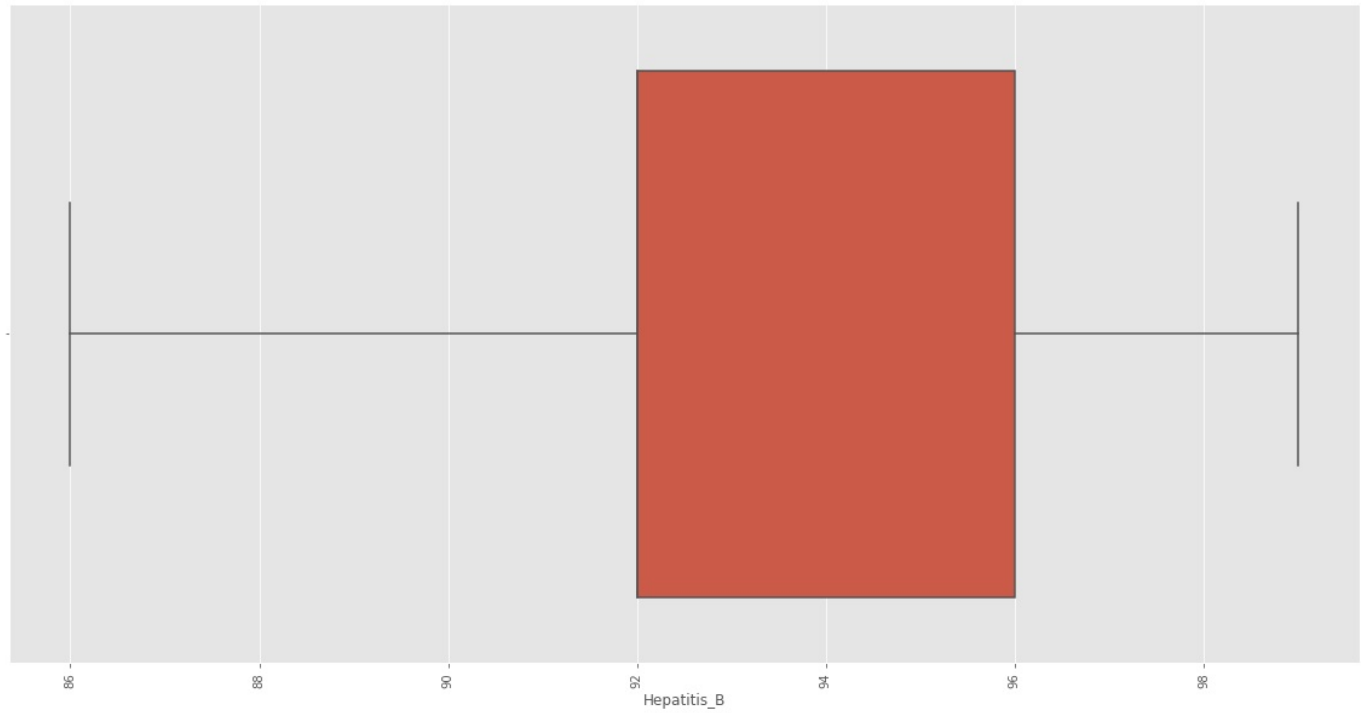


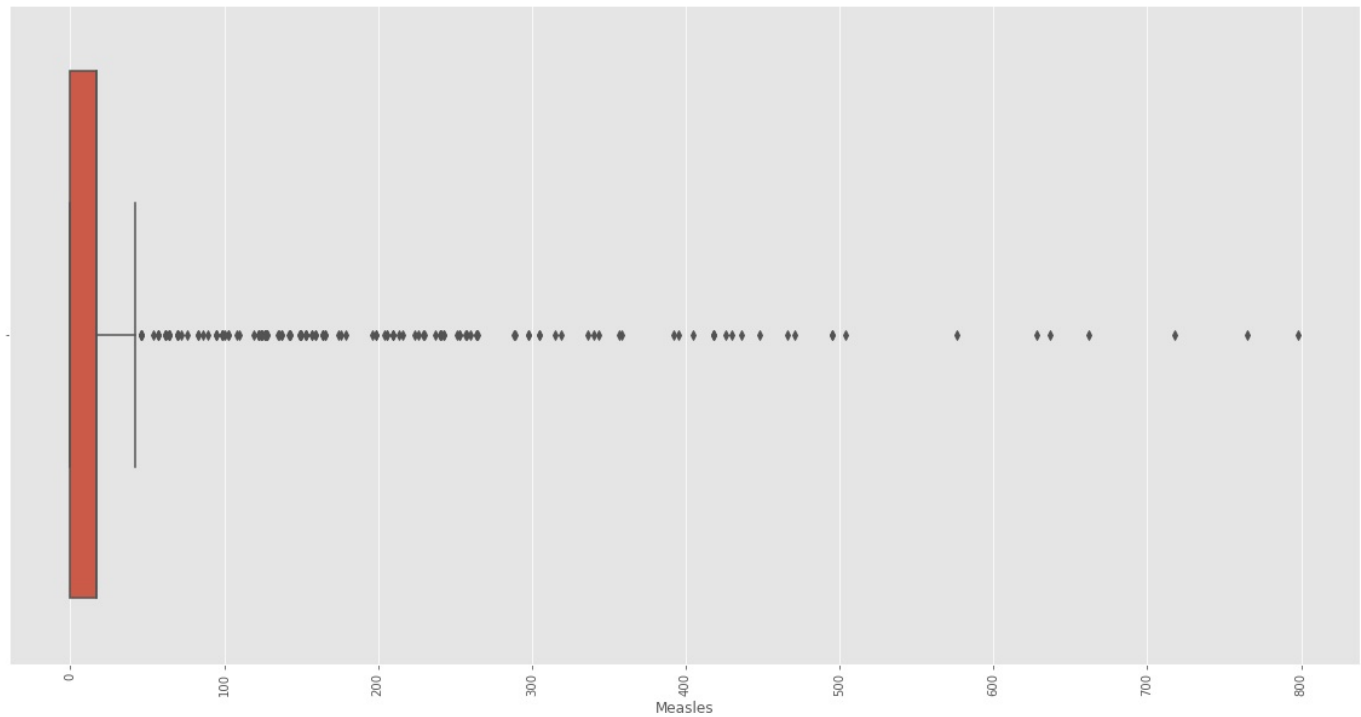

50
60
70
80
90
Life_Expectancy

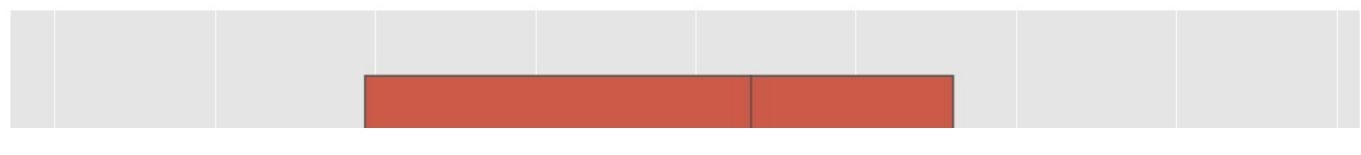

0
100
200
300
400
Adult_Mortality

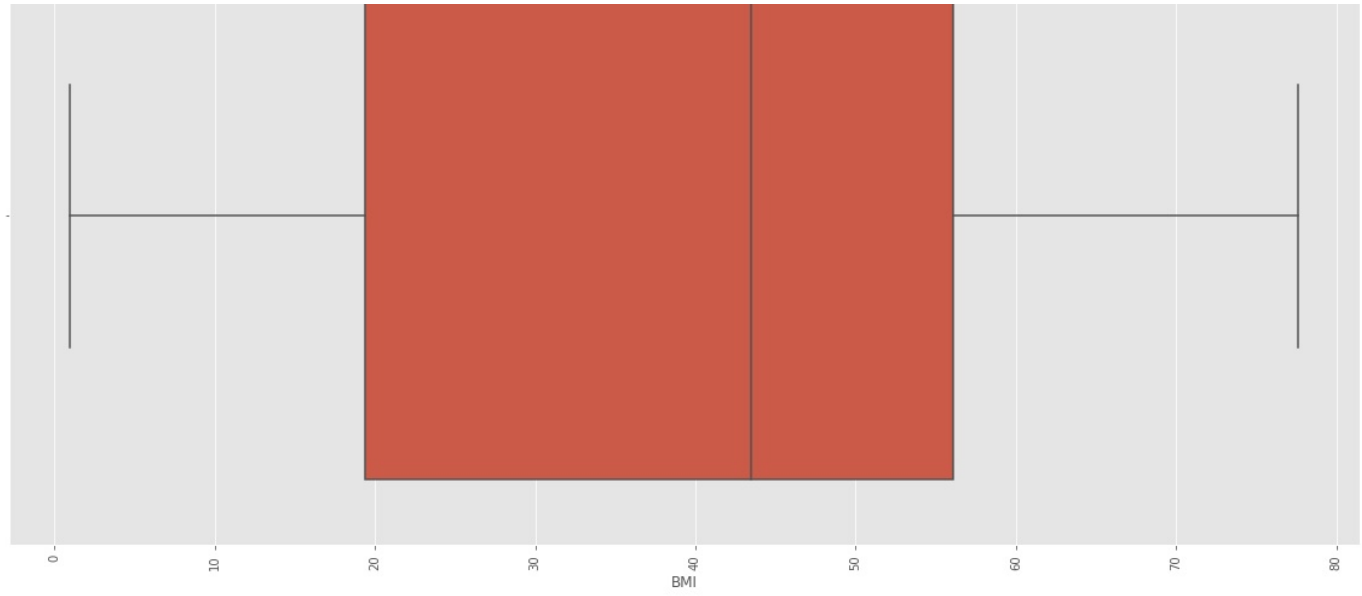

0
1
2
3
4
5
6
7
Infant_Deaths

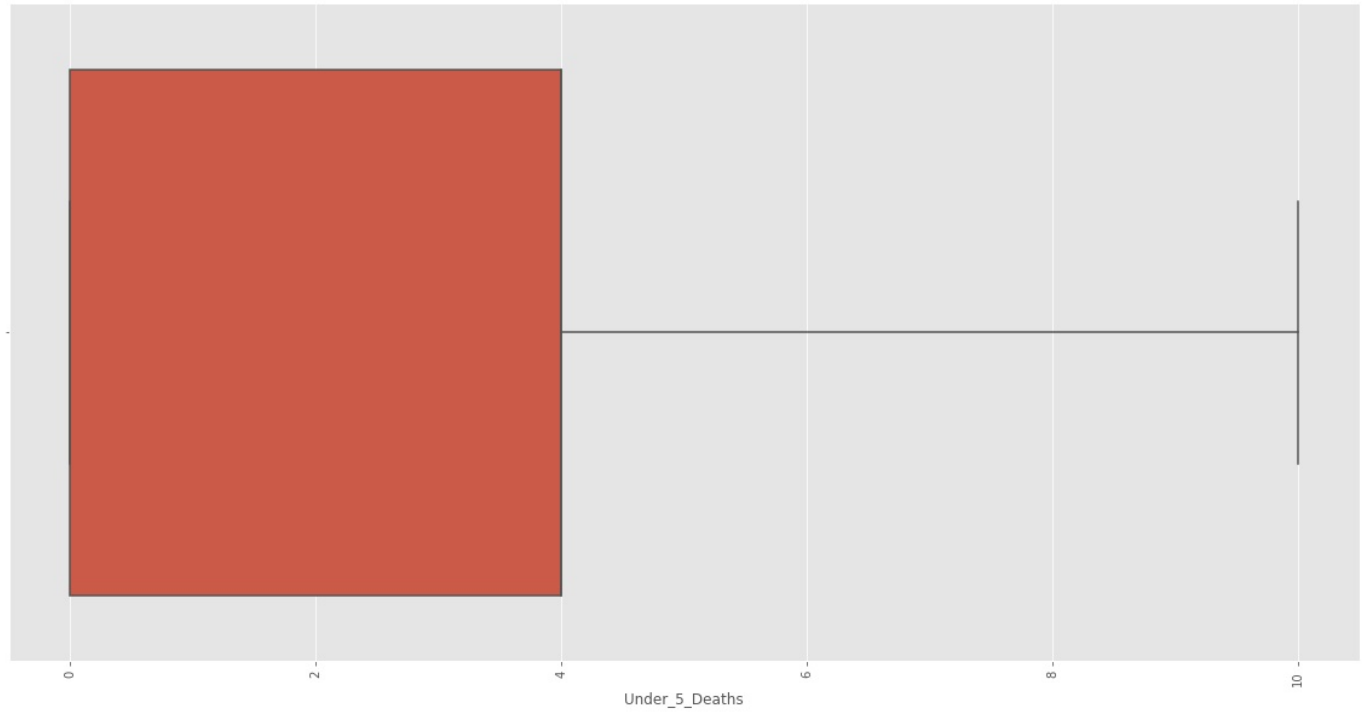

0.0
2.5
5.0
7.5
10.0
12.5
15.0
Alcohol

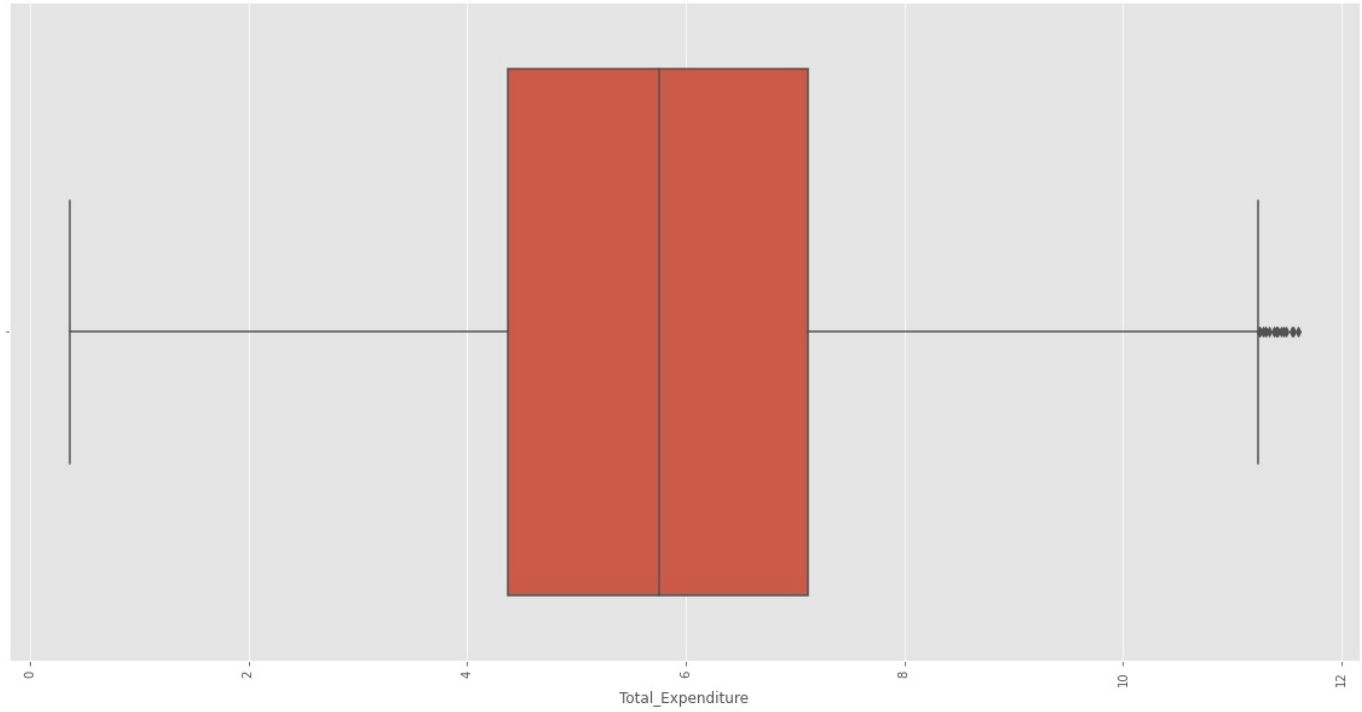

86
88
90
92
94
96
98
Hepatitis_B

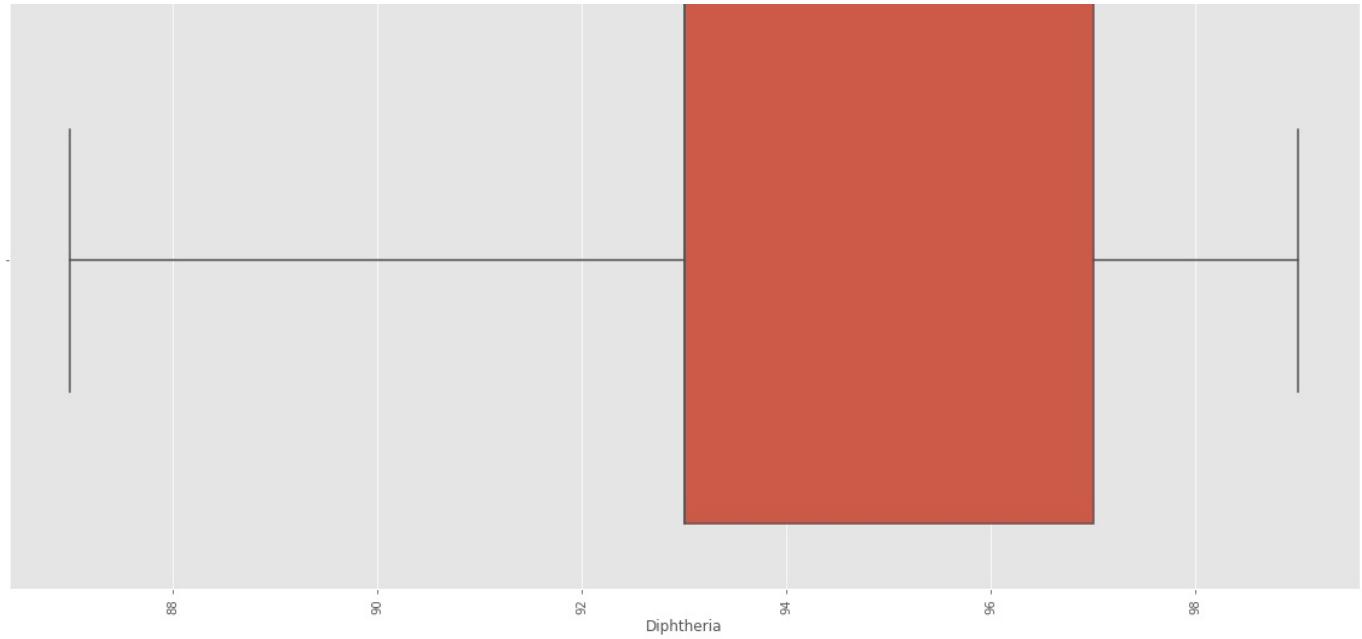

0
100
200
300
400
500
600
700
800
Measles

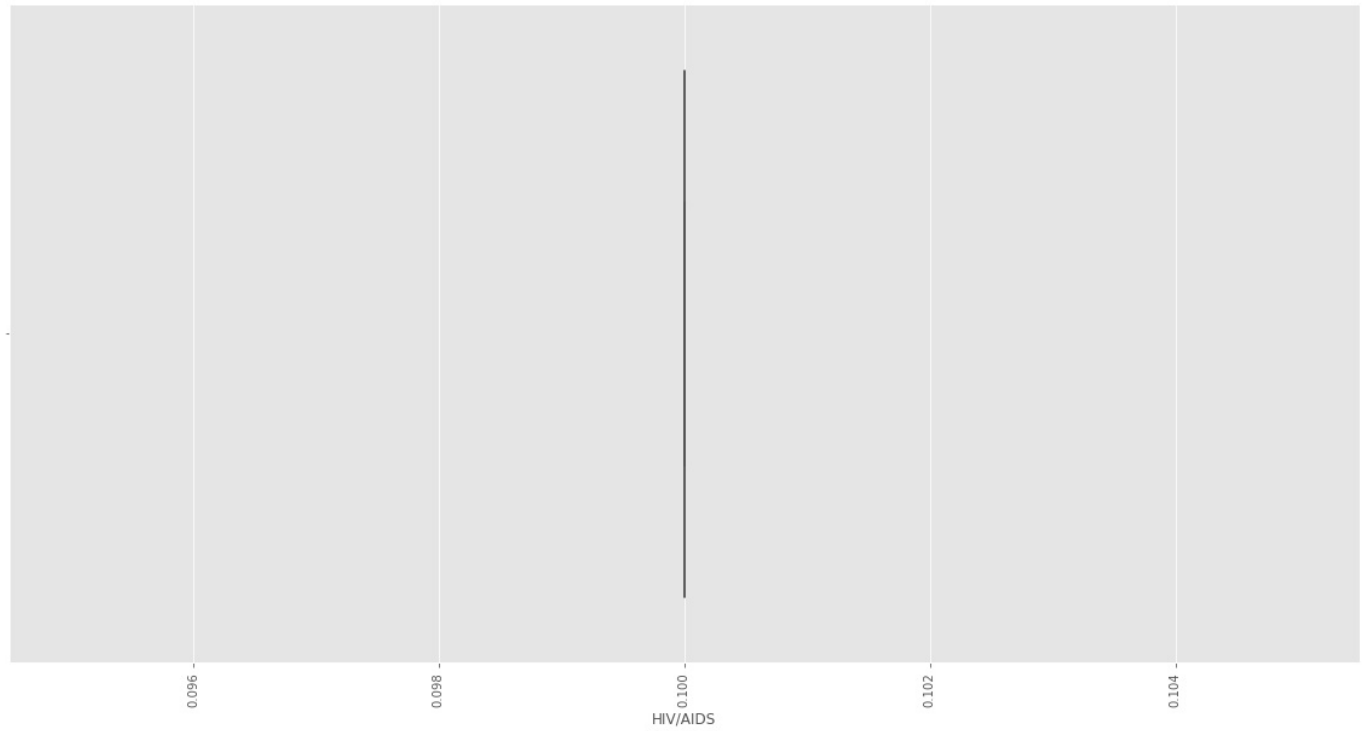


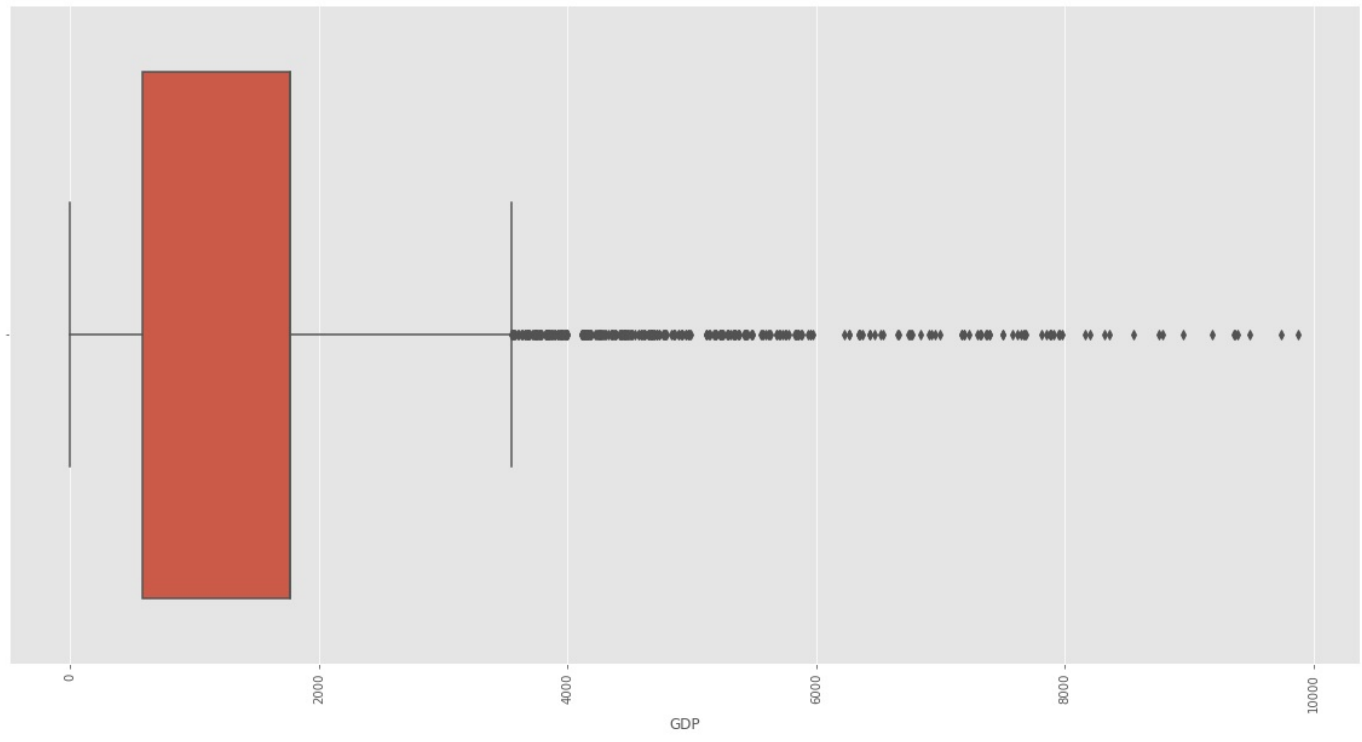

0
10
20
30
40
50
60
70
80
BMI

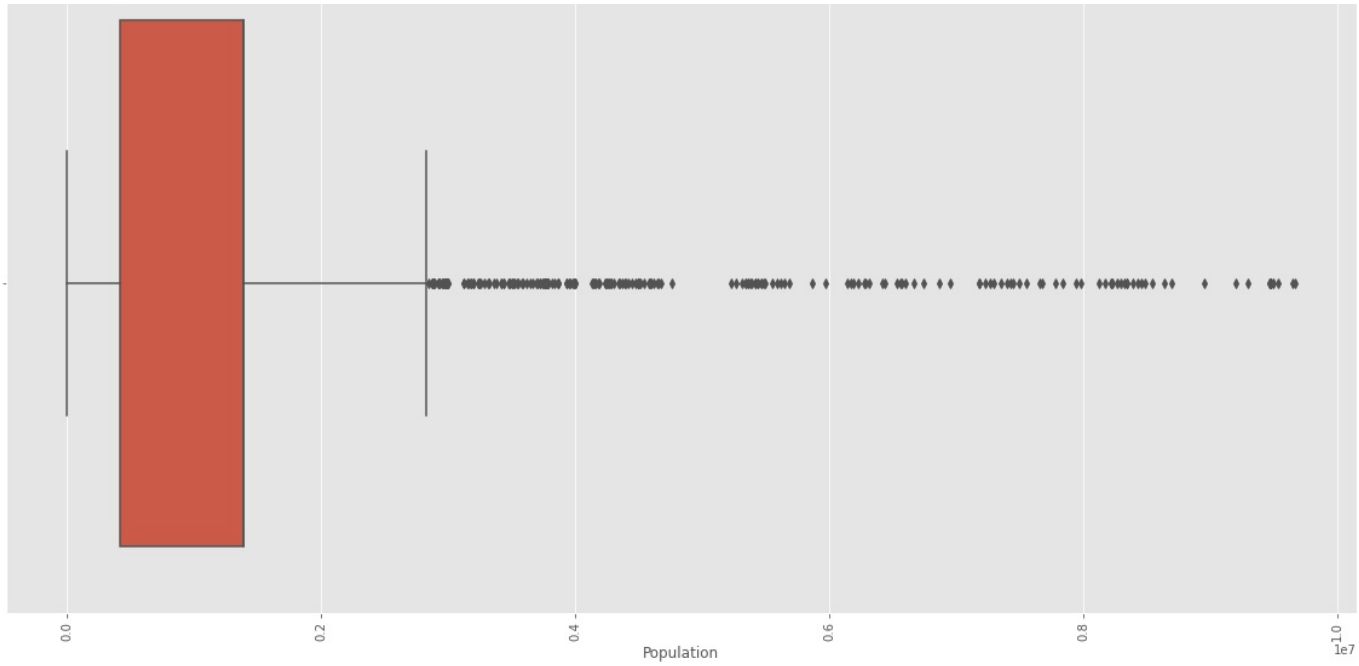

0
2
4
6
8
10
Under_5_Deaths

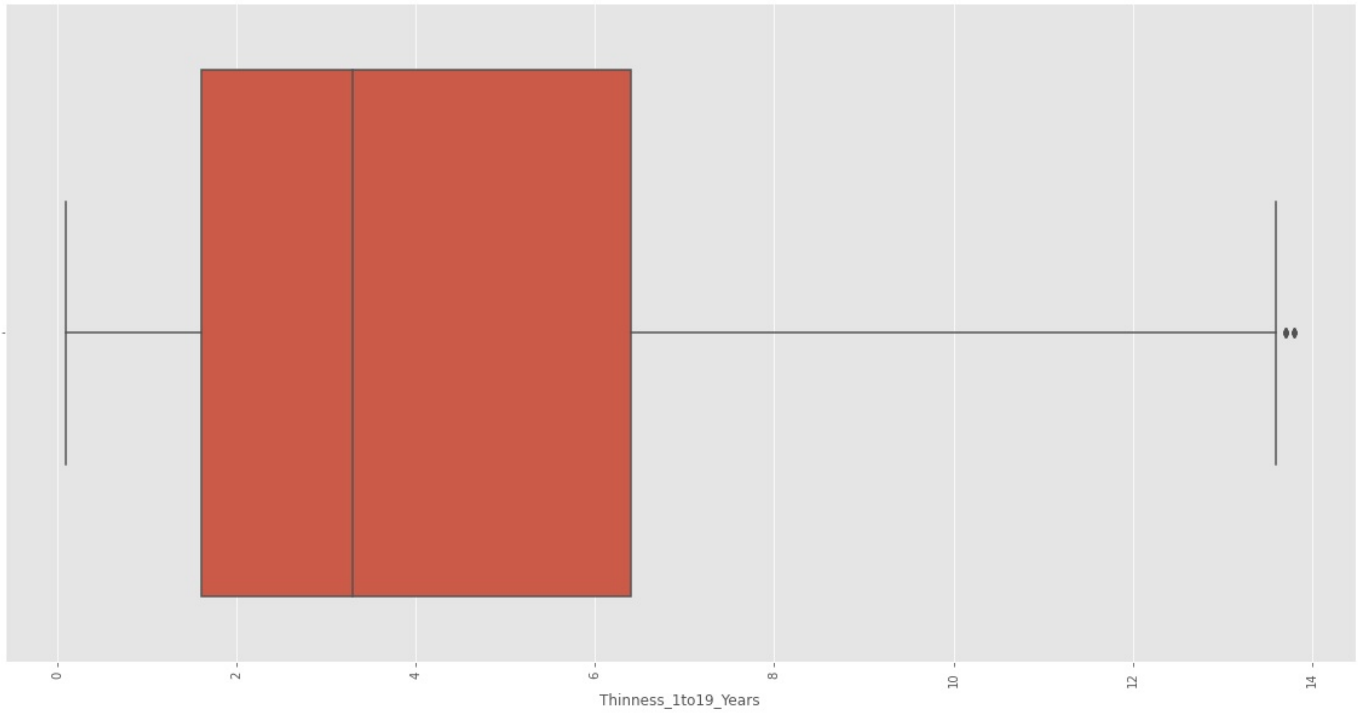

0
2
4
6
8
10
12
Total_Expenditure

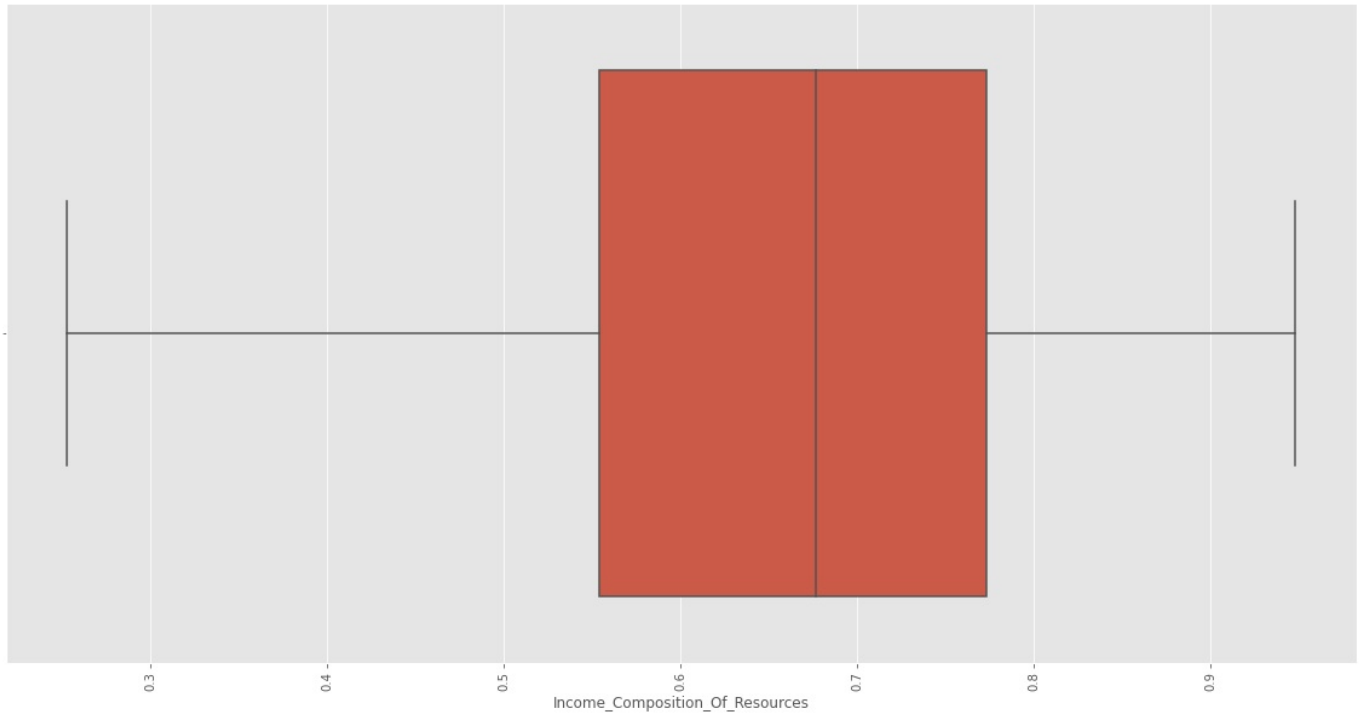

88
90
92
94
96
98
Diphtheria

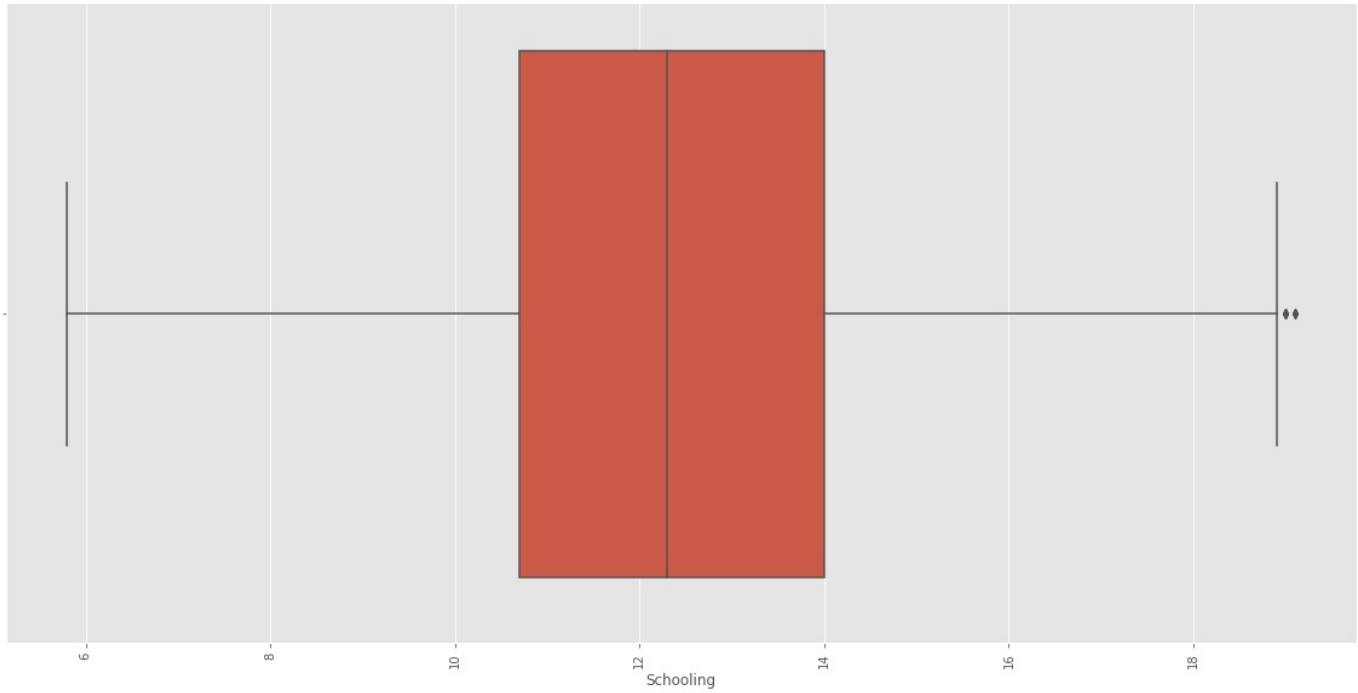

0.096
0.098
0.100
0.102
0.104
HIV/AIDS


0
2000
4000
6000
8000
10000
GDP


0.0
0.2
0.4
0.6
0.8
1.0
1e7
Population


0
2
4
6
8
10
12
14
Thinness_1to19_Years


0.3
0.4
0.5
0.6
0.7
0.8
0.9
Income_Composition_Of_Resources



In [71]:
```
life_expectancy_data.shape
```

Out[71]: (2928, 20)

In [72]:
```
life_expectancy_data
```

Out[72]:

	Country	Year	Status	Life_Expectancy	Adult_Mortality	Infant_Deaths	Alcohol	Hepatitis_B	Measles	BMI	Under_5_Deaths	Total
0	Afghanistan	2015	Developing	65.0	263.0	3	0.01	92.0	17	19.1	4	
