Project 2- 430- An Investigation Into Factors Determining Life Expectancy Differences Across Countries

Names :

1. **Dishant Vaishnav**- 106545424
2. **Ketki Sawant**- 506353861
3. **Praharsh Mehrotra**- 906306085
4. **Teesta Bose**- 206549884

Motivation-

Life expectancy is one of the most important indicators of a country's overall development and quality of life. It reflects not just the state of healthcare but also how social and economic factors shape people's well-being. However, despite progress in many areas, there are still huge gaps in life expectancy between different regions and populations. Understanding why these differences exist and predicting life expectancy more accurately can help governments and organizations make better decisions about where to focus their resources.

This project is motivated by the desire to uncover what really drives life expectancy. By analyzing a combination of health-related factors, economic indicators, and lifestyle variables, I aim to identify which ones matter the most and how they interact. Using regression models and advanced statistical techniques, the goal is to build a reliable model that doesn't just fit the data but also makes meaningful predictions.

Ultimately, this project is about more than just numbers. It's about contributing to a better understanding of the challenges different communities face and finding insights that could lead to tangible improvements in people's lives.

Data Description-

The data for this project has been extracted from the Global Health Observatory (GHO) data repository under World Health Organization (WHO), that 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. (https://www.who.int/data/gho)

Life_expectancy: Average number of years a person is expected to live, reflecting overall health and quality of life in a population. It summarizes the mortality pattern that prevails across all age groups

Adult_Mortality: Probability of dying between ages 15 and 60 per 1000 population, a measure of adult health risks. Disease burden from non-communicable diseases among adults - the most economically productive age span - is rapidly increasing in developing countries due to aging and epidemiological transitions.

Infant_deaths: Infant mortality rate is the probability of a child born in a specific year or period dying before reaching the age of one, if subject to age-specific mortality rates of that period.

Alcohol: Per capita alcohol consumption (liters of pure alcohol), reflecting pattern and trends of alcohol consumption in the adult population (15 years of age and older).

Percentage_expenditure: Level of total expenditure on health (THE) expressed as a percentage of gross domestic product (GDP). It provides information on the level of resources channeled to health relative to a country's wealth.

Hepatitis_B: Immunization coverage (%) for Hepatitis B among children, a measure of preventive healthcare. The percentage of one-year-olds who have received three doses of hepatitis B containing vaccines in a given year.

Measles: Number of confirmed measles cases per 1000 population, including those confirmed clinically, epidemiologically, or by laboratory investigation, indicating disease burden and immunization efforts.

BMI: Average body mass index of the population, representing nutritional status and obesity levels. Under_five_deaths: The probability of a child born in a specific year or period dying before reaching the age of five, reflecting the social, economic and environmental conditions in which children (and others in society) live, including their health care.

Polio: The percentage of one-year-olds who have received three doses of polio containing vaccine in a given year.

Total_expenditure: Level of general government expenditure on health (GGHE) expressed as a percentage of total expenditure on health (THE).

Diphtheria: The percentage of one-year-olds who have received three doses of the combined diphtheria, tetanus toxoid and pertussis

containing vaccine in a given year.

HIV: Deaths per 1000 live births due to HIV/AIDS, reflecting the disease's prevalence and healthcare response.

GDP: Gross Domestic Product per capita, representing the economic strength and wealth of a country.

Population: Total population size, indicating the density and scale of healthcare systems required.

Thinness_1_19_years: Percentage of defined population with a body mass index (BMI) less than 2 standard deviations below the median, reflecting undernutrition among children and adolescents.

Thinness_5_9_years: Percentage of the population aged 5–9 classified as thin, focusing specifically on younger children's nutrition levels.

Income_composition_of_resources: Human Development Index component based on income, showing resource availability and socioeconomic status.

Schooling: Average number of years of education, indicating literacy, awareness, and education's impact on health outcomes.

In [43]:
```
# Importing Libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from numpy import random; random.seed(10) # pre-setting seed
from scipy import stats
from scipy import optimize  # for Box-Cox calculations
from scipy.stats import norm
from matplotlib import rcParams # for ease of resizing plots
# Numpy version matters for scipy

# For model fitting
import statsmodels.api as sm
import statsmodels.stats.api as sms
import statsmodels.formula.api as smf
from statsmodels.stats.multicomp import MultiComparison
from statsmodels.robust.robust_linear_model import RLM
from fitter import Fitter # might require install, numpy version matters
from sklearn.model_selection import train_test_split, cross_val_score, KFold
from sklearn.linear_model import LinearRegression
from sklearn import preprocessing
from sklearn import metrics
from sklearn.impute import KNNImputer
from sklearn.preprocessing import LabelEncoder
from scipy.stats import binom


from sklearn.ensemble import RandomForestRegressor
from boruta import BorutaPy
from BorutaShap import BorutaShap # use .py file version


# Subset regressions & feature selection
from sklearn.linear_model import LinearRegression
from mlxtend.feature_selection import ExhaustiveFeatureSelector as EFS
from mlxtend.feature_selection import SequentialFeatureSelector as SFS

#from scipy.stats import binomtest, ks_2samp
from ydata_profiling import ProfileReport

from statsmodels.stats.diagnostic import linear_reset #FOR RAMSEY RESET

from statsmodels.stats.diagnostic import het_breuschpagan, het_white, het_goldfeldquandt
from statsmodels.iolib.summary2 import summary_col

#using R
import os
os.environ['R_HOME'] = '/Library/Frameworks/R.framework/Resources'
import rpy2
import warnings
warnings.filterwarnings('ignore')
from rpy2.robjects import pandas2ri
import rpy2.rinterface as rinterface
from rpy2 import robjects
pandas2ri.activate()
%load_ext rpy2.ipython
%reload_ext rpy2.ipython

import rpy2.robjects as ro
from rpy2.robjects.packages import importr
```

In [14]:
```
#Import data
raw_data = pd.read_csv("/Users/ketkisawant/Downloads/Life Expectancy Data .csv")
```

In [15]:
```
# Check for NA's

raw_data.isna().sum()
```

Out[15]:
```
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                                 0
GDP                               448
Population                        652
thinness_1_19_years                34
thinness_5_9_years                 34
Income_composition_of_resources   167
Schooling                         163
dtype: int64
```

The dataset has some NA values detected. Thus, we will now use KNN to impute the NA's.

In [16]:
```
#KNN
#replace a missing observation with a weighted average of the closest (using the Euclidean distance)
raw_data_knn_test = raw_data.copy() #creating a copy of cleaned data and assigning it to new data
#helps to ensure old data is not affected

# Separate 'Country' for safekeeping
country_column = raw_data_knn_test['Country']

# Drop 'Country' for KNN purposes but keep it for reattachment later
raw_data_knn_test = raw_data_knn_test.drop(['Country'], axis=1)

# One-hot encode 'Status'
raw_data_knn_test = pd.get_dummies(raw_data_knn_test, columns=['Status'])
# for the computer to understand, .get_dummies will create a new column for every continent and assign values 0

# Create a KNNImputer instance
imputer = KNNImputer(n_neighbors= 3)
#handle missing values by employing KNN imputation
#n=3 takes three near neighbours and takes average to fill it up

# Impute missing values
raw_data_knn_test_imputed = imputer.fit_transform(raw_data_knn_test)

# Convert the result back to a dataframe
raw_data_knn_test_imputed = pd.DataFrame(raw_data_knn_test_imputed, columns=raw_data_knn_test.columns)
raw_data_knn_test_imputed['Country'] = country_column

raw_data_knn_test_imputed
```

Out[16]:

	Year	Life_expectancy	Adult_Mortality	infant_deaths	Alcohol	percentage_expenditure	Hepatitis_B	Measles	BMI	under_f
0	2015.0	65.0	263.0	62.0	0.01	71.279624	65.0	1154.0	19.1	
1	2014.0	59.9	271.0	64.0	0.01	73.523582	62.0	492.0	18.6	
2	2013.0	59.9	268.0	66.0	0.01	73.219243	64.0	430.0	18.1	
3	2012.0	59.5	272.0	69.0	0.01	78.184215	67.0	2787.0	17.6	
4	2011.0	59.2	275.0	71.0	0.01	7.097109	68.0	3013.0	17.2	
...	...	...	...	...	...	...	...	...	...	
2933	2004.0	44.3	723.0	27.0	4.36	0.000000	68.0	31.0	27.1	
2934	2003.0	44.5	715.0	26.0	4.06	0.000000	7.0	998.0	26.7	
2935	2002.0	44.8	73.0	25.0	4.43	0.000000	73.0	304.0	26.3	
2936	2001.0	45.3	686.0	25.0	1.72	0.000000	76.0	529.0	25.9	
2937	2000.0	46.0	665.0	24.0	1.68	0.000000	79.0	1483.0	25.5	

2938 rows × 23 columns

In [18]:
```
# Adding Continents into our data

continents = pd.read_csv("/Users/ketkisawant/Downloads/continents-according-to-our-world-in-data.csv")
raw_data_knn_test_imputed = pd.merge(raw_data_knn_test_imputed, continents, on = 'Country', how = 'left')
raw_data_knn_test_imputed.head()
```

Out[18]:

	Year	Life_expectancy	Adult_Mortality	infant_deaths	Alcohol	percentage_expenditure	Hepatitis_B	Measles	BMI	under_five_
0	2015.0	65.0	263.0	62.0	0.01	71.279624	65.0	1154.0	19.1	
1	2014.0	59.9	271.0	64.0	0.01	73.523582	62.0	492.0	18.6	
2	2013.0	59.9	268.0	66.0	0.01	73.219243	64.0	430.0	18.1	
3	2012.0	59.5	272.0	69.0	0.01	78.184215	67.0	2787.0	17.6	
4	2011.0	59.2	275.0	71.0	0.01	7.097109	68.0	3013.0	17.2	

5 rows × 24 columns

Question 1)

Q1) a)

