Regression Modeling

Defining the Problem Statement

- ### Problem Statement:

"To predict average life expectancy across populations using a multiple linear regression model based on health, economic, and demographic indicators."

- ### Project Aim:

The aim of this project is to build a multiple linear regression model to predict average life expectancy using selected health, demographic, and economic indicators. The model will identify key factors influencing life expectancy and evaluate prediction accuracy using standard performance metrics.

- **Dependent Variable (Target):** Life expectancy
- **Independent variables (After Selecting significant variables):** The following variables are considered as potential predictors for life expectancy:
- ' Adult Mortality',
- 'Alcohol',
- 'percentage expenditure',
- 'Hepatitis B',
- 'Measles',
- 'Polio',
- 'Diphtheria',
- 'HIV/AIDS',
- 'thinness 5-9 years',
- 'Income composition of resources',
- 'Schooling'

Dataset Collection and Overview: Understanding Data Characteristics and Context.

- ### Data Collection

The dataset for this project was collected from Kaggle, a popular platform for sharing datasets related to various domains, including machine learning, data analysis, and artificial intelligence. Kaggle provides access to high-quality datasets contributed by users, researchers, and organizations, making it an invaluable resource for data-driven projects.

Source:

- Kaggle Dataset URL: https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who
- Kaggle Profile/Account:
- This dataset is open-source and freely available to the public, provided that the user abides by Kaggle's terms of service and dataset license.

- ### Data Description:

This project uses health-related data for 193 countries collected from the Global Health Observatory (GHO) under the World Health Organization (WHO). The dataset spans from 2000 to 2015 and includes life expectancy, health, and economic data. The economic data was sourced from the United Nations (UN).

- ### Dataset Structure:

	A	B	C	D	E	F	G	H	I	J	K	L
1	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths
2	Afghanistan	2015	Developing	65	263	62	0.01	71.27962362	65	1154	19.1	83
3	Afghanistan	2014	Developing	59.9	271	64	0.01	73.52358168	62	492	18.6	86
4	Afghanistan	2013	Developing	59.9	268	66	0.01	73.21924272	64	430	18.1	89
5	Afghanistan	2012	Developing	59.5	272	69	0.01	78.1842153	67	2787	17.6	93
6	Afghanistan	2011	Developing	59.2	275	71	0.01	7.097108703	68	3013	17.2	97
7	Afghanistan	2010	Developing	58.8	279	74	0.01	79.67936736	66	1989	16.7	102
8	Afghanistan	2009	Developing	58.6	281	77	0.01	56.76221682	63	2861	16.2	106
9	Afghanistan	2008	Developing	58.1	287	80	0.03	25.87392536	64	1599	15.7	110
10	Afghanistan	2007	Developing	57.5	295	82	0.02	10.91015598	63	1141	15.2	113
11	Afghanistan	2006	Developing	57.3	295	84	0.03	17.17151751	64	1990	14.7	116
12	Afghanistan	2005	Developing	57.3	291	85	0.02	1.388647732	66	1296	14.2	118
13	Afghanistan	2004	Developing	57	293	87	0.02	15.29606643	67	466	13.8	120
14	Afghanistan	2003	Developing	56.7	295	87	0.01	11.08905273	65	798	13.4	122
15	Afghanistan	2002	Developing	56.2	3	88	0.01	16.88735091	64	2486	13	122
16	Afghanistan	2001	Developing	55.3	316	88	0.01	10.5747282	63	8762	12.6	122
17	Afghanistan	2000	Developing	54.8	321	88	0.01	10.42496	62	6532	12.2	122
18	Albania	2015	Developing	77.8	74	0	4.6	364.9752287	99	0	58	0
19	Albania	2014	Developing	77.5	8	0	4.51	428.7490668	98	0	57.2	1
20	Albania	2013	Developing	77.2	84	0	4.76	430.8769785	99	0	56.5	1
21	Albania	2012	Developing	76.9	86	0	5.14	412.4433563	99	9	55.8	1

Life Expectancy Data | Sheet1

The dataset consists of 2938 records (rows) and 22 variables (columns).

- **Columns Description:**

Columns	Description
Country	Country 193 unique values
Year	Year
Status	Developed or Developing status
Life expectancy	Life Expectancy in age
Adult Mortality	Adult Mortality Rates of both sexes (probability of dying between 15 and 60 years per 1000 population)
Infant deaths	Number of Infant Deaths per 1000 population
Alcohol	Alcohol, recorded per capita (15+) consumption (in litres of pure alcohol)
Percentage expenditure	Expenditure on health as a percentage of Gross Domestic Product per capita(%)
Hepatitis B	Hepatitis B (hepb) immunization coverage among 1-year-olds (%)
Measles	Measles - number of reported cases per 1000 population
BMI	Average Body Mass Index of entire population
Under-five deaths	Number of under-five deaths per 1000 population
Polio	Polio (Pol3) immunization coverage among 1-year-olds (%)
Total expenditure	General government expenditure on health as a percentage of total government expenditure (%)
Diphtheria	Diphtheria tetanus toxoid and pertussis (DTP3) immunization coverage among 1-year-olds (%)
HIV/AIDS	Deaths per 1 000 live births HIV/AIDS (0-4 years)
GDP	Gross Domestic Product per capita (in USD)
Population	Population of the country
Thinness 10-19 years	Prevalence of thinness among children and adolescents for Age 10 to 19 (%)
Thinness 5-9 years	Prevalence of thinness among children for Age 5 to 9(%)
Income composition of resources	Human Development Index in terms of income composition of resources (index ranging from 0 to 1)
Schooling	Number of years of Schooling(years)

- **Nature of each variable:**

Variable Name	Data Type
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-five deaths	Int64
Polio	Float64
Total expenditure	Float64
Diphtheria	Float64
HIV/AIDS	Float64
GDP	Float64
Population	Float64
Thinness 10–19 years	Float64
Thinness 5–9 years	Float64
Income composition of resources	Float64

Schooling	Float64

- ## Context of the Data:

The dataset explores factors influencing life expectancy across 193 countries from 2000 to 2015, focusing on variables like immunization rates, mortality, economic indicators (GDP), and social factors. It aims to identify key drivers of life expectancy improvements and guide policy interventions, especially in underdeveloped regions, by analyzing the impact of health and socio-economic factors.

Perform Exploratory Data Analysis (EDA)

- ## Summary Statistics:

```
[274]: print(data.describe())

              Year  Life expectancy  Adult Mortality  infant deaths  \
count  2938.000000      2928.000000      2928.000000    2938.000000
mean   2007.518720        69.224932       164.796448      30.303948
std       4.613841         9.523867       124.292079     117.926501
min    2000.000000        36.300000         1.000000       0.000000
25%    2004.000000        63.100000        74.000000       0.000000
50%    2008.000000        72.100000       144.000000       3.000000
75%    2012.000000        75.700000       228.000000      22.000000
max    2015.000000        89.000000       723.000000    1800.000000

           Alcohol  percentage expenditure  Hepatitis B       Measles  \
count  2744.000000             2938.000000  2385.000000    2938.000000
mean      4.602861              738.251295    80.940461    2419.592240
std       4.052413             1987.914858    25.070016   11467.272489
min       0.010000                0.000000     1.000000       0.000000
25%       0.877500                4.685343    77.000000       0.000000
50%       3.755000               64.912906    92.000000      17.000000
75%       7.702500              441.534144    97.000000     360.250000
max      17.870000            19479.911610    99.000000  212183.000000


               BMI  under-five deaths        Polio  Total expenditure  \
count  2904.000000        2938.000000  2919.000000         2712.00000
mean     38.321247          42.035739    82.550188            5.93819
std      20.044034         160.445548    23.428046            2.49832
min       1.000000           0.000000     3.000000            0.37000
25%      19.300000           0.000000    78.000000            4.26000
50%      43.500000           4.000000    93.000000            5.75500
75%      56.200000          28.000000    97.000000            7.49250
max      87.300000        2500.000000    99.000000           17.60000

        Diphtheria     HIV/AIDS            GDP    Population  \
count  2919.000000  2938.000000    2490.000000  2.286000e+03
mean     82.324084     1.742103    7483.158469  1.275338e+07
std      23.716912     5.077785   14270.169342  6.101210e+07
min       2.000000     0.100000       1.681350  3.400000e+01
25%      78.000000     0.100000     463.935626  1.957932e+05
50%      93.000000     0.100000    1766.947595  1.386542e+06
75%      97.000000     0.800000    5910.806335  7.420359e+06
max      99.000000    50.600000  119172.741800  1.293859e+09
```

```
        thinness  10-19 years  thinness 5-9 years  \
count            2904.000000         2904.000000
mean                4.839704            4.870317
std                 4.420195            4.508882
min                 0.100000            0.100000
25%                 1.600000            1.500000
50%                 3.300000            3.300000
75%                 7.200000            7.200000
max                27.700000           28.600000

        Income composition of resources    Schooling
count                       2771.000000  2775.000000
mean                           0.627551    11.992793
std                            0.210904     3.358920
min                            0.000000     0.000000
25%                            0.493000    10.100000
50%                            0.677000    12.300000
75%                            0.779000    14.300000
max                            0.948000    20.700000
```

- **Identification of missing values:**

[208]:
```python
import matplotlib.pyplot as plt
import seaborn as sns

# Now you can proceed with the visualization
plt.figure(figsize=(10, 8))
sns.heatmap(data.isnull(), cbar=False, cmap='viridis')
print(data.isnull().sum())
plt.title('Missing Values Heatmap')
plt.show()
```

```
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  10-19 years               34
thinness 5-9 years                  34
Income composition of resources    167
Schooling                          163
dtype: int64
```

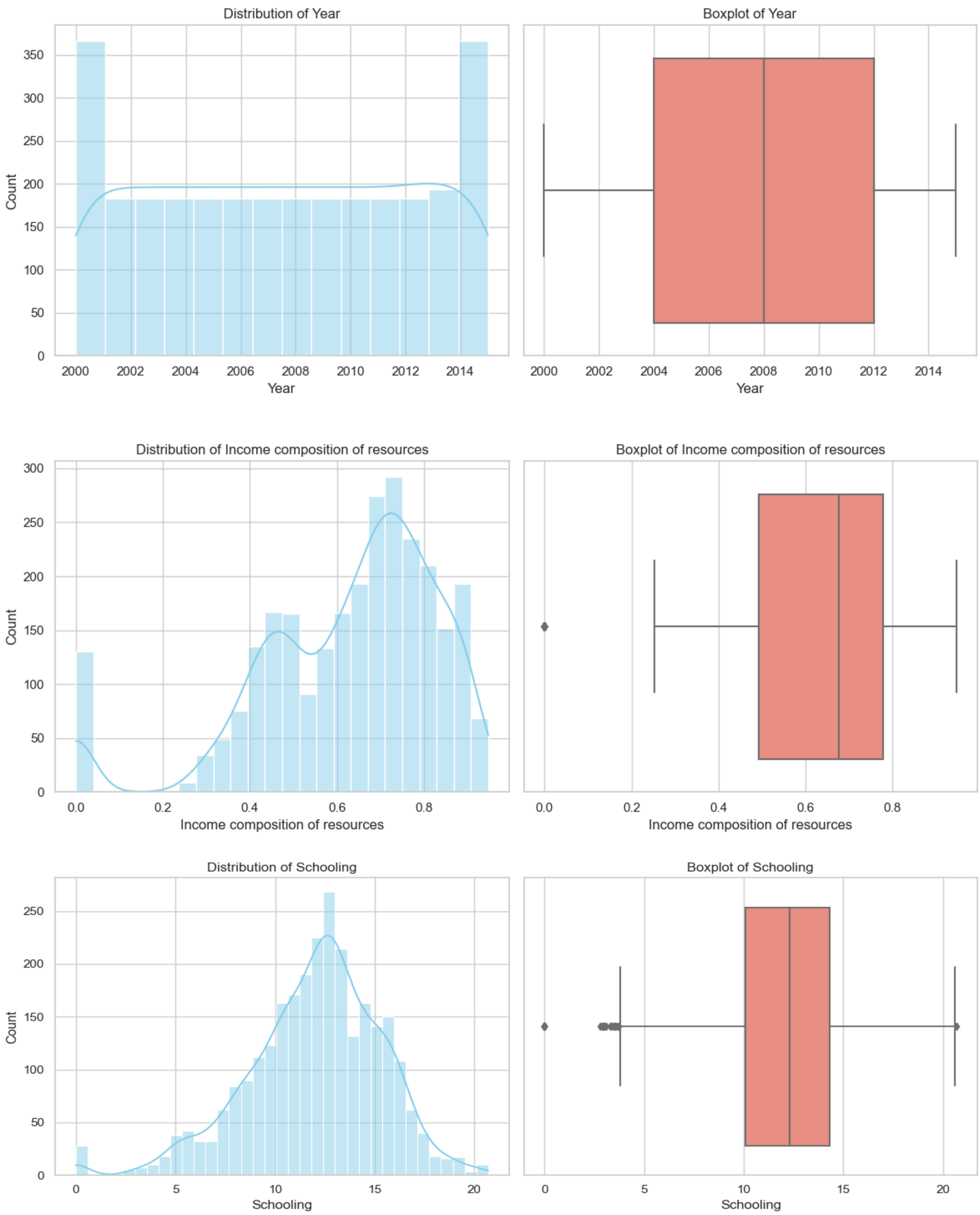
- ## Missing values Heat map:

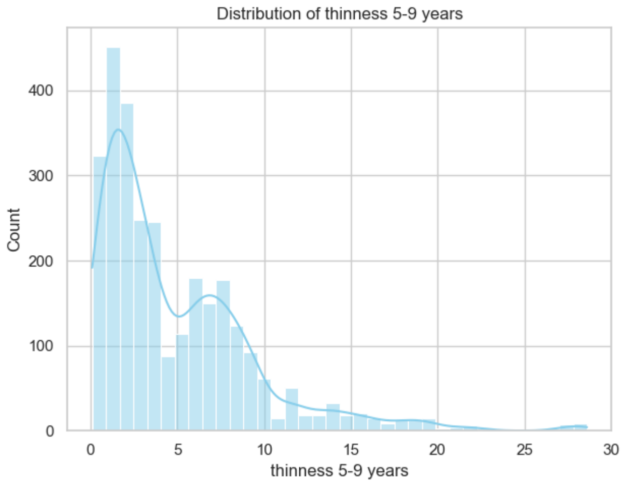
Visualizing data using histograms, boxplots:

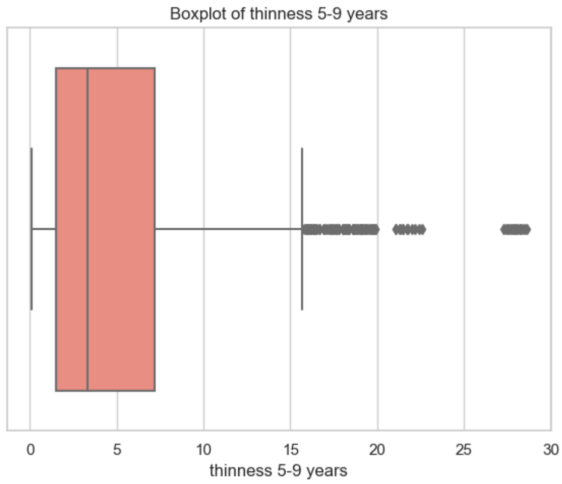
To better understand the behavior of each numerical variable and identify outliers I used Histogram and boxplots.