1	Afghanistan	2014	Developing	59.9	271.0	3	0.01	92.0	17	18.6	4	
2	Afghanistan	2013	Developing	59.9	268.0	3	0.01	92.0	430	18.1	4	
3	Afghanistan	2012	Developing	59.5	272.0	3	0.01	92.0	17	17.6	4	
4	Afghanistan	2011	Developing	59.2	275.0	3	0.01	92.0	17	17.2	4	
...	...	...	...	...	...	...	...	...	...	...	...	
2933	Zimbabwe	2004	Developing	72.1	144.0	3	4.36	92.0	31	27.1	4	
2934	Zimbabwe	2003	Developing	72.1	144.0	3	4.06	92.0	17	26.7	4	
2935	Zimbabwe	2002	Developing	72.1	73.0	3	4.43	92.0	17	26.3	4	
2936	Zimbabwe	2001	Developing	45.3	144.0	3	1.72	92.0	17	25.9	4	
2937	Zimbabwe	2000	Developing	46.0	144.0	3	1.68	92.0	17	25.5	4	

2928 rows × 20 columns

AFTER DROPPING COLUMNS AGAIN SPLIT THE DATA

In [73]:
```
numeric_cols,factor_cols=split_cols(life_expectancy_data)
```

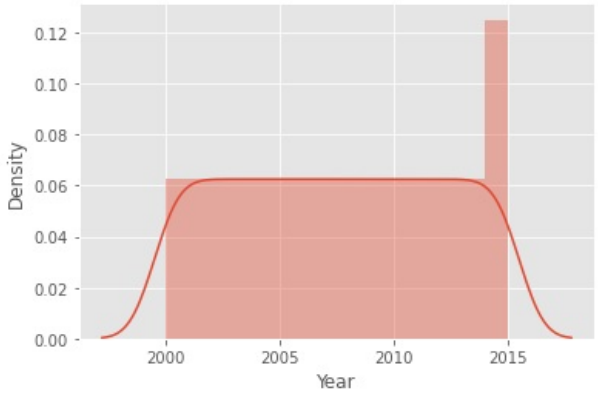
DATA TRANSFORMATION

i. HANDLING SKEWNESS IN DATA

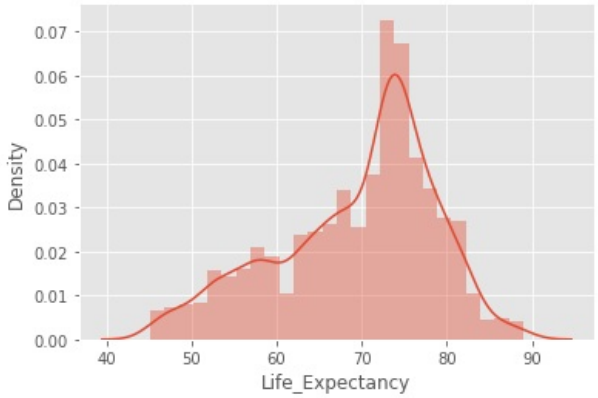
In [74]:
```
for col in numeric_cols:
```

```
    print(col)
    print(life_expectancy_data[col].skew())
    plt.figure()
    sns.distplot(life_expectancy_data[col])
    plt.show()
```

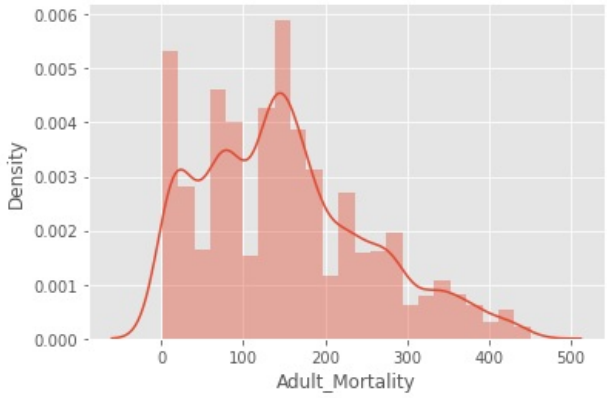
```
Year
0.0
```



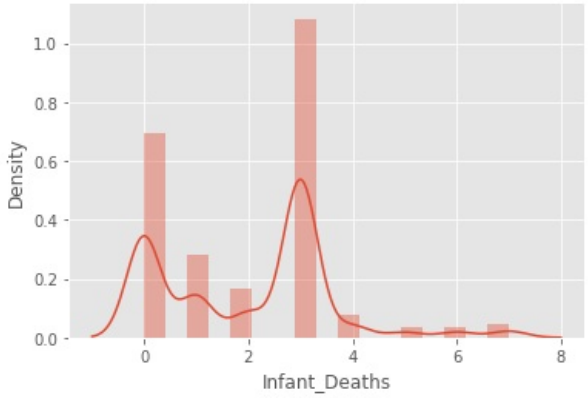

```
Life_Expectancy
-0.6013760092700127
```



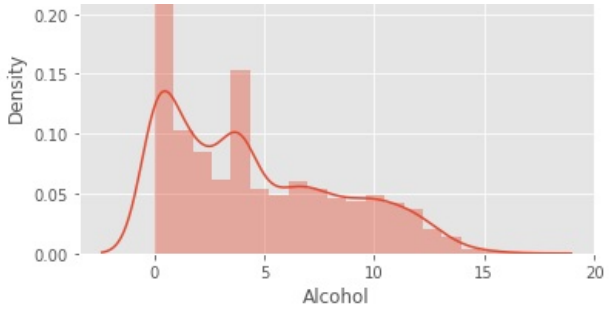

```
Adult_Mortality
0.6290789414075955
```



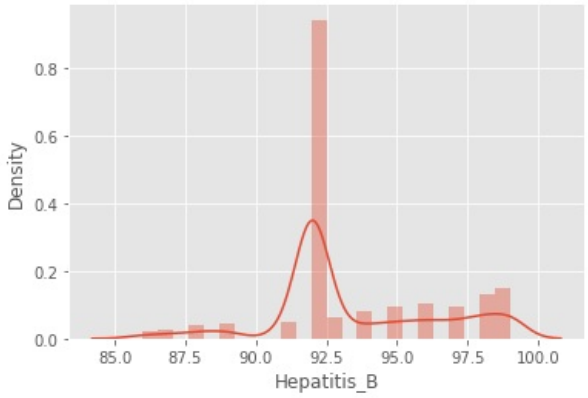

```
Infant_Deaths
0.41297304274113633
```