Boruta Algorithm

In [96]:
```
### Boruta using BorutaPy

raw_data_knn_test_imputed_2 = raw_data_knn_test_imputed.copy()

# Prepare data
raw_data_knn_test_imputed_y = raw_data_knn_test_imputed_2['Life_expectancy'].values
raw_data_knn_test_imputed_x = raw_data_knn_test_imputed_2.drop(['Life_expectancy', 'Country', 'Continent', 'Sta

#Boruta algorithm
rf = RandomForestRegressor(max_depth=5)
boruta = BorutaPy(rf, n_estimators='auto', verbose=2, random_state=10, two_step=True) # two_step uses Bonferron

# Run algorithm, X and y must by numpy arrays
boruta.fit(raw_data_knn_test_imputed_x, raw_data_knn_test_imputed_y)
```

```
Iteration:      1 / 100
Confirmed:      0
Tentative:      18
Rejected:       0
Iteration:      2 / 100
Confirmed:      0
Tentative:      18
Rejected:       0
Iteration:      3 / 100
Confirmed:      0
Tentative:      18
Rejected:       0
Iteration:      4 / 100
Confirmed:      0
```

```
Tentative:      18
Rejected:       0
Iteration:      5 / 100
Confirmed:      0
Tentative:      18
Rejected:       0
Iteration:      6 / 100
Confirmed:      0
Tentative:      18
Rejected:       0
Iteration:      7 / 100
Confirmed:      0
Tentative:      18
Rejected:       0
Iteration:      8 / 100
Confirmed:      15
Tentative:      3
Rejected:       0
Iteration:      9 / 100
Confirmed:      15
Tentative:      3
Rejected:       0
Iteration:      10 / 100
Confirmed:      15
Tentative:      3
Rejected:       0
Iteration:      11 / 100
Confirmed:      15
Tentative:      3
Rejected:       0
Iteration:      12 / 100
Confirmed:      15
Tentative:      3
Rejected:       0
Iteration:      13 / 100
Confirmed:      15
Tentative:      3
Rejected:       0
Iteration:      14 / 100
Confirmed:      15
Tentative:      3
Rejected:       0
Iteration:      15 / 100
Confirmed:      15
Tentative:      3
Rejected:       0
Iteration:      16 / 100
Confirmed:      16
Tentative:      2
Rejected:       0
Iteration:      17 / 100
Confirmed:      16
Tentative:      2
Rejected:       0
Iteration:      18 / 100
Confirmed:      16
Tentative:      2
Rejected:       0
Iteration:      19 / 100
Confirmed:      16
Tentative:      2
Rejected:       0
Iteration:      20 / 100
Confirmed:      16
Tentative:      2
Rejected:       0
Iteration:      21 / 100
Confirmed:      16
Tentative:      2
Rejected:       0
Iteration:      22 / 100
Confirmed:      16
Tentative:      2
Rejected:       0
Iteration:      23 / 100
Confirmed:      16
Tentative:      2
Rejected:       0
Iteration:      24 / 100
Confirmed:      16
Tentative:      2
Rejected:       0
Iteration:      25 / 100
```

```
Confirmed:      16
Tentative:      2
Rejected:       0
Iteration:      26 / 100
Confirmed:      16
Tentative:      2
Rejected:       0
Iteration:      27 / 100
Confirmed:      16
Tentative:      2
Rejected:       0
Iteration:      28 / 100
Confirmed:      16
Tentative:      2
Rejected:       0
Iteration:      29 / 100
Confirmed:      16
Tentative:      2
Rejected:       0
Iteration:      30 / 100
Confirmed:      16
Tentative:      2
Rejected:       0
Iteration:      31 / 100
Confirmed:      16
Tentative:      2
Rejected:       0
Iteration:      32 / 100
Confirmed:      16
Tentative:      2
Rejected:       0
Iteration:      33 / 100
Confirmed:      16
Tentative:      2
Rejected:       0
Iteration:      34 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      35 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      36 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      37 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      38 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      39 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      40 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      41 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      42 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      43 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      44 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      45 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
```

```
Iteration:      46 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      47 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      48 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      49 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      50 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      51 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      52 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      53 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      54 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      55 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      56 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      57 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      58 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      59 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      60 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      61 / 100
Confirmed:      16
Tentative:      1
Rejected:       1
Iteration:      62 / 100
Confirmed:      17
Tentative:      0
Rejected:       1


BorutaPy finished running.

Iteration:      63 / 100
Confirmed:      17
Tentative:      0
Rejected:       1
```

Out[96]:
```
BorutaPy(estimator=RandomForestRegressor(max_depth=5, n_estimators=116,
                                         random_state=RandomState(MT19937) at 0x2CE3853E6A8),
         n_estimators='auto',
         random_state=RandomState(MT19937) at 0x2CE3853E6A8, verbose=2)
```

In [97]:
```
# Compile feature decisions
boruta_keep = ['Confirmed' if c else
```

```
                'Tentative' if t else
                'Reject' for c,t in zip(boruta.support_, boruta.support_weak_)]

# Combine results and print
boruta_features = zip(raw_data_knn_test_imputed_2.drop(['Life_expectancy', 'Country', 'Continent', 'Status_Deve
for feat in boruta_features:
    print('Feature: {:<20} Rank: {},  Keep: {}'.format(feat[0], feat[1], feat[2]))
```

```
Feature: Adult_Mortality      Rank: 1,  Keep: Confirmed
Feature: infant_deaths        Rank: 1,  Keep: Confirmed
Feature: Alcohol              Rank: 1,  Keep: Confirmed
Feature: percentage_expenditure Rank: 1,  Keep: Confirmed
Feature: Hepatitis_B          Rank: 1,  Keep: Confirmed
Feature: Measles              Rank: 1,  Keep: Confirmed
Feature: BMI                  Rank: 1,  Keep: Confirmed
Feature: under_five_deaths    Rank: 1,  Keep: Confirmed
Feature: Polio                Rank: 1,  Keep: Confirmed
Feature: Total_expenditure    Rank: 1,  Keep: Confirmed
Feature: Diphtheria           Rank: 1,  Keep: Confirmed
Feature: HIV                  Rank: 1,  Keep: Confirmed
Feature: GDP                  Rank: 1,  Keep: Confirmed
Feature: Population           Rank: 2,  Keep: Reject
Feature: thinness_1_19_years  Rank: 1,  Keep: Confirmed
Feature: thinness_5_9_years   Rank: 1,  Keep: Confirmed
Feature: Income_composition_of_resources Rank: 1,  Keep: Confirmed
Feature: Schooling            Rank: 1,  Keep: Confirmed
```

In [7]:

```
# Boruta using BorutaShap
raw_data_knn_test_imputed_2_X2 = raw_data_knn_test_imputed_2.drop(['Life_expectancy', 'Country', 'Continent', '

# Run Boruta algorithm
Feature_Selector = BorutaShap(importance_measure='shap', classification=False)
Feature_Selector.fit(X=raw_data_knn_test_imputed_2_X2, y= raw_data_knn_test_imputed_y, n_trials=100, random_sta

# Plot results
Feature_Selector.plot(which_features='all')
plt.show()

# Print results
boruta_keep2 = ['Confirmed' if p in Feature_Selector.accepted else
                'Tentative' if p in Feature_Selector.tentative else
                'Reject' for p in Feature_Selector.all_columns]

boruta_features2 = zip(Feature_Selector.all_columns,
                       len(Feature_Selector.shap_values) - np.argsort(Feature_Selector.shap_values),
                       Feature_Selector.shap_values,
                       boruta_keep2)

for feat in boruta_features2:
    print('Feature: {:<20} Rank: {} ({:.3f}),  Keep: {}'.format(feat[0], feat[1], feat[2], feat[3]))
```

```
  0%|          | 0/100 [00:00<?, ?it/s]
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[7], line 6
      4 # Run Boruta algorithm
      5 Feature_Selector = BorutaShap(importance_measure='shap', classification=False)
----> 6 Feature_Selector.fit(X=raw_data_knn_test_imputed_2_X2, y= raw_data_knn_test_imputed_y, n_trials=100, ran
dom_state=90095)
      8 # Plot results
      9 Feature_Selector.plot(which_features='all')

File /opt/anaconda3/lib/python3.12/site-packages/BorutaShap.py:473, in BorutaShap.fit(self, X, y, sample_weight,
n_trials, random_state, sample, train_or_test, normalize, verbose, stratify)
    471         self.hits += hits
    472         self.history_hits = np.vstack((self.history_hits, self.hits))
--> 473         self.test_features(iteration=trial+1)
    475 self.store_feature_importance()
    476 self.calculate_rejected_accepted_tentative(verbose=verbose)

File /opt/anaconda3/lib/python3.12/site-packages/BorutaShap.py:934, in BorutaShap.test_features(self, iteration)
    919 def test_features(self, iteration):
    921     """
    922     For each feature with an undetermined importance perform a two-sided test of equality
    923     with the maximum shadow value to determine if it is statistcally better
   (...)
    931         Two arrays of the names of the accepted and rejected columns at that instance
    932     """
--> 934     acceptance_p_values = self.binomial_H0_test(self.hits,
    935                                                 n=iteration,
    936                                                 p=0.5,
    937                                                 alternative='greater')
    939     regect_p_values = self.binomial_H0_test(self.hits,
    940                                             n=iteration,
    941                                             p=0.5,
    942                                             alternative='less')
    944     # [1] as function returns a tuple

File /opt/anaconda3/lib/python3.12/site-packages/BorutaShap.py:885, in BorutaShap.binomial_H0_test(array, n, p,
alternative)
    878 @staticmethod
    879 def binomial_H0_test(array, n, p, alternative):
    880     """
    881     Perform a test that the probability of success is p.
    882     This is an exact, two-sided test of the null hypothesis
    883     that the probability of success in a Bernoulli experiment is p
    884     """
--> 885     return [binom_test(x, n=n, p=p, alternative=alternative) for x in array]

NameError: name 'binom_test' is not defined
```

In [8]:

```
### Boruta using BorutaShap
## Note: takes about 10 minutes


raw_data_knn_test_imputed_2_X2 = raw_data_knn_test_imputed_2.drop(['Life_expectancy', 'Country', 'Continent', '

# Run Boruta algorithm
Feature_Selector = BorutaShap(importance_measure='shap', classification=False)
Feature_Selector.fit(X=raw_data_knn_test_imputed_2_X2, y= raw_data_knn_test_imputed_y, n_trials=100, random_sta

# Plot results
Feature_Selector.plot(which_features='all')
plt.show()

# Print results
boruta_keep2 = ['Confirmed' if p in Feature_Selector.accepted else
                'Tentative' if p in Feature_Selector.tentative else
                'Reject' for p in Feature_Selector.all_columns]

boruta_features2 = zip(Feature_Selector.all_columns,
                       len(Feature_Selector.shap_values) - np.argsort(Feature_Selector.shap_values),
                       Feature_Selector.shap_values,
                       boruta_keep2)
for feat in boruta_features2:
    print('Feature: {:<20} Rank: {} ({:.3f}),  Keep: {}'.format(feat[0], feat[1], feat[2], feat[3]))
```

```
  0%|          | 0/100 [00:00<?, ?it/s]
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[8], line 9
      7 # Run Boruta algorithm
      8 Feature_Selector = BorutaShap(importance_measure='shap', classification=False)
----> 9 Feature_Selector.fit(X=raw_data_knn_test_imputed_2_X2, y= raw_data_knn_test_imputed_y, n_trials=100, ran
dom_state=90095)
     11 # Plot results
     12 Feature_Selector.plot(which_features='all')

File /opt/anaconda3/lib/python3.12/site-packages/BorutaShap.py:473, in BorutaShap.fit(self, X, y, sample_weight,
n_trials, random_state, sample, train_or_test, normalize, verbose, stratify)
    471         self.hits += hits
    472         self.history_hits = np.vstack((self.history_hits, self.hits))
--> 473         self.test_features(iteration=trial+1)
    475 self.store_feature_importance()
    476 self.calculate_rejected_accepted_tentative(verbose=verbose)

File /opt/anaconda3/lib/python3.12/site-packages/BorutaShap.py:934, in BorutaShap.test_features(self, iteration)
    919 def test_features(self, iteration):
    921     """
    922     For each feature with an undetermined importance perform a two-sided test of equality
    923     with the maximum shadow value to determine if it is statistcally better
   (...)
    931         Two arrays of the names of the accepted and rejected columns at that instance
    932     """
--> 934     acceptance_p_values = self.binomial_H0_test(self.hits,
    935                                                 n=iteration,
    936                                                 p=0.5,
    937                                                 alternative='greater')
    939     regect_p_values = self.binomial_H0_test(self.hits,
    940                                             n=iteration,
    941                                             p=0.5,
    942                                             alternative='less')
    944     # [1] as function returns a tuple

File /opt/anaconda3/lib/python3.12/site-packages/BorutaShap.py:885, in BorutaShap.binomial_H0_test(array, n, p,
alternative)
    878 @staticmethod
    879 def binomial_H0_test(array, n, p, alternative):
    880     """
    881     Perform a test that the probability of success is p.
    882     This is an exact, two-sided test of the null hypothesis
    883     that the probability of success in a Bernoulli experiment is p
    884     """
--> 885     return [binom_test(x, n=n, p=p, alternative=alternative) for x in array]

NameError: name 'binom_test' is not defined
```