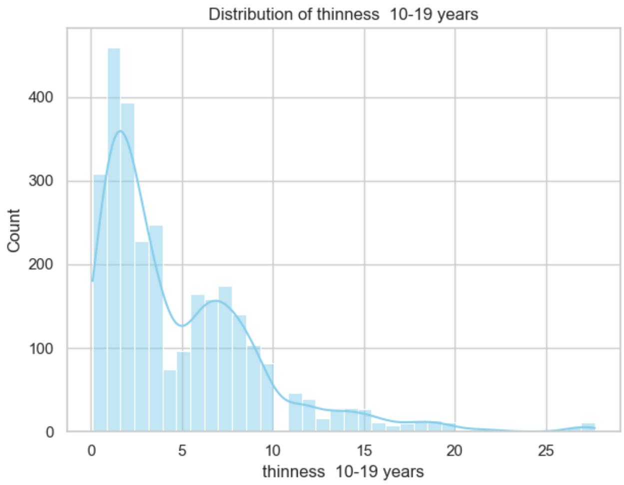
```python
import matplotlib.pyplot as plt
import seaborn as sns

# Select only numeric columns (excluding categorical ones like 'Country', 'Status' if still present)
numeric_cols = data.select_dtypes(include=['float64', 'int64']).columns

# Plot distribution and boxplot for each numeric column
for col in numeric_cols:
    plt.figure(figsize=(12, 5))

    # Histogram + KDE
    plt.subplot(1, 2, 1)
    sns.histplot(data[col], kde=True, color='skyblue')
    plt.title(f'Distribution of {col}')

    # Boxplot for outliers
    plt.subplot(1, 2, 2)
    sns.boxplot(x=data[col], color='salmon')
    plt.title(f'Boxplot of {col}')

    plt.tight_layout()
    plt.show()
```

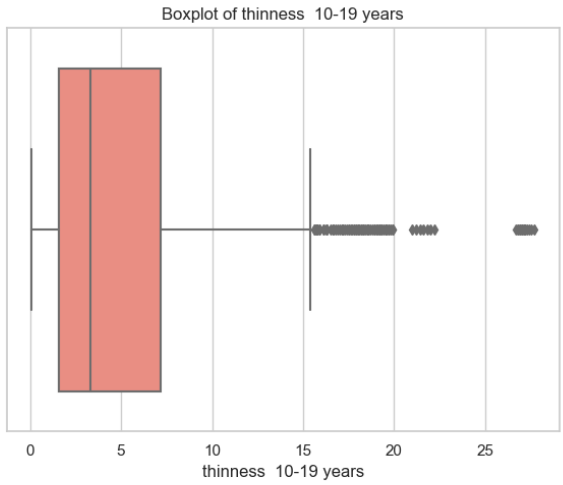
Histogram and Boxplot:

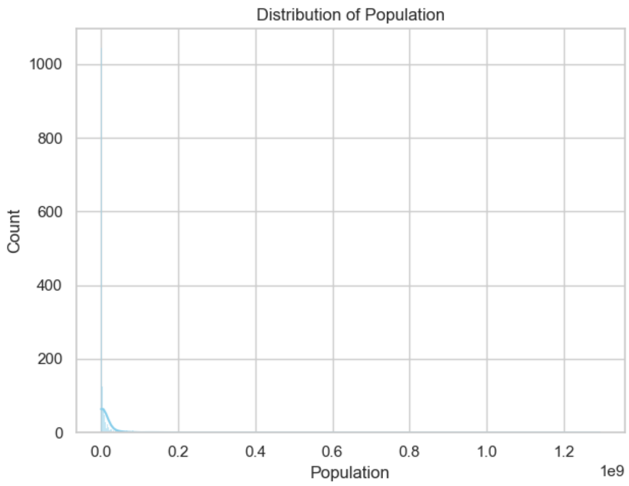


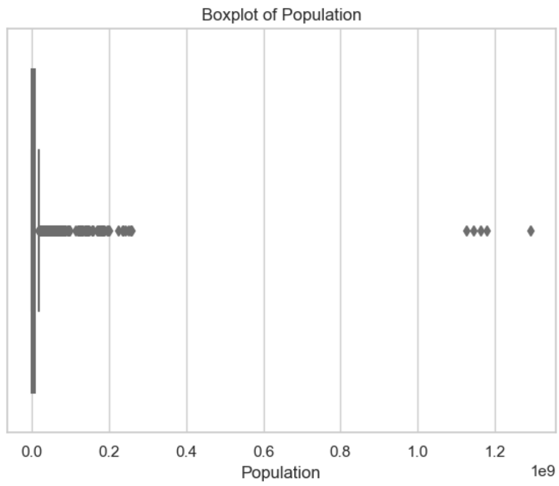

Distribution of thinness 5-9 years
Count
0
100
200
300
400
0
5
10
15
20
25
30
thinness 5-9 years

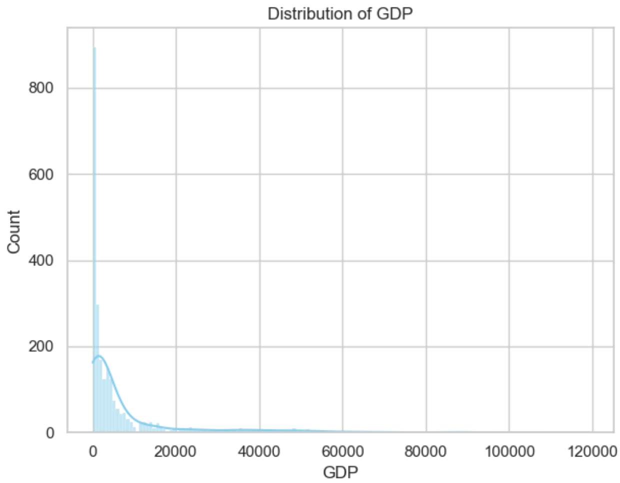

Boxplot of thinness 5-9 years
0
5
10
15
20
25
30
thinness 5-9 years

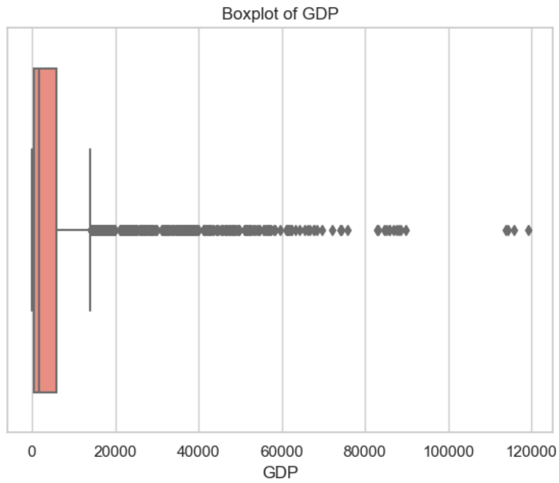

Distribution of thinness 10-19 years
Count
0
100
200
300
400
0
5
10
15
20
25
thinness 10-19 years

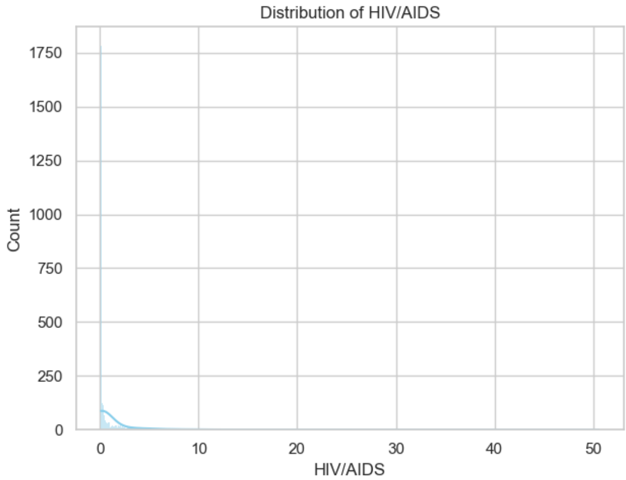

Boxplot of thinness 10-19 years
0
5
10
15
20
25
thinness 10-19 years

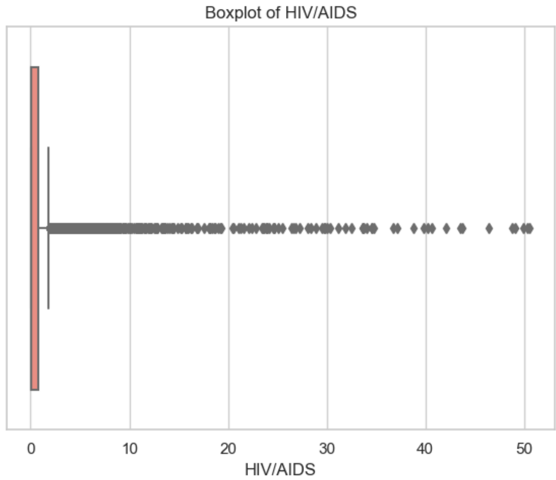

Distribution of Population
Count
0
200
400
600
800
1000
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e9
Population

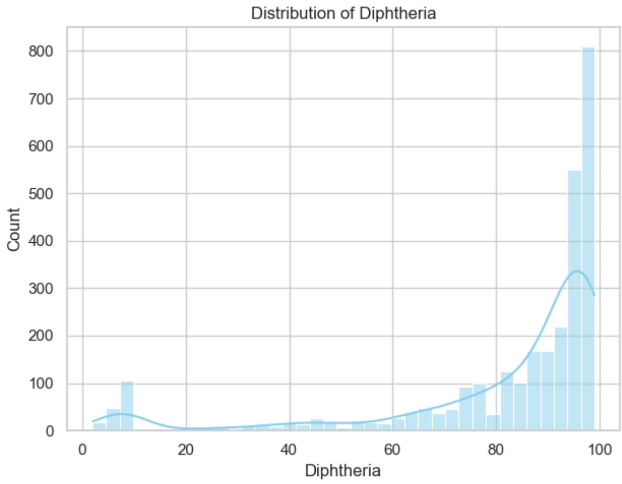

Boxplot of Population
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e9
Population