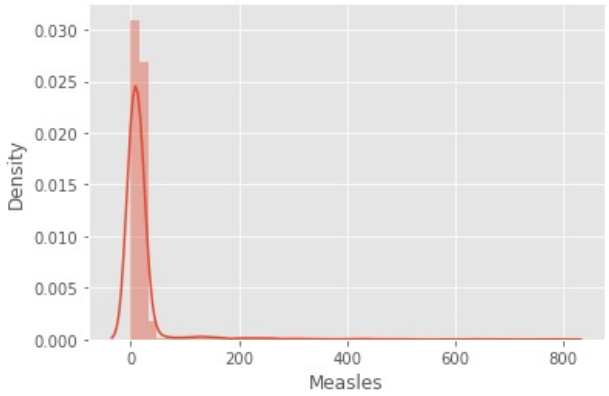

```
Alcohol
0.6325245252894824
```



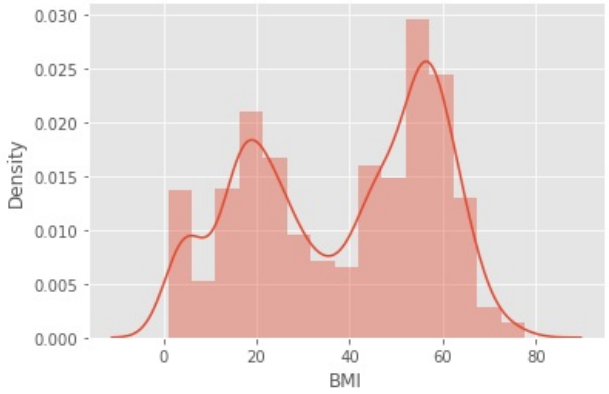
Hepatitis_B
0.3170740090254279



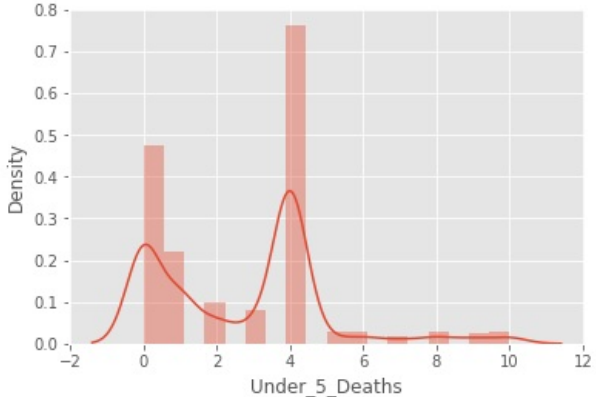
Measles
7.175438138361072



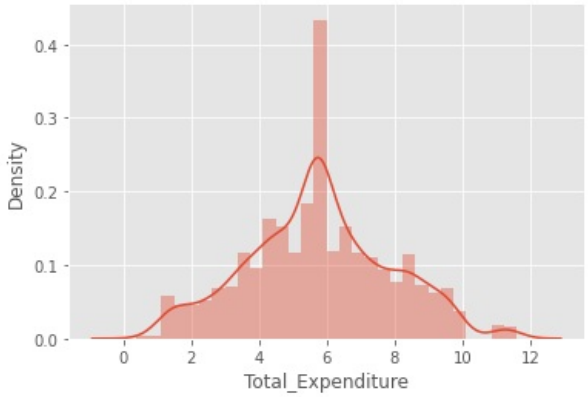
BMI
-0.24005789699158706



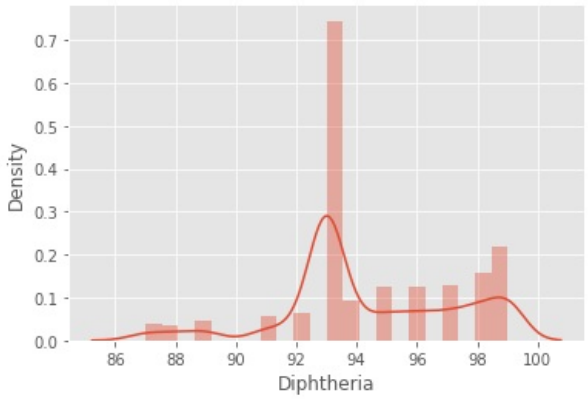
Under_5_Deaths
0.680924636138983



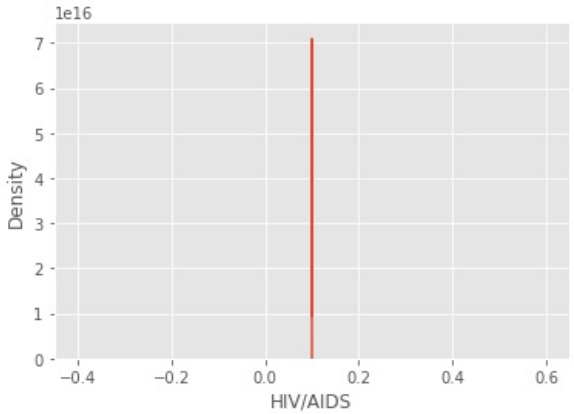
Total_Expenditure
0.09054307904784355



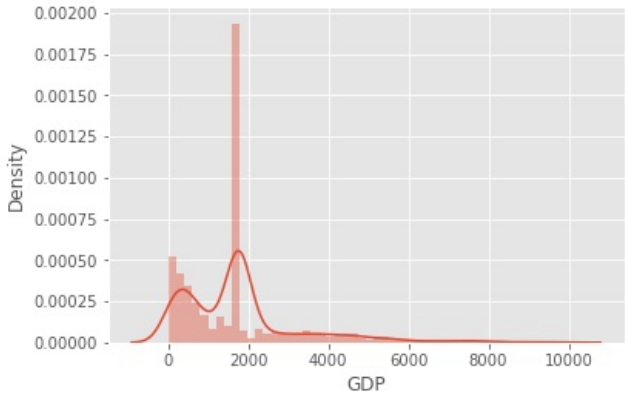
Diphtheria
-0.12917355694856375



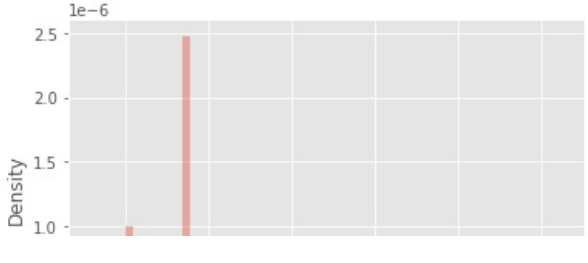
HIV/AIDS
0

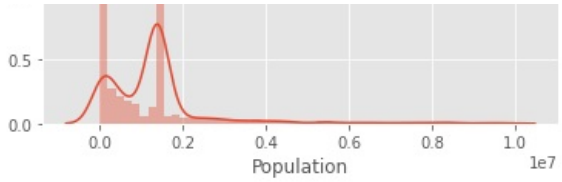


GDP
1.8276718797534568

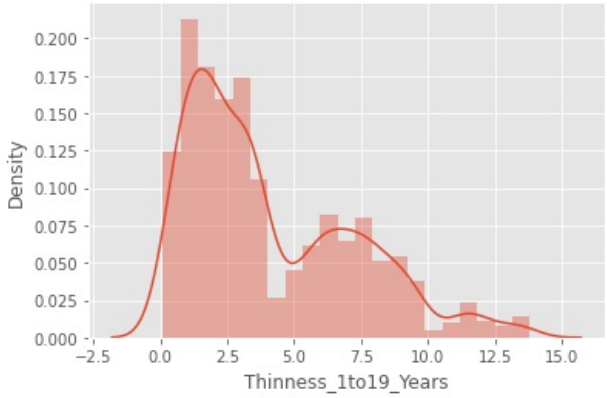


Population
3.012756855521179

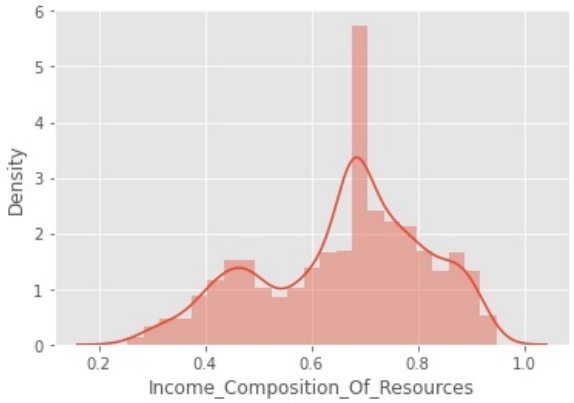




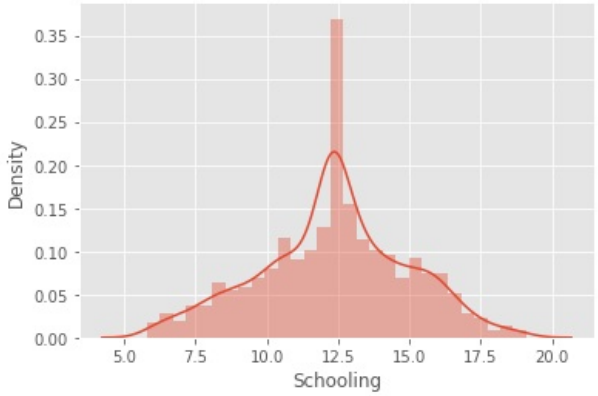
Thinness_1to19_Years
0.8899945398181692



Income_Composition_Of_Resources
-0.41567732776137184



Schooling
-0.10717384788793256



In [75]:
```
life_expectancy_data[numeric_cols].skew()
```

Out[75]:
```
Year                              0.000000
Life_Expectancy                  -0.601376
Adult_Mortality                   0.629079
Infant_Deaths                     0.412973
Alcohol                           0.632525
Hepatitis_B                       0.317074
Measles                           7.175438
BMI                              -0.240058
Under_5_Deaths                    0.680925
Total_Expenditure                 0.090543
Diphtheria                       -0.129174
HIV/AIDS                          0.000000
GDP                               1.827672
Population                        3.012757
Thinness_1to19_Years              0.889995
Income_Composition_Of_Resources  -0.415677
Schooling                        -0.107174
dtype: float64
```

In [76]:
```
skew_cols = life_expectancy_data[numeric_cols].apply(lambda x: skew(x)).sort_values(ascending=False)
skew_cols = abs(skew_cols)
high_skew = skew_cols[skew_cols > 0.5]
print(high_skew)
skew_index = high_skew.index
# Normalize skewed feature
life_expectancy_data[skew_index] = np.sqrt(life_expectancy_data[skew_index])
```

```
Measles                 7.171762
Population              3.011213
GDP                     1.826735
HIV/AIDS                1.000000
Thinness_1to19_Years    0.889539
Under_5_Deaths          0.680576
Alcohol                 0.632200
Adult_Mortality         0.628757
Life_Expectancy         0.601068
dtype: float64
```

In [77]:
```
life_expectancy_data[numeric_cols].skew()
```

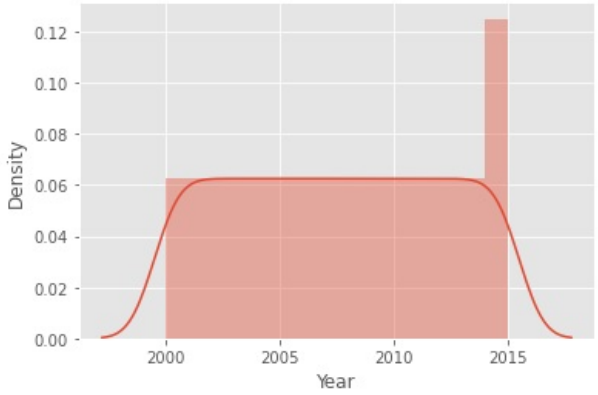
Out[77]:
```
Year                              0.000000
Life_Expectancy                  -0.740892
Adult_Mortality                  -0.196334
Infant_Deaths                     0.412973
Alcohol                          -0.145969
Hepatitis_B                       0.317074
Measles                           2.621747
BMI                              -0.240058
Under_5_Deaths                   -0.396111
Total_Expenditure                 0.090543
Diphtheria                       -0.129174
HIV/AIDS                          0.000000
GDP                               0.367358
Population                        0.495676
Thinness_1to19_Years              0.261144
Income_Composition_Of_Resources  -0.415677
```

```
Schooling                        -0.107174
dtype: float64
```

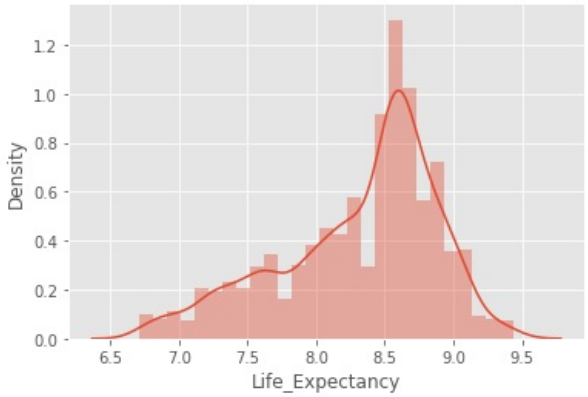
In [78]:

```
for col in numeric_cols:
    print(col)
    print(life_expectancy_data[col].skew())
    plt.figure()
    sns.distplot(life_expectancy_data[col])
    plt.show()
```

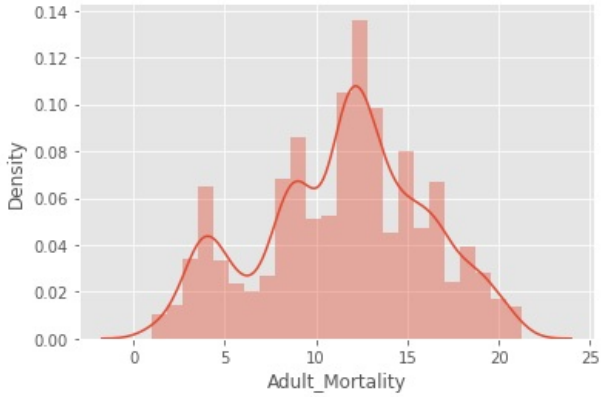
```
Year
0.0
```



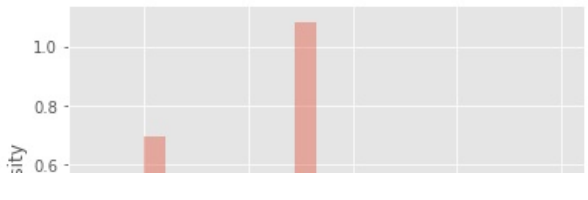

```
Life_Expectancy
-0.7408922904102437
```

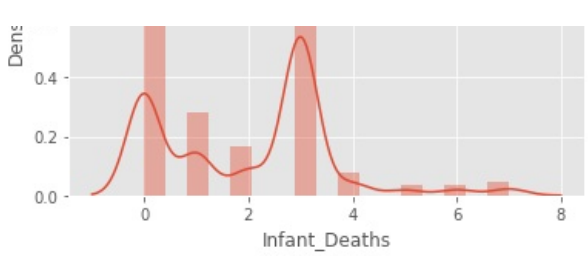



```
Adult_Mortality
-0.19633376401693398
```

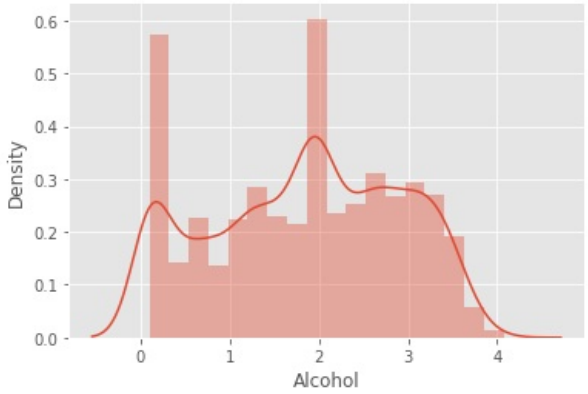



```
Infant_Deaths
0.41297304274113633
```

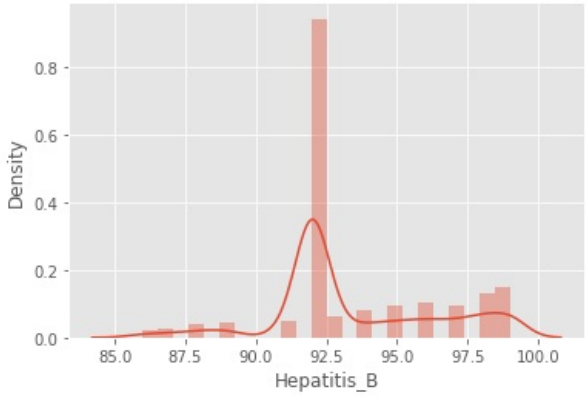




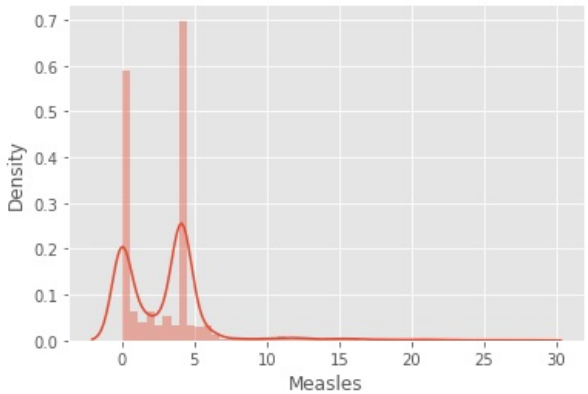
Alcohol
-0.14596876187732108



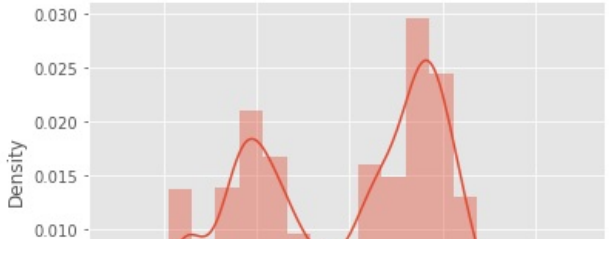
Hepatitis_B
0.3170740090254279

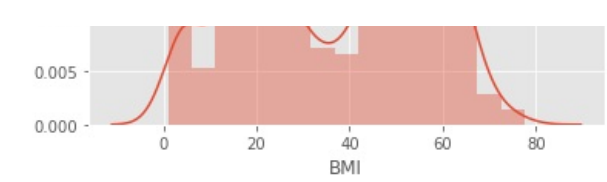


Measles
2.6217473908685283

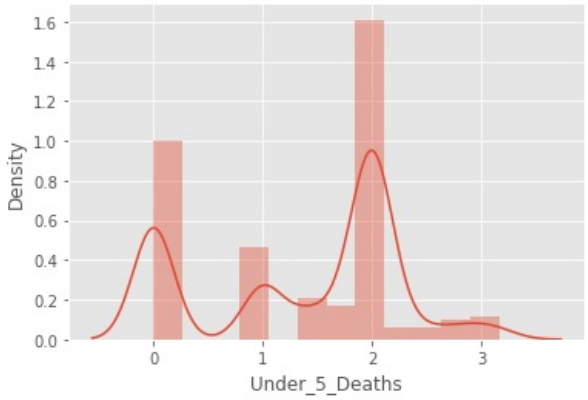


BMI
-0.24005789699158706

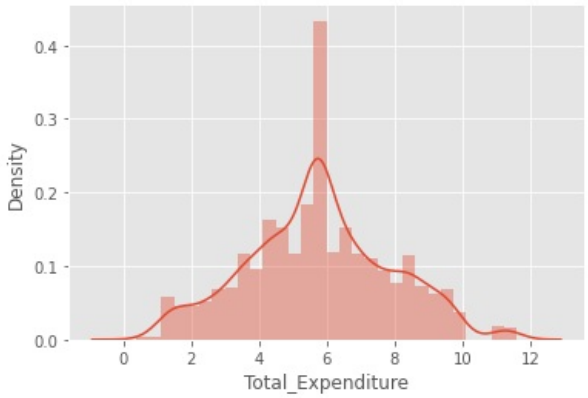




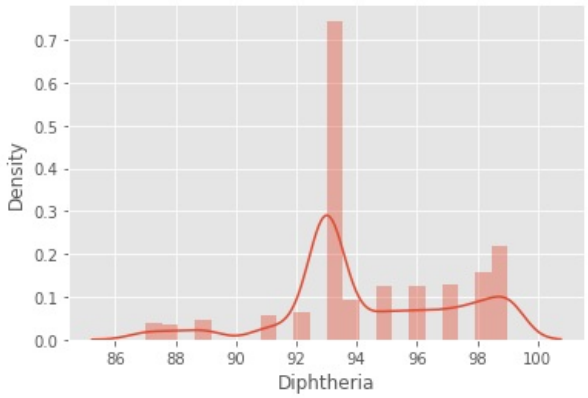
Under_5_Deaths
-0.39611059519960173



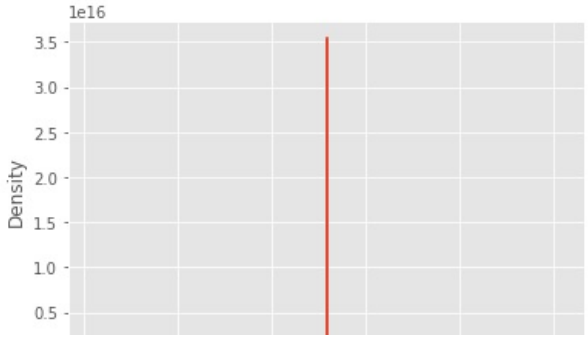
Total_Expenditure
0.09054307904784355

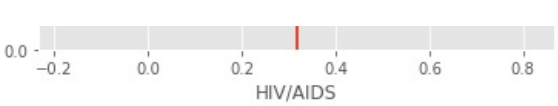


Diphtheria
-0.12917355694856375

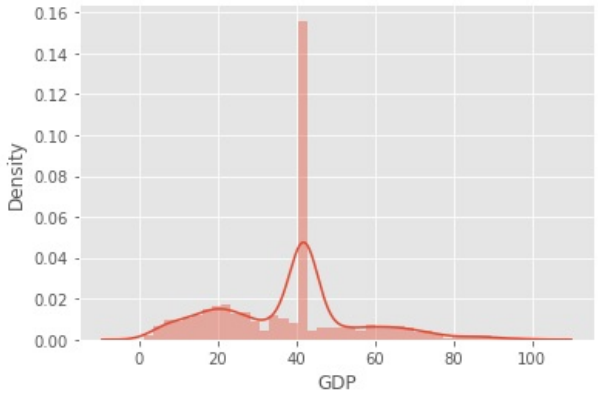


HIV/AIDS
0

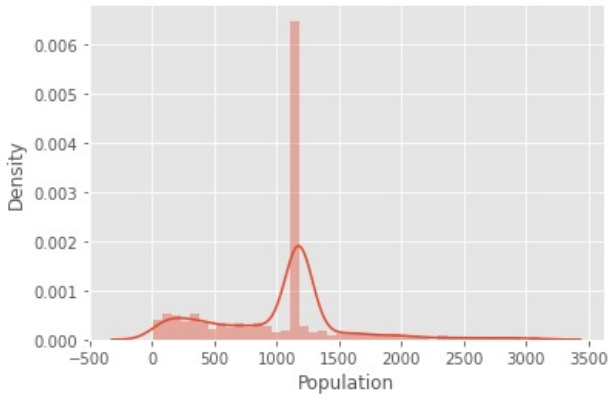




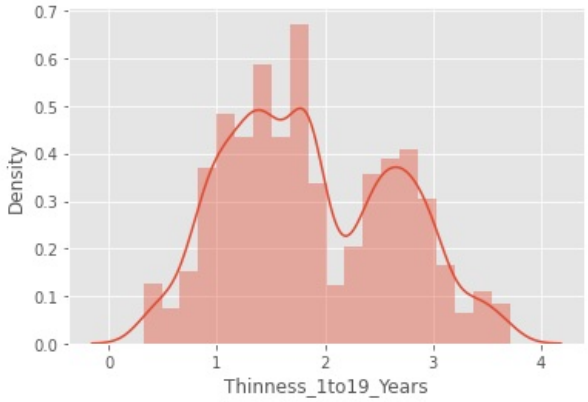
GDP
0.3673584640850442



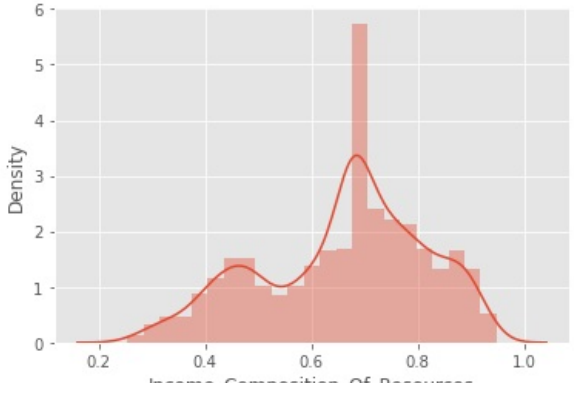
Population
0.4956763342584033



Thinness_1to19_Years
0.2611436061962901

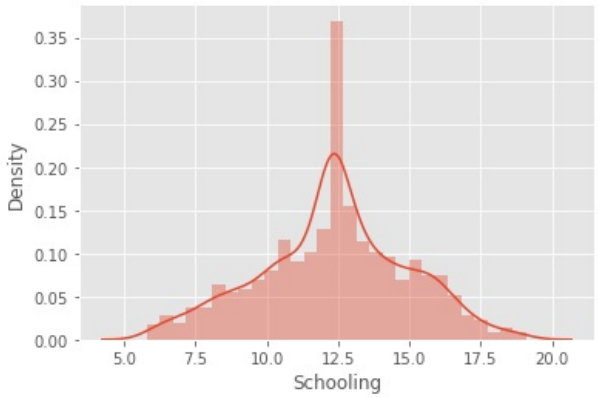


Income_Composition_Of_Resources
-0.41567732776137184



Income_Composition_Of_Resources

```
Schooling
-0.10717384788793256
```



CONVERTING DATA TO DUMMY VARIABLES (ONE HOT ENCODING) AND LABEL ENCODING

Here country and status have no such order so we will apply one hot encoding to these columns`

In [79]:
```
factor_cols
```

Out[79]:
```
array(['Country', 'Status', 'Region'], dtype=object)
```

In [80]:
```
life_expectancy_data.Region.nunique()+ life_expectancy_data.Status.nunique()
```

Out[80]:
```
9
```

In [81]:
```
life_expectancy_dummies_cat=pd.get_dummies(life_expectancy_data[factor_cols],drop_first=True)
```

In [82]:
```
len(life_expectancy_dummies_cat.columns)
```

Out[82]:
```
189
```

In [83]:
```
len(numeric_cols)
```

Out[83]:
```
17
```

In [84]:
```
life_expectancy_data = pd.concat([life_expectancy_data[numeric_cols],life_expectancy_dummies_cat],axis =1)
life_expectancy_data.head()
```

Out[84]:

	Year	Life_Expectancy	Adult_Mortality	Infant_Deaths	Alcohol	Hepatitis_B	Measles	BMI	Under_5_Deaths	Total_Expenditure	...	Country
0	2015	8.062258	16.217275	3	0.1	92.0	4.123106	19.1	2.0	8.16	...	
1	2014	7.739509	16.462078	3	0.1	92.0	4.123106	18.6	2.0	8.18	...	
2	2013	7.739509	16.370706	3	0.1	92.0	20.736441	18.1	2.0	8.13	...	
3	2012	7.713624	16.492423	3	0.1	92.0	4.123106	17.6	2.0	8.52	...	
4	2011	7.694154	16.583124	3	0.1	92.0	4.123106	17.2	2.0	7.87	...	