These are the following conclusions from the Boruta Algorithm:

1. Important Features: The algorithm identified 18 features as important for the model. These include:

Health-related metrics: Adult_Mortality, infant_deaths, Hepatitis_B, Measles, Polio, HIV, BMI, under_five_deaths, and thinness indicators (1-19 years and 5-9 years). Economic indicators: GDP, Income_composition_of_resources, Schooling, and Total_expenditure. Other indicators: Alcohol, Diphtheria, percentage_expenditure. Rejected Features: The only feature rejected as unimportant is Population. This suggests that population size does not directly contribute to predicting life expectancy in this dataset.

2. Feature Importance Visualization:

The box plot shows the Z-scores of importance for the features. Features with green bars are confirmed important, and features with blue bars are unimportant or tentative. In this case: The confirmed important features have a distinct and consistent Z-score range. The rejected features (like Population) have lower Z-scores and are clearly separated. Tentative Features: None of the features remain tentative, indicating a clear categorization of all variables after the Boruta process.

3. Interpretation: The Boruta algorithm has effectively ranked and confirmed the most significant variables for predicting life expectancy. It highlights that health indicators (like Adult_Mortality and HIV) and socio-economic factors (like GDP and Schooling) play a crucial role. Meanwhile, the exclusion of Population suggests it may not have a strong direct correlation with life expectancy within the context of this dataset.

Top 10 Predictors from Boruta are:

1. Total_expenditure
2. thinness_5_9_years
3. Income_composition_of_resources
4. infant_deaths
5. Diphtheria
6. Polio

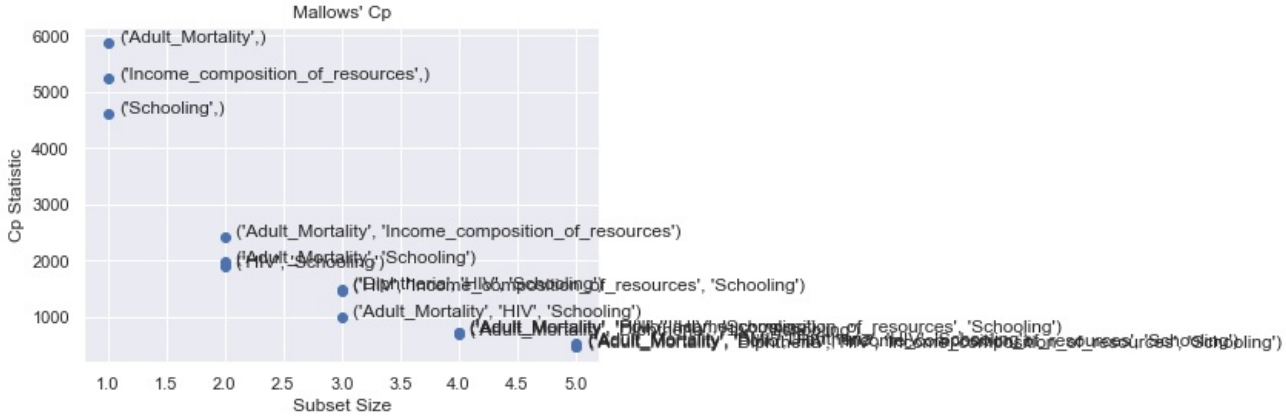
7. GDP
8. Alcohol
9. BMI
10. Measles

We now, look at Mallow's CP and Stepwise regression to further check for good quantitative predictors

Mallow's CP

In [98]:
```
# Mallow's CP
raw_data_knn_test_imputed_2_X2 = raw_data_knn_test_imputed_2.drop(['Life_expectancy', 'Country', 'Continent', 'S

#full regression
exp_reg = smf.ols('Life_expectancy ~ Adult_Mortality + infant_deaths + Alcohol + percentage_expenditure + Hepat
exp_reg_sig2 = exp_reg.mse_resid
exp_n, _ = raw_data_knn_test_imputed.shape
exp_reg.summary()

# Negative Mallows' cp, so we want largest
def negative_mallows(estimator, X, y):
    y_pred = estimator.predict(X)
    sse = np.sum((y - y_pred) ** 2)
    p = estimator.n_features_in_ + 1 # number features doesn't include bias term
    cp = sse / exp_reg_sig2 + 2*p - exp_n
    return -cp

# Set up regression
exp_reg_Y = raw_data_knn_test_imputed['Life_expectancy'].values
exp_reg_X = raw_data_knn_test_imputed.drop(['Life_expectancy', 'Country', 'Continent', 'Status_Developing', 'Sta
lr = LinearRegression(fit_intercept=True)

# Using ExhaustiveFeatureSelector from mlxtend
exp_efs = EFS(lr,
              min_features=1,
              max_features=5,
              scoring=negative_mallows,
              cv=None)


# Run regressions
exp_efs.fit(raw_data_knn_test_imputed_2_X2, raw_data_knn_test_imputed_y)
print('Best subset:', [exp_efs.feature_names[i] for i in exp_efs.best_idx_])

# Extract Mallows' Cp data
exp_efs_features = [exp_efs.subsets_[p]['feature_names'] for p in exp_efs.subsets_]
exp_efs_subset = [len(exp_efs.subsets_[p]['feature_idx']) for p in exp_efs.subsets_]
exp_efs_cp = [ -exp_efs.subsets_[p]['avg_score'] for p in exp_efs.subsets_]
exp_mallows = pd.DataFrame({'subsets':exp_efs_subset, 'Cp':exp_efs_cp}, index=exp_efs_features)

# Top 3 best, by subset
exp_mallows_best = exp_mallows.groupby('subsets')[['subsets','Cp']].apply(lambda x: x.nsmallest(3, 'Cp'))
exp_mallows_best

# Plotting best subsets
plt.scatter(exp_mallows_best['subsets'], exp_mallows_best['Cp'])
plt.title("Mallows' Cp")
plt.xlabel('Subset Size')
plt.ylabel('Cp Statistic')
for i,sub in enumerate(exp_mallows_best['subsets']):
    subset = exp_mallows_best.index[i][1]
    plt.annotate(subset, (sub + 0.1, exp_mallows_best.iloc[i]['Cp'] + 0.3)) # to plot labels
plt.tight_layout()
plt.show()

# And the associated sorted data frame
exp_mallows_best.sort_values('Cp')
```

```
Features: 12615/12615
Best subset: ['Adult_Mortality', 'Diphtheria', 'HIV', 'Income_composition_of_resources', 'Schooling']
```



Out[98]:

		subsets	Cp
subsets			
5	**(Adult_Mortality, Diphtheria, HIV, Income_composition_of_resources, Schooling)**	5	469.140211
	(Adult_Mortality, Polio, HIV, Income_composition_of_resources, Schooling)	5	493.154932
	(Adult_Mortality, BMI, Diphtheria, HIV, Schooling)	5	527.289601
4	**(Adult_Mortality, Diphtheria, HIV, Schooling)**	4	694.350171
	(Adult_Mortality, HIV, Income_composition_of_resources, Schooling)	4	725.879930
	(Adult_Mortality, Polio, HIV, Schooling)	4	734.432236
3	**(Adult_Mortality, HIV, Schooling)**	3	1005.673970
	(HIV, Income_composition_of_resources, Schooling)	3	1463.864175
	(Diphtheria, HIV, Schooling)	3	1503.476882
2	**(HIV, Schooling)**	2	1890.598862
	(Adult_Mortality, Schooling)	2	1974.941901
	(Adult_Mortality, Income_composition_of_resources)	2	2430.273512
1	**(Schooling,)**	1	4619.414173
	(Income_composition_of_resources,)	1	5230.699260
	(Adult_Mortality,)	1	5876.474729

Based on the Mallow's CP, the best subset of variables to explain Life Expectancy variations are ['Adult_Mortality', 'Diphtheria', 'HIV', 'Income_composition_of_resources', 'Schooling']. Do however, note that this is assuming the top 5 predictors since we are unable to compute 10 to computational power limitations.s

Stepwise Regression

Next, we run a stepwise regression which adds and subtracts predictors iteratively and determines the best set of predictors to be used. We will run both a forward and backward selection algorithm for robustness of our results.