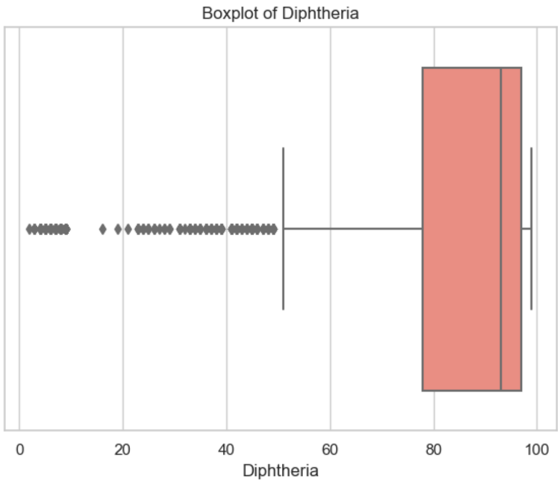

Distribution of GDP
Count
GDP

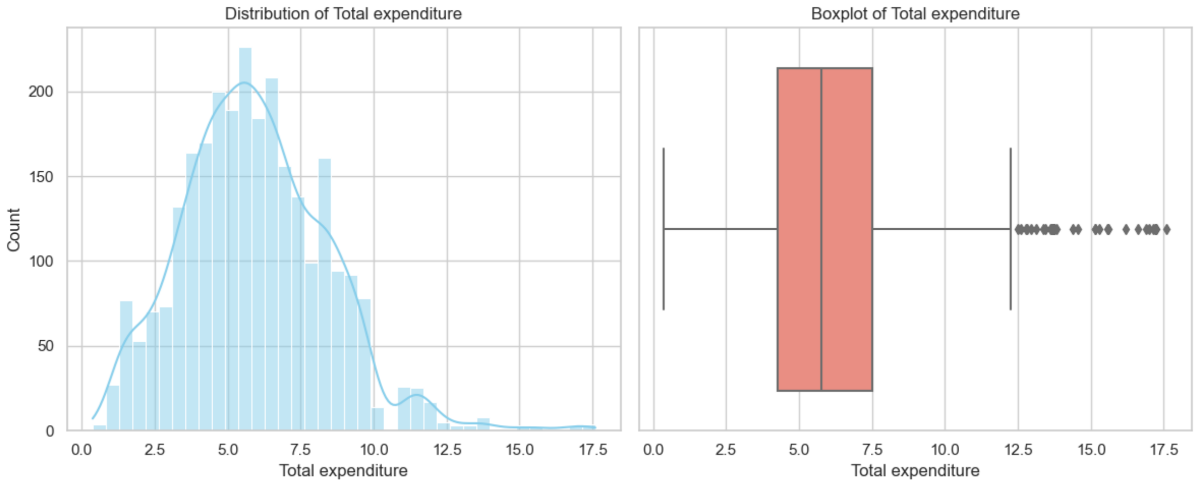

Boxplot of GDP
GDP

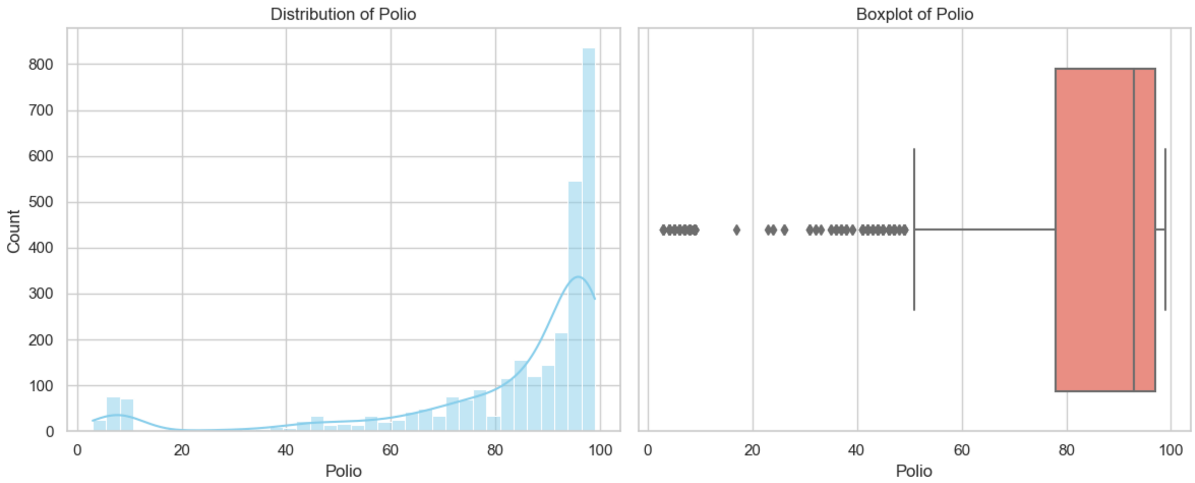

Distribution of HIV/AIDS
Count
HIV/AIDS

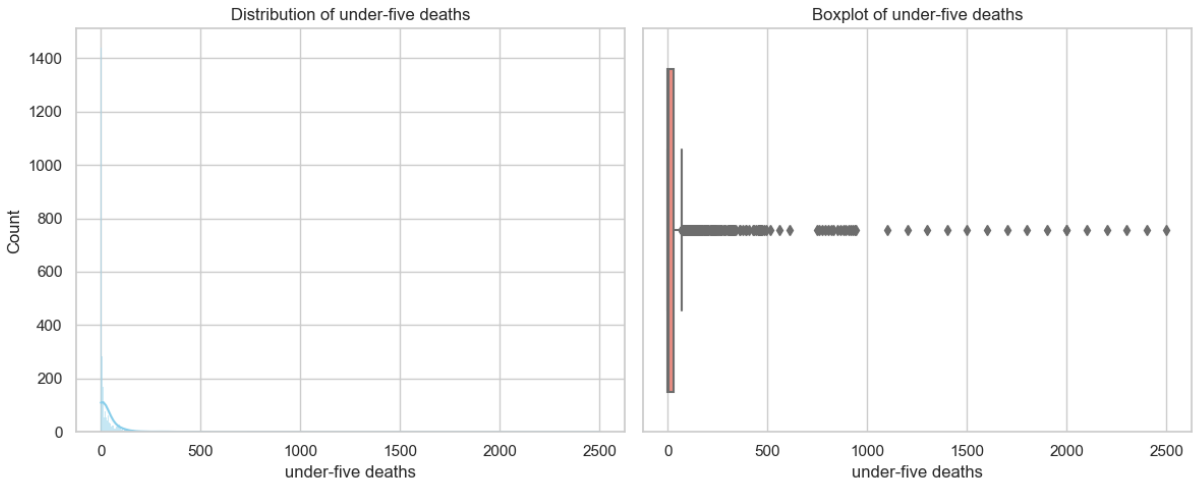

Boxplot of HIV/AIDS
HIV/AIDS

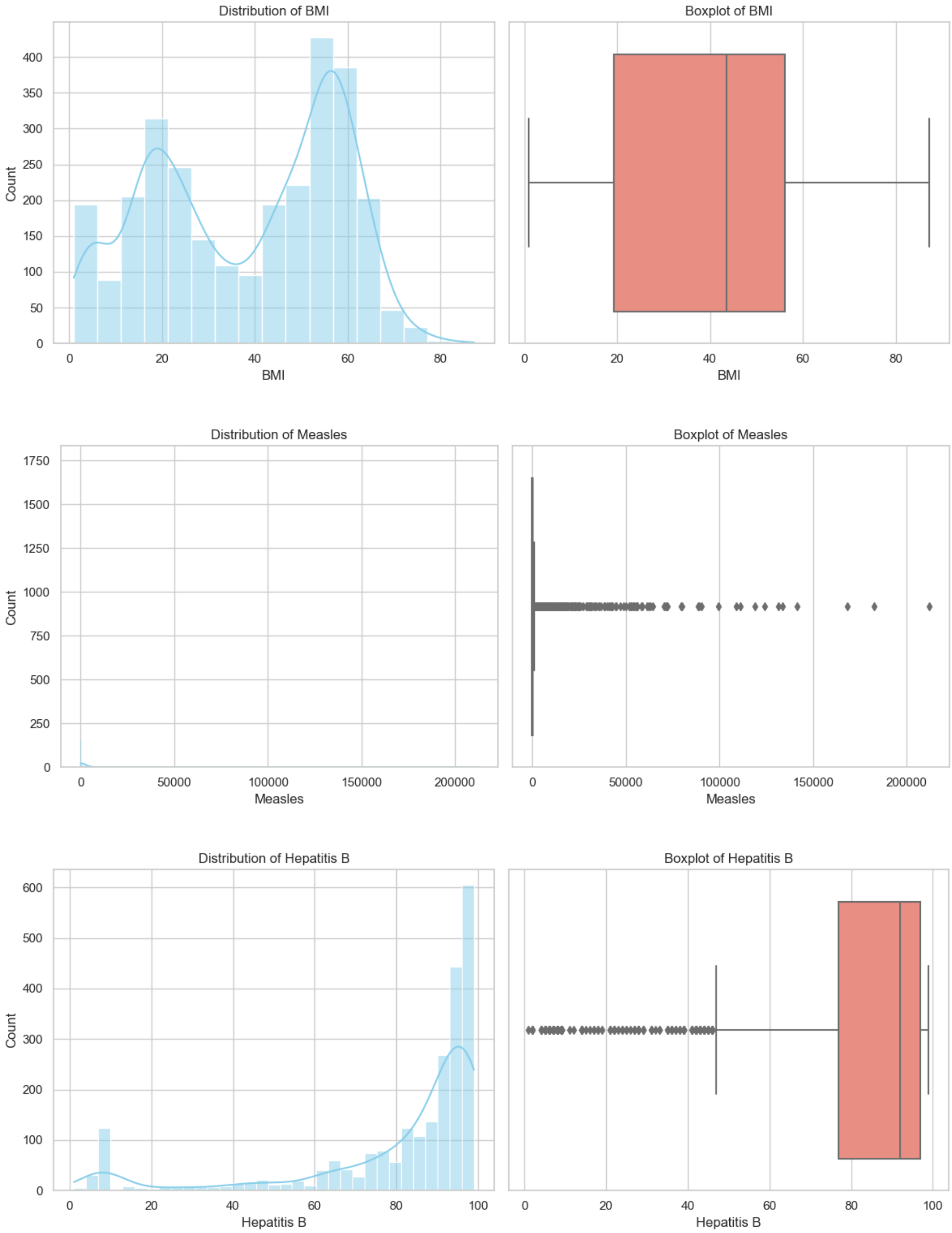

Distribution of Diphtheria
Count
Diphtheria

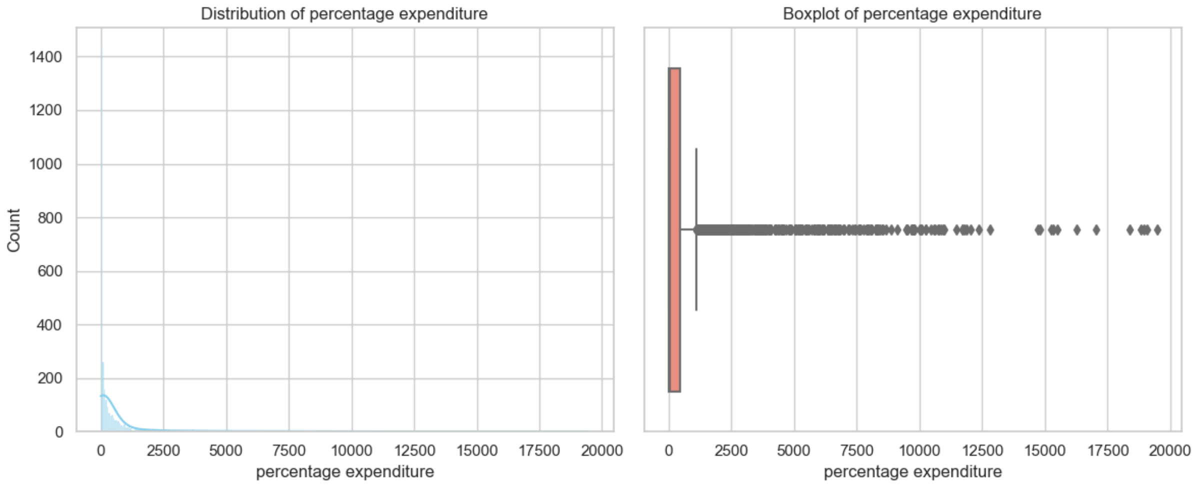

Boxplot of Diphtheria
Diphtheria

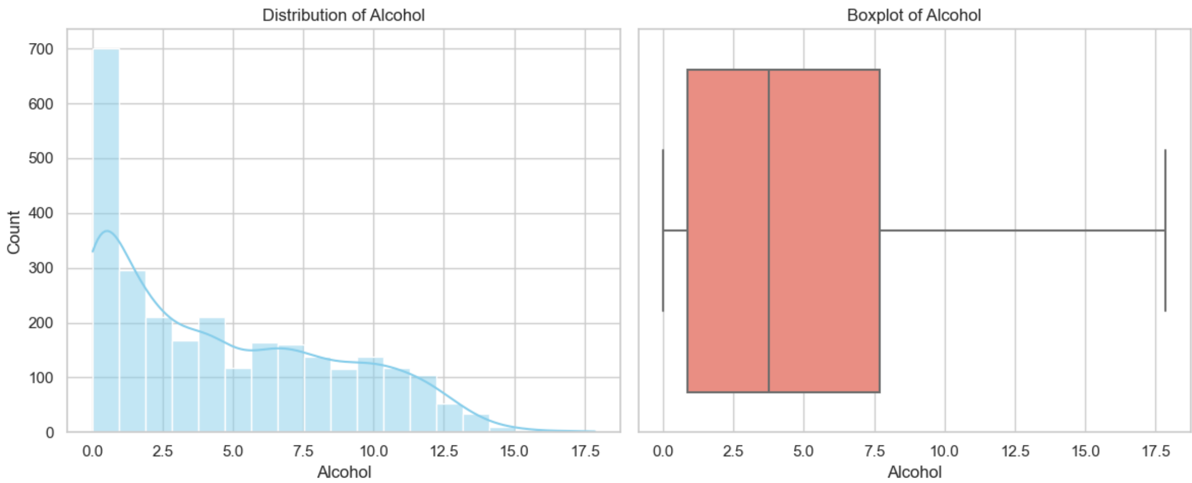

Distribution of Total expenditure
Count
0
50
100
150
200
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5
Total expenditure
Boxplot of Total expenditure
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5
Total expenditure

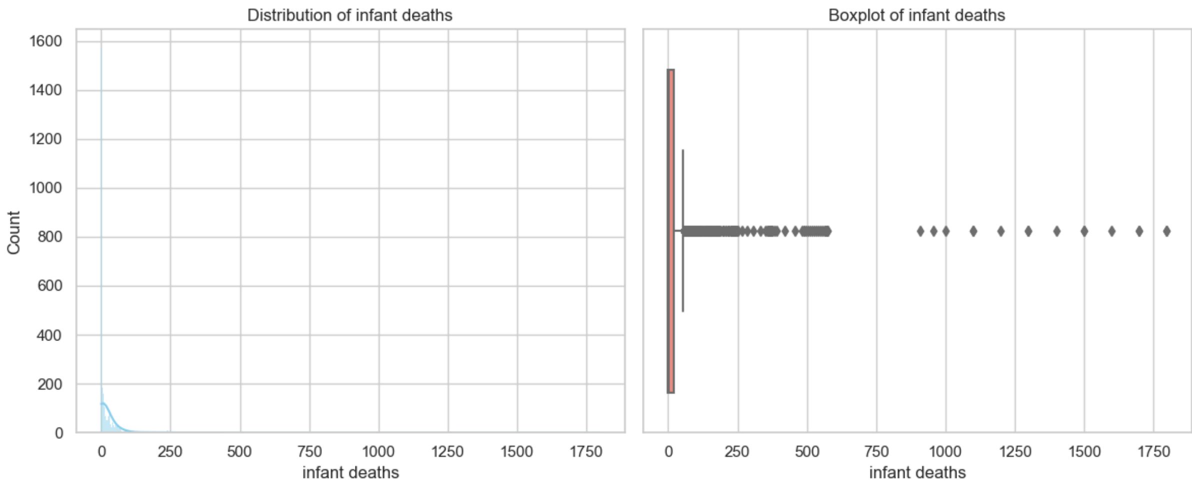

Distribution of Polio
Count
0
100
200
300
400
500
600
700
800
0 20 40 60 80 100
Polio
Boxplot of Polio
0 20 40 60 80 100
Polio

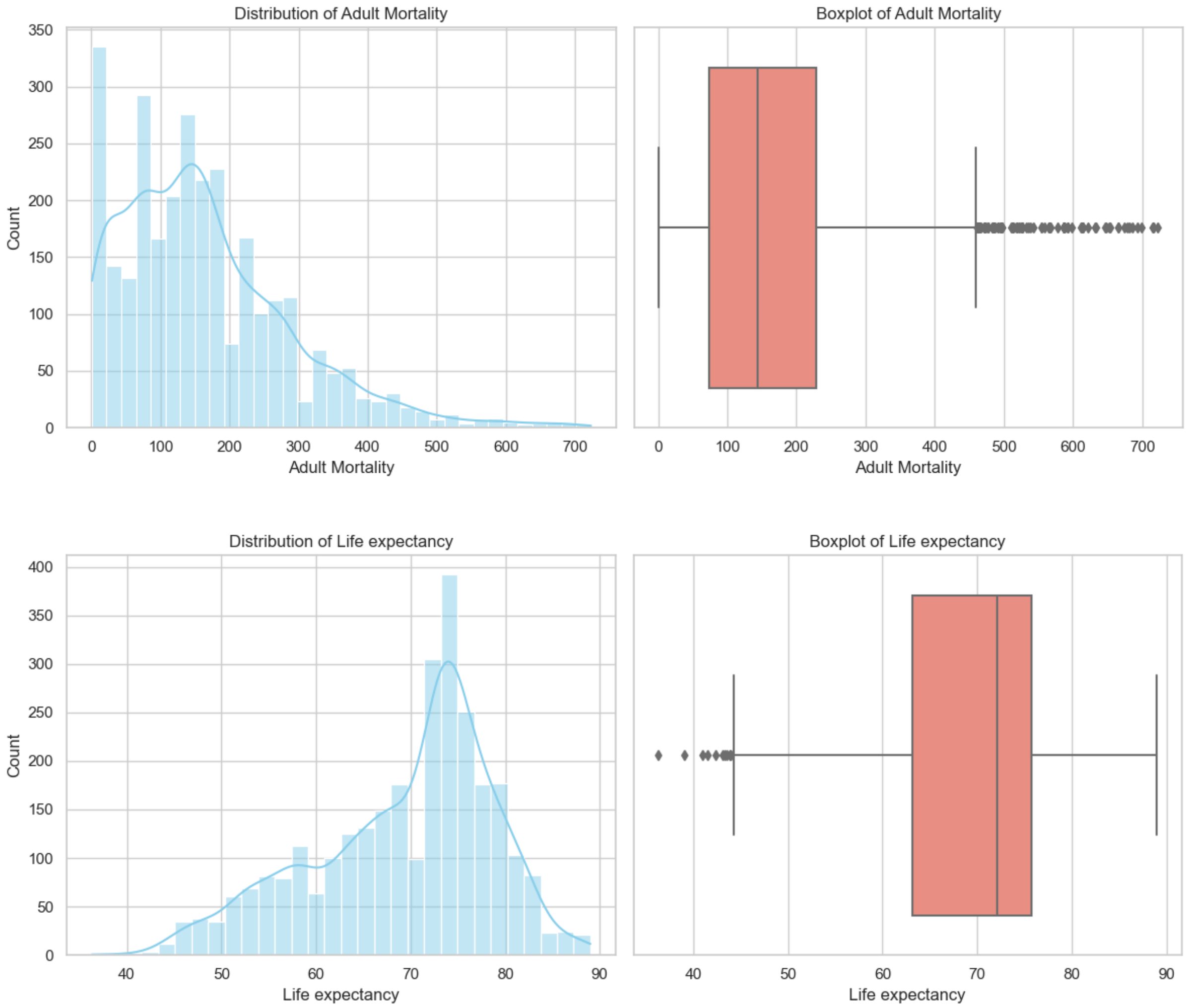

Distribution of under-five deaths
Count
0
200
400
600
800
1000
1200
1400
0 500 1000 1500 2000 2500
under-five deaths
Boxplot of under-five deaths
0 500 1000 1500 2000 2500
under-five deaths