5 rows × 206 columns

In [85]:
```
life_expectancy_data.shape
```

Out[85]: (2928, 206)

TAKE BACKUP OF ORIGINAL DATA

In [86]:
```
data = life_expectancy_data.copy()
```

In [87]:
```
data = data.sample(frac=1)
```

In [88]:
```
data.reset_index(drop=True,inplace = True)
```

In [89]:
```
data.dropna(inplace=True)
```

In [90]:
```
data
```

Out[90]:

	Year	Life_Expectancy	Adult_Mortality	Infant_Deaths	Alcohol	Hepatitis_B	Measles	BMI	Under_5_Deaths	Total_Expenditure	...	Coun
0	2006	8.608136	1.000000	1	1.349074	92.0	4.123106	61.4	1.000000	8.83	...	
1	2006	7.155418	20.832667	3	1.095445	92.0	4.123106	18.9	2.000000	6.51	...	
2	2004	8.899438	9.000000	0	3.091925	92.0	1.000000	59.9	1.000000	8.30	...	
3	2014	8.809086	10.908712	1	2.596151	92.0	0.000000	57.1	1.000000	8.30	...	
4	2013	7.899367	16.763055	3	0.100000	92.0	3.464102	47.2	2.000000	4.78	...	
...	...	...	...	...	...	...	...	...	...	...	...	
2923	2012	7.924645	5.656854	7	0.100000	92.0	3.605551	54.2	3.000000	3.25	...	
2924	2007	8.514693	11.489125	2	0.100000	98.0	4.123106	57.4	1.732051	2.63	...	
2925	2005	7.021396	12.000000	3	1.526434	92.0	4.123106	18.4	2.000000	7.56	...	
2926	2007	8.173127	14.696938	3	1.862794	92.0	0.000000	47.3	2.000000	4.96	...	
2927	2013	8.252272	14.798649	3	2.147091	89.0	4.123106	24.3	2.000000	4.56	...	

2928 rows × 206 columns

In [91]:
```
pd.unique(life_expectancy_data.Year)
```

Out[91]:
```
array([2015, 2014, 2013, 2012, 2011, 2010, 2009, 2008, 2007, 2006, 2005,
       2004, 2003, 2002, 2001, 2000], dtype=int64)
```

CHECK FOR DUPLICATE DATA

In [92]:
```
data.duplicated().sum()
```

Out[92]: 0

ii. STANDARDIZE THE DATA TO USE FOR MODELS

In [149…
```
def stdData(data,y,std):
    D = data.copy()
```

```
    if std == "ss":
        tr = preprocessing.StandardScaler()
    elif std == "minmax":
        tr = preprocessing.MinMaxScaler()
    else:
        return("Invalid type specifies")

    D=D.drop(y,axis=1)
    D.iloc[:,:]=tr.fit_transform(D.iloc[:,:])

    #D[y] = data[y]
    D[y] = np.log(data[y])
    return (D)
```

In [150…
```
df_ss = stdData(data,"Life_Expectancy","ss")
```

In [151…
```
df_ss.head()
```

Out[151…

	Year	Adult_Mortality	Infant_Deaths	Alcohol	Hepatitis_B	Measles	BMI	Under_5_Deaths	Total_Expenditure	Diphtheria	...	Co
0	-0.325396	-2.332066	-0.623864	-0.476544	-0.483049	0.332244	1.163830	-0.395720	1.446959	-0.481853	...	
1	-0.325396	2.092030	0.598400	-0.716838	-0.483049	0.332244	-0.976760	0.683611	0.352925	-0.481853	...	
2	-0.759257	-0.547497	-1.234996	1.174674	-0.483049	-0.598797	1.088279	-0.395720	1.197029	0.210257	...	
3	1.410048	-0.121718	-0.623864	0.704966	-0.483049	-0.896910	0.947252	-0.395720	1.197029	-0.481853	...	
4	1.193118	1.184217	0.598400	-1.659946	-0.483049	0.135786	0.448621	0.683611	-0.462885	-0.481853	...	

5 rows × 206 columns

In [152…
```
df_mm = stdData(data,"Life_Expectancy","minmax")
```

In [153…
```
df_mm.head()
```

Out[153…

	Year	Adult_Mortality	Infant_Deaths	Alcohol	Hepatitis_B	Measles	BMI	Under_5_Deaths	Total_Expenditure	Diphtheria	...	Count
0	0.400000	0.000000	0.142857	0.314481	0.461538	0.145956	0.788512	0.316228	0.753339	0.500000	...	
1	0.400000	0.980032	0.428571	0.250625	0.461538	0.145956	0.233681	0.632456	0.546750	0.500000	...	
2	0.266667	0.395320	0.000000	0.753282	0.461538	0.035400	0.768930	0.316228	0.706144	0.666667	...	
3	0.933333	0.489639	0.142857	0.628460	0.461538	0.000000	0.732376	0.316228	0.706144	0.500000	...	
4	0.866667	0.778932	0.428571	0.000000	0.461538	0.122628	0.603133	0.632456	0.392698	0.500000	...	

5 rows × 206 columns

- PERFORMED STANDARD SCALING AND MIMMAX SCALING ON DATA OBSERVED THAT STANDARD SCALING GIVES BETTER PERFORMANCE OF MODEL

- ALSO IT IS A UNITLESS SCALING METHOD SO OUR DATA GETS SCALED AND BECOMES UNITLESS

SPLIT DATA INTO TRAIN AND TEST

In [154…
```
def splitdata(data,y,ratio=0.3):

    trainx,testx,trainy,testy = train_test_split(data.drop(y,1),
                                                 data[y],
                                                 test_size = ratio,random_state=0)

    return(trainx,testx,trainy,testy)
```

In [155…
```
trainx,testx,trainy,testy = splitdata(df_ss,'Life_Expectancy')
print(trainx.shape, trainy.shape, testx.shape, testy.shape)
```

```
(2049, 205) (2049,) (879, 205) (879,)
```

In [156…
```
trainx1 =trainx.copy()
trainy1 =trainy.copy()
testx1 =testx.copy()
testy1 =testy.copy()
```

LINEAR REGRESSION (m1)

OLS METHOD

In [157…
```
trainx1.head()
```

Out[157…

	Year	Adult_Mortality	Infant_Deaths	Alcohol	Hepatitis_B	Measles	BMI	Under_5_Deaths	Total_Expenditure	Diphtheria	...	C
1928	0.976187	0.327578	-1.234996	1.165467	0.868473	-0.896910	1.224270	-1.475051	0.786766	1.248422	...	
2497	0.325396	1.753083	0.598400	0.544644	-0.483049	3.826097	-0.649376	0.683611	-0.029044	-0.481853	...	
261	0.108465	-0.358141	0.598400	0.203065	0.530592	0.332244	-0.593972	0.683611	-0.552482	0.902367	...	
1775	0.759257	-0.462544	-1.234996	0.440801	1.882114	0.680559	0.846519	-0.395720	-0.024328	1.594477	...	
2587	0.542326	0.007755	-0.623864	0.794567	0.530592	0.501367	0.690382	-0.395720	-2.009622	-0.827908	...	

5 rows × 205 columns

In [158…
```
trainx1 = sm.add_constant(trainx1)
```

In [159…
```
trainx1.head()
```

Out[159…

	const	Year	Adult_Mortality	Infant_Deaths	Alcohol	Hepatitis_B	Measles	BMI	Under_5_Deaths	Total_Expenditure	...	Count
1928	1.0	0.976187	0.327578	-1.234996	1.165467	0.868473	-0.896910	1.224270	-1.475051	0.786766	...	
2497	1.0	0.325396	1.753083	0.598400	0.544644	-0.483049	3.826097	-0.649376	0.683611	-0.029044	...	
261	1.0	0.108465	-0.358141	0.598400	0.203065	0.530592	0.332244	-0.593972	0.683611	-0.552482	...	
1775	1.0	0.759257	-0.462544	-1.234996	0.440801	1.882114	0.680559	0.846519	-0.395720	-0.024328	...	
2587	1.0	0.542326	0.007755	-0.623864	0.794567	0.530592	0.501367	0.690382	-0.395720	-2.009622	...	

5 rows × 206 columns

In [160…
```
m1_ols = sm.OLS(trainy1,trainx1).fit()
print(m1_ols.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Life_Expectancy   R-squared:                       0.929
Model:                            OLS   Adj. R-squared:                  0.921
Method:                 Least Squares   F-statistic:                     122.4
Date:                Fri, 14 Jan 2022   Prob (F-statistic):               0.00
Time:                        14:29:11   Log-Likelihood:                 5206.3
No. Observations:                2049   AIC:                         -1.002e+04
Df Residuals:                    1851   BIC:                            -8903.
Df Model:                         197
Covariance Type:            nonrobust
================================================================================================================
                                                   coef    std err          t      P>|t|      [0.
025      0.975]
----------------------------------------------------------------------------------------------------------------
```

```
---------------
const                                                 2.1152      0.000   4696.741      0.000       2.
114       2.116
Year                                                  0.0112      0.001     19.692      0.000       0.
010       0.012
Adult_Mortality                                       0.0019      0.001      3.227      0.001       0.
001       0.003
Infant_Deaths                                         0.0007      0.002      0.348      0.728      -0.
003       0.004
Alcohol                                              -0.0034      0.001     -3.344      0.001      -0.
005      -0.001
Hepatitis_B                                          -0.0008      0.001     -0.922      0.357      -0.
002       0.001
Measles                                               0.0004      0.001      0.694      0.488      -0.
001       0.001
BMI                                                  -0.0005      0.001     -0.727      0.467      -0.
002       0.001
Under_5_Deaths                                       -0.0014      0.003     -0.504      0.614      -0.
007       0.004
Total_Expenditure                                     0.0006      0.001      0.882      0.378      -0.
001       0.002
Diphtheria                                            0.0003      0.001      0.360      0.719      -0.
001       0.002
HIV/AIDS                                          -5.51e-17   2.51e-18    -21.910      0.000      -6e
-17   -5.02e-17
GDP                                                   0.0007      0.001      1.167      0.244      -0.
000       0.002
Population                                            0.0006      0.001      1.005      0.315      -0.
001       0.002
Thinness_1to19_Years                                 -0.0022      0.001     -2.044      0.041      -0.
004   -8.78e-05
Income_Composition_Of_Resources                       0.0014      0.002      0.652      0.514      -0.
003       0.006
Schooling                                            -0.0025      0.001     -2.038      0.042      -0.
005   -9.55e-05
Country_Albania                                       0.0035      0.000      7.795      0.000       0.
003       0.004
Country_Algeria                                       0.0032      0.000      7.261      0.000       0.
002       0.004
Country_Angola                                       -0.0096      0.000    -22.076      0.000      -0.
010      -0.009
Country_Antigua and Barbuda                           0.0037      0.001      6.824      0.000       0.
003       0.005
Country_Argentina                                     0.0038      0.001      7.314      0.000       0.
003       0.005
Country_Armenia                                       0.0037      0.001      7.475      0.000       0.
003       0.005
Country_Australia                                     0.0041      0.000      8.421      0.000       0.
003       0.005
Country_Austria                                       0.0032      0.000      6.612      0.000       0.
002       0.004
Country_Azerbaijan                                    0.0022      0.000      4.643      0.000       0.
001       0.003
Country_Bahamas                                       0.0031      0.001      5.392      0.000       0.
002       0.004
Country_Bahrain                                       0.0046      0.001      7.186      0.000       0.
003       0.006
Country_Bangladesh                                    0.0067      0.001     10.561      0.000       0.
005       0.008
Country_Barbados                                      0.0034      0.001      6.178      0.000       0.
002       0.004
Country_Belarus                                       0.0022      0.001      4.308      0.000       0.
001       0.003
Country_Belgium                                       0.0030      0.000      6.259      0.000       0.
002       0.004
Country_Belize                                        0.0006      0.001      1.096      0.273      -0.
000       0.002
Country_Benin                                        -0.0037      0.001     -6.492      0.000      -0.
005      -0.003
Country_Bhutan                                        0.0045      0.001      5.558      0.000       0.
003       0.006
Country_Bolivia (Plurinational State of)             -0.0008      0.001     -1.393      0.164      -0.
002       0.000
Country_Bosnia and Herzegovina                        0.0040      0.001      6.748      0.000       0.
003       0.005
Country_Botswana                                     -0.0037      0.000     -7.836      0.000      -0.
005      -0.003
Country_Brazil                                        0.0028      0.000      6.375      0.000       0.
002       0.004
Country_Brunei Darussalam                             0.0109      0.001     11.951      0.000       0.
009       0.013
Country_Bulgaria                                     -0.0009      0.000     -2.059      0.040      -0.
002   -4.34e-05
```

```
Country_Burkina Faso                                -0.0045      0.000     -9.599      0.000      -0.
005      -0.004
Country_Burundi                                     -0.0052      0.001     -9.462      0.000      -0.
006      -0.004
Country_Cabo Verde                                   0.0056      0.001      9.259      0.000       0.
004       0.007
Country_Cambodia                                     0.0043      0.001      6.693      0.000       0.
003       0.006
Country_Cameroon                                    -0.0059      0.000    -12.217      0.000      -0.
007      -0.005
Country_Canada                                       0.0072      0.001     13.557      0.000       0.
006       0.008
Country_Central African Republic                   -0.0098      0.001    -19.518      0.000      -0.
011      -0.009
Country_Chad                                        -0.0089      0.001    -16.054      0.000      -0.
010      -0.008
Country_Chile                                        0.0059      0.001     11.515      0.000       0.
005       0.007
Country_China                                        0.0099      0.001     14.305      0.000       0.
009       0.011
Country_Colombia                                     0.0026      0.000      5.693      0.000       0.
002       0.003
Country_Comoros                                     -0.0012      0.000     -2.694      0.007      -0.
002      -0.000
Country_Congo                                       -0.0029      0.001     -5.125      0.000      -0.
004      -0.002
Country_Costa Rica                                   0.0050      0.000     10.529      0.000       0.
004       0.006
Country_Croatia                                      0.0008      0.000      1.682      0.093      -0.
000       0.002
Country_Cuba                                         0.0051      0.000     11.035      0.000       0.
004       0.006
Country_Cyprus                                       0.0023      0.001      4.470      0.000       0.
001       0.003
Country_Czechia                                      0.0012      0.000      2.501      0.012       0.
000       0.002
Country_Côte dIvoire                                -0.0084      0.000    -18.793      0.000      -0.
009      -0.008
Country_Democratic Peoples Republic of Korea         0.0072      0.001     10.073      0.000       0.
006       0.009
Country_Democratic Republic of the Congo            -0.0049      0.000    -10.091      0.000      -0.
006      -0.004
Country_Denmark                                      0.0021      0.001      4.102      0.000       0.
001       0.003
Country_Djibouti                                    -0.0021      0.000     -4.333      0.000      -0.
003      -0.001
Country_Dominican Republic                           0.0021      0.001      3.633      0.000       0.
001       0.003
Country_Ecuador                                      0.0032      0.001      5.634      0.000       0.
002       0.004
Country_Egypt                                        0.0020      0.000      4.338      0.000       0.
001       0.003
Country_El Salvador                                  0.0019      0.001      3.547      0.000       0.
001       0.003
Country_Equatorial Guinea                           -0.0049      0.000    -10.182      0.000      -0.
006      -0.004
Country_Eritrea                                     -0.0012      0.001     -1.851      0.064      -0.
002    7.16e-05
Country_Estonia                                      0.0030      0.001      5.668      0.000       0.
002       0.004
Country_Ethiopia                                    -0.0031      0.001     -5.936      0.000      -0.
004      -0.002
Country_Fiji                                         0.0009      0.001      1.705      0.088      -0.
000       0.002
Country_Finland                                      0.0063      0.001      9.738      0.000       0.
005       0.008
Country_France                                       0.0069      0.001     11.416      0.000       0.
006       0.008
Country_Gabon                                       -0.0004      0.000     -0.927      0.354      -0.
001       0.000
Country_Gambia                                      -0.0023      0.000     -4.661      0.000      -0.
003      -0.001
Country_Georgia                                      0.0040      0.000      9.096      0.000       0.
003       0.005
Country_Germany                                      0.0034      0.001      6.757      0.000       0.
002       0.004
Country_Ghana                                       -0.0020      0.000     -4.704      0.000      -0.
003      -0.001
Country_Greece                                       0.0064      0.001     12.159      0.000       0.
005       0.007
Country_Grenada                                      0.0026      0.001      4.413      0.000       0.
001       0.004
Country_Guatemala                                    0.0014      0.000      2.868      0.004       0.
```

```
000       0.002
Country_Guinea                                -0.0049      0.001     -9.334      0.000      -0.
006      -0.004
Country_Guinea-Bissau                         -0.0050      0.001     -9.392      0.000      -0.
006      -0.004
Country_Guyana                                -0.0013      0.000     -3.063      0.002      -0.
002      -0.000
Country_Haiti                                 -0.0038      0.000     -7.803      0.000      -0.
005      -0.003
Country_Honduras                               0.0024      0.000      4.786      0.000       0.
001       0.003
Country_Hungary                               -0.0003      0.001     -0.504      0.614      -0.
001       0.001
Country_Iceland                                0.0034      0.001      5.899      0.000       0.
002       0.005
Country_India                                  0.0048      0.001      7.423      0.000       0.
004       0.006
Country_Indonesia                              0.0058      0.001      8.547      0.000       0.
004       0.007
Country_Iran (Islamic Republic of)             0.0033      0.000      6.960      0.000       0.
002       0.004
Country_Iraq                                   0.0016      0.001      3.014      0.003       0.
001       0.003
Country_Ireland                                0.0026      0.001      4.899      0.000       0.
002       0.004
Country_Israel                                 0.0072      0.001     11.100      0.000       0.
006       0.008
Country_Italy                                  0.0037      0.000      7.919      0.000       0.
003       0.005
Country_Jamaica                                0.0029      0.000      6.947      0.000       0.
002       0.004
Country_Japan                                  0.0105      0.001     13.741      0.000       0.
009       0.012
Country_Jordan                                 0.0029      0.000      6.160      0.000       0.
002       0.004
Country_Kazakhstan                             0.0007      0.001      1.276      0.202      -0.
000       0.002
Country_Kenya                                 -0.0038      0.000     -8.545      0.000      -0.
005      -0.003
Country_Kiribati                               0.0044      0.001      5.295      0.000       0.
003       0.006
Country_Kuwait                                 0.0032      0.000      6.459      0.000       0.
002       0.004
Country_Kyrgyzstan                             0.0017      0.000      3.797      0.000       0.
001       0.003
Country_Lao Peoples Democratic Republic        0.0029      0.001      4.187      0.000       0.
002       0.004
Country_Latvia                                -0.0007      0.000     -1.604      0.109      -0.
002       0.000
Country_Lebanon                                0.0035      0.001      6.300      0.000       0.
002       0.005
Country_Lesotho                               -0.0062      0.001    -11.456      0.000      -0.
007      -0.005
Country_Liberia                               -0.0038      0.000     -7.882      0.000      -0.
005      -0.003
Country_Libya                                  0.0027      0.000      6.171      0.000       0.
002       0.004
Country_Lithuania                             -0.0009      0.001     -1.779      0.075      -0.
002    9.65e-05
Country_Luxembourg                             0.0028      0.000      5.604      0.000       0.
002       0.004
Country_Madagascar                            -0.0008      0.001     -1.381      0.168      -0.
002       0.000
Country_Malawi                                -0.0051      0.000    -10.256      0.000      -0.
006      -0.004
Country_Malaysia                               0.0095      0.001     12.945      0.000       0.
008       0.011
Country_Maldives                               0.0106      0.001     12.678      0.000       0.
009       0.012
Country_Mali                                  -0.0055      0.000    -11.589      0.000      -0.
006      -0.005
Country_Malta                                  0.0027      0.000      5.665      0.000       0.
002       0.004
Country_Mauritania                            -0.0030      0.000     -6.559      0.000      -0.
004      -0.002
Country_Mauritius                              0.0053      0.001      8.696      0.000       0.
004       0.006
Country_Mexico                                 0.0039      0.001      7.570      0.000       0.
003       0.005
Country_Micronesia (Federated States of)      -0.0001      0.001     -0.198      0.843      -0.
001       0.001
Country_Mongolia                               0.0051      0.001      6.783      0.000       0.
004       0.007
```

```
Country_Montenegro                                0.0028      0.001      4.909      0.000       0.
002       0.004
Country_Morocco                                   0.0025      0.000      5.677      0.000       0.
002       0.003
Country_Mozambique                               -0.0066      0.001    -13.114      0.000      -0.
008      -0.006
Country_Myanmar                                   0.0045      0.001      6.664      0.000       0.
003       0.006
Country_Namibia                                  -0.0016      0.000     -3.577      0.000      -0.
003      -0.001
Country_Nepal                                     0.0053      0.001      8.532      0.000       0.
004       0.007
Country_Netherlands                               0.0030      0.001      5.259      0.000       0.
002       0.004
Country_New Zealand                               0.0037      0.000      7.574      0.000       0.
003       0.005
Country_Nicaragua                                 0.0026      0.001      4.932      0.000       0.
002       0.004
Country_Niger                                    -0.0037      0.001     -7.207      0.000      -0.
005      -0.003
Country_Nigeria                                  -0.0073      0.000    -15.259      0.000      -0.
008      -0.006
Country_Norway                                    0.0027      0.001      5.125      0.000       0.
002       0.004
Country_Oman                                      0.0040      0.000      7.994      0.000       0.
003       0.005
Country_Pakistan                                  0.0041      0.001      6.291      0.000       0.
003       0.005
Country_Panama                                    0.0040      0.000      8.538      0.000       0.
003       0.005
Country_Papua New Guinea                         -0.0036      0.001     -6.822      0.000      -0.
005      -0.003
Country_Paraguay                                  0.0021      0.000      4.786      0.000       0.
001       0.003
Country_Peru                                      0.0028      0.000      5.567      0.000       0.
002       0.004
Country_Philippines                               0.0059      0.001      8.248      0.000       0.
005       0.007
Country_Poland                                    0.0008      0.000      1.760      0.079   -8.83e
-05       0.002
Country_Portugal                                  0.0028      0.000      5.741      0.000       0.
002       0.004
Country_Qatar                                     0.0049      0.001      8.112      0.000       0.
004       0.006
Country_Republic of Korea                         0.0127      0.001     17.580      0.000       0.
011       0.014
Country_Republic of Moldova                       0.0023      0.000      4.979      0.000       0.
001       0.003
Country_Romania                                   0.0006      0.001      0.949      0.343      -0.
001       0.002
Country_Russian Federation                       -0.0004      0.000     -0.931      0.352      -0.
001       0.000
Country_Rwanda                                   -0.0026      0.000     -5.525      0.000      -0.
004      -0.002
Country_Saint Lucia                               0.0030      0.001      4.993      0.000       0.
002       0.004
Country_Saint Vincent and the Grenadines          0.0026      0.001      4.893      0.000       0.
002       0.004
Country_Samoa                                     0.0034      0.001      6.449      0.000       0.
002       0.004
Country_Sao Tome and Principe                     0.0012      0.001      2.046      0.041    4.95e
-05       0.002
Country_Saudi Arabia                              0.0035      0.001      6.559      0.000       0.
002       0.005
Country_Senegal                                  -0.0008      0.000     -1.803      0.072      -0.
002    7.38e-05
Country_Serbia                                    0.0029      0.000      6.456      0.000       0.
002       0.004
Country_Seychelles                                0.0049      0.001      7.552      0.000       0.
004       0.006
Country_Sierra Leone                             -0.0018      0.001     -3.358      0.001      -0.
003      -0.001
Country_Singapore                                 0.0098      0.001     12.216      0.000       0.
008       0.011
Country_Slovakia                                  0.0001      0.000      0.233      0.816      -0.
001       0.001
Country_Slovenia                                  0.0026      0.000      5.921      0.000       0.
002       0.003
Country_Solomon Islands                           0.0059      0.001      7.545      0.000       0.
004       0.007
Country_Somalia                                  -0.0065      0.001    -10.799      0.000      -0.
008      -0.005
Country_South Africa                             -0.0038      0.000     -8.074      0.000      -0.
```

```
005      -0.003
Country_South Sudan                                    -0.0081      0.000    -18.031      0.000      -0.
009      -0.007
Country_Spain                                           0.0037      0.000      8.598      0.000       0.
003       0.004
Country_Sri Lanka                                       0.0094      0.001     12.839      0.000       0.
008       0.011
Country_Sudan                                          -0.0035      0.000     -7.410      0.000      -0.
004      -0.003
Country_Suriname                                        0.0009      0.001      1.663      0.097      -0.
000       0.002
Country_Swaziland                                      -0.0081      0.000    -18.507      0.000      -0.
009      -0.007
Country_Sweden                                          0.0034      0.000      6.992      0.000       0.
002       0.004
Country_Switzerland                                     0.0038      0.001      7.422      0.000       0.
003       0.005
Country_Syrian Arab Republic                            0.0071      0.001      9.888      0.000       0.
006       0.008
Country_Tajikistan                                     -0.0001      0.000     -0.308      0.758      -0.
001       0.001
Country_Thailand                                        0.0095      0.001     12.219      0.000       0.
008       0.011
Country_The former Yugoslav republic of Macedonia       0.0025      0.001      4.995      0.000       0.
002       0.003
Country_Timor-Leste                                     0.0050      0.001      7.232      0.000       0.
004       0.006
Country_Togo                                           -0.0042      0.000     -8.808      0.000      -0.
005      -0.003
Country_Tonga                                           0.0024      0.000      4.772      0.000       0.
001       0.003
Country_Trinidad and Tobago                             0.0014      0.001      2.545      0.011       0.
000       0.002
Country_Tunisia                                         0.0040      0.000      8.134      0.000       0.
003       0.005
Country_Turkey                                          0.0037      0.000      7.410      0.000       0.
003       0.005
Country_Turkmenistan                                   -0.0009      0.001     -1.719      0.086      -0.
002       0.000
Country_Uganda                                         -0.0041      0.001     -8.088      0.000      -0.
005      -0.003
Country_Ukraine                                         0.0019      0.001      3.682      0.000       0.
001       0.003
Country_United Arab Emirates                            0.0044      0.000      9.424      0.000       0.
003       0.005
Country_United Kingdom of Great Britain and Northern Ireland     0.0030      0.001      5.734      0.000       0.
002       0.004
Country_United Republic of Tanzania                    -0.0044      0.000     -9.682      0.000      -0.
005      -0.003
Country_United States of America                        0.0025      0.001      5.029      0.000       0.
002       0.004
Country_Uruguay                                         0.0039      0.001      7.088      0.000       0.
003       0.005
Country_Uzbekistan                                      0.0008      0.000      1.753      0.080   -9.27e
-05       0.002
Country_Vanuatu                                         0.0017      0.001      2.967      0.003       0.
001       0.003
Country_Venezuela (Bolivarian Republic of)              0.0027      0.001      5.322      0.000       0.
002       0.004
Country_Viet Nam                                        0.0098      0.001     14.169      0.000       0.
008       0.011
Country_Yemen                                          -0.0023      0.000     -4.957      0.000      -0.
003      -0.001
Country_Zambia                                         -0.0049      0.000    -11.080      0.000      -0.
006      -0.004
Country_Zimbabwe                                       -0.0034      0.001     -6.098      0.000      -0.
005      -0.002
Status_Developing                                      -0.0165      0.001    -20.471      0.000      -0.
018      -0.015
Region_Asia                                            -0.0181      0.002     -8.841      0.000      -0.
022      -0.014
Region_Latin America                                    0.0150      0.001     25.333      0.000       0.
014       0.016
Region_North America and Oceania                        0.0072      0.000     14.841      0.000       0.
006       0.008
Region_Western Asia and North Africa                    0.0102      0.000     22.125      0.000       0.
009       0.011
Region_the European Region                              0.0170      0.001     19.095      0.000       0.
015       0.019
Region_the European and Central Asian Region            0.0057      0.000     15.380      0.000       0.
005       0.006
==============================================================================
Omnibus:                      873.635   Durbin-Watson:                   2.016
```

```
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            30030.913
Skew:                           1.347   Prob(JB):                         0.00
Kurtosis:                      21.561   Cond. No.                     1.32e+16
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 7.91e-29. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

VALIDATING ASSUMPTIONS FOR LINEAR REGRESSION

i . Residual mean = 0

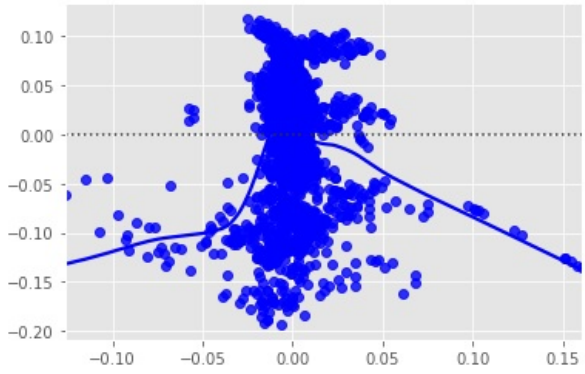
In [161…
```
np.mean(m1_ols.resid)
```

Out[161…
```
2.9354535179156896e-15
```

ii. Residuals have a constant variance (homoscedasticity)

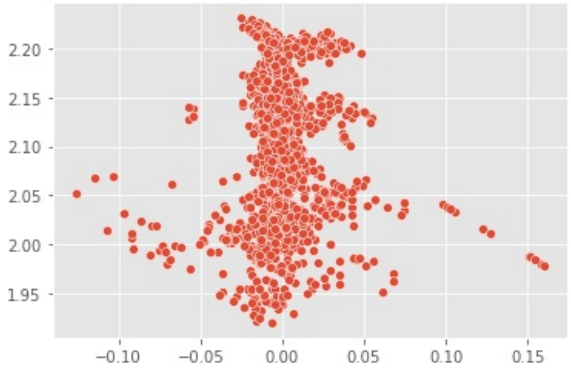
In [162…
```
sns.residplot(m1_ols.resid,m1_ols.predict(),color='blue',lowess=True)
```

Out[162…
```
<AxesSubplot:>
```



In [163…
```
sns.scatterplot(m1_ols.resid,m1_ols.predict())
```

Out[163…
```
<AxesSubplot:>
```



- The errors shows heteroscedasticity

HYPOTHESIS TESTING FOR HETEROSCEDASTICITY

H0: homoscedasticity

H1: heteroscedasticity

i) Breusch-Pagan test against heteroscedasticity

In [164…
```
import statsmodels.stats.api as stats