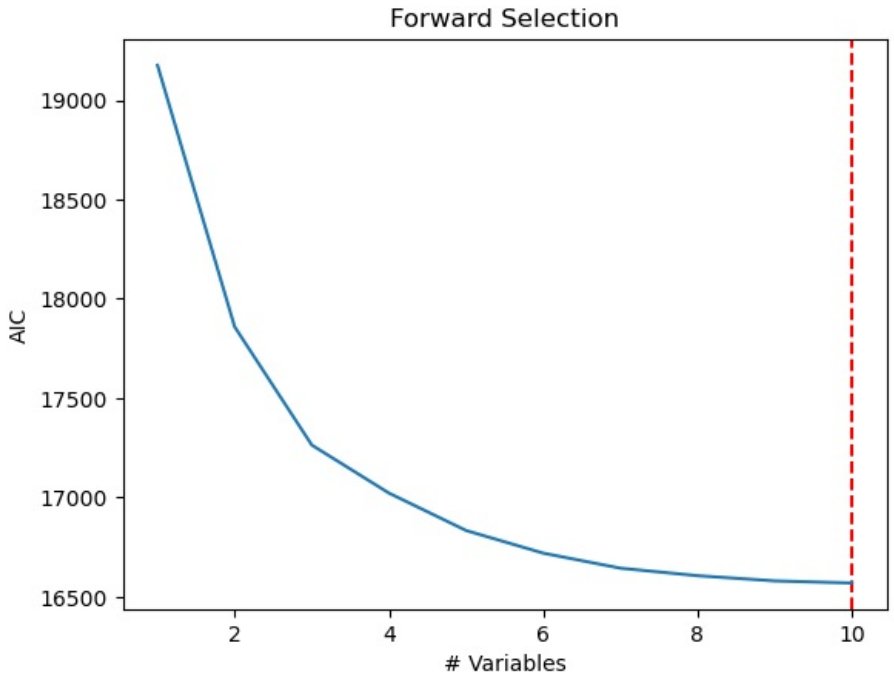
Forward Selection

In [19]:
```
# Example 30: Stepwise Regressions (Family Income)

# Negative AIC, so we want largest
def negative_AIC(estimator, X, y):
    model = sm.OLS(endog=y, exog=sm.add_constant(X)).fit()
    return -model.aic

# Could also use (negative) BIC: (-)model.bic
# Note that BIC is more restrictive
```

```
# Set up regression
exp_reg_Y = raw_data_knn_test_imputed['Life_expectancy'].values
exp_reg_X = raw_data_knn_test_imputed.drop(['Life_expectancy', 'Country', 'Continent', 'Status_Developing', 'Sta
lr = LinearRegression(fit_intercept=True)


### FIRST: Forward Selection

# Using `SequentialFeatureSelector` from `mlxtend`
exp_forward = SFS(lr,
                  forward=True,
                  k_features=(1,10),
                  scoring=negative_AIC,
                  cv=None)

# Run regressions
exp_forward.fit(exp_reg_X, exp_reg_Y)
print('Best subset:', [n for n in exp_forward.k_feature_names_], '\n')

# Extract subset data
exp_fwd_metrics = exp_forward.get_metric_dict()
exp_fwd_features = [exp_fwd_metrics[p]['feature_names'] for p in exp_fwd_metrics]
exp_fwd_subset = [len(exp_fwd_metrics[p]['feature_idx']) for p in exp_fwd_metrics]
exp_fwd_aic = [ -exp_fwd_metrics[p]['avg_score'] for p in exp_fwd_metrics]
exp_forward_df = pd.DataFrame({'subsets':exp_fwd_subset, 'AIC':exp_fwd_aic}, index=exp_fwd_features)

# Ranked by AIC
print(exp_forward_df.sort_values(by='AIC'))

# Plotting
plt.axvline(x=np.argmin(exp_forward_df['AIC']) + 1, linestyle='--', color='red')
plt.plot(exp_forward_df['subsets'], exp_forward_df['AIC'])
plt.title('Forward Selection')
plt.xlabel('# Variables')
plt.ylabel('AIC')
plt.show()

# Best model has 3 features
```

```
Best subset: ['Adult_Mortality', 'BMI', 'Polio', 'Total_expenditure', 'Diphtheria', 'HIV', 'GDP', 'thinness_1_19
_years', 'Income_composition_of_resources', 'Schooling']
```

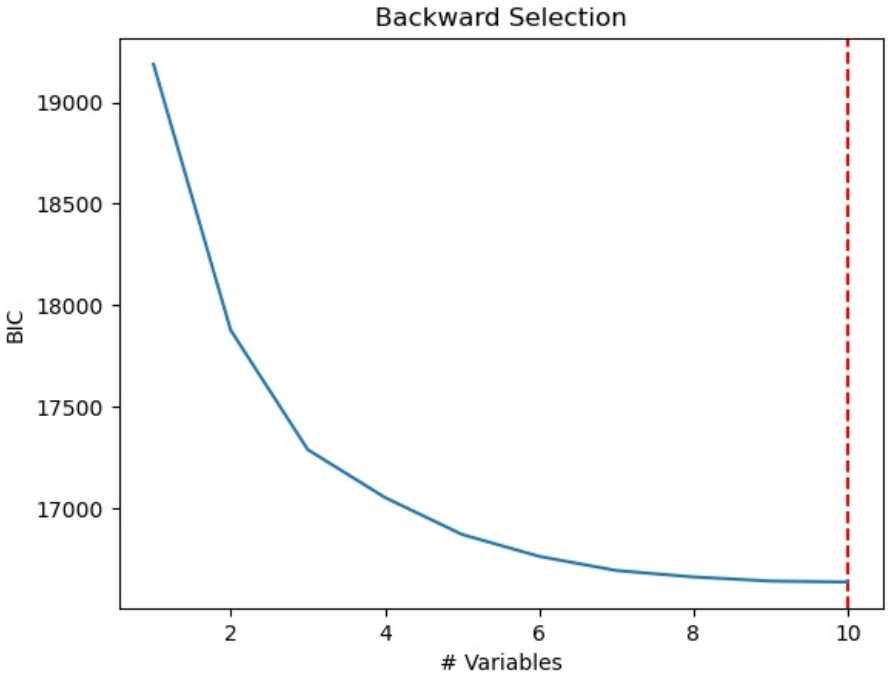
	subsets	AIC
(Adult_Mortality, BMI, Polio, Total_expenditure...	10	16569.675643
(Adult_Mortality, BMI, Polio, Diphtheria, HIV, ...	9	16580.355769
(Adult_Mortality, BMI, Polio, Diphtheria, HIV, ...	8	16606.531270
(Adult_Mortality, BMI, Diphtheria, HIV, GDP, In...	7	16644.276829
(Adult_Mortality, BMI, Diphtheria, HIV, Income_...	6	16719.926622
(Adult_Mortality, Diphtheria, HIV, Income_compo...	5	16833.907601
(Adult_Mortality, Diphtheria, HIV, Schooling)	4	17022.225290
(Adult_Mortality, HIV, Schooling)	3	17263.959153
(HIV, Schooling)	2	17859.051310
(Schooling,)	1	19175.292845



Backward Selection

In [20]:
```
### SECOND: Backward Selection

# Negative BIC, so we want largest
def negative_BIC(estimator, X, y):
    model = sm.OLS(endog=y, exog=sm.add_constant(X)).fit()
    return -model.bic

# Initialize sequential selection
exp_backward = SFS(lr,
                   forward=True,
                   k_features=(1,10),
                   scoring=negative_BIC, # trying with BIC
                   cv=None)

# Run regressions
exp_backward.fit(exp_reg_X, exp_reg_Y)
print('Best subset:', [n for n in exp_backward.k_feature_names_], '\n')

# Extract subset data
exp_bwd_metrics = exp_backward.get_metric_dict()
exp_bwd_features = [exp_bwd_metrics[p]['feature_names'] for p in exp_bwd_metrics]
exp_bwd_subset = [len(exp_bwd_metrics[p]['feature_idx']) for p in exp_bwd_metrics]
exp_bwd_bic = [ -exp_bwd_metrics[p]['avg_score'] for p in exp_bwd_metrics]
exp_backward_df = pd.DataFrame({'subsets':exp_bwd_subset, 'BIC':exp_bwd_bic}, index=exp_bwd_features)

# Ranked by BIC
print(exp_backward_df.sort_values(by='BIC'))

# Plotting
plt.axvline(x=np.argmin(exp_backward_df['BIC']) + 1, linestyle='--', color='red')
plt.plot(exp_backward_df['subsets'], exp_backward_df['BIC'])
plt.title('Backward Selection')
plt.xlabel('# Variables')
plt.ylabel('BIC')
plt.show()

# Again, best model has 3 features
```

```
Best subset: ['Adult_Mortality', 'BMI', 'Polio', 'Total_expenditure', 'Diphtheria', 'HIV', 'GDP', 'thinness_1_19
_years', 'Income_composition_of_resources', 'Schooling']

                                                  subsets           BIC
(Adult_Mortality, BMI, Polio, Total_expenditure...      10  16635.515971
(Adult_Mortality, BMI, Polio, Diphtheria, HIV, ...       9  16640.210613
(Adult_Mortality, BMI, Polio, Diphtheria, HIV, ...       8  16660.400629
(Adult_Mortality, BMI, Diphtheria, HIV, GDP, In...       7  16692.160704
(Adult_Mortality, BMI, Diphtheria, HIV, Income_...       6  16761.825013
(Adult_Mortality, Diphtheria, HIV, Income_compo...       5  16869.820508
(Adult_Mortality, Diphtheria, HIV, Schooling)            4  17052.152711
(Adult_Mortality, HIV, Schooling)                        3  17287.901090
(HIV, Schooling)                                         2  17877.007763
(Schooling,)                                             1  19187.263814
```



Identifying Key Quantitative Variables

Based on the Boruta algorithm, Mallow's CP, Forward and Backward Selection algorithms, we have identified the following top 10 quantitative variables to be included in our model:

1. HIV
2. Income Composition Of Resources
3. Adult Mortality
4. Schooling
5. BMI
6. Total Expenditure
7. Polio
8. Diphtheria
9. Thinness 1-19 Years
10. GDP

Next, we will examine the factor variables to be included in our model

Q1) b)

To decide on which factor variables to be included in our model, we first look at a MANOVA + Quantitative variables model to determine statistically significant factor variables