Distribution of BMI
Count
BMI
Boxplot of BMI
BMI
Distribution of Measles
Count
Measles
Boxplot of Measles
Measles
Distribution of Hepatitis B
Count
Hepatitis B
Boxplot of Hepatitis B
Hepatitis B


Distribution of percentage expenditure
Count
percentage expenditure
Boxplot of percentage expenditure
percentage expenditure


Distribution of Alcohol
Count
Alcohol
Boxplot of Alcohol
Alcohol


Distribution of infant deaths
Count
infant deaths
Boxplot of infant deaths
infant deaths



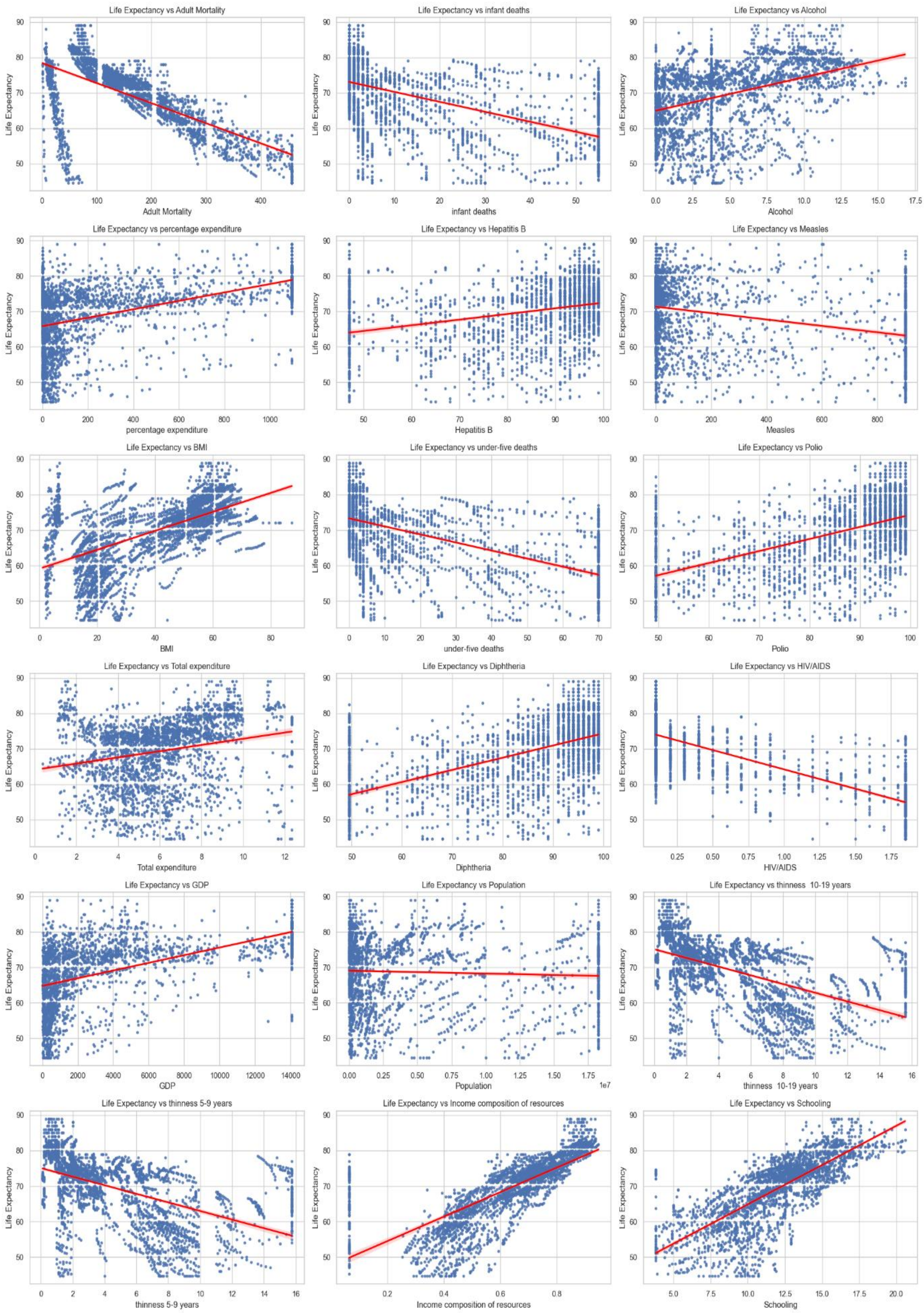
- ## Interpretation:

Most of the variables show skewed distributions—some are left-skewed while others are right-skewed. Boxplots also indicate the presence of outliers in several features.

- ## Scatter plot:

Scatter plots were generated to examine the relationship between life expectancy and each independent variable. These plots provided a visual assessment of linearity, helping to identify potential non-linear patterns, outliers, and the direction of associations.

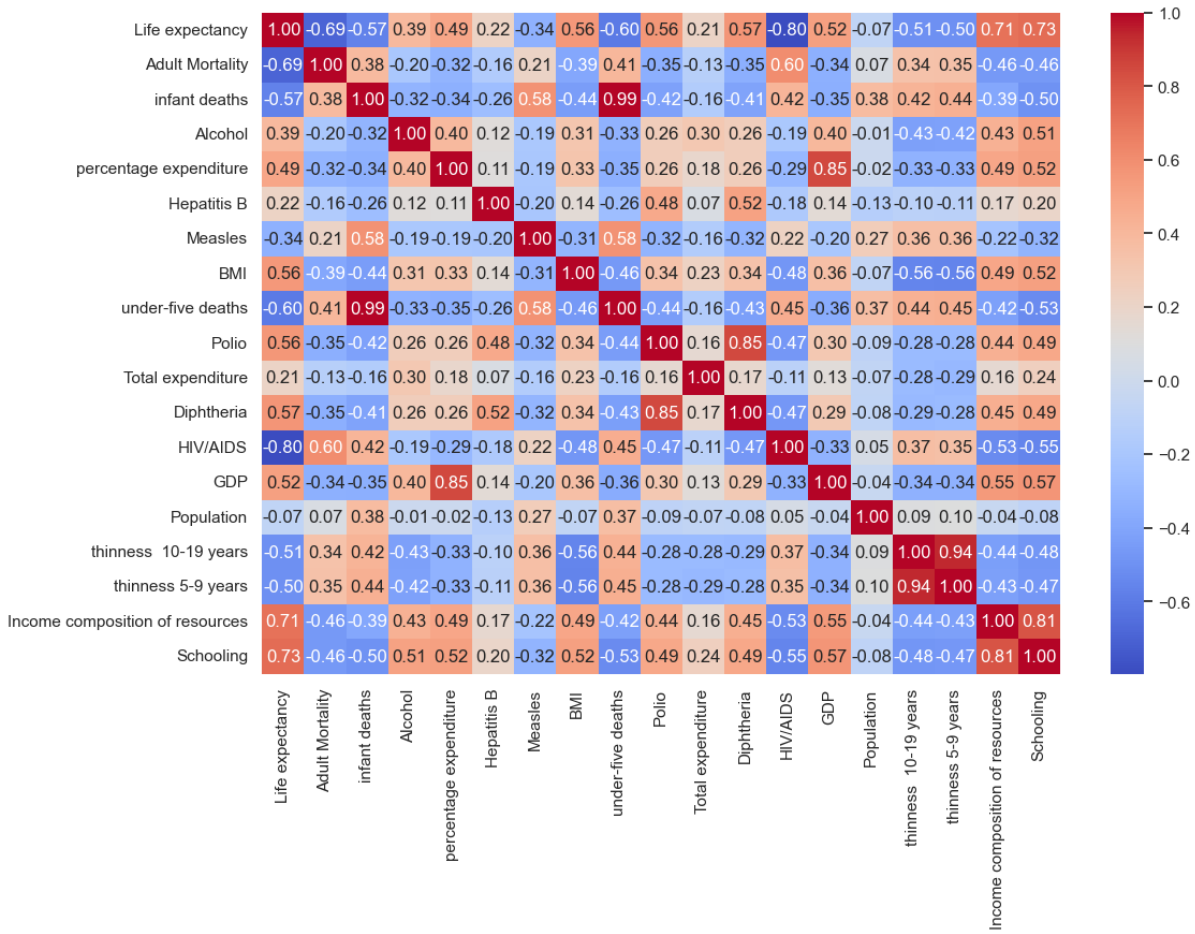
[240]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Select columns for regression plots
cols_to_compare = ['Adult Mortality', 'infant deaths', 'Alcohol', 'percentage expenditure',
                   'Hepatitis B', 'Measles', 'BMI', 'under-five deaths', 'Polio',
                   'Total expenditure', 'Diphtheria', 'HIV/AIDS', 'GDP', 'Population',
                   'thinness  10-19 years', 'thinness 5-9 years',
                   'Income composition of resources', 'Schooling']

# Set plot style
sns.set(style="whitegrid")
plt.figure(figsize=(20, 25))

# Create regression plots
for idx, col in enumerate(cols_to_compare):
    plt.subplot(6, 3, idx + 1)
    sns.regplot(x=col, y='Life expectancy', data=data, scatter_kws={'s': 10}, line_kws={"color": "red"})
    plt.title(f'Life Expectancy vs {col}')
    plt.xlabel(col)
    plt.ylabel('Life Expectancy')

plt.tight_layout()
plt.show()
```


Life Expectancy vs Adult Mortality
Adult Mortality
Life Expectancy vs infant deaths
infant deaths
Life Expectancy vs Alcohol
Alcohol
Life Expectancy vs percentage expenditure
percentage expenditure
Life Expectancy vs Hepatitis B
Hepatitis B
Life Expectancy vs Measles
Measles
Life Expectancy vs BMI
BMI
Life Expectancy vs under-five deaths
under-five deaths
Life Expectancy vs Polio
Polio
Life Expectancy vs Total expenditure
Total expenditure
Life Expectancy vs Diphtheria
Diphtheria
Life Expectancy vs HIV/AIDS
HIV/AIDS
Life Expectancy vs GDP
GDP
Life Expectancy vs Population
Population
1e7
Life Expectancy vs thinness 10-19 years
thinness 10-19 years
Life Expectancy vs thinness 5-9 years
thinness 5-9 years
Life Expectancy vs Income composition of resources
Income composition of resources
Life Expectancy vs Schooling
Schooling
Life Expectancy

- ## Correlation matrix:

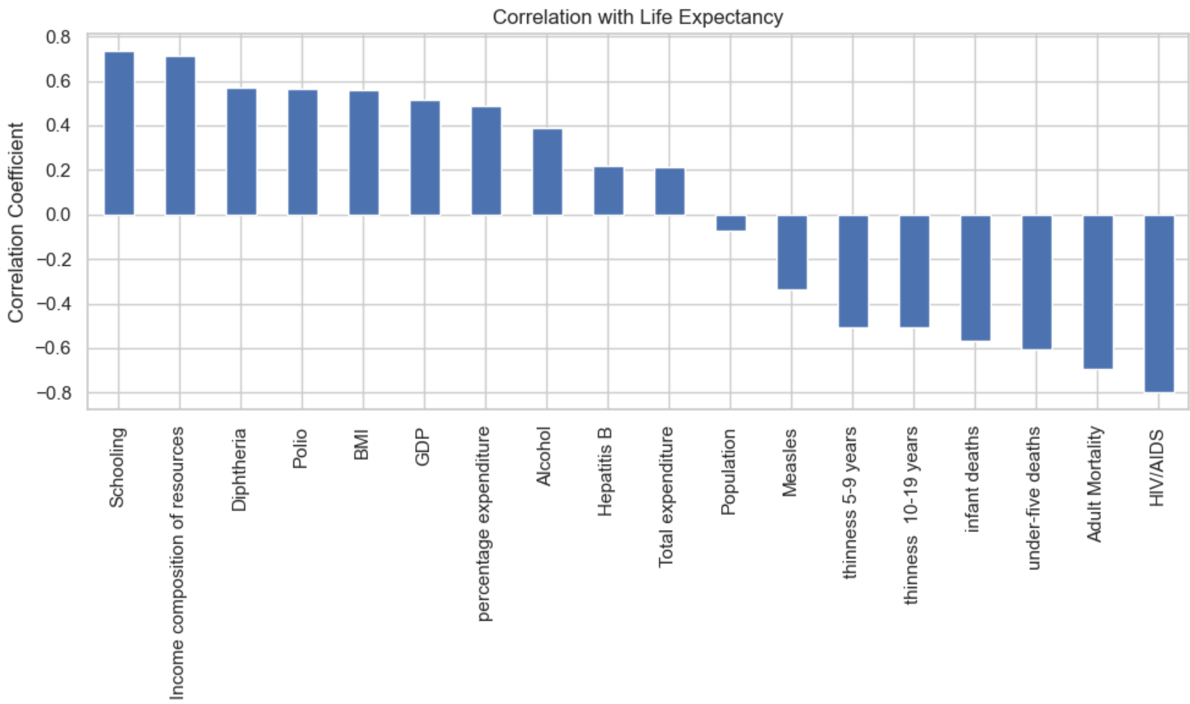
A **correlation matrix** was computed, and a **heatmap** was plotted for visualization. This allows for a quick inspection of both positive and negative correlations among variables.

```
[334]: import seaborn as sns
       import matplotlib.pyplot as plt
       # Calculate the correlation matrix
       correlation_matrix = data.corr()

       # Plot heatmap for feature correlation
       plt.figure(figsize=(12, 8))
       sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", fmt=".2f")
       plt.show()
```



- ## Correlation Plot:

```
[331]: import matplotlib.pyplot as plt

correlation.plot(kind='bar', figsize=(10, 6), title='Correlation with Life Expectancy')
plt.ylabel('Correlation Coefficient')
plt.grid(True)
plt.tight_layout()
plt.show()
```



Data Preprocessing:

- ### Handling Missing Data Using Median Imputation

Missing values in the dataset were handled using median imputation. The median was chosen because many variables are not normally distributed and contain outliers. Unlike the mean, the median is robust to skewed distributions and extreme values, making it a more appropriate measure for imputation in such cases.