pvalue_bp = stats.het_breuschpagan(m1_ols.resid,m1_ols.model.exog)[1]

if pvalue_bp < 0.05:
    print('Reject H0 : Model is Heteroscedastic')
else:
    print('FTR H0: Model is Homoscedastic')
```

```
Reject H0 : Model is Heteroscedastic
```

ii) White's test

In [165…
```
from statsmodels.stats.diagnostic import het_white
pvalue_wt = het_white(m1_ols.resid,m1_ols.model.exog)[1]

if pvalue_wt < 0.05:
    print('Reject H0 : Model is Heteroscedastic')
else:
    print('FTR H0: Model is Homoscedastic')
```

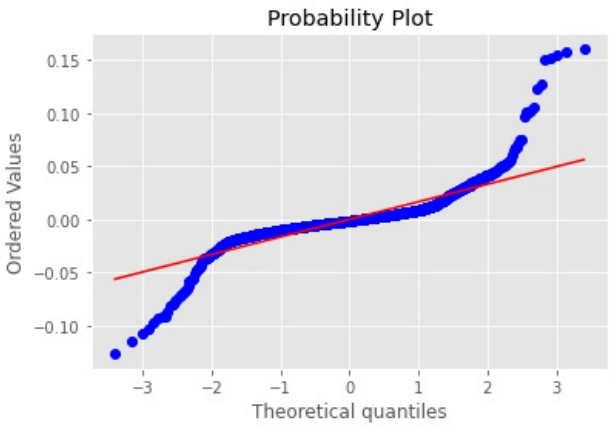
```
FTR H0: Model is Homoscedastic
```

- The White's test says errors are homoscedastic in nature after transformation of data

iii. Residuals have a Normal distribution

In [166…
```
import scipy.stats as spst
spst.probplot(m1_ols.resid,dist='norm',plot=pylab)
```

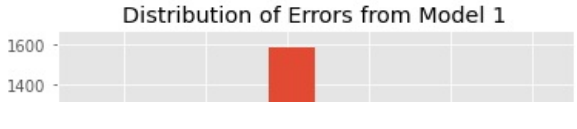
Out[166…
```
((array([-3.3989476 , -3.14834602, -3.00937195, ...,  3.00937195,
         3.14834602,  3.3989476 ]),
  array([-0.12628135, -0.11478696, -0.10724714, ...,  0.15448023,
         0.15826456,  0.16033939])),
 (0.01655217585820162, 2.9326983721898755e-15, 0.8668678296886438))
```

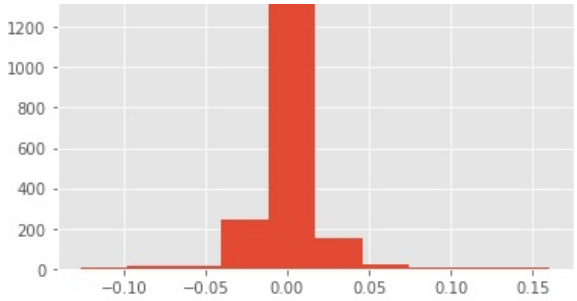


Alternatively, plot a histogram of the residuals to check the distribution

In [167…
```
plt.hist(m1_ols.resid)
plt.title('Distribution of Errors from Model 1')
```

Out[167…
```
Text(0.5, 1.0, 'Distribution of Errors from Model 1')
```





iv. rows > columns

In [168…
```
trainx1.shape
```

Out[168… (2049, 206)

v. Outliers : Check outliers from the boxplot and interpret it accordingly

- The outliers are handled above.

MODEL PREDICTION FUNCTION

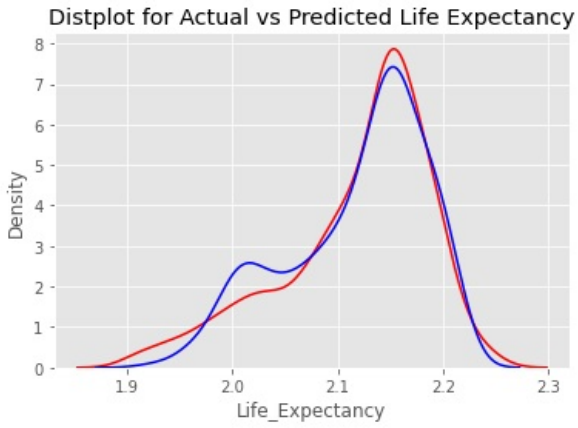
In [169…
```
def model_prediction(model,testx1,testy1,trainx1,trainy1):
    y_train_pred = model.predict(trainx1)
    y_test_pred = model.predict(testx1)

    mse_train = mean_squared_error(trainy1,y_train_pred)
    mse_test = mean_squared_error(testy1,y_test_pred)

    print("Model Train Error\n\t MSE={},\n\tRMSE={}".format(round(mse_train,4),round(np.sqrt(mse_train),4)))
    print("Model Test Error\n\t MSE={},\n\tRMSE={}".format(round(mse_test,4),round(np.sqrt(mse_test),4)))

    r2_train = r2_score(trainy1,y_train_pred)
    r2_test = r2_score(testy1,y_test_pred)
    print("R2 Train : {} , R2 Test : {}".format(round(r2_train,4),round(r2_test,4)))

    ax1 = sns.distplot(testy1,hist=False,color='r',label="Actual Life Expectancy")
    sns.distplot(y_test_pred,hist=False,color='b',ax=ax1,label="Predicted Life Expectancy")
    plt.title("Distplot for Actual vs Predicted Life Expectancy")
    plt.show()

    sns.regplot(testy1,y_test_pred,marker='.',color='yellow',line_kws={'color':'red'},ci=None)
    plt.title("Reg Plot for Actual vs Predicted Life Expectancy")
    plt.show()

    plt.plot(testy1, y_test_pred, "^", color = 'r')
    plt.xlabel('Model Predictions')
    plt.ylabel('True Values')
    plt.show()

    df = pd.DataFrame({'actual':round(testy1,4),'predicted':np.round(y_test_pred,4)})
    print(df)

    return (mse_train,mse_test,r2_train,r2_test,y_train_pred,y_test_pred)
```

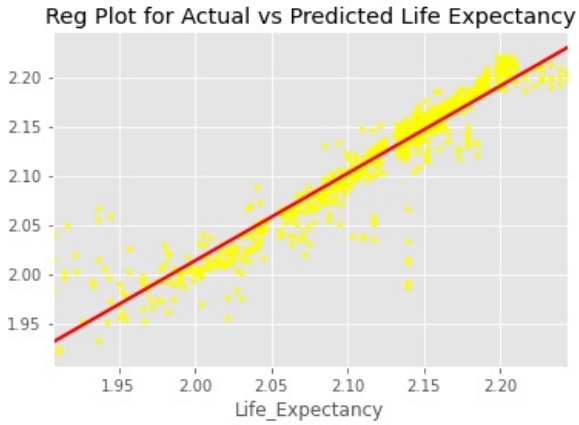
CROSS VALIDATION

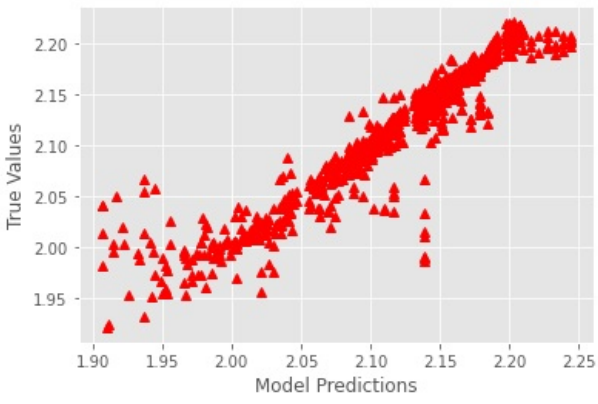
In [170…
```
cv_mse = []
cv_r2 =[]
X = trainx1.values
Y = trainy1.values

folds = 5
kf = KFold(folds)

for train_index,test_index in kf.split(X):
    cv_trainx,cv_trainy = X[train_index], Y[train_index]
    cv_testx, cv_testy = X[test_index], Y[test_index]

    cv_model = sm.OLS(cv_trainy,cv_trainx).fit()
```

```
    cv_pred = cv_model.predict(cv_testx)
    cv_mse.append(mean_squared_error(cv_testy,cv_pred))
    cv_r2.append(r2_score(cv_testy,cv_pred))

print(cv_mse)
print(cv_r2)

cv_mean = np.mean(cv_mse)
cv_mean

cv_r2 = np.mean(cv_r2)
cv_r2

print("Cross Validation \n\t mse1_ols={}, \n\t rmse1_ols={}, \n\t r2_ols ={}".format(round(cv_mean,4),round(np.sq

testx1 = sm.add_constant(testx1, has_constant='add')

mse_train_ols,mse_test_ols,r2_train_ols,r2_test_ols,y_train_pred,y_test_pred=model_prediction(m1_ols,testx1,testy
```

```
[0.0004991003277630389, 0.00044855468343326857, 0.00049183085096292, 0.00040109895788829875, 0.000368609222487760
85]
[0.9032070274099278, 0.9155809008290239, 0.9045382547210712, 0.921941109858623, 0.9217966524190315]
Cross Validation
         mse1_ols=0.0004,
         rmse1_ols=0.021,
         r2_ols =0.9134
Model Train Error
         MSE=0.0004,
        RMSE=0.0191
Model Test Error
         MSE=0.0006,
        RMSE=0.0238
R2 Train : 0.9287 , R2 Test : 0.8851
```








```
actual  predicted
```

```
2216  2.1327     2.1315
836   2.1764     2.1793
2396  2.0676     2.0814
1962  2.1547     2.1491
305   2.1390     2.0667
...      ...        ...
1713  2.1156     2.1278
1499  2.0978     2.0801
2252  2.1621     2.1606
2072  2.1554     2.1531
411   2.1554     2.1639

[879 rows x 2 columns]
```

In [115…

```
def anova_test(x,y,data):
    model = ols('x~y',data=data).fit()
    anova = sm.stats.anova_lm(model,type=2)
    pvalue = anova['PR(>F)'][0]

    if pvalue < 0.05:
        msg = 'Reject H0: Feature {} is significant'.format(x.name)
    else:
        msg = 'FTR H0: Feature {} is insignificant'.format(x.name)

    return(msg)
```

In [116…

```
anova_test(df_ss.Infant_Deaths, df_ss.Life_Expectancy,df_ss)
```

Out[116…

```
'Reject H0: Feature Infant_Deaths is significant'
```

ii. Linear Regression library

In [117…

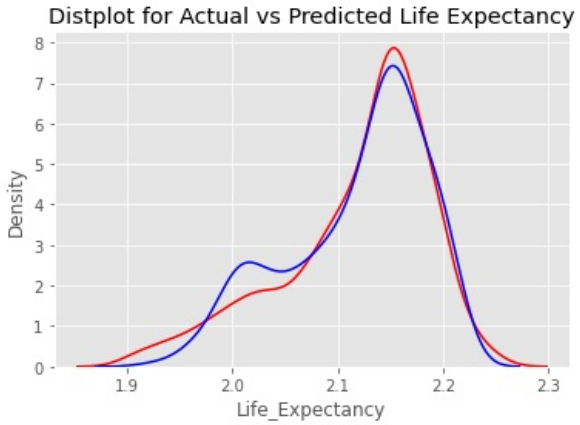
```
m1_lin_reg = LinearRegression()
m1_lin_reg.fit(trainx1,trainy1)

print('Intercept',m1_lin_reg.intercept_)
for col,val in zip(trainx1.columns.values,m1_lin_reg.coef_):
    print(col,val)

mse_train_lin_reg,mse_test_lin_reg,r2_train_lin_reg,r2_test_lin_reg,y_train_pred,y_test_pred=model_prediction(m1_
```

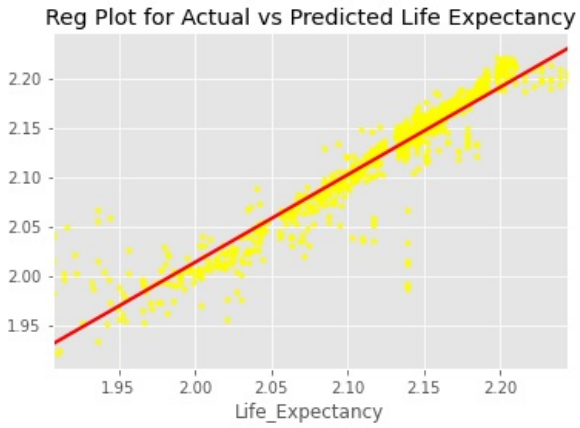
```
Intercept 1998946.4355725725
const -1998944.3203623428
Year 0.011170525540718556
Adult_Mortality 0.0018949598188706361
Infant_Deaths 0.0006382790047510826
Alcohol -0.0034388161582500423
Hepatitis_B -0.0007892020491385918
Measles 0.00037909126225283557
BMI -0.0005433321333838453
Under_5_Deaths -0.001410309363206054
Total_Expenditure 0.0006196628553514025
Diphtheria 0.0003089901278763522
HIV/AIDS 503266085.57972664
GDP 0.0007030526745589252
Population 0.000647485825977821
Thinness_1to19_Years -0.0021606189583719027
Income_Composition_Of_Resources 0.001426671882374067
Schooling -0.002531903765907551
Country_Albania -508965680.40312046
Country_Algeria 1028708897.7385685
Country_Angola -2699841755.1982493
Country_Antigua and Barbuda -630374575.7486792
Country_Argentina -630374575.7485131
Country_Armenia 6539704489.297501
Country_Australia 2785674655.7071805
Country_Austria 4072283391.0191793
Country_Azerbaijan 6539704489.295895
Country_Bahamas -630374575.7492238
Country_Bahrain 1028708897.7398711
Country_Bangladesh 0.006689800406168087
Country_Barbados -630374575.7489376
Country_Belarus 6539704489.295959
```

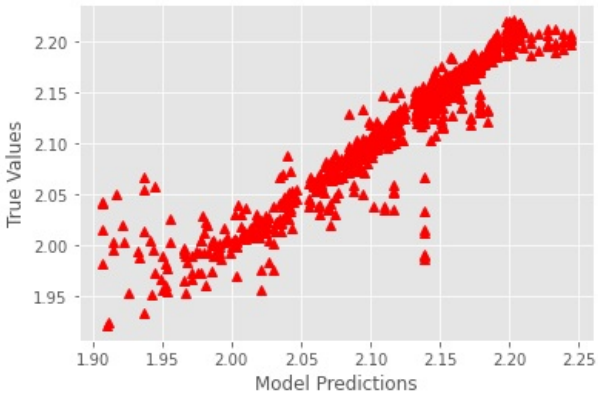
Country_Belgium 4072283391.018977
Country_Belize -630374575.7517205
Country_Benin -2699841755.1923747
Country_Bhutan 0.0045280265061421665
Country_Bolivia (Plurinational State of) -630374575.7530619
Country_Bosnia and Herzegovina -508965680.4027308
Country_Botswana -2699841755.192413
Country_Brazil -630374575.7494458
Country_Brunei Darussalam 0.010861953894500348
Country_Bulgaria 4072283391.01509
Country_Burkina Faso -2699841755.1931624
Country_Burundi -2699841755.1939344
Country_Cabo Verde -2699841755.1830792
Country_Cambodia 0.004346014221720833
Country_Cameroon -2699841755.194646
Country_Canada -1795574415.7123861
Country_Central African Republic -2699841755.1984863
Country_Chad -2699841755.197523
Country_Chile -630374575.7464463
Country_China 0.0099372108972035
Country_Colombia -630374575.7497637
Country_Comoros -2699841755.189878
Country_Congo -2699841755.191555
Country_Costa Rica -630374575.7472485
Country_Croatia 4072283391.016806
Country_Cuba -630374575.747222
Country_Cyprus 4072283391.0183067
Country_Czechia 4072283391.017159
Country_Côte dIvoire -2699841755.1971045
Country_Democratic Peoples Republic of Korea 0.007197185358234814
Country_Democratic Republic of the Congo -2699841755.1935863
Country_Denmark 4072283391.0181236
Country_Djibouti -2699841755.1907353
Country_Dominican Republic -630374575.7502321
Country_Ecuador -630374575.7491074
Country_Egypt 1028708897.7373149
Country_El Salvador -630374575.75049
Country_Equatorial Guinea -2699841755.193592
Country_Eritrea -2699841755.1898737
Country_Estonia -508965680.40366435
Country_Ethiopia -2699841755.1917963
Country_Fiji -1795574415.7186694
Country_Finland -508965680.40038943
Country_France -508965680.39976907
Country_Gabon -2699841755.1890945
Country_Gambia -2699841755.1909666
Country_Georgia 6539704489.29771
Country_Germany 4072283391.0193944
Country_Ghana -2699841755.190694
Country_Greece -508965680.4002746
Country_Grenada -630374575.7497689
Country_Guatemala -630374575.7508837
Country_Guinea -2699841755.193596
Country_Guinea-Bissau -2699841755.1937103
Country_Guyana -630374575.7536733
Country_Haiti -630374575.756149
Country_Honduras -630374575.749977
Country_Hungary 4072283391.015773
Country_Iceland 4072283391.0194335
Country_India 0.004814603364034191
Country_Indonesia 0.0057919937403996605
Country_Iran (Islamic Republic of) 1028708897.7386518
Country_Iraq 1028708897.7369317
Country_Ireland 4072283391.0186143
Country_Israel 1028708897.74254
Country_Italy 4072283391.0196943
Country_Jamaica -630374575.7494315
Country_Japan 4581249071.433154
Country_Jordan 1028708897.73822
Country_Kazakhstan 6539704489.294393
Country_Kenya -2699841755.192506
Country_Kiribati 0.004422976320442586
Country_Kuwait 1028708897.7385759
Country_Kyrgyzstan 6539704489.29547
Country_Lao Peoples Democratic Republic 0.0029142602158477708
Country_Latvia 4072283391.0152745
Country_Lebanon 1028708897.7388176
Country_Lesotho -2699841755.1948986
Country_Liberia -2699841755.1924543
Country_Libya 1028708897.7380999
Country_Lithuania 4072283391.0151224
Country_Luxembourg 4072283391.018745
Country_Madagascar -2699841755.189493

Country_Malawi -2699841755.193782
Country_Malaysia 0.009470888021477483
Country_Maldives 0.010577309909286466
Country_Mali -2699841755.194163
Country_Malta 4072283391.0186763
Country_Mauritania 1028708897.7323729
Country_Mauritius -2699841755.1834054
Country_Mexico -630374575.7483709
Country_Micronesia (Federated States of) -1795574415.7196486
Country_Mongolia 0.005117695559350351
Country_Montenegro -508965680.4039084
Country_Morocco 1028708897.7378739
Country_Mozambique -2699841755.1952457
Country_Myanmar 0.004543060401045707
Country_Namibia -2699841755.190274
Country_Nepal 0.0053156466077223085
Country_Netherlands 4072283391.019043
Country_New Zealand 2785674655.7068086
Country_Nicaragua -630374575.7497098
Country_Niger -2699841755.1923895
Country_Nigeria -2699841755.1960244
Country_Norway 4072283391.01872
Country_Oman 1028708897.73931
Country_Pakistan 0.0041026981594421115
Country_Panama -630374575.74828
Country_Papua New Guinea -1795574415.7231903
Country_Paraguay -630374575.7502067
Country_Peru -630374575.7495577
Country_Philippines 0.005973365309186171
Country_Poland 4072283391.016801
Country_Portugal 4072283391.0187907
Country_Qatar 1028708897.7402763
Country_Republic of Korea 0.012721399510247804
Country_Republic of Moldova 6539704489.296009
Country_Romania 4072283391.0165386
Country_Russian Federation -508965680.40714663
Country_Rwanda -2699841755.191263
Country_Saint Lucia -630374575.7493467
Country_Saint Vincent and the Grenadines -630374575.7496756
Country_Samoa -1795574415.7162068
Country_Sao Tome and Principe -2699841755.18749
Country_Saudi Arabia 1028708897.738858
Country_Senegal -2699841755.1895223
Country_Serbia -508965680.40382206
Country_Seychelles -2699841755.1837325
Country_Sierra Leone -2699841755.190486
Country_Singapore 4581249071.43255
Country_Slovakia 4072283391.016141
Country_Slovenia 4072283391.018644
Country_Solomon Islands 0.0058843159491209236
Country_Somalia -2699841755.1951413
Country_South Africa -2699841755.1924157
Country_South Sudan 1028708897.7272748
Country_Spain 4072283391.019707
Country_Sri Lanka 0.009428136338879917
Country_Sudan 1028708897.7318792
Country_Suriname -630374575.7514302
Country_Swaziland -2699841755.1968307
Country_Sweden 4072283391.019416
Country_Switzerland 4072283391.019791
Country_Syrian Arab Republic 0.007087445022896352
Country_Tajikistan 6539704489.293612
Country_Thailand 0.009555865742346844
Country_The former Yugoslav republic of Macedonia -508965680.4041759
Country_Timor-Leste 0.004962969007547226
Country_Togo -2699841755.1928506
Country_Tonga -1795574415.7171762
Country_Trinidad and Tobago -630374575.7509432
Country_Tunisia 1028708897.7393556
Country_Turkey 1028708897.7390633
Country_Turkmenistan 6539704489.292863
Country_Uganda -2699841755.192826
Country_Ukraine 6539704489.295621
Country_United Arab Emirates 1028708897.7398244
Country_United Kingdom of Great Britain and Northern Ireland 4072283391.019018
Country_United Republic of Tanzania -2699841755.1930585
Country_United States of America 2785674655.7056375
Country_Uruguay -630374575.7483886
Country_Uzbekistan 6539704489.294552
Country_Vanuatu -1795574415.717881
Country_Venezuela (Bolivarian Republic of) -630374575.7495716
Country_Viet Nam 0.009774605459755144
Country_Yemen 1028708897.7330629