Evaluating MANOVA + Quant Models

In [21]:
```
# Estimating Manova + Quant Model
raw_data_knn_test_imputed['Year'] = raw_data_knn_test_imputed['Year'].astype('str')
maova_quant_total = smf.ols('Life_expectancy~ HIV + Income_composition_of_resources + Schooling + Adult_Mortali
print(maova_quant_total.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Life_expectancy   R-squared:                       0.848
Model:                            OLS   Adj. R-squared:                  0.846
Method:                 Least Squares   F-statistic:                     521.5
Date:                Wed, 20 Nov 2024   Prob (F-statistic):               0.00
Time:                        14:11:48   Log-Likelihood:                -8022.8
No. Observations:                2938   AIC:                         1.611e+04
Df Residuals:                    2906   BIC:                         1.630e+04
Df Model:                          31
Covariance Type:            nonrobust
=====================================================================================================
                                        coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------------------
Intercept                            54.0379      0.629     85.857      0.000      52.804      55.272
Continent[T.Asia]                     4.2809      0.232     18.448      0.000       3.826       4.736
Continent[T.Europe]                   4.1348      0.322     12.846      0.000       3.504       4.766
Continent[T.North America]            5.7190      0.291     19.646      0.000       5.148       6.290
Continent[T.Oceania]                  3.1464      0.376      8.370      0.000       2.409       3.884
Continent[T.South America]            4.3723      0.352     12.435      0.000       3.683       5.062
Year[T.2001.0]                       -0.4889      0.391     -1.250      0.211      -1.256       0.278
Year[T.2002.0]                       -0.5614      0.391     -1.435      0.152      -1.329       0.206
Year[T.2003.0]                       -0.6842      0.392     -1.748      0.081      -1.452       0.083
Year[T.2004.0]                       -0.6098      0.392     -1.554      0.120      -1.379       0.159
Year[T.2005.0]                       -0.5302      0.393     -1.350      0.177      -1.300       0.240
Year[T.2006.0]                       -0.6078      0.394     -1.541      0.123      -1.381       0.166
Year[T.2007.0]                       -0.6653      0.395     -1.684      0.092      -1.440       0.109
Year[T.2008.0]                       -0.4498      0.396     -1.135      0.257      -1.227       0.328
Year[T.2009.0]                       -0.2241      0.398     -0.564      0.573      -1.004       0.555
Year[T.2010.0]                       -0.2666      0.397     -0.671      0.502      -1.045       0.512
Year[T.2011.0]                       -0.2794      0.399     -0.700      0.484      -1.062       0.503
Year[T.2012.0]                       -0.2939      0.400     -0.734      0.463      -1.078       0.491
Year[T.2013.0]                        0.0195      0.396      0.049      0.961      -0.757       0.796
Year[T.2014.0]                        0.0610      0.401      0.152      0.879      -0.725       0.847
Year[T.2015.0]                        0.2743      0.401      0.683      0.494      -0.513       1.061
HIV                                  -0.3899      0.017    -22.817      0.000      -0.423      -0.356
Income_composition_of_resources       5.9545      0.595     10.002      0.000       4.787       7.122
Schooling                             0.6322      0.039     16.150      0.000       0.555       0.709
Adult_Mortality                      -0.0152      0.001    -20.215      0.000      -0.017      -0.014
BMI                                   0.0242      0.005      5.104      0.000       0.015       0.034
Total_expenditure                     0.0468      0.032      1.484      0.138      -0.015       0.109
Polio                                 0.0271      0.004      6.619      0.000       0.019       0.035
Diphtheria                            0.0287      0.004      7.011      0.000       0.021       0.037
thinness_1_19_years                  -0.0800      0.021     -3.740      0.000      -0.122      -0.038
GDP                                4.965e-05   6.16e-06      8.061      0.000    3.76e-05    6.17e-05
Status_Developing                    -1.7255      0.281     -6.144      0.000      -2.276      -1.175
==============================================================================
Omnibus:                      117.407   Durbin-Watson:                   0.662
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              358.907
Skew:                          -0.048   Prob(JB):                     1.16e-78
Kurtosis:                       4.710   Cond. No.                     2.51e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.51e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Based on the results, we can see that among the 3 factor variables, Status_Developing and Continent appear to have statistically significant impacts on Life Expectancy at a 5% level while Year does not seem to have a statistically significant impact. We also look at Boxplots and Tukey SHD multi comparisons for each factor variable.

1. Year

In [22]:
```
# Check distribution Of Factor variables
cols = ['Year', 'Continent', 'Status_Developing']
for col in cols:

    print(f"{raw_data_knn_test_imputed[col].value_counts()} \n\n")
```

```
Year
2013.0    193
2015.0    183
2014.0    183
2012.0    183
2011.0    183
2010.0    183
2009.0    183
2008.0    183
2007.0    183
2006.0    183
2005.0    183
2004.0    183
2003.0    183
2002.0    183
2001.0    183
2000.0    183
Name: count, dtype: int64


Continent
Africa           864
Asia             736
Europe           642
North America    338
South America    192
Oceania          166
Name: count, dtype: int64


Status_Developing
1.0    2426
0.0     512
Name: count, dtype: int64
```

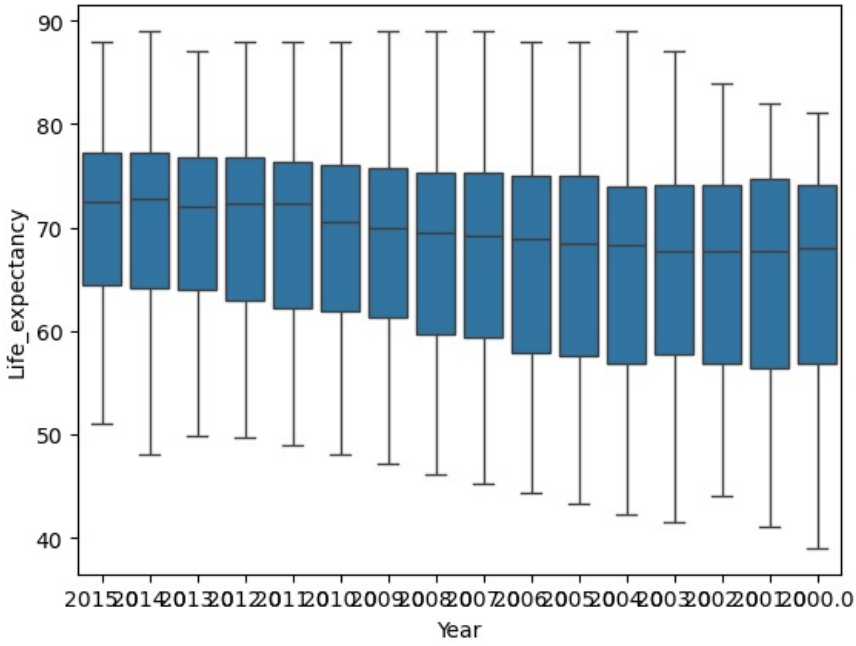
Based on this, it is clear that the Continent factor is unevenly distributed which may cause our regression to be biased. Thus, we will only consider data points from Asia, Europe and Africa in our data. While, this is also true for Status_Developing, we will keep it as it is given that this may be the real nature of the world and population data itself.

In [23]:
```
# Remove North America, SA, Oceania

raw_data_knn_test_imputed = raw_data_knn_test_imputed[raw_data_knn_test_imputed['Continent'].isin(['Africa','As
```

In [24]:
```
#Group By Year
raw_data_knn_test_imputed.groupby('Year')['Life_expectancy'].mean()
```

Out[24]:
```
Year
2000.0    65.220000
2001.0    65.715714
2002.0    65.875000
2003.0    66.016429
2004.0    66.324286
2005.0    66.952857
2006.0    67.380714
2007.0    67.835000
2008.0    68.234286
2009.0    68.833571
2010.0    69.149286
2011.0    69.693571
2012.0    70.060000
2013.0    70.504695
2014.0    70.754286
2015.0    70.759286
Name: Life_expectancy, dtype: float64
```

In [25]:
```
# Plotting Boxplot

sns.boxplot(x='Year', y='Life_expectancy', data=raw_data_knn_test_imputed)
plt.show()
```

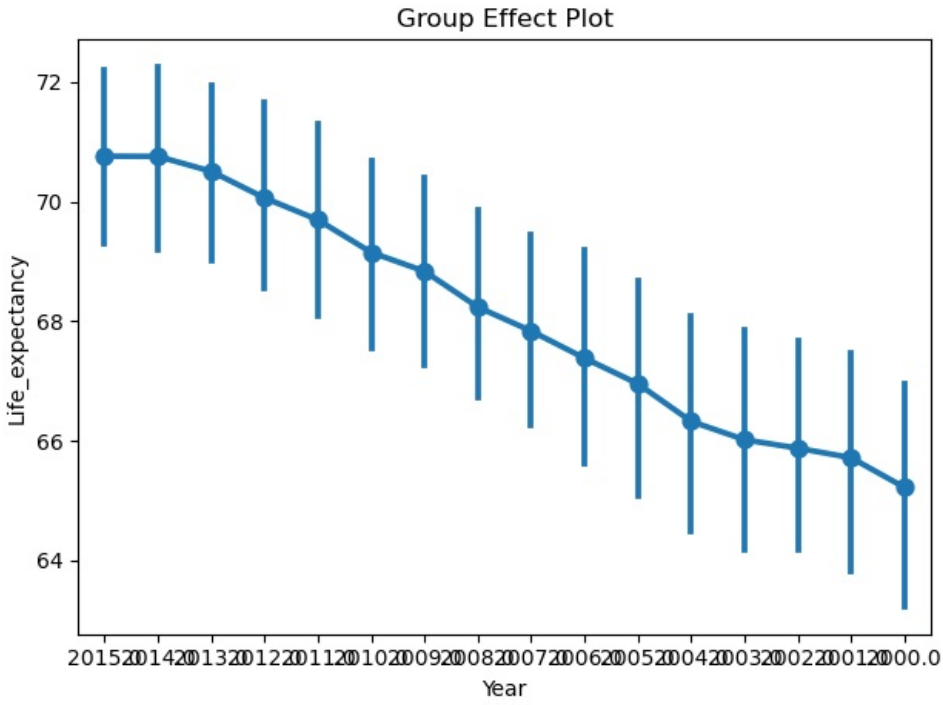


Based on boxplot, we cannot see any significant statistical difference caused by year either. Here the reference group is 2000 and it appears from the boxplot that year does not seem to have a differentiated impact.

In [26]:
```
# ANOVA as multiple comparison
anova_year_2 = MultiComparison(raw_data_knn_test_imputed['Life_expectancy'], raw_data_knn_test_imputed['Year'])

# We can also do pair-wise comparisons, with p-value corrections
## H0: no significant difference in means
print(anova_year_2.tukeyhsd())

# We can represent it visually too
sns.pointplot(raw_data_knn_test_imputed, x='Year', y='Life_expectancy')
plt.title('Group Effect Plot')
plt.tight_layout()
```