```
median_value = data['Life expectancy'].median()
data['Life expectancy'].fillna(median_value, inplace=True)


print(data['Life expectancy'].isnull().sum())
print(data.isnull().sum())
```

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

[281]:

```
median_value = data['Adult Mortality'].median()
data['Adult Mortality'].fillna(median_value, inplace=True)


print(data['Adult Mortality'].isnull().sum())
print(data.isnull().sum())
```

```
0
Country                              0
Year                                 0
Status                               0
Life expectancy                      0
Adult Mortality                      0
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  10-19 years               34
thinness 5-9 years                  34
Income composition of resources    167
Schooling                          163
```

[290]:

```
median_value = data['Alcohol'].median()
data['Alcohol'].fillna(median_value, inplace=True)


print(data['Alcohol'].isnull().sum())
print(data.isnull().sum())
```

```
0
Country                              0
Year                                 0
Status                               0
Life expectancy                      0
Adult Mortality                      0
infant deaths                        0
Alcohol                              0
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  10-19 years               34
thinness 5-9 years                  34
Income composition of resources    167
Schooling                          163
dtype: int64
```

```
median_value = data['Hepatitis B'].median()
data['Hepatitis B'].fillna(median_value, inplace=True)


print(data['Hepatitis B'].isnull().sum())
print(data.isnull().sum())
```

```
0
Country                             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-five deaths                   0
Polio                               0
Total expenditure                 226
Diphtheria                         19
HIV/AIDS                            0
GDP                               448
Population                        652
thinness  10-19 years              34
thinness 5-9 years                 34
Income composition of resources   167
Schooling                         163
dtype: int64
```

```
[221]: median_value = data['BMI'].median()
       data['BMI'].fillna(median_value, inplace=True)


       print(data['BMI'].isnull().sum())
       print(data.isnull().sum())
```

```
0
Country                             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-five deaths                   0
Polio                               0
Total expenditure                 226
Diphtheria                         19
HIV/AIDS                            0
GDP                               448
Population                        652
thinness  10-19 years              34
thinness 5-9 years                 34
Income composition of resources   167
Schooling                         163
dtype: int64
```

```
[222]: median_value = data['Polio'].median()
       data['Polio'].fillna(median_value, inplace=True)


       print(data['Polio'].isnull().sum())
       print(data.isnull().sum())
```

```
0
Country                              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-five deaths                    0
Polio                                0
Total expenditure                  226
Diphtheria                          19
HIV/AIDS                             0
GDP                                448
Population                         652
thinness  10-19 years               34
thinness 5-9 years                  34
Income composition of resources    167
Schooling                          163
dtype: int64
```

```
: median_value = data['Total expenditure'].median()
  data['Total expenditure'].fillna(median_value, inplace=True)


  print(data['Total expenditure'].isnull().sum())
  print(data.isnull().sum())
```

```
0
Country                              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-five deaths                    0
Polio                                0
Total expenditure                    0
Diphtheria                          19
HIV/AIDS                             0
GDP                                448
Population                         652
thinness  10-19 years               34
thinness 5-9 years                  34
Income composition of resources    167
Schooling                          163
dtype: int64
```

[233]:
```
median_value = data['Diphtheria'].median()
data['Diphtheria'].fillna(median_value, inplace=True)
print(data['Diphtheria'].isnull().sum())
print(data.isnull().sum())
```

```
0
Country                            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-five deaths                  0
Polio                              0
Total expenditure                  0
Diphtheria                         0
HIV/AIDS                           0
GDP                              448
Population                       652
thinness  10-19 years             34
thinness 5-9 years                34
Income composition of resources  167
Schooling                        163
dtype: int64
```

[234]:
```
median_value = data['GDP'].median()
data['GDP'].fillna(median_value, inplace=True)
print(data['GDP'].isnull().sum())
print(data.isnull().sum())
```

```
0
Country                            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-five deaths                  0
Polio                              0
Total expenditure                  0
Diphtheria                         0
HIV/AIDS                           0
GDP                                0
Population                       652
thinness  10-19 years             34
thinness 5-9 years                34
Income composition of resources  167
Schooling                        163
dtype: int64
```

[235]:
```
median_value = data['Population'].median()
data['Population'].fillna(median_value, inplace=True)
print(data['Population'].isnull().sum())
print(data.isnull().sum())
```

```
0
Country                            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-five deaths                  0
Polio                              0
Total expenditure                  0
Diphtheria                         0
HIV/AIDS                           0
GDP                                0
Population                         0
thinness  10-19 years             34
thinness 5-9 years                34
Income composition of resources  167
Schooling                        163
dtype: int64
```

[236]:
```
median_value = data['thinness  10-19 years'].median()
data['thinness  10-19 years'].fillna(median_value, inplace=True)
print(data['thinness  10-19 years'].isnull().sum())
print(data.isnull().sum())
```

```
0
Country                            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-five deaths                  0
Polio                              0
Total expenditure                  0
Diphtheria                         0
HIV/AIDS                           0
GDP                                0
Population                         0
thinness  10-19 years              0
thinness 5-9 years                34
Income composition of resources  167
Schooling                        163
dtype: int64
```

[237]:
```
median_value = data['thinness 5-9 years'].median()
data['thinness 5-9 years'].fillna(median_value, inplace=True)
print(data['thinness 5-9 years'].isnull().sum())
print(data.isnull().sum())
```

```
0
Country                              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-five deaths                    0
Polio                                0
Total expenditure                    0
Diphtheria                           0
HIV/AIDS                             0
GDP                                  0
Population                           0
thinness  10-19 years                0
thinness 5-9 years                   0
Income composition of resources    167
Schooling                          163
dtype: int64
```

[238]:
```
median_value = data['Income composition of resources'].median()
data['Income composition of resources'].fillna(median_value, inplace=True)
print(data['Income composition of resources'].isnull().sum())
print(data.isnull().sum())
```

```
0
Country                              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-five deaths                    0
Polio                                0
Total expenditure                    0
Diphtheria                           0
HIV/AIDS                             0
GDP                                  0
Population                           0
thinness  10-19 years                0
thinness 5-9 years                   0
Income composition of resources      0
Schooling                          163
dtype: int64
```

```
•[239]: median_value = data['Schooling'].median()
        data['Schooling'].fillna(median_value, inplace=True)
        print(data['Schooling'].isnull().sum())
        print(data.isnull().sum())
```

```
0
Country                            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-five deaths                  0
Polio                              0
Total expenditure                  0
Diphtheria                         0
HIV/AIDS                           0
GDP                                0
Population                         0
thinness  10-19 years              0
thinness 5-9 years                 0
Income composition of resources    0
Schooling                          0
dtype: int64
```

- ## Outlier Treatment Using IQR (Capping)

Outliers were treated using the IQR method by capping values below the lower bound and above the upper bound. This helps minimize the impact of extreme values without removing data points.

```
[278]: # Select only numeric columns
       numeric_cols = data.select_dtypes(include=['float64', 'int64']).columns

       # Apply IQR capping for each numeric column
       for col in numeric_cols:
           Q1 = data[col].quantile(0.25)
           Q3 = data[col].quantile(0.75)
           IQR = Q3 - Q1
           lower_bound = Q1 - 1.5 * IQR
           upper_bound = Q3 + 1.5 * IQR

           # Cap values outside IQR range
           data[col] = data[col].apply(lambda x: lower_bound if x < lower_bound else upper_bound if x > upper_bound else x)

       print("Outliers in 'data' have been capped using the IQR method.")
```

```
Outliers in 'data' have been capped using the IQR method.
```

- **Perform encoding for categorical variables:**

Since the objective of this study is to predict life expectancy based on socio-economic factors and the only categorical variable, 'Country', was excluded from the analysis, there was no need to apply encoding techniques. All remaining features in the dataset are numerical, making the dataset suitable for regression analysis without additional preprocessing for categorical data.

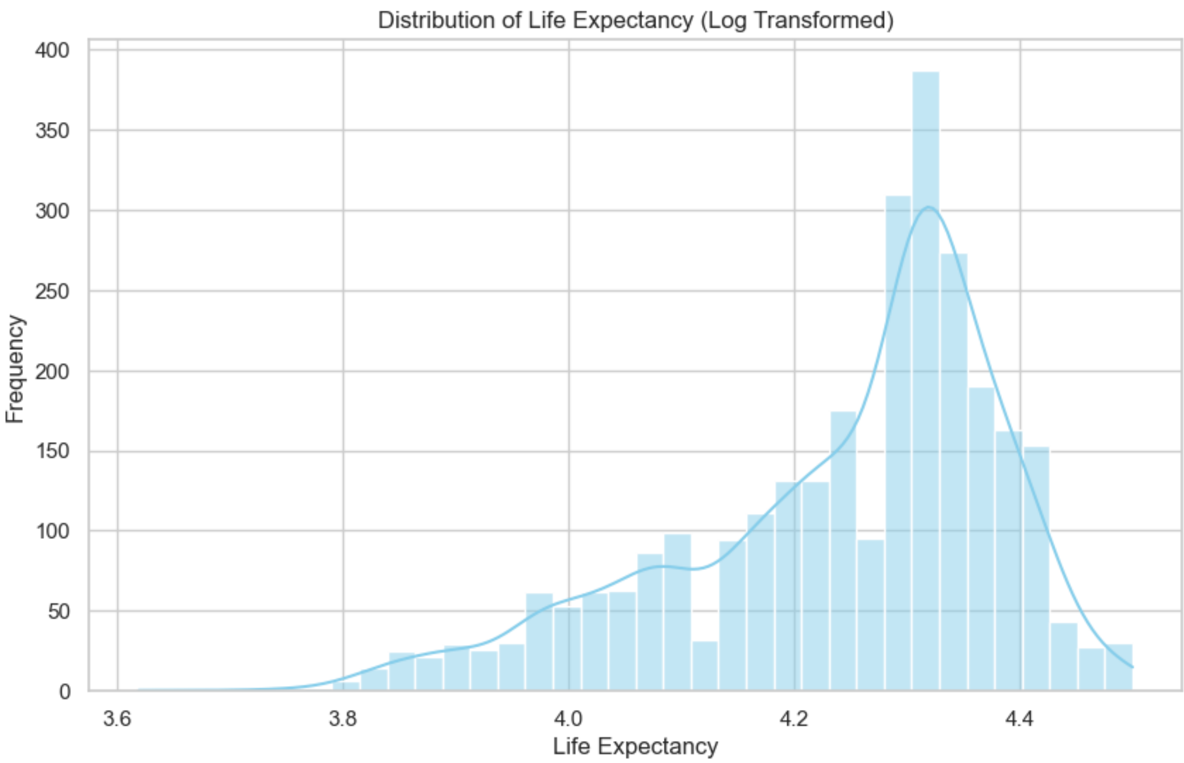
```
[241]: # Permanently drop object (string) columns from `data`
data.drop(columns=data.select_dtypes(include=['object']).columns, inplace=True)

# Now print data
print(data)
```

- **Scaling/normalization:**

Scaling or normalization was considered for the **life expectancy** variable, but after applying transformations, the histogram showed no significant change in its distribution. This suggests that **life expectancy** is already appropriately scaled and does not require further transformation for modeling purposes.

```
[109]: # import numpy as np

# # Log transform the 'Life expectancy' column
# data['Life expectancy'] = np.log1p(data['Life expectancy'])

# # Print the first few rows to check the transformed values
# print(data[['Life expectancy']].head())
```

```
   Life expectancy
0         4.189655
1         4.109233
2         4.109233
3         4.102643
4         4.097672
```



- ### **Creating new variables (feature engineering):**

A new variable, thinness_combined, was created by averaging the values from two existing columns:

- Thinness 10-19 years
- Thinness 5-9 years

```
1          62.0       492  18.6    58.0
2          64.0       430  18.1    62.0
3          67.0      2787  17.6    67.0
4          68.0      3013  17.2    68.0

           GDP  Population  Income composi
0   584.259210  33736494.0
1   612.696514    327582.0
2   631.744976  31731688.0
3   669.959000   3696958.0
4    63.537231   2978599.0

   thinness_combined
0              17.25
1              17.50
2              17.70
3              17.95
4              18.20
```

```
[248]: # Combine the two existing columns
data['thinness_combined'] = (data['thinness  10-19 years'] + data['thinness 5-9 years']) / 2

# Drop the originals to avoid redundancy
data = data.drop(columns=['thinness  10-19 years', 'thinness 5-9 years'])

# Check the updated data
print(data.head())
```

- **Correlation Coefficient Analysis**

To better understand the relationships between the independent variables and the target variable (**Life Expectancy**), I calculated the **Pearson correlation coefficients**. This helps identify Which features are strongly or weakly related to the target variable.

```
[2938 rows x 19 columns]

[330]: # Calculate Pearson correlation with Life expectancy
correlation = data.corr()['Life expectancy'].drop('Life expectancy').sort_values(ascending=False)

# Display the correlation values
print("Pearson Correlation with Life Expectancy:")
print(correlation)

Pearson Correlation with Life Expectancy:
Schooling                          0.733127
Income composition of resources    0.711743
Diphtheria                         0.568627
Polio                              0.562833
BMI                                0.557762
GDP                                0.517416
percentage expenditure             0.487679
Alcohol                            0.389846
Hepatitis B                        0.220424
Total expenditure                  0.212735
Population                        -0.074000
Measles                           -0.336993
thinness 5-9 years                -0.504555
thinness  10-19 years             -0.506580
infant deaths                     -0.566998
under-five deaths                 -0.604007
Adult Mortality                   -0.691454
HIV/AIDS                          -0.796704
Name: Life expectancy, dtype: float64
```

- **Variance Inflation Factor (VIF) Analysis(Multicollinearity Diagnosis):**

To detect multicollinearity among the independent variables, I calculated the Variance Inflation Factor (VIF) for each predictor.

VIF values were categorized as follows:

- Low: VIF < 5
- Moderate: VIF between 5 and 10
- High: VIF > 10

The results are summarized below:

- Low VIF features (no multicollinearity concern):
  Adult Mortality, Alcohol, percentage expenditure, Hepatitis B, Measles, BMI, Polio, Total expenditure, Diphtheria, HIV/AIDS, GDP, Population, Income composition of resources, and Schooling.
- Moderate VIF features (potential concern):
  Life expectancy, thinness 10-19 years, and thinness 5-9 years.
- High VIF features (indicates multicollinearity):
  infant deaths and under-five deaths.

To improve the reliability of the regression estimates, I removed the features with high VIF values, namely infant deaths and under-five deaths. These variables showed strong linear dependence with other predictors and could distort the model's results.

[333]:
```
import pandas as pd
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor

# Assuming 'data' is your DataFrame with all the features
# Step 1: Add constant (intercept) column to the data for VIF calculation
X = sm.add_constant(data)  # Adds constant to your DataFrame

# Step 2: Calculate VIF for each feature
vif_data = pd.DataFrame()
vif_data['Feature'] = X.columns
vif_data['VIF'] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

# Step 3: Classify features based on VIF values
vif_data['VIF Category'] = pd.cut(vif_data['VIF'],
                                  bins=[0, 5, 10, float('inf')],
                                  labels=['Low', 'Moderate', 'High'],
                                  right=False)