```
Country_Zambia -2699841755.1935515
Country_Zimbabwe -2699841755.1920705
Status_Developing 23605416779.03482
Region_Asia -12988135088.275557
Region_Latin America -10529930143.005266
Region_North America and Oceania -2787945005.84444
Region_Western Asia and North Africa -16120240175.369741
Region_the European Region -11936016449.707314
Region_the European and Central Asian Region -29790343815.818714
Model Train Error
         MSE=0.0004,
        RMSE=0.0191
Model Test Error
         MSE=0.0006,
        RMSE=0.0238
R2 Train : 0.9287 , R2 Test : 0.8852
```







	actual	predicted
2216	2.1327	2.1320
836	2.1764	2.1790
2396	2.0676	2.0815
1962	2.1547	2.1494
305	2.1390	2.0666
...	...	...
1713	2.1156	2.1282
1499	2.0978	2.0802
2252	2.1621	2.1605
2072	2.1554	2.1531
411	2.1554	2.1637

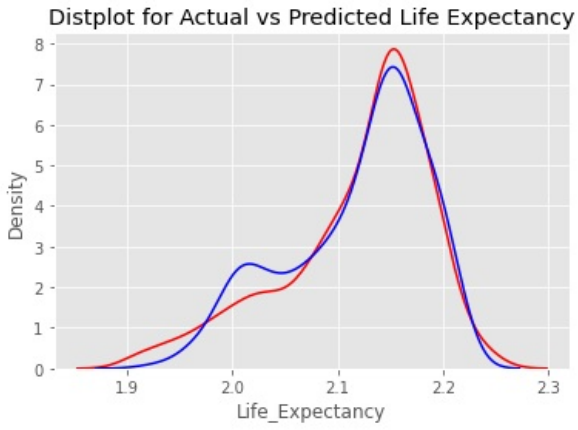
[879 rows x 2 columns]

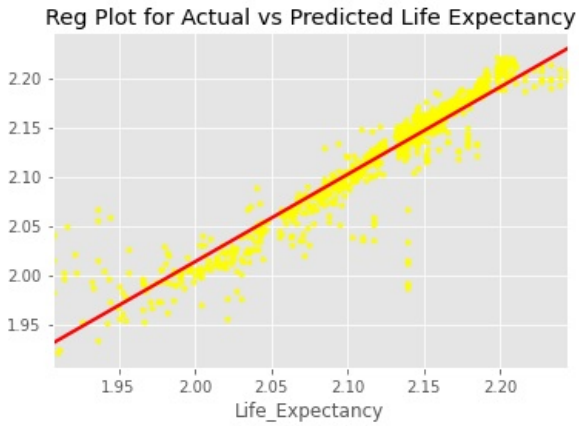
normalise = True only when data is not scaled else do normalise = False no need of n_jobs parameter as we have no huge data

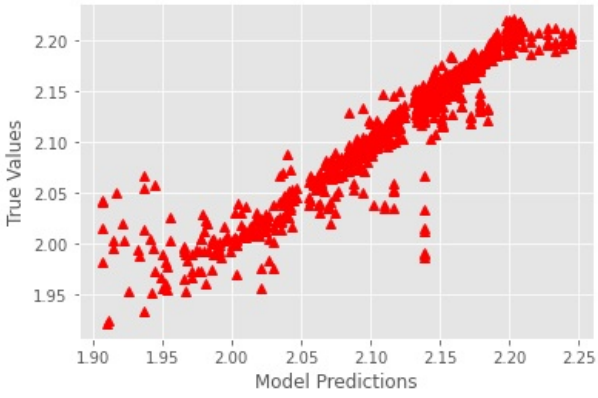
In [118…

```
m1_lin_reg = LinearRegression(normalize=False,fit_intercept = True)
m1_lin_reg.fit(trainx1,trainy1)

mse_train_lin_reg,mse_test_lin_reg,r2_train_lin_reg,r2_test_lin_reg,y_train_pred,y_test_pred=model_prediction(m1_
```

```
Model Train Error
         MSE=0.0004,
        RMSE=0.0191
Model Test Error
         MSE=0.0006,
        RMSE=0.0238
R2 Train : 0.9287 , R2 Test : 0.8852
```





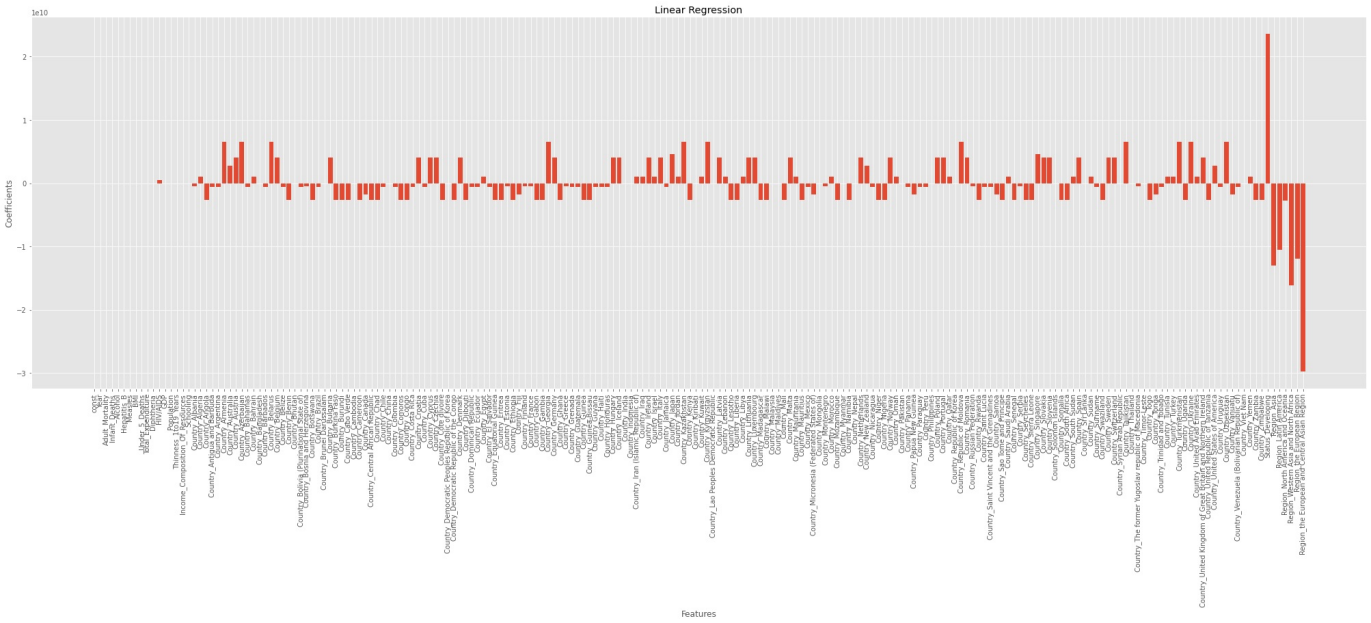


	actual	predicted
2216	2.1327	2.1320
836	2.1764	2.1790
2396	2.0676	2.0815
1962	2.1547	2.1494
305	2.1390	2.0666
...	...	...
1713	2.1156	2.1282
1499	2.0978	2.0802
2252	2.1621	2.1605
2072	2.1554	2.1531
411	2.1554	2.1637

[879 rows x 2 columns]

In [119…

```
plt.figure(figsize =(35,10))
plt.bar(trainx1.columns,m1_lin_reg.coef_)
plt.title('Linear Regression')
plt.xlabel('Features')
plt.ylabel('Coefficients')
plt.xticks(rotation=90)
plt.show()
```

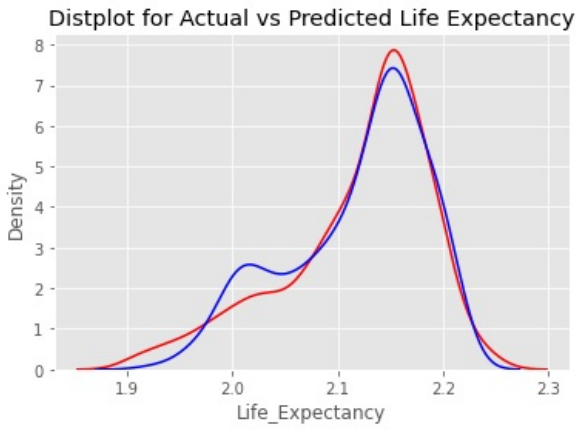


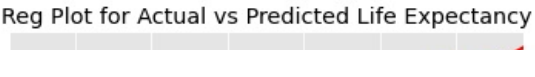
Ridge Regression (l2 regularization) (m1.1)

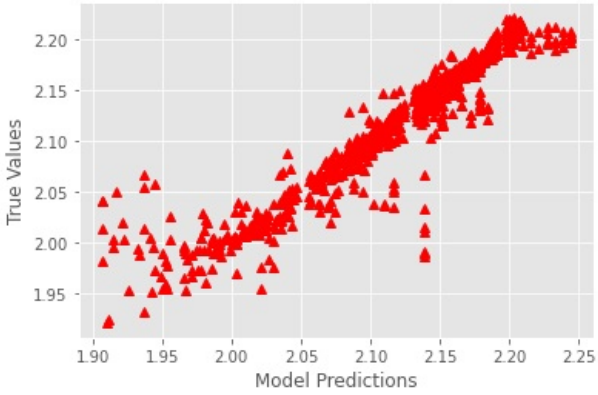
In [120…

```
m1_ridge_reg = Ridge(alpha = 0.1)
m1_ridge_reg.fit(trainx1,trainy1)

mse_train_ridge_reg,mse_test_ridge_reg,r2_train_ridge_reg,r2_test_ridge_reg,y_train_pred,y_test_pred=model_predic

plt.figure(figsize =(35,10))
plt.bar(trainx1.columns,m1_ridge_reg.coef_)
plt.title('Ridge Regression')
plt.xlabel('Features')
plt.ylabel('Coefficients')
plt.xticks(rotation=90)
plt.show()
```

```
Model Train Error
         MSE=0.0004,
        RMSE=0.0191
Model Test Error
         MSE=0.0006,
        RMSE=0.0238
R2 Train : 0.9287 , R2 Test : 0.885
```

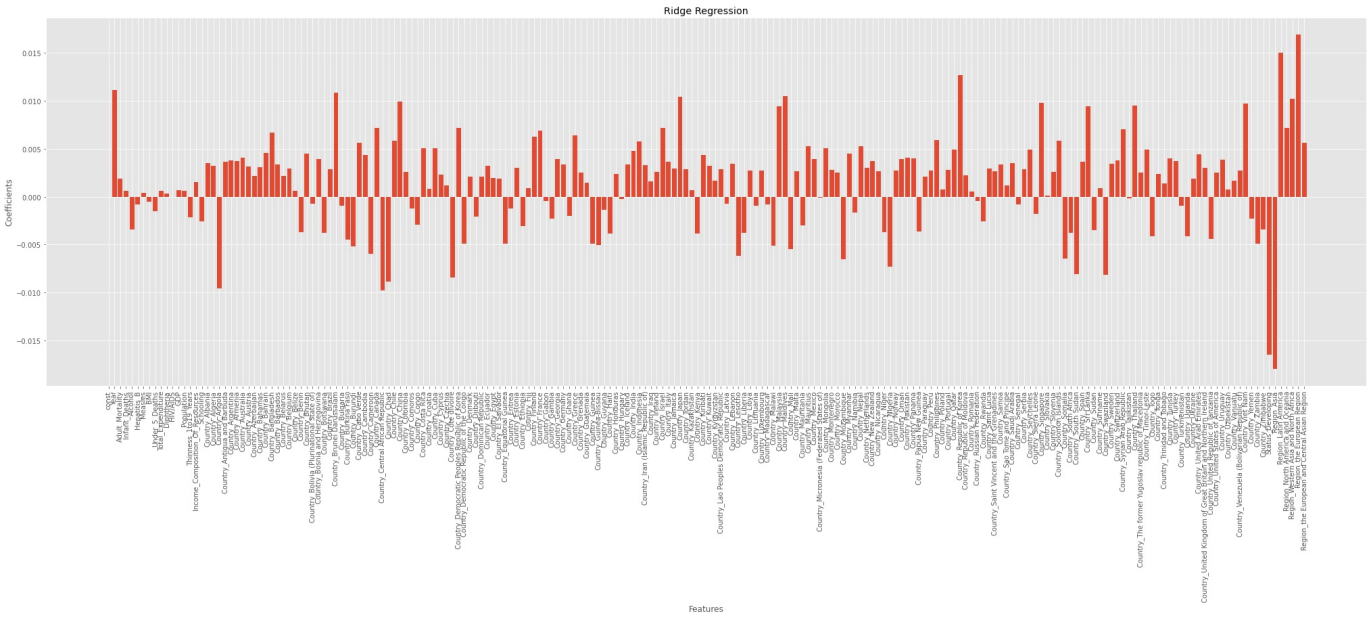






	actual	predicted
2216	2.1327	2.1315
836	2.1764	2.1793
2396	2.0676	2.0814
1962	2.1547	2.1491
305	2.1390	2.0667
...	...	...
1713	2.1156	2.1278
1499	2.0978	2.0801
2252	2.1621	2.1606
2072	2.1554	2.1531
411	2.1554	2.1639

[879 rows x 2 columns]



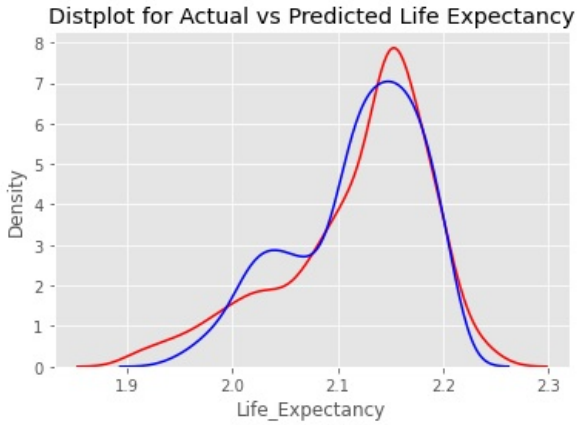
Lasso Regression (l1 regularization) (m1.2)

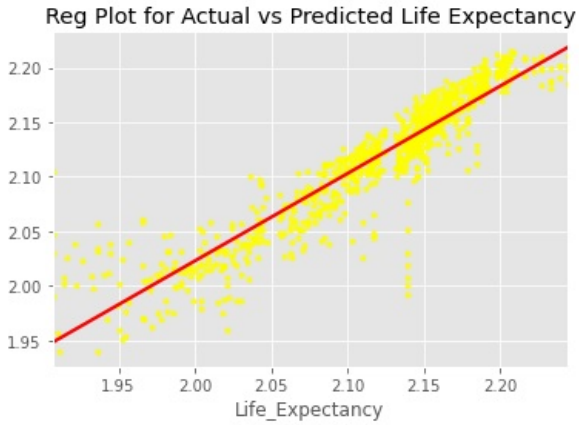
In [121…

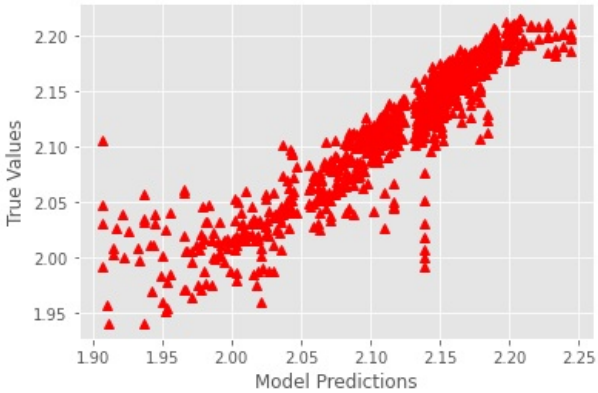
```
m1_lasso_reg = Lasso(alpha = 0.001)
m1_lasso_reg.fit(trainx1,trainy1)

mse_train_lasso_reg,mse_test_lasso_reg,r2_train_lasso_reg,r2_test_lasso_reg,y_train_pred,y_test_pred=model_predic
```

```
plt.figure(figsize =(35,10))
plt.bar(trainx1.columns,m1_lasso_reg.coef_)
plt.title('Lasso Regression')
plt.xlabel('Features')
plt.ylabel('Coefficients')
plt.xticks(rotation=90)
plt.show()
```

```
Model Train Error
         MSE=0.0006,
        RMSE=0.024
Model Test Error
         MSE=0.0007,
        RMSE=0.0273
R2 Train : 0.8868 , R2 Test : 0.8489
```

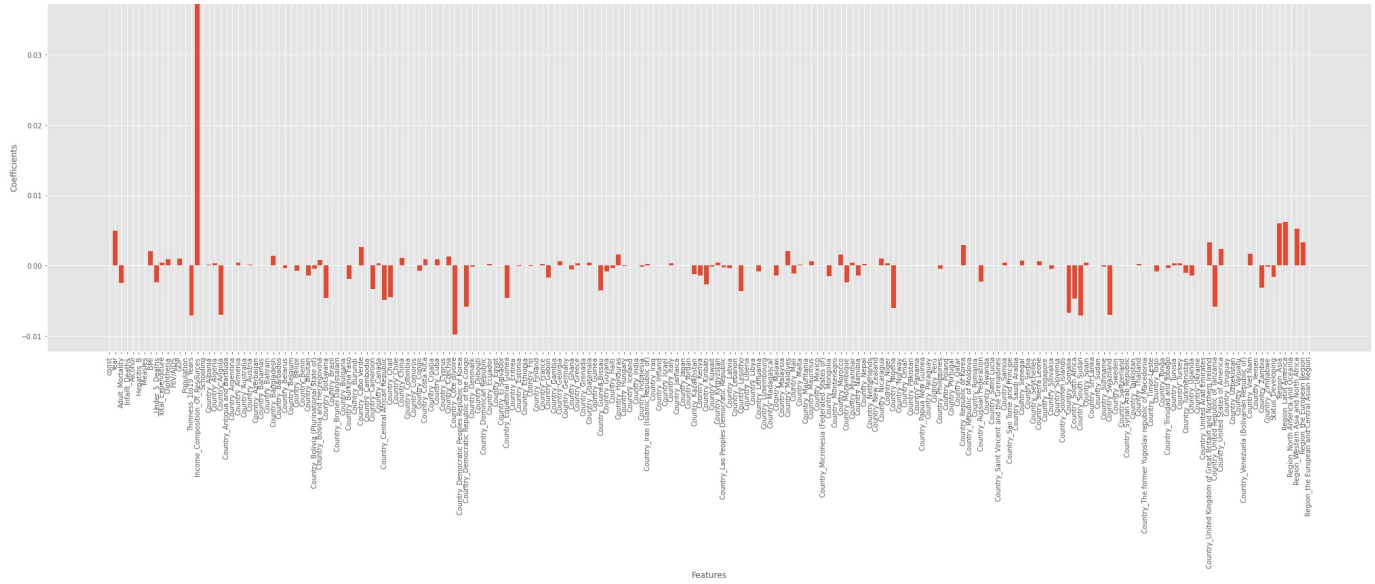






	actual	predicted
2216	2.1327	2.1067
836	2.1764	2.1829
2396	2.0676	2.1035
1962	2.1547	2.1390
305	2.1390	2.0766
...	...	...
1713	2.1156	2.1078
1499	2.0978	2.0665
2252	2.1621	2.1618
2072	2.1554	2.1499
411	2.1554	2.1759

[879 rows x 2 columns]



- OLS REGRESSION/LINEAR REGRESSION GIVES AN OVERFIT MODEL SO WE HAVE TUNED IT THAT IS USED REGULARIZATION METHODS LIKE RIDGE AND LASSO WHICH PENALISES THE ALPHA PARAMETER AND REDUCES THE HIGH VARIANCE IN OUR MODEL BY INTRODUCING SOME BIAS TO IT.

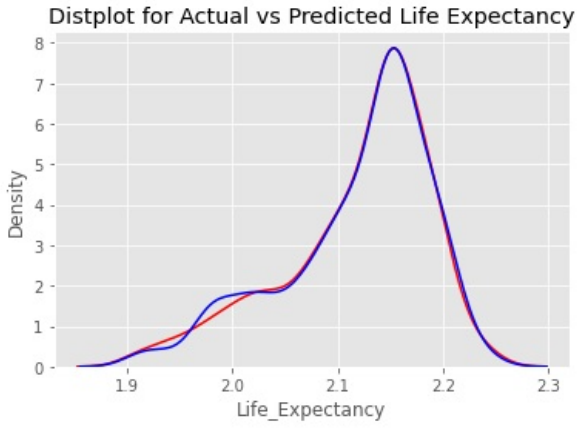
- HERE RIDGE HAS NO EFFECT ON MODEL PERFORMANCE BUT LASSO REDUCES THE VARIANCE BY PENALISING THE COEFFICIENT TO 0.

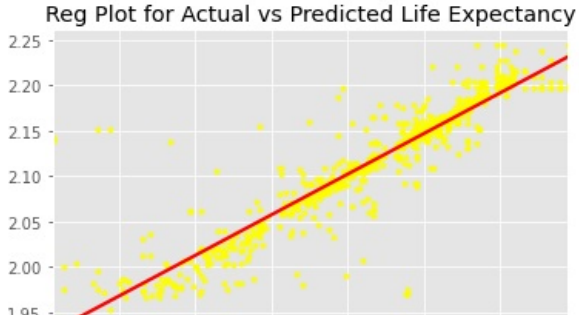
DECISION TREE (m2)

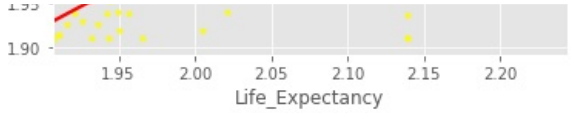
In [122…

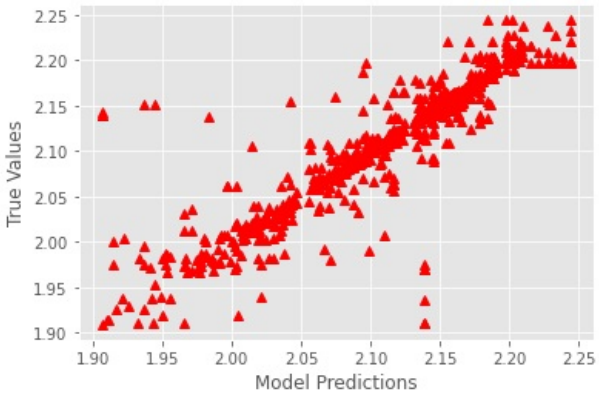
```
m2_dt_reg = DecisionTreeRegressor().fit(trainx1,trainy1)

m2_dt_reg.fit(trainx1,trainy1)

mse_train_dt_reg,mse_test_dt_reg,r2_train_dt_reg,r2_test_dt_reg,y_train_pred,y_test_pred=model_prediction(m2_dt_r
```