```
Multiple Comparison of Means - Tukey HSD, FWER=0.05
=================================================
group1 group2 meandiff p-adj   lower  upper  reject
-------------------------------------------------
2000.0 2001.0   0.4957    1.0 -3.6508 4.6423  False
2000.0 2002.0    0.655    1.0 -3.4916 4.8016  False
2000.0 2003.0   0.7964    1.0 -3.3501  4.943  False
2000.0 2004.0   1.1043 0.9999 -3.0423 5.2508  False
2000.0 2005.0   1.7329 0.9888 -2.4137 5.8794  False
2000.0 2006.0   2.1607  0.921 -1.9858 6.3073  False
2000.0 2007.0    2.615 0.7231 -1.5316 6.7616  False
2000.0 2008.0   3.0143 0.4768 -1.1323 7.1608  False
2000.0 2009.0   3.6136 0.1742  -0.533 7.7601  False
2000.0 2010.0   3.9293 0.0863 -0.2173 8.0758  False
2000.0 2011.0   4.4736 0.0201   0.327 8.6201   True
2000.0 2012.0     4.84 0.0065  0.6934 8.9866   True
2000.0 2013.0   5.2847 0.0013  1.1528 9.4166   True
2000.0 2014.0   5.5343 0.0006  1.3877 9.6808   True
2000.0 2015.0   5.5393 0.0005  1.3927 9.6858   True
2001.0 2002.0   0.1593    1.0 -3.9873 4.3058  False
2001.0 2003.0   0.3007    1.0 -3.8458 4.4473  False
2001.0 2004.0   0.6086    1.0  -3.538 4.7551  False
2001.0 2005.0   1.2371 0.9997 -2.9094 5.3837  False
2001.0 2006.0    1.665 0.9925 -2.4816 5.8116  False
2001.0 2007.0   2.1193 0.9321 -2.0273 6.2658  False
2001.0 2008.0   2.5186 0.7759  -1.628 6.6651  False
2001.0 2009.0   3.1179 0.4138 -1.0287 7.2644  False
2001.0 2010.0   3.4336 0.2469  -0.713 7.5801  False
2001.0 2011.0   3.9779 0.0767 -0.1687 8.1244  False
2001.0 2012.0   4.3443 0.0292  0.1977 8.4908   True
2001.0 2013.0    4.789 0.0072   0.657 8.9209   True
2001.0 2014.0   5.0386 0.0033   0.892 9.1851   True
2001.0 2015.0   5.0436 0.0033   0.897 9.1901   True
2002.0 2003.0   0.1414    1.0 -4.0051  4.288  False
2002.0 2004.0   0.4493    1.0 -3.6973 4.5958  False
2002.0 2005.0   1.0779    1.0 -3.0687 5.2244  False
2002.0 2006.0   1.5057 0.9974 -2.6408 5.6523  False
2002.0 2007.0     1.96 0.9647 -2.1866 6.1066  False
2002.0 2008.0   2.3593 0.8513 -1.7873 6.5058  False
2002.0 2009.0   2.9586 0.5117  -1.188 7.1051  False
2002.0 2010.0   3.2743 0.3255 -0.8723 7.4208  False
2002.0 2011.0   3.8186 0.1118  -0.328 7.9651  False
```

2002.0	2012.0	4.185	0.0452	0.0384	8.3316	True
2002.0	2013.0	4.6297	0.012	0.4978	8.7616	True
2002.0	2014.0	4.8793	0.0057	0.7327	9.0258	True
2002.0	2015.0	4.8843	0.0056	0.7377	9.0308	True
2003.0	2004.0	0.3079	1.0	-3.8387	4.4544	False
2003.0	2005.0	0.9364	1.0	-3.2101	5.083	False
2003.0	2006.0	1.3643	0.9992	-2.7823	5.5108	False
2003.0	2007.0	1.8186	0.9822	-2.328	5.9651	False
2003.0	2008.0	2.2179	0.9038	-1.9287	6.3644	False
2003.0	2009.0	2.8171	0.6009	-1.3294	6.9637	False
2003.0	2010.0	3.1329	0.405	-1.0137	7.2794	False
2003.0	2011.0	3.6771	0.1526	-0.4694	7.8237	False
2003.0	2012.0	4.0436	0.0652	-0.103	8.1901	False
2003.0	2013.0	4.4883	0.0184	0.3563	8.6202	True
2003.0	2014.0	4.7379	0.009	0.5913	8.8844	True
2003.0	2015.0	4.7429	0.0088	0.5963	8.8894	True
2004.0	2005.0	0.6286	1.0	-3.518	4.7751	False
2004.0	2006.0	1.0564	1.0	-3.0901	5.203	False
2004.0	2007.0	1.5107	0.9974	-2.6358	5.6573	False
2004.0	2008.0	1.91	0.9719	-2.2366	6.0566	False
2004.0	2009.0	2.5093	0.7807	-1.6373	6.6558	False
2004.0	2010.0	2.825	0.596	-1.3216	6.9716	False
2004.0	2011.0	3.3693	0.2771	-0.7773	7.5158	False
2004.0	2012.0	3.7357	0.1345	-0.4108	7.8823	False
2004.0	2013.0	4.1804	0.044	0.0485	8.3123	True
2004.0	2014.0	4.43	0.0229	0.2834	8.5766	True
2004.0	2015.0	4.435	0.0225	0.2884	8.5816	True
2005.0	2006.0	0.4279	1.0	-3.7187	4.5744	False
2005.0	2007.0	0.8821	1.0	-3.2644	5.0287	False
2005.0	2008.0	1.2814	0.9996	-2.8651	5.428	False
2005.0	2009.0	1.8807	0.9756	-2.2658	6.0273	False
2005.0	2010.0	2.1964	0.9105	-1.9501	6.343	False
2005.0	2011.0	2.7407	0.6484	-1.4058	6.8873	False
2005.0	2012.0	3.1071	0.4202	-1.0394	7.2537	False
2005.0	2013.0	3.5518	0.1923	-0.5801	7.6838	False
2005.0	2014.0	3.8014	0.1162	-0.3451	7.948	False
2005.0	2015.0	3.8064	0.1149	-0.3401	7.953	False
2006.0	2007.0	0.4543	1.0	-3.6923	4.6008	False
2006.0	2008.0	0.8536	1.0	-3.293	5.0001	False
2006.0	2009.0	1.4529	0.9983	-2.6937	5.5994	False
2006.0	2010.0	1.7686	0.9864	-2.378	5.9151	False
2006.0	2011.0	2.3129	0.8701	-1.8337	6.4594	False
2006.0	2012.0	2.6793	0.6856	-1.4673	6.8258	False
2006.0	2013.0	3.124	0.4037	-1.008	7.2559	False
2006.0	2014.0	3.3736	0.275	-0.773	7.5201	False
2006.0	2015.0	3.3786	0.2726	-0.768	7.5251	False
2007.0	2008.0	0.3993	1.0	-3.7473	4.5458	False
2007.0	2009.0	0.9986	1.0	-3.148	5.1451	False
2007.0	2010.0	1.3143	0.9995	-2.8323	5.4608	False
2007.0	2011.0	1.8586	0.9781	-2.288	6.0051	False
2007.0	2012.0	2.225	0.9015	-1.9216	6.3716	False
2007.0	2013.0	2.6697	0.6857	-1.4622	6.8016	False
2007.0	2014.0	2.9193	0.5364	-1.2273	7.0658	False
2007.0	2015.0	2.9243	0.5333	-1.2223	7.0708	False
2008.0	2009.0	0.5993	1.0	-3.5473	4.7458	False
2008.0	2010.0	0.915	1.0	-3.2316	5.0616	False
2008.0	2011.0	1.4593	0.9982	-2.6873	5.6058	False
2008.0	2012.0	1.8257	0.9815	-2.3208	5.9723	False
2008.0	2013.0	2.2704	0.883	-1.8615	6.4023	False
2008.0	2014.0	2.52	0.7752	-1.6266	6.6666	False
2008.0	2015.0	2.525	0.7725	-1.6216	6.6716	False
2009.0	2010.0	0.3157	1.0	-3.8308	4.4623	False
2009.0	2011.0	0.86	1.0	-3.2866	5.0066	False
2009.0	2012.0	1.2264	0.9998	-2.9201	5.373	False
2009.0	2013.0	1.6711	0.992	-2.4608	5.8031	False
2009.0	2014.0	1.9207	0.9705	-2.2258	6.0673	False
2009.0	2015.0	1.9257	0.9698	-2.2208	6.0723	False
2010.0	2011.0	0.5443	1.0	-3.6023	4.6908	False
2010.0	2012.0	0.9107	1.0	-3.2358	5.0573	False
2010.0	2013.0	1.3554	0.9992	-2.7765	5.4873	False
2010.0	2014.0	1.605	0.9949	-2.5416	5.7516	False
2010.0	2015.0	1.61	0.9947	-2.5366	5.7566	False
2011.0	2012.0	0.3664	1.0	-3.7801	4.513	False
2011.0	2013.0	0.8111	1.0	-3.3208	4.9431	False
2011.0	2014.0	1.0607	1.0	-3.0858	5.2073	False
2011.0	2015.0	1.0657	1.0	-3.0808	5.2123	False
2012.0	2013.0	0.4447	1.0	-3.6872	4.5766	False
2012.0	2014.0	0.6943	1.0	-3.4523	4.8408	False
2012.0	2015.0	0.6993	1.0	-3.4473	4.8458	False
2013.0	2014.0	0.2496	1.0	-3.8823	4.3815	False
2013.0	2015.0	0.2546	1.0	-3.8773	4.3865	False
2014.0	2015.0	0.005	1.0	-4.1416	4.1516	False



Thus, we can conclude while year might not be relevant for small differences, it does have some impact over longer periods versus reference groups. However, given the previous results of no statistical significance in the model and also, no apparent effects from the Boxplot, it is possible that the year longer term effects, may be captured by other variables in the MANOVA + Quant Model. Thus, we have decided to not include year in our regression.

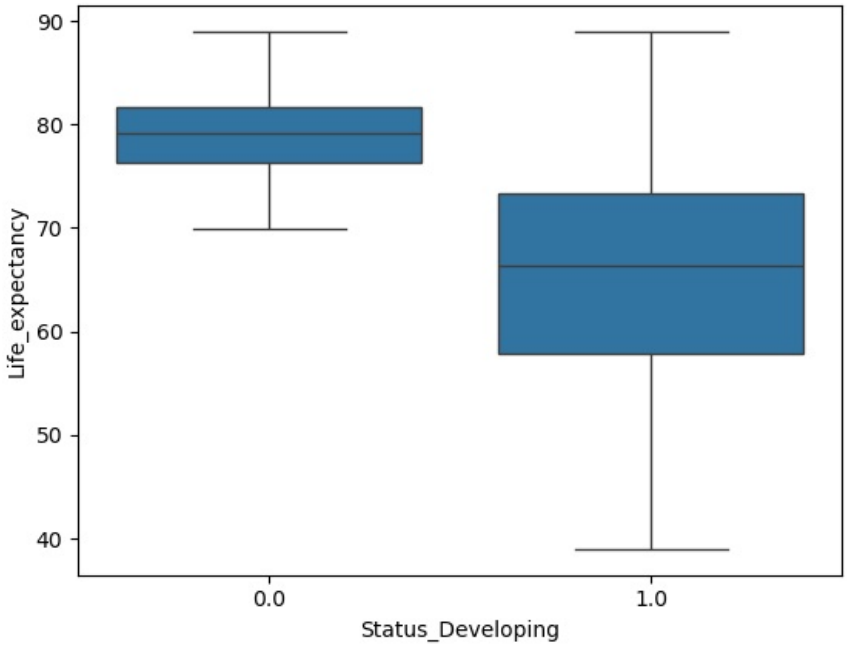
2. Status_Developing

In [27]:
```
# Checking for variable balance