# Step 4: Print the VIF data categorized
print(vif_data)

# Optional: Filter and print only the VIF categories
low_vif = vif_data[vif_data['VIF Category'] == 'Low']
moderate_vif = vif_data[vif_data['VIF Category'] == 'Moderate']
high_vif = vif_data[vif_data['VIF Category'] == 'High']

print("\nLow VIF Features:")
print(low_vif[['Feature', 'VIF']])

print("\nModerate VIF Features:")
print(moderate_vif[['Feature', 'VIF']])

print("\nHigh VIF Features:")
print(high_vif[['Feature', 'VIF']])
```

VIF Output:

	Feature	VIF	VIF Category
0	const	391.345774	High
1	Life expectancy	6.646245	Moderate
2	Adult Mortality	1.992800	Low
3	infant deaths	108.767232	High
4	Alcohol	1.603751	Low
5	percentage expenditure	3.712921	Low
6	Hepatitis B	1.430425	Low
7	Measles	1.607607	Low
8	BMI	1.839843	Low
9	under-five deaths	115.237966	High
10	Polio	3.750949	Low
11	Total expenditure	1.196128	Low
12	Diphtheria	4.000554	Low
13	HIV/AIDS	3.122778	Low
14	GDP	3.972407	Low
15	Population	1.238457	Low
16	thinness 10-19 years	9.196377	Moderate
17	thinness 5-9 years	9.328908	Moderate
18	Income composition of resources	3.416675	Low
19	Schooling	4.038131	Low

Low VIF Features:

	Feature	VIF
2	Adult Mortality	1.992800
4	Alcohol	1.603751
5	percentage expenditure	3.712921
6	Hepatitis B	1.430425
7	Measles	1.607607
8	BMI	1.839843
10	Polio	3.750949
11	Total expenditure	1.196128
12	Diphtheria	4.000554
13	HIV/AIDS	3.122778
14	GDP	3.972407
15	Population	1.238457
18	Income composition of resources	3.416675
19	Schooling	4.038131

Moderate VIF Features:

	Feature	VIF
1	Life expectancy	6.646245
16	thinness 10-19 years	9.196377
17	thinness 5-9 years	9.328908

High VIF Features:

	Feature	VIF
0	const	391.345774
3	infant deaths	108.767232
9	under-five deaths	115.237966

- **Removing the columns with hight ViF:**

```
[337]: # Remove features with high VIF
features_to_remove = ['infant deaths', 'under-five deaths']
data = data.drop(columns=features_to_remove)

# Print the cleaned data
print("Data after removing highly collinear features:")
print(data)
```

- **VIF after removing the columns:**

```
Low VIF Features:
                          Feature       VIF
2                 Adult Mortality  1.990446
3                         Alcohol  1.592532
4          percentage expenditure  3.697459
5                     Hepatitis B  1.423808
6                         Measles  1.358211
7                             BMI  1.814020
8                           Polio  3.743507
9               Total expenditure  1.184984
10                     Diphtheria  4.000064
11                       HIV/AIDS  3.071296
12                            GDP  3.966725
13                     Population  1.095107
16  Income composition of resources  3.374572
17                      Schooling  3.998505

Moderate VIF Features:
                 Feature       VIF
1        Life expectancy  6.227033
14  thinness  10-19 years  9.177173
15     thinness 5-9 years  9.276811

High VIF Features:
  Feature         VIF
0   const  351.116381
```

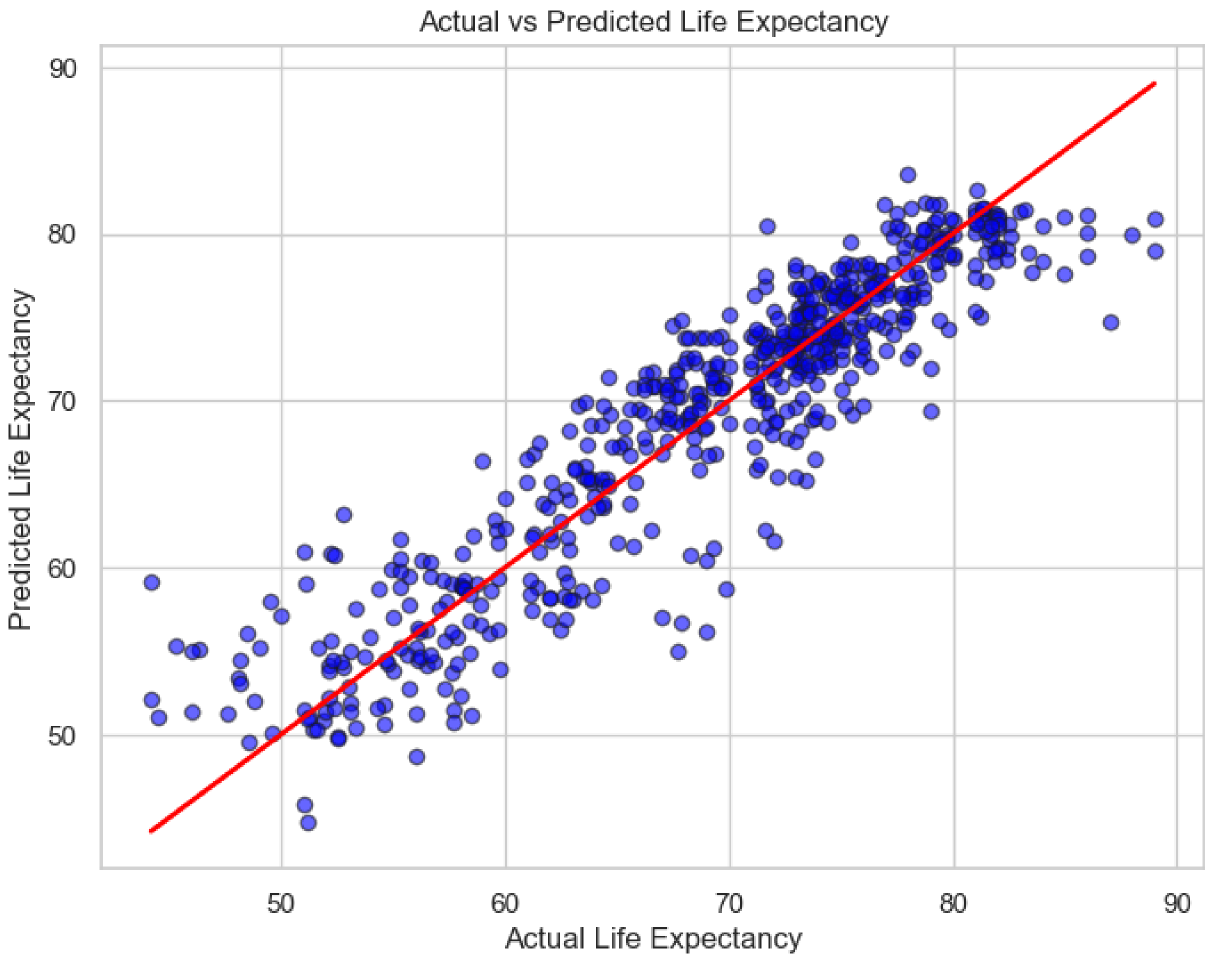
Model Building:

- **Building a Linear Regression model.**

[343]:
```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

# Define features and target
X = data.drop(columns=['Life expectancy'])
y = data['Life expectancy']

# Split the data into training and testing sets (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create and fit the linear regression model
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)
```

[343]:
```
▾ LinearRegression
LinearRegression()
```

[345]:
```
import matplotlib.pyplot as plt

# Predict on test set
y_pred = lr_model.predict(X_test)

# Plot actual vs predicted values
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, color='blue', edgecolors='k', alpha=0.6)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], color='red', linewidth=2)
plt.title('Actual vs Predicted Life Expectancy')
plt.xlabel('Actual Life Expectancy')
plt.ylabel('Predicted Life Expectancy')
plt.grid(True)
plt.show()
```



[349]:
```
# Get intercept and coefficients
intercept = lr_model.intercept_
coefficients = lr_model.coef_

# Get feature names
features = X.columns

# Print regression equation
equation = f"Life expectancy = {intercept:.2f}"
for coef, feature in zip(coefficients, features):
    equation += f" + ({coef:.2f} × {feature})"

print("Linear Regression Equation:")
print(equation)
```

```
Linear Regression Equation:
Life expectancy = 61.07 + (-0.02 x Adult Mortality) + (0.08 x Alcohol) + (0.00 x percentage expenditure) + (-0.03 x Hepatitis B) + (-0.00 x Measles) + (0.01 x BMI) + (0.03 x Polio) + (0.04 x Total expenditure)
+ (0.06 x Diphtheria) + (-5.40 x HIV/AIDS) + (0.00 x GDP) + (-0.00 x Population) + (0.03 x thinness  10-19 years) + (-0.19 x thinness 5-9 years) + (8.35 x Income composition of resources) + (0.30 x Schooling)
```

- **Linear Regression Equation:**

Life expectancy = 61.07 + (-0.02 × Adult Mortality) + (0.08 × Alcohol) + (0.00 × percentage expenditure) + (-0.03 × Hepatitis B) + (-0.00 × Measles) + (0.01 × BMI) + (0.03 × Polio) + (0.04 × Total expenditure) + (0.06 × Diphtheria) + (-5.40 × HIV/AIDS) + (0.00 × GDP) + (-

0.00 × Population) + (0.03 × thinness 10-19 years) + (-0.19 × thinness 5-9 years) + (8.35 × Income composition of resources) + (0.30 × Schooling)

- ## Interpretation of Intercept and Slopes (Regression coefficients)
  - **Intercept (61.07)**:
    The intercept represents the **baseline life expectancy** when all predictor variables are set to zero. While this scenario is not realistic in practice, it serves as the starting point for the regression model. In this case, if all independent variables were zero, the predicted life expectancy would be **61.07 years**.
  - **Slopes (Coefficients)**:

    Each slope shows the **change in life expectancy** (in years) associated with a **one-unit increase** in the corresponding variable, holding all other variables constant. For example:
  - A **1-unit increase in HIV/AIDS prevalence** leads to a **5.40-year decrease** in life expectancy (slope = -5.40).
  - A **1-unit increase in income composition of resources** increases life expectancy by **8.35 years** (slope = 8.35).
  - A **1-year increase in schooling** is associated with a **0.30-year increase** in life expectancy.

- ## Model Summary

Displaying the summary of the fitted model for key statistics and parameter estimates.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Life expectancy   R-squared:                       0.837
Model:                            OLS   Adj. R-squared:                  0.836
Method:                 Least Squares   F-statistic:                     798.8
Date:                Sun, 04 May 2025   Prob (F-statistic):               0.00
Time:                        23:20:45   Log-Likelihood:                -6500.4
No. Observations:                2350   AIC:                         1.303e+04
Df Residuals:                    2334   BIC:                         1.312e+04
Df Model:                          15
Covariance Type:            nonrobust
====================================================================================
                          coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
const                  61.0657      0.800     76.305      0.000      59.496      62.635
Adult Mortality        -0.0181      0.001    -20.050      0.000      -0.020      -0.016
Alcohol                 0.0810      0.026      3.167      0.002       0.031       0.131
```

```
 percentage expenditure            0.0016      0.000      4.127      0.000       0.001       0.002
 Hepatitis B                      -0.0280      0.006     -4.589      0.000      -0.040      -0.016
 Measles                          -0.0012      0.000     -4.546      0.000      -0.002      -0.001
 BMI                               0.0072      0.005      1.334      0.182      -0.003       0.018
 Polio                             0.0268      0.010      2.769      0.006       0.008       0.046
 Total expenditure                 0.0397      0.037      1.069      0.285      -0.033       0.113
 Diphtheria                        0.0563      0.010      5.680      0.000       0.037       0.076
 HIV/AIDS                         -5.3770      0.166    -32.347      0.000      -5.703      -5.051
 GDP                             4.404e-05   3.36e-05     1.311      0.190   -2.18e-05       0.000
 Population                     -1.386e-09   1.42e-08    -0.098      0.922   -2.92e-08    2.64e-08
 Income composition of resources   8.3206      0.749     11.115      0.000       6.853       9.789
 Schooling                         0.2978      0.051      5.876      0.000       0.198       0.397
 thinness_combined                -0.1607      0.028     -5.830      0.000      -0.215      -0.107
 ==============================================================================
 Omnibus:                        72.077   Durbin-Watson:                   2.032
 Prob(Omnibus):                   0.000   Jarque-Bera (JB):              163.413
 Skew:                           -0.142   Prob(JB):                     3.28e-36
 Kurtosis:                        4.260   Cond. No.                     7.40e+07
```

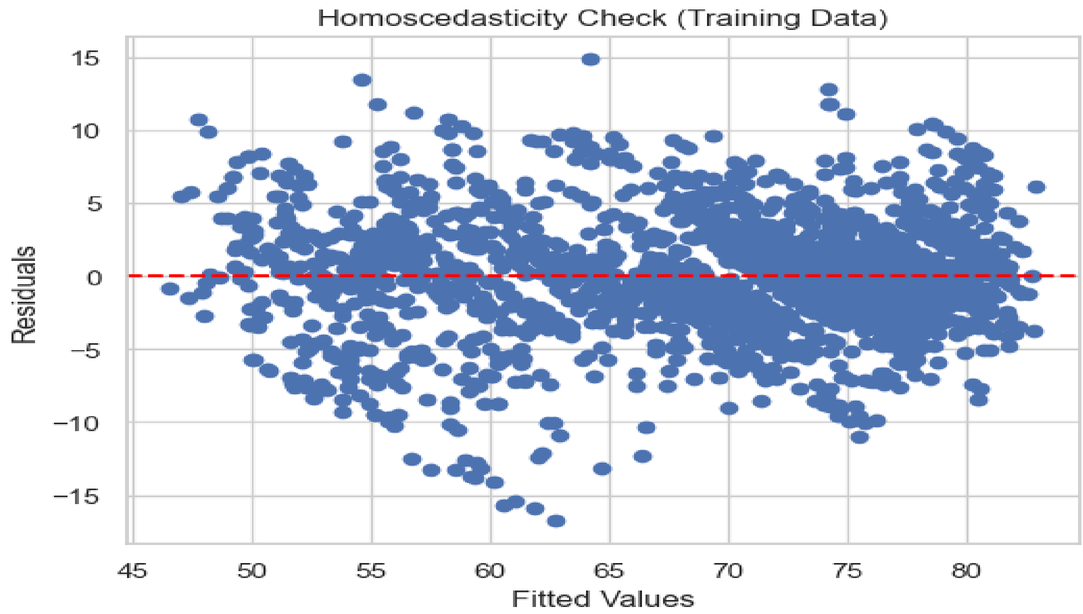
Several variables, such as 'BMI', 'GDP', 'Population', 'Total expenditure', have p-values greater than 0.05, indicating they do not significantly contribute to the model. These variables may be removed to improve model performance and focus on statistically significant predictors.

- ## Residual Analysis on Training Data:

  - **Shapiro-Wilk Test**: Testing the normality of residuals statistically.
    - Shapiro-Wilk Test (Training Data): Statistic: 0.9858, p-value: 0.0000
    - The **p-value (0.0000)** is less than 0.05, indicating that the residuals **are not normally distributed**. This suggests a potential violation of the normality assumption.
  - **Durbin-Watson Test**: Assessing the independence of residuals.
    - Durbin-Watson Statistic (Training Data): 2.0308 The **statistic (2.0308)** is very close to 2, indicating that **residuals are independent** and there is **no significant autocorrelation**

- **Homoscedasticity**

Verifying constant variance of residuals.