```
Model Train Error
         MSE=0.0,
        RMSE=0.0
Model Test Error
         MSE=0.001,
        RMSE=0.0313
R2 Train : 1.0 , R2 Test : 0.8008
```









	actual	predicted
2216	2.1327	2.1061
836	2.1764	2.1726
2396	2.0676	2.0636
1962	2.1547	2.1561
305	2.1390	1.9698
...	...	...
1713	2.1156	2.1199
1499	2.0978	2.1001
2252	2.1621	2.1621
2072	2.1554	2.1520
411	2.1554	2.2213

[879 rows x 2 columns]

- DECISION TREE IS PRONE TO OVERFITTING AS WE CAN OBSERVE HERE

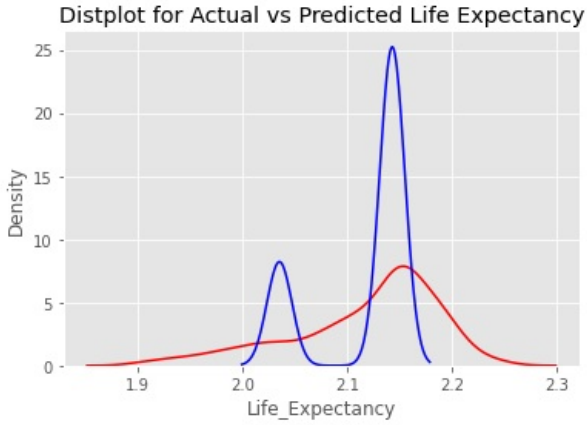
- NEED TO TUNE MODEL FOR REDUCING THE OVERFIT SCENARIO

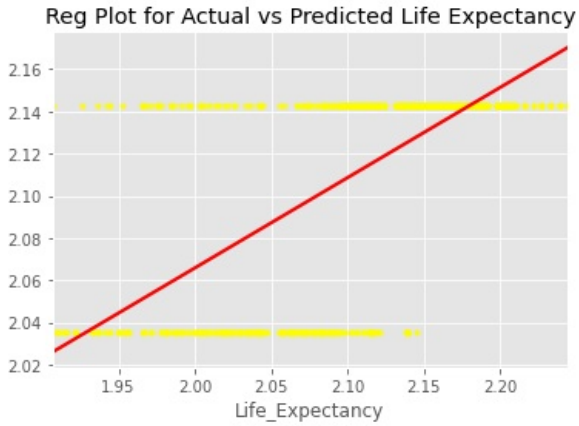
DECISION TREE TUNING (m2.1)

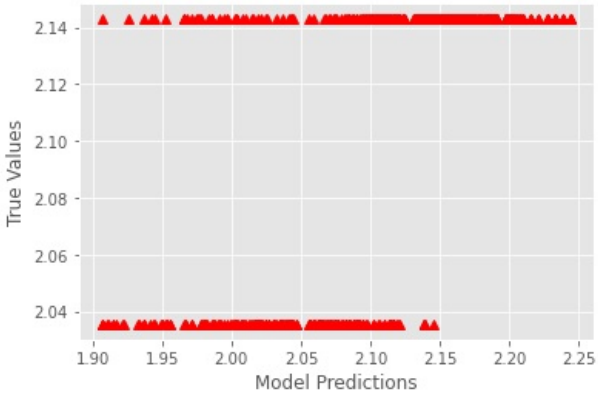
In [123…

```
m2_dt_reg = DecisionTreeRegressor()

param_grid={"splitter":["best","random"],
            "max_depth" : [1,3,5,7,9],
           "min_samples_leaf":[1,2,3,4,5,6,7],
           "max_features":["auto","log2","sqrt",None],
            "min_samples_split":[0.1,3,5,7,9],
            "max_leaf_nodes":[None,10,20,30,40,50]
     }
grid = GridSearchCV(m2_dt_reg,param_grid=param_grid,cv= 5,scoring='f1')
grid.fit(trainx1,trainy1)
#y_train_prob = grid_clf.predict_proba(trainx1)[:,0]
#y_test_prob = grid_clf.predict_proba(testx1)[:,0]


bp = grid.best_params_
print(bp)


m2_dt_reg_ht = DecisionTreeRegressor(splitter= bp["splitter"],
    max_depth=bp['max_depth'],
                                min_samples_leaf=bp['min_samples_leaf'],
                                max_features=bp['max_features']).fit(trainx1,trainy1)

mse_train_dt_reg_ht,mse_test_dt_reg_ht,r2_train_dt_reg_ht,r2_test_dt_reg_ht,y_train_pred,y_test_pred=model_predic
```

```
{'max_depth': 1, 'max_features': 'auto', 'max_leaf_nodes': None, 'min_samples_leaf': 1, 'min_samples_split': 0.1,
'splitter': 'best'}
Model Train Error
         MSE=0.0028,
        RMSE=0.0531
Model Test Error
         MSE=0.0029,
        RMSE=0.0534
R2 Train : 0.4467 , R2 Test : 0.4203
```







	actual	predicted
2216	2.1327	2.1429
836	2.1764	2.1429
2396	2.0676	2.1429
1962	2.1547	2.1429
305	2.1390	2.0355
...	...	...
1713	2.1156	2.1429
1499	2.0978	2.0355
2252	2.1621	2.1429
2072	2.1554	2.1429
411	2.1554	2.1429

[879 rows x 2 columns]

- DECISION TREE TUNING HAS NOT GIVEN ASATISFCATORY PERFORMANCE EVEN AFTER HYPERTUNING

In [124…

```
grid.best_params_,grid.best_estimator_
```

Out[124…

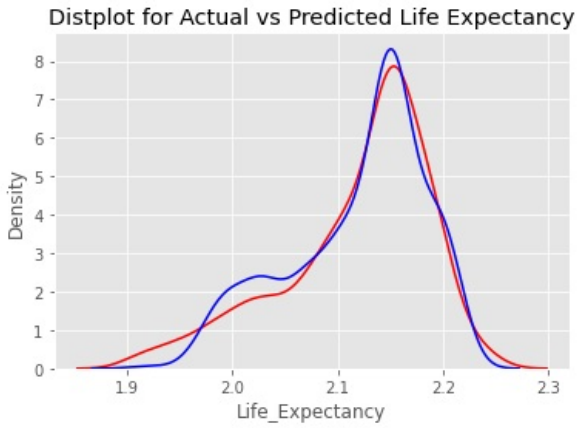
```
({'max_depth': 1,
  'max_features': 'auto',
  'max_leaf_nodes': None,
  'min_samples_leaf': 1,
  'min_samples_split': 0.1,
  'splitter': 'best'},
 DecisionTreeRegressor(max_depth=1, max_features='auto', min_samples_split=0.1))
```

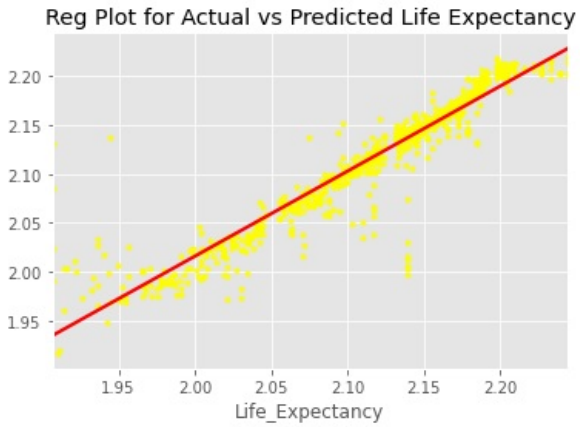
RANDOM FOREST (m3)

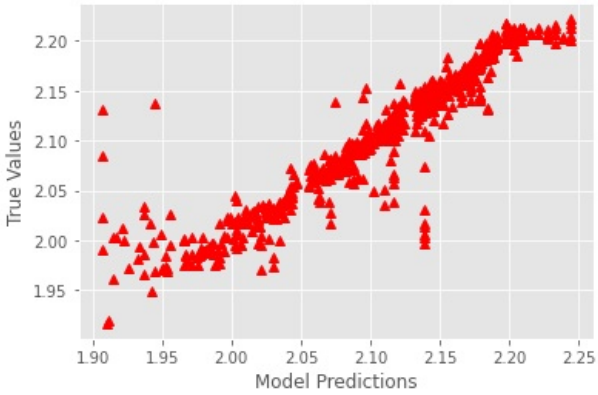
In [125…

```
m3_rf = RandomForestRegressor().fit(trainx1,trainy1)

m3_rf.estimators_

mse_train_rf_reg,mse_test_rf_reg,r2_train_rf_reg,r2_test_rf_reg,y_train_pred,y_test_pred=model_prediction(m3_rf,t
```

```
Model Train Error
         MSE=0.0001,
        RMSE=0.0073
Model Test Error
         MSE=0.0005,
        RMSE=0.0233
R2 Train : 0.9896 , R2 Test : 0.8896
```







	actual	predicted
2216	2.1327	2.1170
836	2.1764	2.1727
2396	2.0676	2.0601
1962	2.1547	2.1512
305	2.1390	2.0302
...	...	...
1713	2.1156	2.1181
1499	2.0978	2.0906
2252	2.1621	2.1590
2072	2.1554	2.1522

```
411   2.1554     2.1832

[879 rows x 2 columns]
```

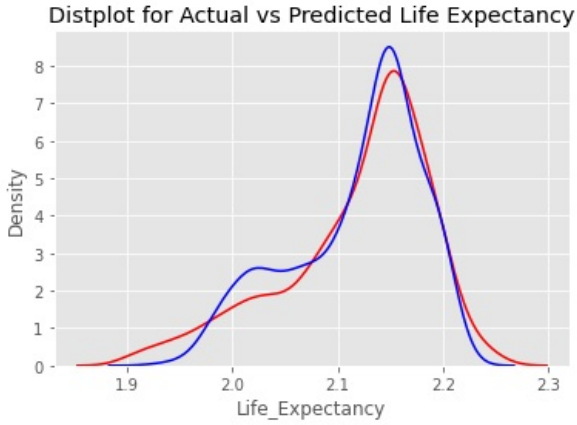
RANDOM FOREST MODEL HAS PERFORMED WELL TO REDUCE THE OVERFITTING WE WILL FURTHER TUNE IT

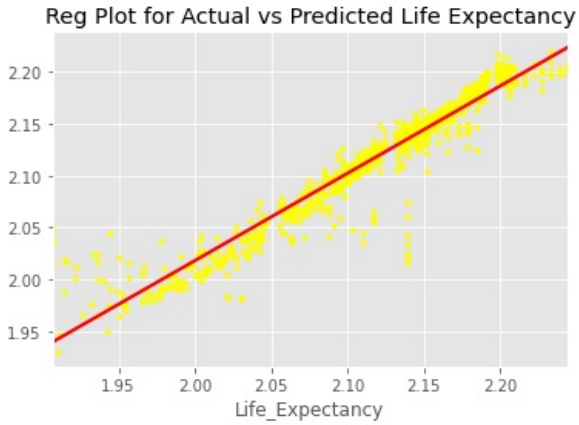
HYPER TUNING RANDOM FOREST (m3.1)

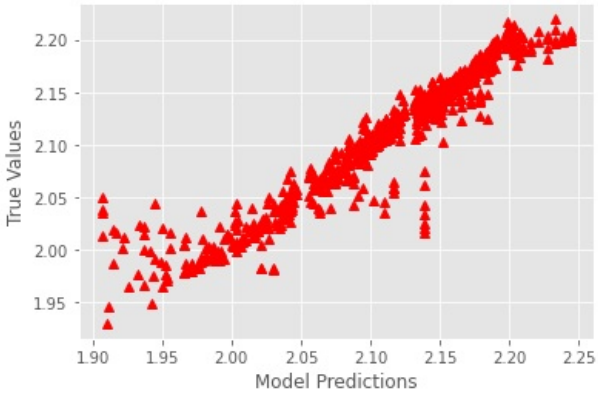
In [126…

```
m3_rf_ht = RandomForestRegressor(n_estimators=500,max_features=2)
m3_rf_ht.fit(trainx1,trainy1)


mse_train_rf_reg_ht,mse_test_rf_reg_ht,r2_train_rf_reg_ht,r2_test_rf_reg_ht,y_train_pred,y_test_pred=model_predic
```

```
Model Train Error
         MSE=0.0001,
        RMSE=0.0073
Model Test Error
         MSE=0.0005,
        RMSE=0.0214
R2 Train : 0.9896 , R2 Test : 0.9072
```








```
      actual  predicted
2216  2.1327     2.1211
836   2.1764     2.1738
2396  2.0676     2.0768
```

```
1962  2.1547     2.1517
305   2.1390     2.0747
...      ...        ...
1713  2.1156     2.1186
1499  2.0978     2.0804
2252  2.1621     2.1530
2072  2.1554     2.1530
411   2.1554     2.1523

[879 rows x 2 columns]
```

- RANDOM FOREST AFTER TUNING CAN BE CONSIDERED AS A GOOD MODEL

Important features

i. Method 1

In [127…

```
feat = pd.DataFrame({'feature':trainx1.columns,
                     'score':m3_rf_ht.feature_importances_})

feat = feat.sort_values('score',ascending=False)
print(feat.head(5))
```

```
                            feature     score
15  Income_Composition_Of_Resources  0.100693
2                   Adult_Mortality  0.078274
16                        Schooling  0.071774
7                               BMI  0.071190
14             Thinness_1to19_Years  0.070111
```

ii. RFE

In [128…

```
rfe = RFE(m3_rf,n_features_to_select=5).fit(trainx1,trainy1)

feat = pd.DataFrame({'feature':trainx1.columns,
                     'support':rfe.support_,
                     'rank':rfe.ranking_ })
feat = feat.sort_values('rank')
print(feat.head(5))
```

```
                            feature  support  rank
2                   Adult_Mortality     True     1
15  Income_Composition_Of_Resources     True     1
4                           Alcohol     True     1
14             Thinness_1to19_Years     True     1
16                        Schooling     True     1
```

Commonly we can say that Adult_Mortality,Income_Composition_Of_Resources,BMI,Schooling,Thinness_1to19_Years these are the highly important features

ADABOOST REGRESSOR (m4)

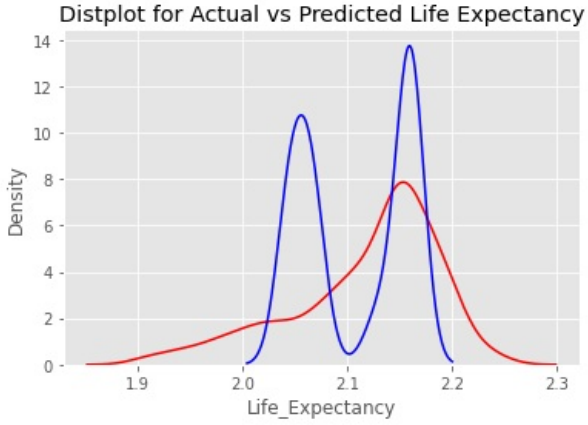
ADABOOST REGRESSOR WITH DECISION TREE AS ITS BASE (m4.1)

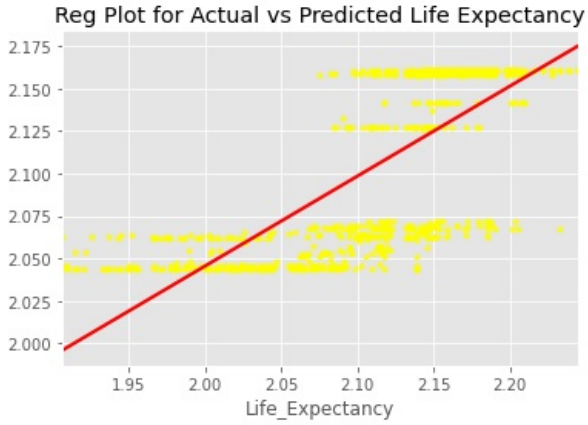
In [129…

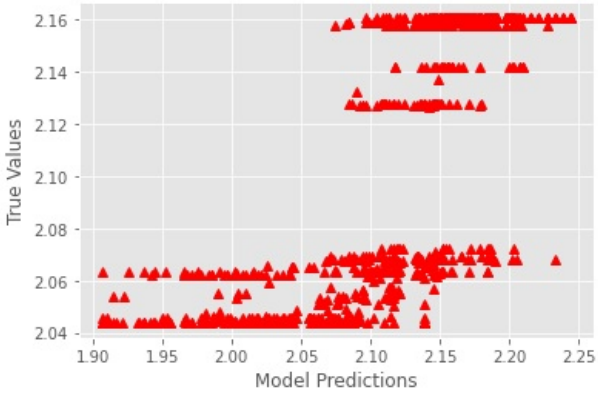
```
m4_ab = AdaBoostRegressor(DecisionTreeRegressor(criterion="mse",max_depth=1,
                                                min_samples_split=0.1,
                                                min_samples_leaf=1,max_features="auto",splitter= 'best'),n_estimator

mse_train_ab_reg,mse_test_ab_reg,r2_train_ab_reg,r2_test_ab_reg,y_train_pred,y_test_pred=model_prediction(m4_ab,t
```

```
Model Train Error
         MSE=0.0024,
        RMSE=0.0487
Model Test Error
         MSE=0.0024,
        RMSE=0.0488
```

```
R2 Train : 0.5341 , R2 Test : 0.5166
```







	actual	predicted
2216	2.1327	2.0620
836	2.1764	2.1607
2396	2.0676	2.0675
1962	2.1547	2.1591
305	2.1390	2.0509
...	...	...
1713	2.1156	2.0633
1499	2.0978	2.0439
2252	2.1621	2.1591
2072	2.1554	2.1607
411	2.1554	2.1591

[879 rows x 2 columns]

- DECISION TREE INDIVIDUALLY HAD A VERY DISSATISFACTORY PERFORMACE

- ADABOOST HAS NOT SHOWN ANY MAJOR IMPROVEMENT IN PERFORMANCE USING DECISION TREE AS ITS BASE MODEL

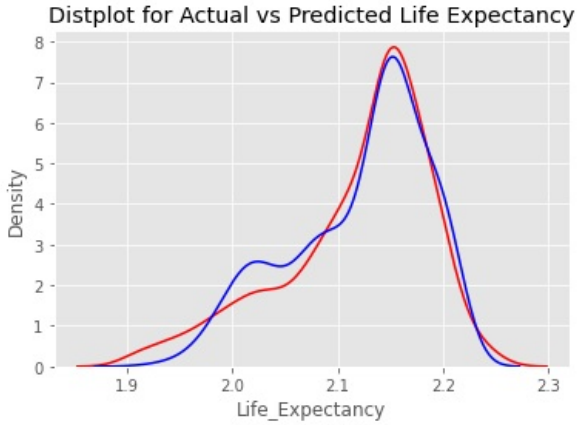
- HENCE WE CAN REMOVE DECISION TREE AND ADABOOST WITH DECISION TREE MODELS FROM OUR EVALAUTION

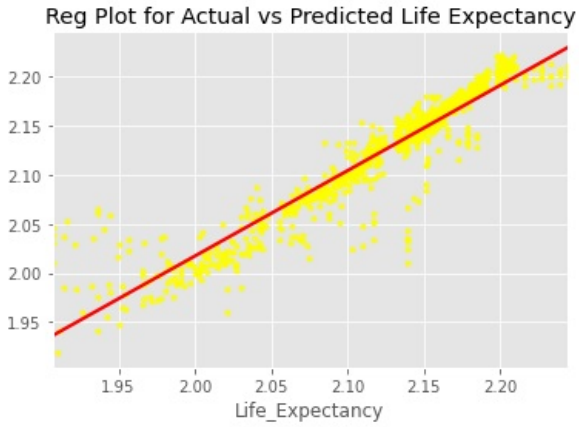
- ADABOOST REGRESSOR WITH LINEAR REGRESSION AS ITS BASE (m4.2)

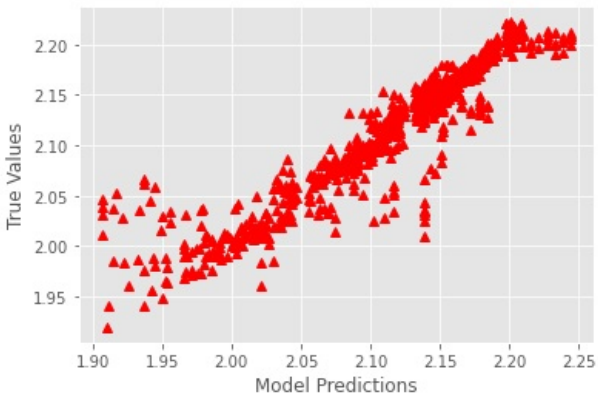
In [173…

```
m4_ab_lin = AdaBoostRegressor(LinearRegression(fit_intercept= True),n_estimators=500).fit(trainx1,trainy1)

mse_train_ab_reg_lin,mse_test_ab_reg_lin,r2_train_ab_reg_lin,r2_test_ab_reg_lin,y_train_pred,y_test_pred = model_
```