raw_data_knn_test_imputed.groupby('Status_Developing')['Life_expectancy'].mean()
```

Out[27]:
```
Status_Developing
0.0    79.073060
1.0    65.216179
Name: Life_expectancy, dtype: float64
```

Differences between mean for developed and developing countries

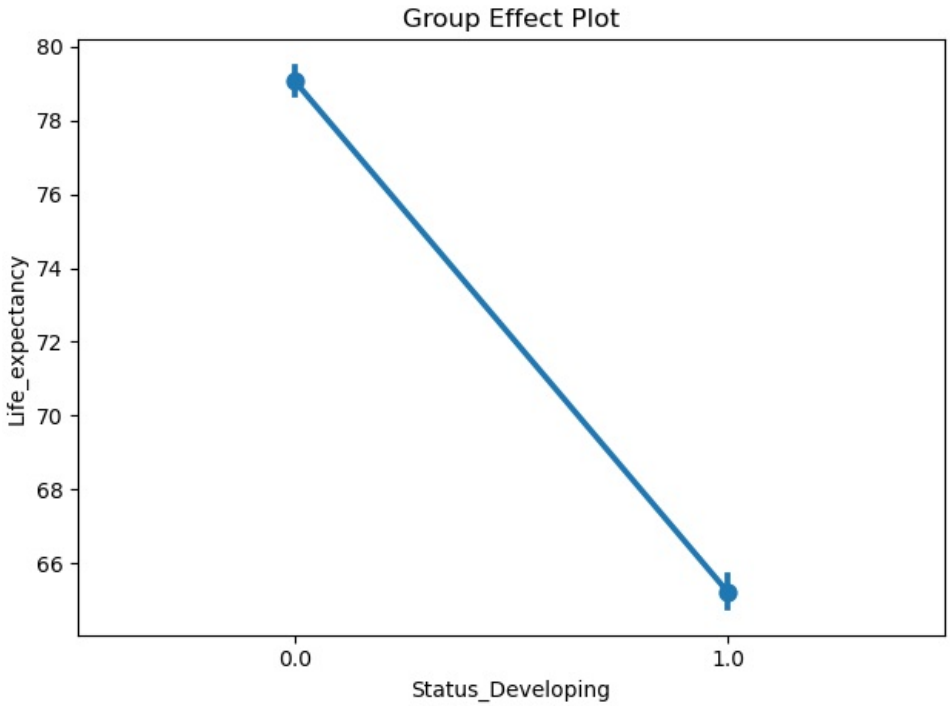
In [28]:
```
# Plotting Boxplot

sns.boxplot(x='Status_Developing', y='Life_expectancy', data=raw_data_knn_test_imputed)
plt.show()
```



In [29]:
```
# ANOVA as multiple comparison
anova_developing_2 = MultiComparison(raw_data_knn_test_imputed['Life_expectancy'], raw_data_knn_test_imputed['S
```

```
# We can also do pair-wise comparisons, with p-value corrections
## H0: no significant difference in means
print(anova_developing_2.tukeyhsd())

# We can represent it visually too
sns.pointplot(raw_data_knn_test_imputed, x='Status_Developing', y='Life_expectancy')
plt.title('Group Effect Plot')
plt.tight_layout()
```

```
 Multiple Comparison of Means - Tukey HSD, FWER=0.05
====================================================
group1 group2 meandiff p-adj  lower    upper   reject
----------------------------------------------------
   0.0    1.0 -13.8569   0.0 -14.7342 -12.9796   True
----------------------------------------------------
```



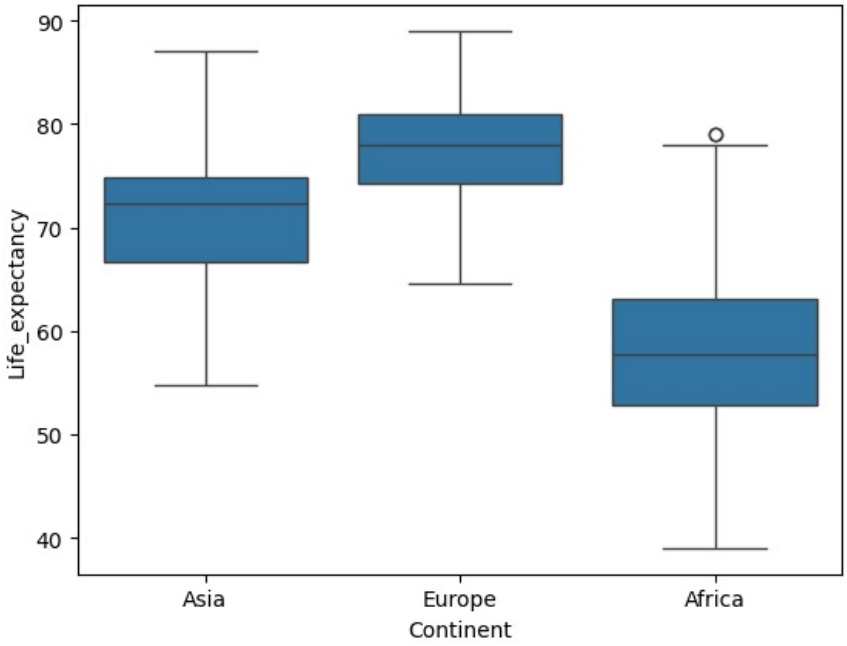
While it is not apparent from the boxplot, based on both MANOVA + Quant model and the Tukey SHD test, Status_Developing shows a statistically significant impact and so, we will include this in our regression

3. Continent

In [30]:
```
#Group By Developing
raw_data_knn_test_imputed.groupby('Continent')['Life_expectancy'].mean()
```

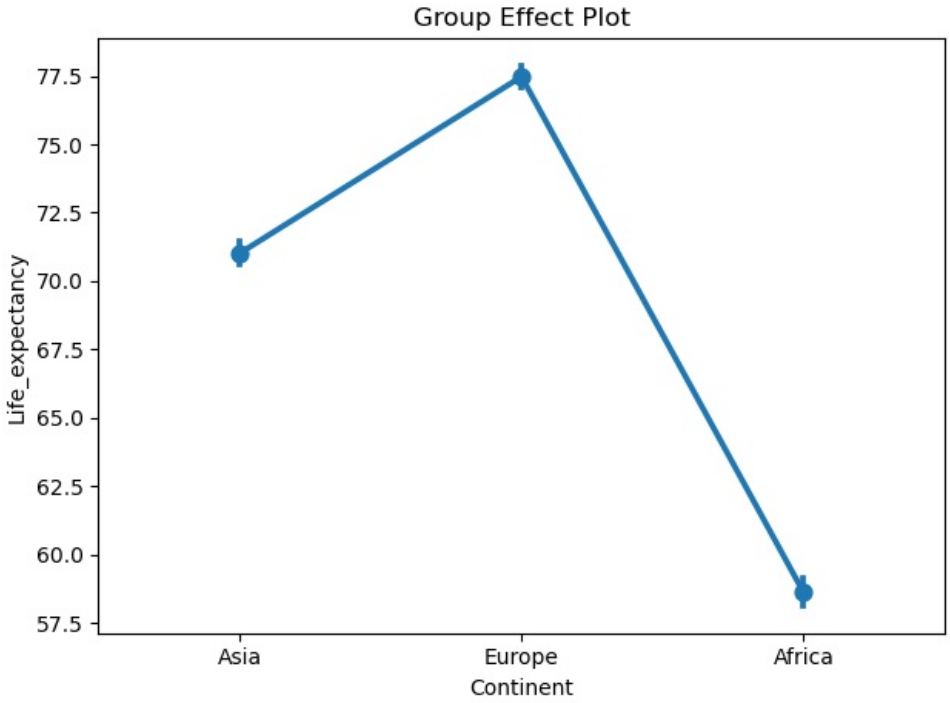
Out[30]:
```
Continent
Africa    58.611921
Asia      71.010326
Europe    77.476584
Name: Life_expectancy, dtype: float64
```

In [31]:
```
# Plotting Boxplot

sns.boxplot(x='Continent', y='Life_expectancy', data=raw_data_knn_test_imputed)
plt.show()
```



In [32]:
```
# ANOVA as multiple comparison
anova_continent_2 = MultiComparison(raw_data_knn_test_imputed['Life_expectancy'], raw_data_knn_test_imputed['Co

# We can also do pair-wise comparisons, with p-value corrections
## H0: no significant difference in means
print(anova_continent_2.tukeyhsd())

# We can represent it visually too
sns.pointplot(raw_data_knn_test_imputed, x='Continent', y='Life_expectancy')
plt.title('Group Effect Plot')
plt.tight_layout()
```