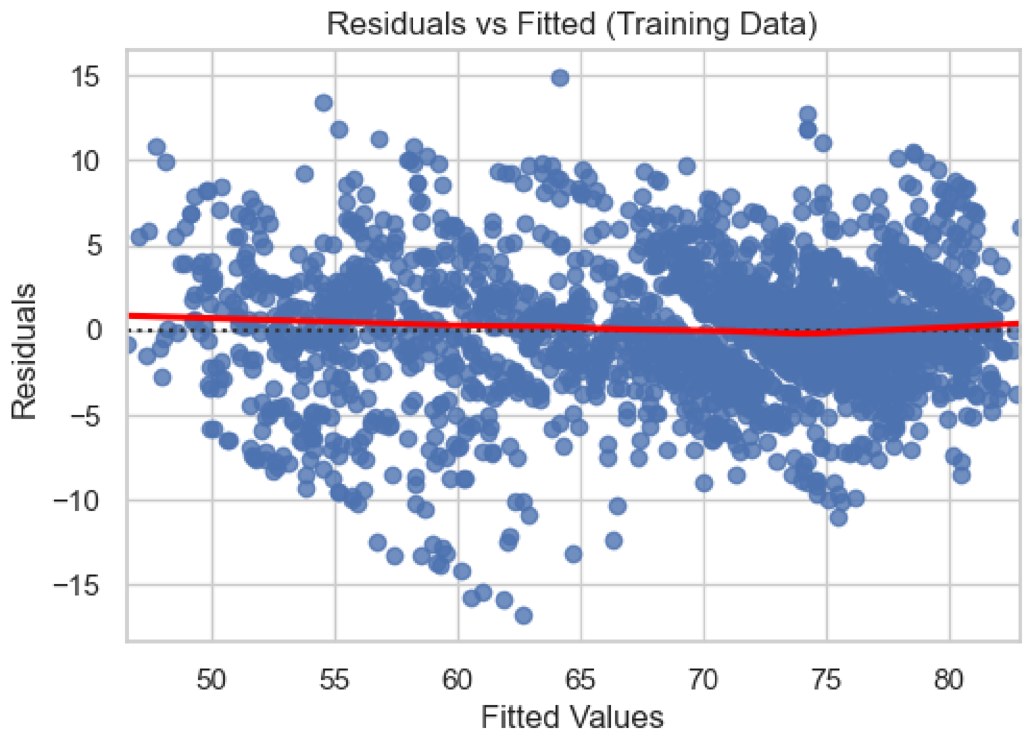


- **Q-Q Plot**

Checking the normality assumption visually.

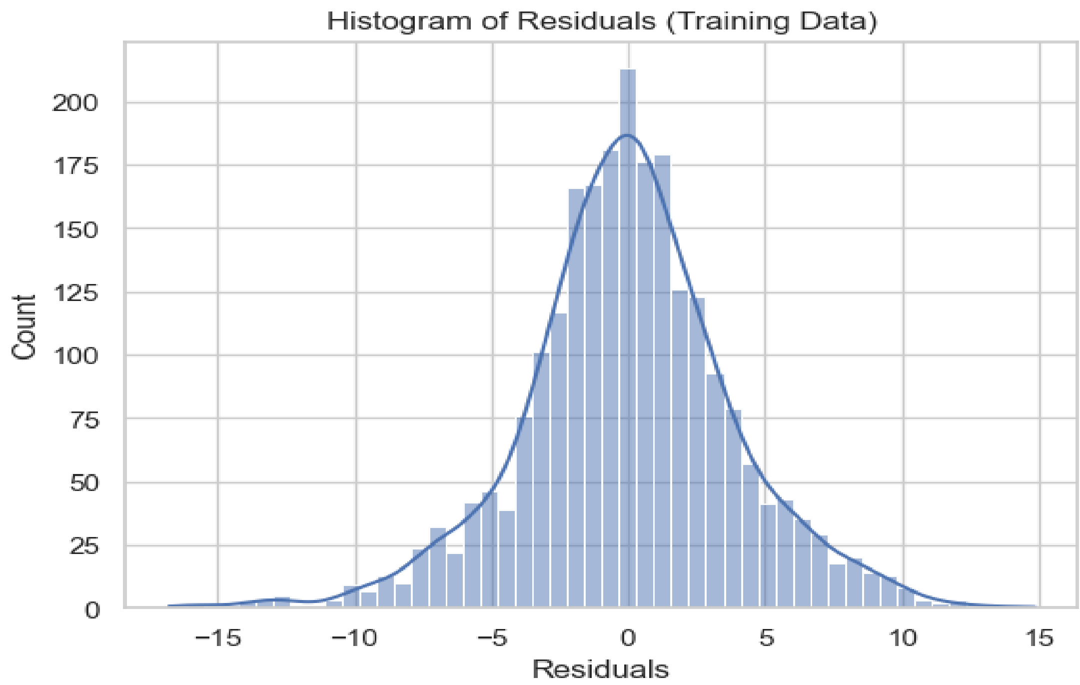
Residuals vs Fitted

Checking for linearity in the residuals.



- **Histogram**

Assessing the normality of residuals.



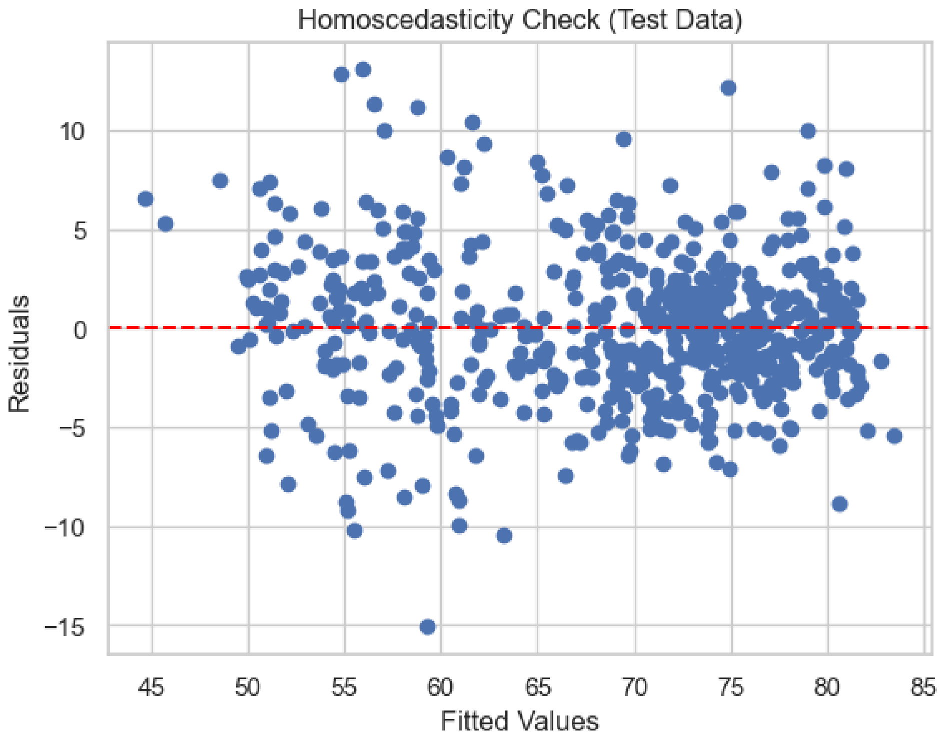
Residual Analysis on test data:

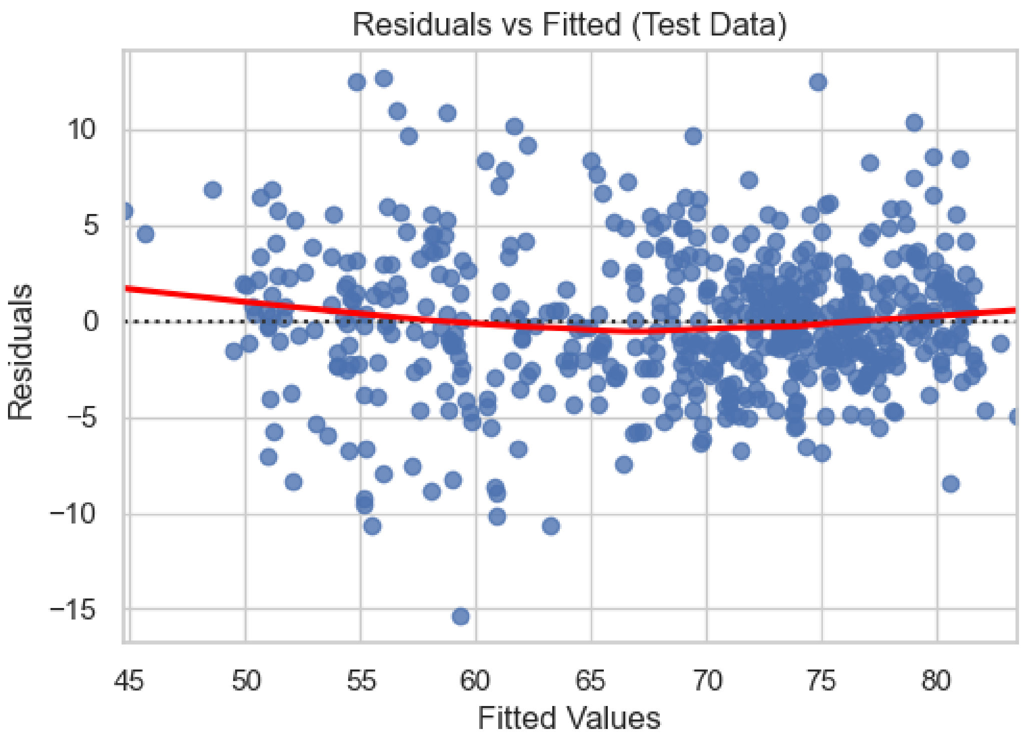
- **Durbin-Watson Test**: Assessing the independence of residuals.

Durbin-Watson Statistic (Test Data): 1.9757

A Durbin-Watson statistic of 1.9757 is very close to 2, which indicates that there is no significant autocorrelation in the residuals of the test data.

Residuals are approximately independent, suggesting that the model does not suffer from autocorrelation issues.


Homoscedasticity Check (Test Data)
Residuals
Fitted Values
10
5
0
−5
−10
−15
45
50
55
60
65
70
75
80
85


Residuals vs Fitted (Test Data)
Residuals
Fitted Values
10
5
0
−5
−10
−15
45
50
55
60
65
70
75
80

- **Removing the columns with high p values (p-values greater than 0.05)**

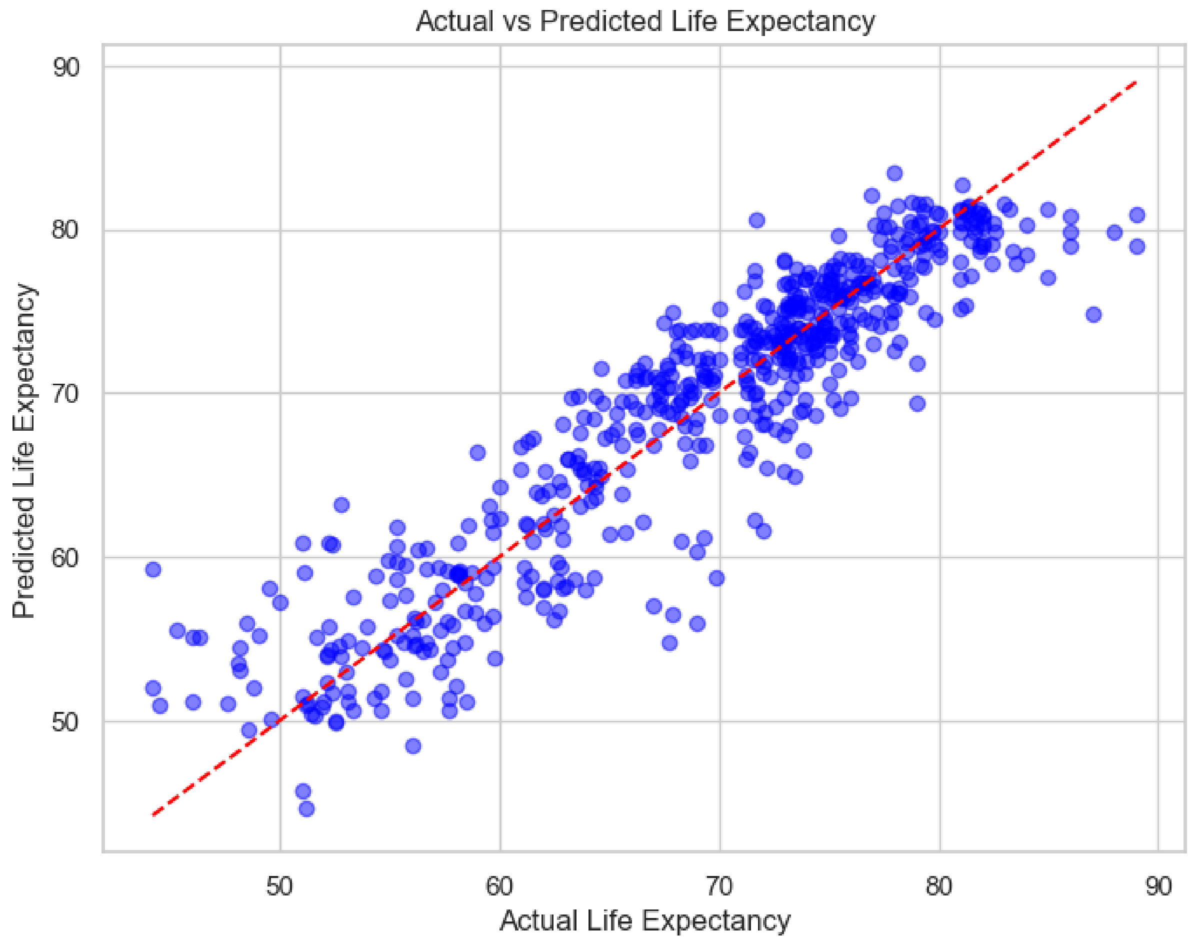
```
# Step 1: Drop the insignificant variables from the data
X_removed = X.drop(columns=['BMI', 'GDP', 'Population','Total expenditure'])

# Step 2: Add constant to the new training and test data
X_train_removed = sm.add_constant(X_removed.loc[X_train.index])
X_test_removed = sm.add_constant(X_removed.loc[X_test.index])


# Step 2: Display the remaining columns
print("Remaining Columns after Removal:")
print(X_removed.columns)
```

- **Rebuilding the model after removal of insignificant columns**

```
[365]: from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

# Define features and target
X = data.drop(columns=['Life expectancy'])
y = data['Life expectancy']

# Split the data into training and testing sets (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X_removed, y, test_size=0.2, random_state=42)

# Create and fit the linear regression model
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)
```

```
[365]: ▾ LinearRegression
LinearRegression()
```

```
[367]: import matplotlib.pyplot as plt
# Step 1: Predict on test data
y_pred = lr_model.predict(X_test)
# Plot actual vs predicted values
plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, color='blue', alpha=0.5)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], color='red', linestyle='--')
plt.xlabel("Actual Life Expectancy")
plt.ylabel("Predicted Life Expectancy")
plt.title("Actual vs Predicted Life Expectancy")
plt.show()
```




```
[377]: print("Linear Regression Equation:")
print(f"Life expectancy = {lr_model.intercept_:.2f} + ", end="")
for feature, coef in zip(X_removed.columns, lr_model.coef_):
    print(f"({coef:.4f} * {feature}) + ", end="")
print("...")

Linear Regression Equation:
Life expectancy = 61.39 + (-0.0181 * Adult Mortality) + (0.0856 * Alcohol) + (0.0020 * percentage expenditure) + (-0.0279 * Hepatitis B) + (-0.0012 * Measles) + (0.0270 * Polio) + (0.0567 * Diphtheria) + (-5.4
209 * HIV/AIDS) + (-0.1795 * thinness 5-9 years) + (8.4195 * Income composition of resources) + (0.3144 * Schooling) + ...
```

- **Linear Regression Equation (After removal of insignificant columns)**

Life expectancy = 61.39 + (-0.0181 * Adult Mortality) + (0.0856 * Alcohol) + (0.0020 * percentage expenditure) + (-0.0279 * Hepatitis B) + (-0.0012 * Measles) + (0.0270 * Polio) + (0.0567 * Diphtheria) + (-5.4209 * HIV/AIDS) + (-0.1795 * thinness 5-9 years) + (8.4195 * Income composition of resources) + (0.3144 * Schooling) + ...