```
Model Train Error
         MSE=0.0004,
        RMSE=0.0205
Model Test Error
         MSE=0.0006,
        RMSE=0.0245
R2 Train : 0.9179 , R2 Test : 0.8784
```







	actual	predicted
2216	2.1327	2.1316
836	2.1764	2.1763
2396	2.0676	2.0813
1962	2.1547	2.1482
305	2.1390	2.0669
...	...	...
1713	2.1156	2.1411
1499	2.0978	2.0821
2252	2.1621	2.1692
2072	2.1554	2.1535
411	2.1554	2.1793

[879 rows x 2 columns]

GRADIENT BOOST REGRESSOR (m5)

GRADIENT BOOST REGRESSOR (m5)

In [131…

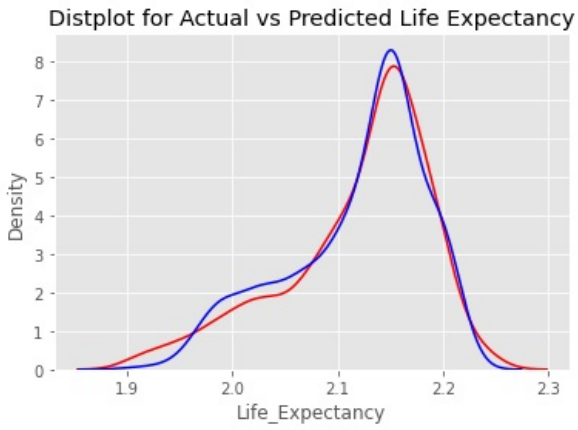
```
params = {'n_estimators':[50,100,150],
          'learning_rate':[0.005,0.05],
          'max_depth':[3,4,5,6,7],
          'subsample':[0.25,0.5],
         'min_samples_split':[0.1,0.2] }

m5_bg_reg = GradientBoostingRegressor()

grid = GridSearchCV(m5_bg_reg,param_grid=params,cv=5,n_jobs=-1,
                    scoring='r2').fit(trainx1,trainy1)

bp = grid.best_params_
print(bp)
```

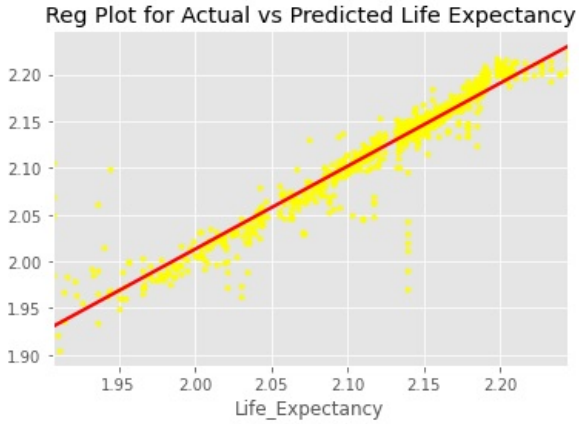
```
{'learning_rate': 0.05, 'max_depth': 7, 'min_samples_split': 0.1, 'n_estimators': 150, 'subsample': 0.5}
```

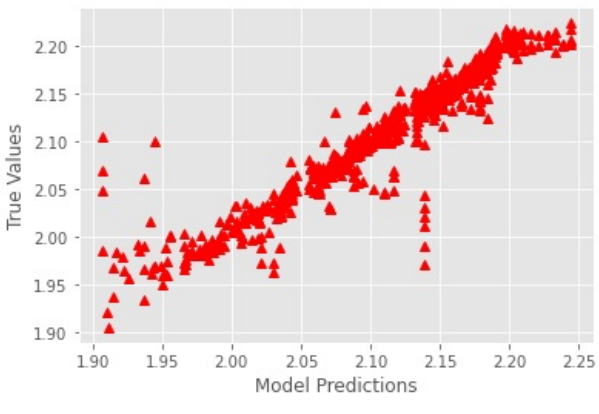
In [132…

```
m5_gb_reg_ht = GradientBoostingRegressor(n_estimators=bp['n_estimators'],
                                learning_rate=bp['learning_rate'],
                                max_depth=bp['max_depth'],
                                subsample=bp['subsample']).fit(trainx1,trainy1)

mse_train_gb_reg_ht,mse_test_gb_reg_ht,r2_train_gb_reg_ht,r2_test_gb_reg_ht,y_train_pred,y_test_pred=model_predic
```

```
Model Train Error
         MSE=0.0001,
        RMSE=0.008
Model Test Error
         MSE=0.0005,
        RMSE=0.022
R2 Train : 0.9874 , R2 Test : 0.9015
```







```
      actual  predicted
2216  2.1327     2.1100
836   2.1764     2.1726
2396  2.0676     2.0559
1962  2.1547     2.1468
305   2.1390     2.0432
...      ...        ...
1713  2.1156     2.1195
1499  2.0978     2.0907
2252  2.1621     2.1607
2072  2.1554     2.1540
411   2.1554     2.1841

[879 rows x 2 columns]
```

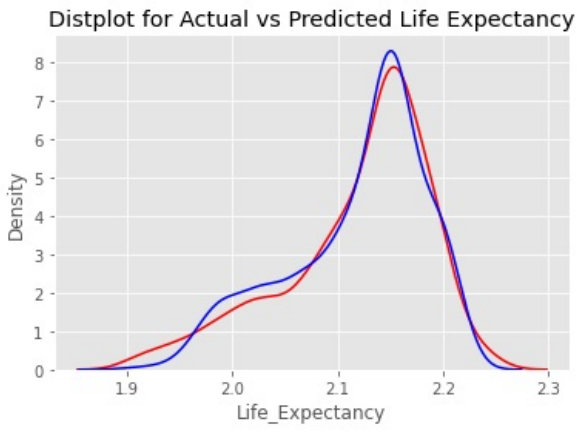
- GRADIENT BOOST HAS PERFORMED VERY WELL IN TERMS OF ADABOOST (LINEAR REGRESSION MODEL)

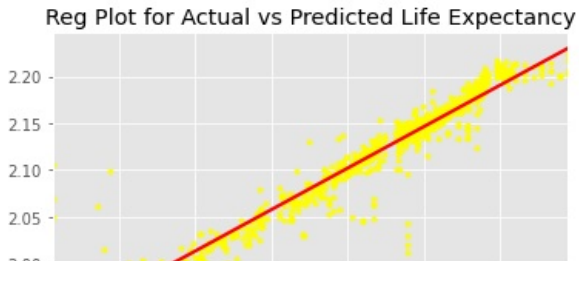
XGBOOST REGRESSOR (m6)

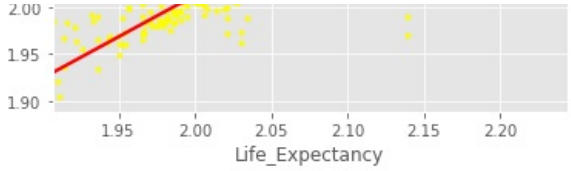
In [133…

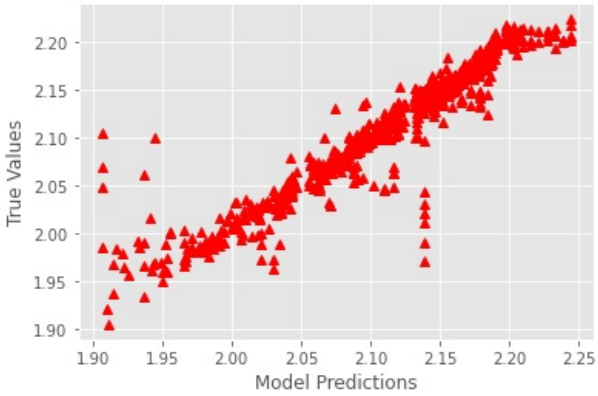
```
params = {'eta':[0.1,0.3,0.5,0.8],
          'max_depth':[4,5,6,7],
          'subsample':[0.5,0.75,0.95]
          # 'base_score':[0.45,0.5,0.55]
          }
          # 'learning_rate':[0.001,0.03,0.15] }

m6_xgb_reg = XGBRegressor()
grid = GridSearchCV(m6_xgb_reg,param_grid=params,scoring='r2',
                    n_jobs=-1,cv=5).fit(trainx1,trainy1)

bp = grid.best_params_; bp

m6_xgb_reg_ht = XGBRegressor(booster='gbtree',objective='reg:squarederror',
                   eval_metric='error', eta=bp['eta'],
                   max_depth=bp['max_depth'],
                   subsample=bp['subsample']).fit(trainx1,trainy1)

mse_train_xgb_reg_ht,mse_test_xgb_reg_ht,r2_train_xgb_reg_ht,r2_test_xgb_reg_ht,y_train_pred,y_test_pred=model_pr
```

```
Model Train Error
         MSE=0.0001,
        RMSE=0.008
Model Test Error
         MSE=0.0005,
        RMSE=0.022
R2 Train : 0.9874 , R2 Test : 0.9015
```









	actual	predicted
2216	2.1327	2.1100
836	2.1764	2.1726
2396	2.0676	2.0559
1962	2.1547	2.1468
305	2.1390	2.0432
...	...	...
1713	2.1156	2.1195
1499	2.0978	2.0907
2252	2.1621	2.1607
2072	2.1554	2.1540
411	2.1554	2.1841

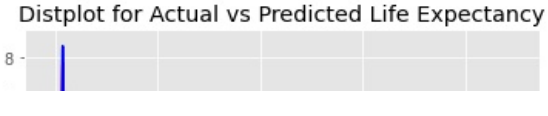
[879 rows x 2 columns]

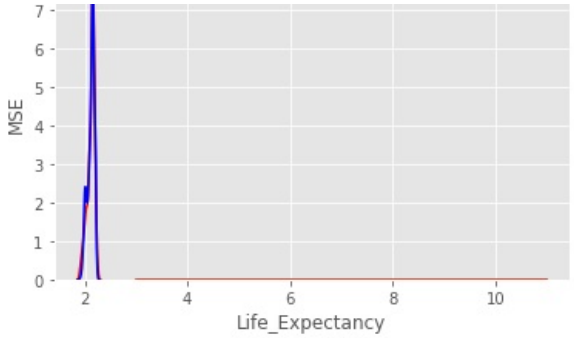
KNN REGRESSOR (m7)

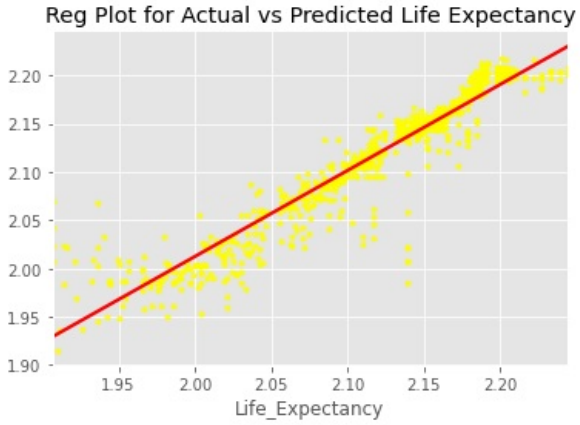
In [176…

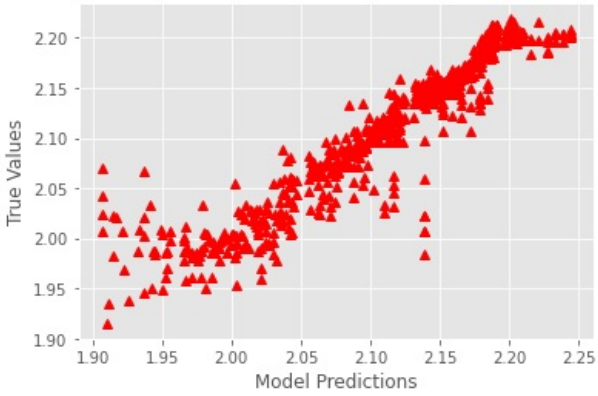
```
cv_mse = []
list_n = np.arange(3,12); list_n

for n in list_n:
    model=neighbors.KNeighborsRegressor(n_neighbors=n).fit(trainx1,trainy1)
    pred=model.predict(testx1)
    cv_mse.append(mean_squared_error(testy1,pred))

print(cv_mse)

bestK = list_n[cv_mse.index(min(cv_mse))]
print("best K = ",bestK)

# plot neighbours - Errors to determine best K
plt.plot(list_n,cv_mse)
plt.xlabel("Neighbours")
plt.ylabel("MSE")
plt.title("MSE - Neighbours")

# build and predict using the best K
m7_knn_reg = neighbors.KNeighborsRegressor(n_neighbors=bestK).fit(trainx1,trainy1)

mse_train_knn_reg,mse_test_knn_reg,r2_train_knn_reg,r2_test_knn_reg,y_train_pred,y_test_pred=model_prediction(m7_
```

```
[0.0005855200463595223, 0.000576151791862008, 0.0005485406448805298, 0.0005772160009326681, 0.0005984595614741194
, 0.0006326662616890531, 0.0006880213798937464, 0.0007697246388242426, 0.0008870950548610714]
best K =  5
Model Train Error
         MSE=0.0003,
        RMSE=0.0168
Model Test Error
         MSE=0.0005,
        RMSE=0.0234
R2 Train : 0.9445 , R2 Test : 0.8885
```









	actual	predicted
2216	2.1327	2.1419
836	2.1764	2.1712
2396	2.0676	2.0956
1962	2.1547	2.1561
305	2.1390	2.0590
...	...	...
1713	2.1156	2.1227
1499	2.0978	2.0735
2252	2.1621	2.1527
2072	2.1554	2.1570
411	2.1554	2.1501

[879 rows x 2 columns]

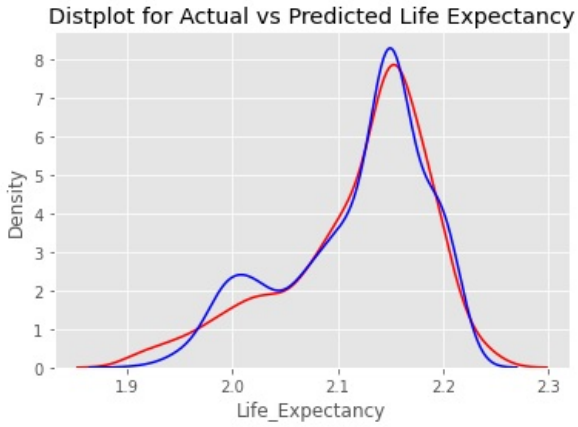
SVM REGRESSOR (m8)

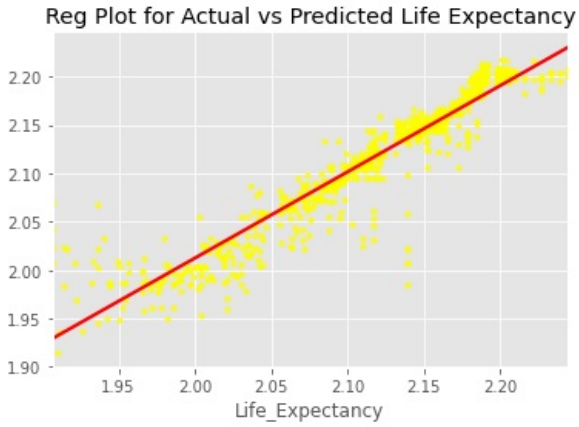
In [177…

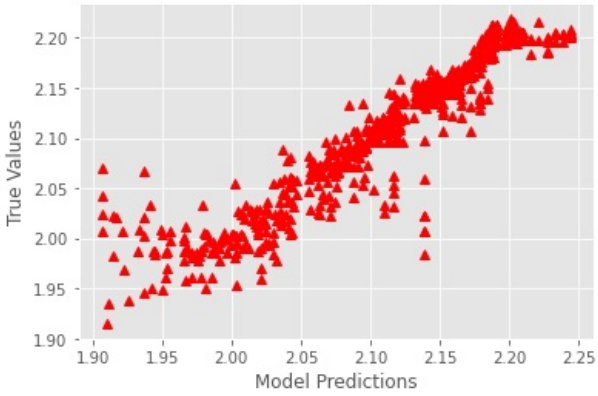
```
lim=6
lov_c = np.logspace(-3,2,lim)
lov_g = np.random.random(lim)

params = [{ 'kernel':['linear'], 'C':lov_c,
            'kernel':['sigmoid'],'C':lov_c,'gamma':lov_g,
            'kernel':['poly'],'C':lov_c,'gamma':lov_g,
            'kernel':['rbf'],'C':lov_c,'gamma':lov_g}]

m8_svm_reg = svm.SVR()

grid = GridSearchCV(m8_svm_reg,param_grid=params,
                    scoring='neg_mean_squared_error',
```

```
                    cv=3,n_jobs=-1).fit(trainx1,trainy1)

bp = grid.best_params_

m8_svm_reg_ht = svm.SVR(kernel = bp["kernel"],C=bp["C"],gamma=bp["gamma"]).fit(trainx1,trainy1)

mse_train_svm_reg_ht,mse_test_svm_reg_ht,r2_train_svm_reg_ht,r2_test_svm_reg_ht,y_train_pred,y_test_pred=model_pr
```

```
Model Train Error
         MSE=0.0003,
        RMSE=0.0168
Model Test Error
         MSE=0.0005,
        RMSE=0.0234
R2 Train : 0.9445 , R2 Test : 0.8885
```







	actual	predicted
2216	2.1327	2.1419
836	2.1764	2.1712
2396	2.0676	2.0956
1962	2.1547	2.1561
305	2.1390	2.0590
...	...	...
1713	2.1156	2.1227
1499	2.0978	2.0735
2252	2.1621	2.1527
2072	2.1554	2.1570
411	2.1554	2.1501

[879 rows x 2 columns]

SVM AND KNN ALMOST GIVES SIMILAR PERFORMANCE

RESULT OF ALL MODELS

In [178…
```
result = pd.DataFrame({"Models":["Linear Regression ols(m1)","Linear Regression(m1)","Ridge Regression(m1.1)","La
                       "MSE":[mse_test_ols,mse_test_lin_reg,mse_test_ridge_reg,mse_test_lasso_reg,mse_test_dt_reg
                       "RMSE":[np.sqrt(mse_test_ols),np.sqrt(mse_test_lin_reg),np.sqrt(mse_test_ridge_reg),np.sqr
                       "R2 SQUARE":[r2_test_ols,r2_test_lin_reg,r2_test_ridge_reg,r2_test_lasso_reg,r2_test_dt_re
```

In [179…
```
result
```

Out[179…

	Models	MSE	RMSE	R2 SQUARE
0	Linear Regression ols(m1)	0.000566	0.023784	0.885053
1	Linear Regression(m1)	0.000565	0.023774	0.885158
2	Ridge Regression(m1.1)	0.000566	0.023785	0.885046
3	Lasso Regression(m1.2)	0.000744	0.027271	0.848883
4	Decision Tree(m2)	0.000980	0.031311	0.800790
5	Decision Tree Tuned(m2.1)	0.002853	0.053411	0.420332
6	Random Forest(m3)	0.000543	0.023311	0.889580
7	Random Forest Tuned (m3.1)	0.000457	0.021376	0.907152
8	Adaboost Regressor with Decision Tree(m4.1)	0.002379	0.048776	0.516577
9	Adaboost Regressor with Linear Regression(m4.2)	0.000598	0.024463	0.878405
10	GradientBoost Regressor(m5)	0.000485	0.022019	0.901487
11	XGBoost Regressor(m6)	0.000485	0.022019	0.901487
12	KNN Regressor(m7)	0.000549	0.023421	0.888540
13	SVM Regressor(m8)	0.000549	0.023421	0.888540

CONCLUSION

i. METRICS EVALUATION -

LIFE EXPECTANCY PREDICTION NEEDS TO BE TOWARDS HIGH ACCURACY AS IT A CRUCIAL FACTOR

R SQUARE IS USED FOR FINDING ACCURACY OF MODEL IT DEPICTS THE CLOSENESS OF THE DATA POINTS TO TREND LINE MADE BY THE MODEL.

THIS HELPS TO MAKE A LINK BETWEEN THE INDEPENDENT AND TARGET VARIABLES

MSE FOCUSES ON LARGER ERRORS,AS WHEN WE ARE SQUARING THE ERROR THE EFFECT OF LARGE ERRORS BECOMES MORE PROMINENT

MY MODEL HAVE NO LARGE ERRORS

CAN USE RMSE INSTEAD

CONSIDERING ABOVE POINTS I PREFER TO CHOOSE R SQUARE FACTOR FOR MY MODEL ALONG WITH RMSE

ii. MODEL SELECTION CRITERIA AND MODEL PERFORMANCE EVALUATION -

AMONG THE ABOVE MODELS THE BEST ACCURATE RESULT IS GIVEN BY BOOSTING MODELS (GRADIENT BOOST AND XGBOOST)

ALSO I CHOOSE LINEAR REGRESSION MODEL AMONG REST MODELS APART FROM BOOSTING

REASON BEING ALL GIVE RESULT ON A SIMILAR BASIS SO OCAM RAZOR PRINCIPLE SAYS THAT CHOOSE THE SIMPLEST ONE AMONG ALL

THIS WAY LIFE EXPECTANCY PREDICTION CAN FURTHER BE TESTED ON UNSEEN DATA USING THESE BEST MODEL

iii. FUTURE SCOPE-

- Life expectancy prediction can be very useful when examining health metrics between different populations that have varying underlying risk profiles

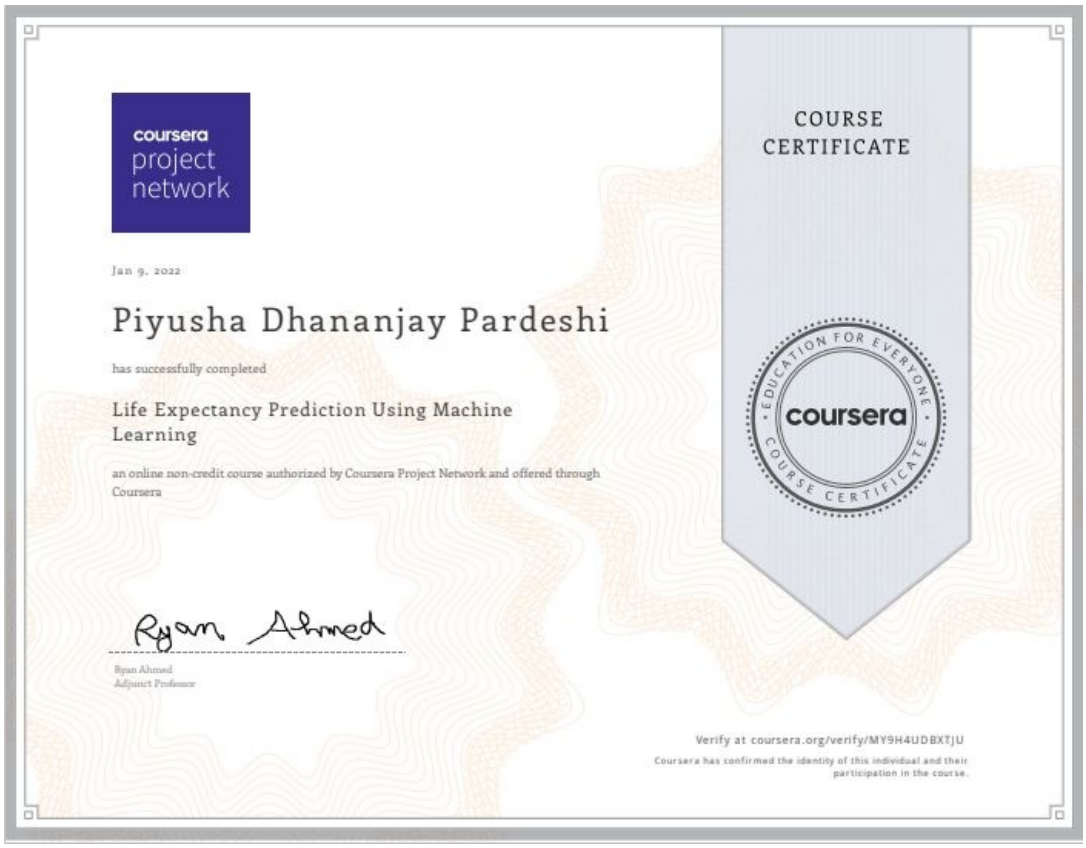
- It can be used in hospitals to predict a new born babies lifespan

- Also various medical fields and health care centers can make use of these factors

- Countries mortality rate can reduced by improving the vaccinations ,doses and many other factors affecting the mortality rates.

- Various factors such as gender, pollution level in region,covid,heart disease,smokeing habbits can all be added in the features to increase overall accuracy of prediction of data

iv .COURSERA PROJECT COMPLETION CERTIFICATION



v.PURPOSE -

I ACHEIVED THIS COURSERA LIFE EXPECTANCY PREDICTION USING MACHINE LEARNING CERTIFICATE INORDER TO IMPROVE MY SKILLS IN BUILDING A REAL TIME PROBLEM MODEL AND LEARN NEW THINGS

COURSERA CERTIFICATION WILL HELP ME IN SHOW CASING MY PROJECT CONTRIBUTION IN ALL MY INTERVIEWS

In [139…

```
from IPython.display import Image
Image(url='https://thumbs.gfycat.com/AccomplishedBelatedGarpike-small.gif')
```

Out[139…