- **Interpretation of regression coefficients:**

Intercept (61.39):

If all predictor variables are zero (hypothetically), the model predicts a baseline life expectancy of 61.39 years. Though not practically meaningful, it anchors the model.

Key Predictor Interpretations (Slope Coefficients):

- Adult Mortality (−0.0181):
  For each 1-unit increase in adult mortality, life expectancy decreases by 0.0181 years, holding other factors constant.
- Alcohol (0.0856):
  Higher alcohol consumption is associated with a slight increase in life expectancy, possibly reflecting better healthcare in moderate-drinking populations.
- Percentage Expenditure (0.0020):
  A higher percentage of GDP spent on health correlates with increased life expectancy, though the effect per unit is small.
- Hepatitis B (−0.0279):
  A higher rate of Hepatitis B vaccination is surprisingly associated with a slight decrease, possibly due to collinearity or confounding.
- Measles (−0.0012):
  More measles cases correlate with lower life expectancy, reflecting poorer healthcare access.
- Polio (0.0270) & Diphtheria (0.0567):
  Higher immunization rates are associated with higher life expectancy, indicating better public health infrastructure.
- HIV/AIDS (−5.4209):
  A very strong negative effect — every unit increase in HIV/AIDS rate reduces life expectancy by 5.42 years.
- Thinness 5–9 Years (−0.1795):
  Higher child malnutrition rates lead to lower life expectancy.
- Income Composition of Resources (8.4195):
  Strongest positive influence — higher income equality or access to resources boosts life expectancy by over 8 years.
- Schooling (0.3144):
  Each additional year of schooling increases life expectancy by 0.31 years, showing the power of education on health.

- ## **Model summary after removal of insignificant columns:**

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Life expectancy   R-squared:                       0.837
Model:                            OLS   Adj. R-squared:                  0.836
Method:                 Least Squares   F-statistic:                     1091.
Date:                Sun, 04 May 2025   Prob (F-statistic):               0.00
Time:                        18:41:44   Log-Likelihood:                -6500.8
No. Observations:                2350   AIC:                         1.303e+04
Df Residuals:                    2338   BIC:                         1.309e+04
Df Model:                          11
Covariance Type:            nonrobust
===================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------------
const                           61.3888      0.750     81.899      0.000      59.919      62.859
Adult Mortality                 -0.0181      0.001    -20.009      0.000      -0.020      -0.016
Alcohol                          0.0856      0.025      3.409      0.001       0.036       0.135
percentage expenditure           0.0020      0.000      8.075      0.000       0.002       0.002
Hepatitis B                     -0.0279      0.006     -4.598      0.000      -0.040      -0.016
Measles                         -0.0012      0.000     -4.742      0.000      -0.002      -0.001
Polio                            0.0270      0.010      2.797      0.005       0.008       0.046
Diphtheria                       0.0567      0.010      5.740      0.000       0.037       0.076
HIV/AIDS                        -5.4209      0.163    -33.243      0.000      -5.741      -5.101
thinness 5-9 years              -0.1795      0.025     -7.258      0.000      -0.228      -0.131
Income composition of resources  8.4195      0.740     11.377      0.000       6.968       9.871
Schooling                        0.3144      0.050      6.284      0.000       0.216       0.413
==============================================================================
Omnibus:                       70.597   Durbin-Watson:                   2.031
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              156.728
Skew:                          -0.146   Prob(JB):                     9.27e-35
Kurtosis:                       4.231   Cond. No.                     5.16e+03
==============================================================================
```

The multiple linear regression model explains 83.7% of the variance in life expectancy ($R^2 = 0.837$), indicating a strong fit. All predictors are statistically significant ($p < 0.05$), suggesting they meaningfully contribute to explaining life expectancy. The Durbin-Watson statistic (2.031) indicates no autocorrelation in residuals. However, the Shapiro-Wilk and Jarque-Bera tests ($p < 0.001$) suggest the residuals deviate from normality, although this is often tolerable in large samples. Overall, the model is statistically robust with meaningful predictors.

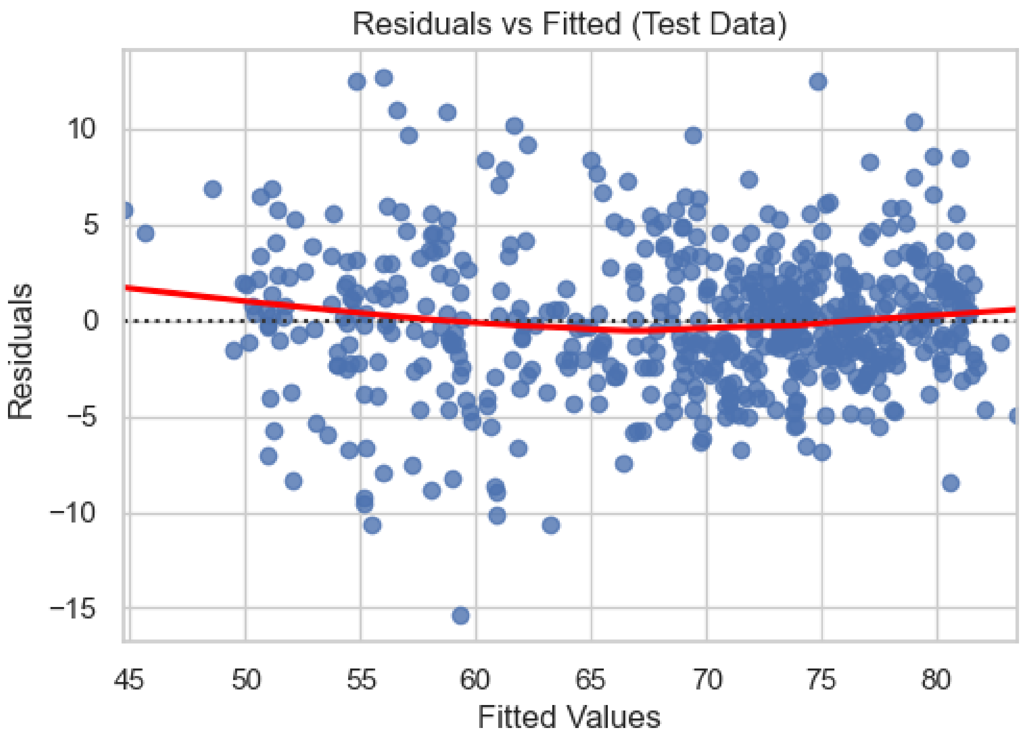
- ## **Model assumptions:**

1. Linearity

Assumption: The relationship between independent variables and the dependent variable is linear.

Check: Residuals vs Fitted plot showed no major curvature or systematic pattern.

Conclusion: Linearity assumption is reasonably satisfied.



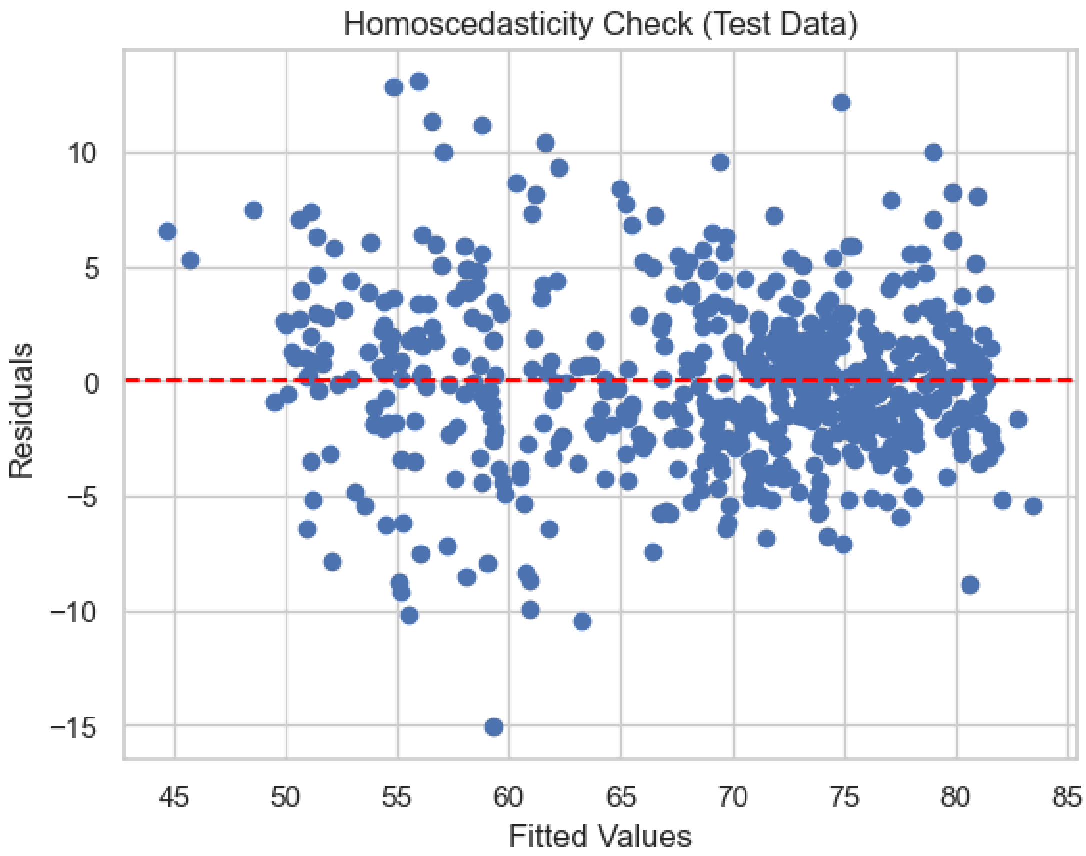
2. Independence of Errors

Assumption: Residuals (errors) are independent of each other.
Check:

- Durbin-Watson Statistic (Training = 2.03, Test = 1.98)
  Conclusion: Values near 2 indicate no autocorrelation, so this assumption is met.

3. Homoscedasticity (Constant Variance of Errors)

Assumption: Residuals have constant variance across all levels of the predicted values.
Check: Residuals vs Fitted plots show no funnel-shaped patterns.
Conclusion: Homoscedasticity is reasonably satisfied.



4. Normality of Residuals

Assumption: Residuals are normally distributed.
Check:

- Histogram and Q-Q Plot show some deviation.
- Shapiro-Wilk Test ($p < 0.001$) and Jarque-Bera Test ($p < 0.001$) suggest non-normality. **Conclusion:** Minor violation observed; however, with a **large sample size (n = 2350)**, normality is **less critical** due to the **Central Limit Theorem**.

- ### 4. Multicollinearity (No High Correlation Among Independent Variables)

Assumption: Independent variables should not be highly correlated with each other (i.e., no multicollinearity).
Check: Variance Inflation Factor (VIF) was calculated for all predictors. A VIF below 5 generally indicates no multicollinearity concern.
Conclusion: VIF values for all features were low to moderate, with no values exceeding

critical thresholds. Therefore, multicollinearity is not a concern in this model.

- ## Model Evaluation metric’s:

```
•[58]: from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

# Assuming y_test and y_pred are already defined
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print(f"Mean Absolute Error (MAE): {mae:.2f}")
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"Root MSE (RMSE): {rmse:.2f}")
print(f"R-squared (R²): {r2:.3f}")
from sklearn.metrics import r2_score

# Inputs:
# y_test = actual values
# y_pred = predicted values from model
# n = number of observations
# p = number of predictors
r2 = r2_score(y_test, y_pred)
n = len(y_test)          # number of observations
p = X_test.shape[1]      # number of predictors
adjusted_r2 = 1 - (1 - r2) * ((n - 1) / (n - p - 1))
print("Adjusted R²:", adjusted_r2)

Mean Absolute Error (MAE): 2.76
Mean Squared Error (MSE): 13.47
Root MSE (RMSE): 3.67
R-squared (R²): 0.844
Adjusted R²: 0.8412331196828167
```

Model Evaluation Interpretation:

- **Mean Absolute Error (MAE)**: 2.75 – On average, the model's predictions are off by 2.75 years of life expectancy. This is quite reasonable.
- **Mean Squared Error (MSE)**: 13.43 – This indicates that on average, the squared differences between the actual and predicted values are 13.43. The squared error penalty is higher for larger errors.
- **Root Mean Squared Error (RMSE)**: 3.66 – This indicates that typical prediction errors are around 3.66 years, which is a reasonable error in the context of life expectancy.
- **R-squared (R^2)**: 0.845 – The model explains 84.5% of the variability in life expectancy, suggesting a strong fit to the data.
- **Adjusted R^2**: 0.841 – Adjusted R^2 accounts for the number of predictors used in the model, slightly adjusting the R^2 value down, but it still shows a very strong fit.

Metric	Value
RMSE	3.66
MAE	2.75
R^2 Score	0.845
Adjusted R^2	0.841

Conclusion and Recommendations

- ### Summary of Findings
  - A multiple linear regression model was developed to predict life expectancy based on socio-economic and health indicators.
  - The final model explains 83.7% of the variance in life expectancy ($R^2 = 0.837$), showing a strong fit.
  - All included predictors are statistically significant ($p < 0.05$), and the model evaluation metrics (MAE = 2.75, RMSE = 3.66) indicate reasonably accurate predictions.
  - The most influential variables include:
  - HIV/AIDS (strong negative impact),
  - Income composition of resources (strong positive impact),
  - Schooling (moderate positive impact).
- ### Limitations
  - Normality assumption of residuals is violated (Shapiro-Wilk $p < 0.001$), though this is less concerning due to the large sample size.
  - Potential data quality issues or missing variables (e.g., environmental factors, healthcare access) may limit model accuracy.

- **Recommendations for Future Work**

  - Conduct feature engineering and explore non-linear models (e.g., Random Forest, Gradient Boosting) for improved accuracy.
  - Include additional variables such as healthcare quality, dietary patterns, or pollution levels.
  - Apply cross-validation techniques for more robust performance estimation.
  - Address residual normality using transformations or by applying generalized linear models (GLMs).