```
Multiple Comparison of Means - Tukey HSD, FWER=0.05
==================================================
group1 group2 meandiff p-adj  lower   upper  reject
--------------------------------------------------
Africa   Asia  12.3984   0.0 11.6315 13.1653   True
Africa Europe  18.8647   0.0  18.068 19.6613   True
  Asia Europe   6.4663   0.0  5.6406  7.2919   True
--------------------------------------------------
```



While it is not apparent from the boxplot, based on both MANOVA + Quant model and the Tukey SHD test, Continent shows a statistically significant impact and so, we will include this in our regression

Our final factor variables are:

1. Status_Developing
2. Continent

Question 2)

Q2) a)

In this analysis, we conducted a comprehensive descriptive examination of the dataset variables using first a basic description, Q-Q plots, histograms, KDE plots, scatterplots, and a correlation heatmap to understand their distribution and relationships.

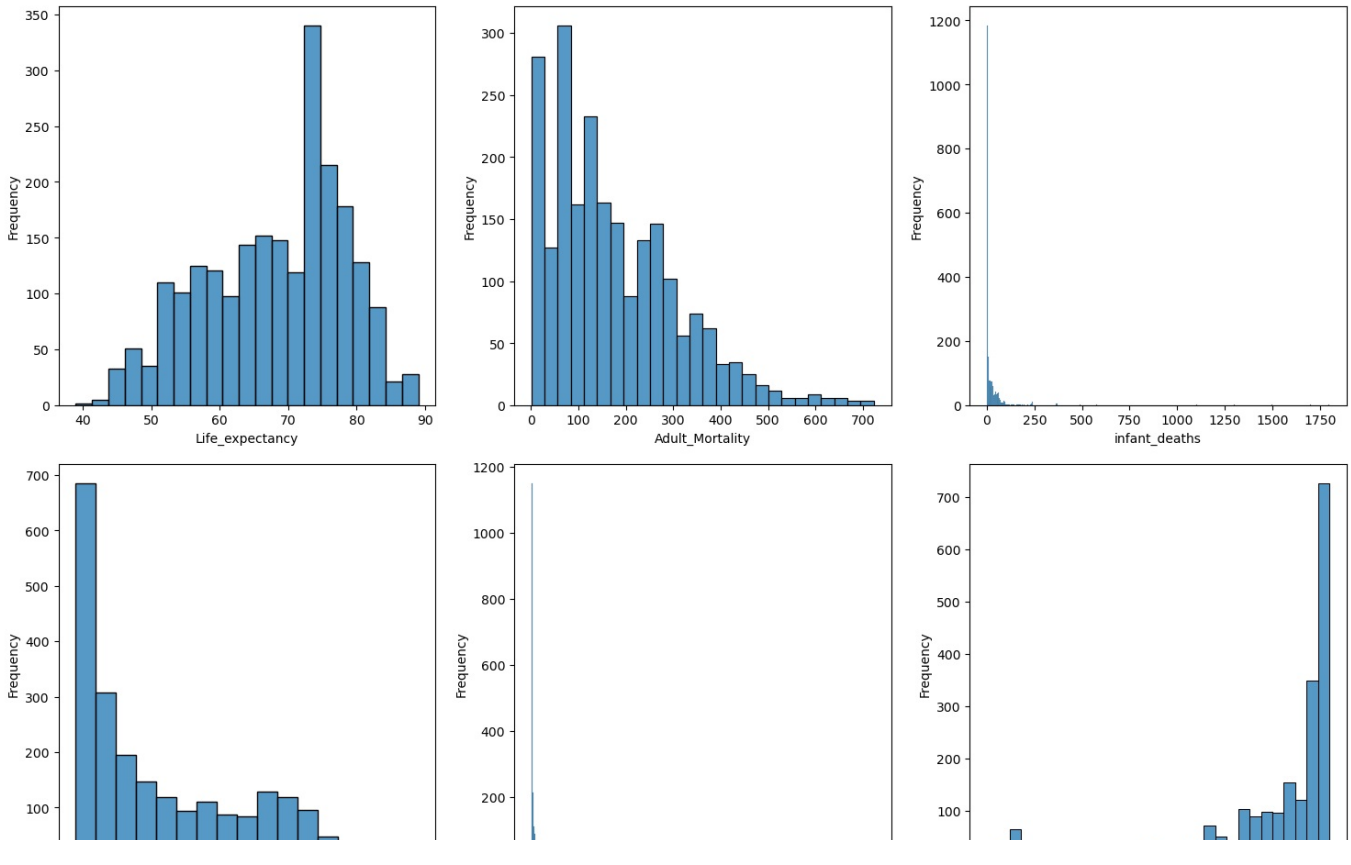
In [33]:
```
raw_data_knn_test_imputed.describe()
```

Out[33]:

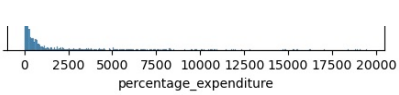
	Life_expectancy	Adult_Mortality	infant_deaths	Alcohol	percentage_expenditure	Hepatitis_B	Measles	B
count	2242.000000	2242.000000	2242.000000	2242.000000	2242.000000	2242.000000	2242.000000	2242.000(
mean	68.083973	174.883289	37.797056	4.341723	785.736345	80.942462	3147.189563	35.0984
std	10.253640	134.713525	133.930495	4.232839	2114.184385	23.489296	13035.371385	19.853(
min	39.000000	1.000000	0.000000	0.010000	0.000000	1.000000	0.000000	1.000(
25%	59.700000	72.000000	1.000000	0.530000	4.124363	75.000000	2.000000	18.000(
50%	69.900000	142.000000	4.000000	2.796667	57.119175	92.000000	51.000000	29.700(
75%	75.600000	259.000000	29.000000	7.557500	407.196346	96.916667	787.500000	55.500(
max	89.000000	723.000000	1800.000000	17.870000	19479.911610	99.000000	212183.000000	71.400(

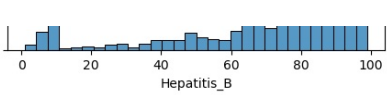
8 rows × 21 columns

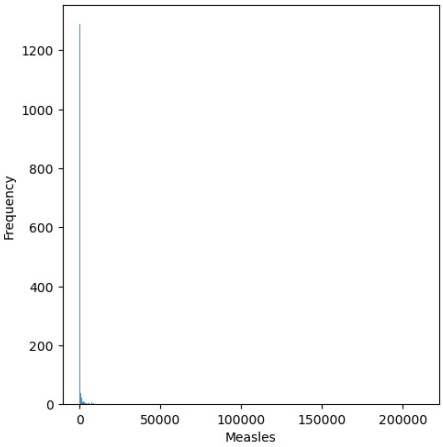
In [34]:
```
# Plotting Histograms

import math

# Calculate grid size dynamically based on the number of columns
n_columns = len(raw_data_knn_test_imputed.columns.drop(['Year', 'Continent', 'Country', 'Status_Developing', 'S
n_rows = math.ceil(n_columns / 3)  # 3 columns per row

plt.figure(figsize=(15, 5 * n_rows))
plt.subplots_adjust(hspace=0.5)

for i, column in enumerate(raw_data_knn_test_imputed.columns.drop(['Year', 'Continent', 'Country', 'Status_Deve
    plt.subplot(n_rows, 3, i + 1)  # Adjust grid size dynamically
    sns.histplot(raw_data_knn_test_imputed[column], bins='fd')
    plt.xlabel(column)
    plt.ylabel('Frequency')

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

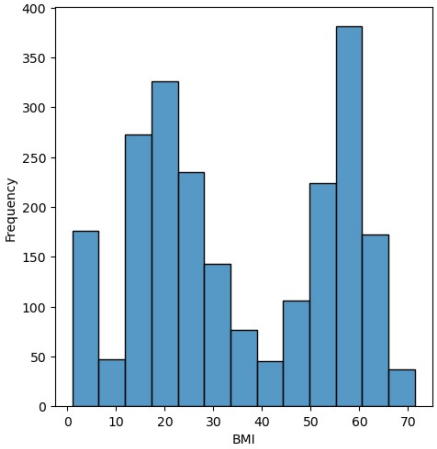


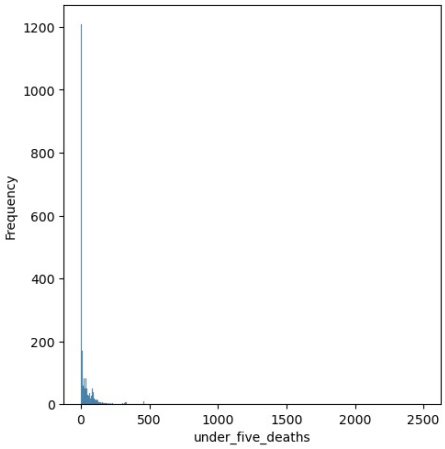

Alcohol

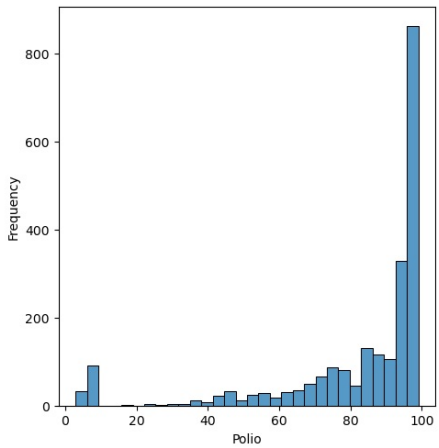

percentage_expenditure

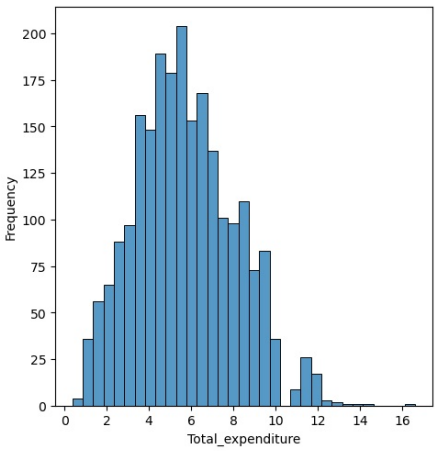

Hepatitis_B

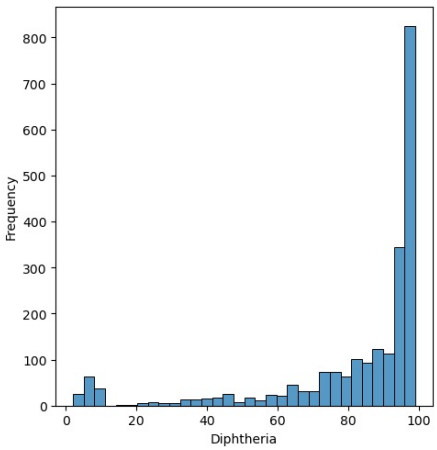

Frequency
Measles

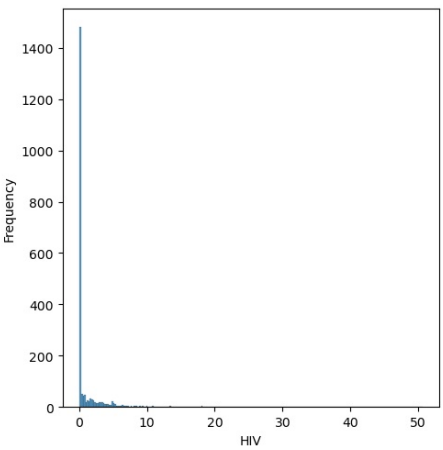

Frequency
BMI

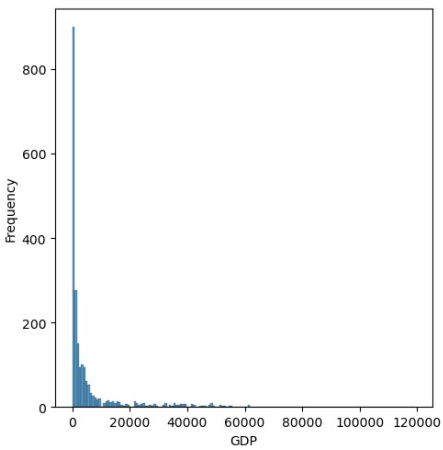

Frequency
under_five_deaths

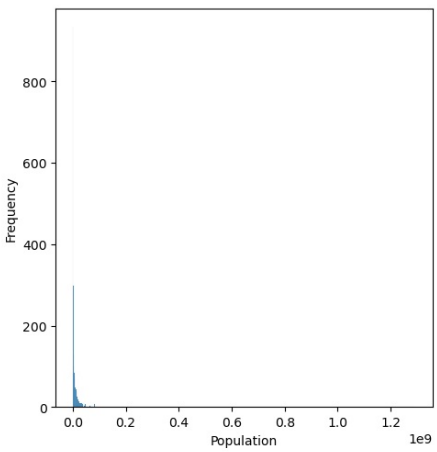

Frequency
Polio

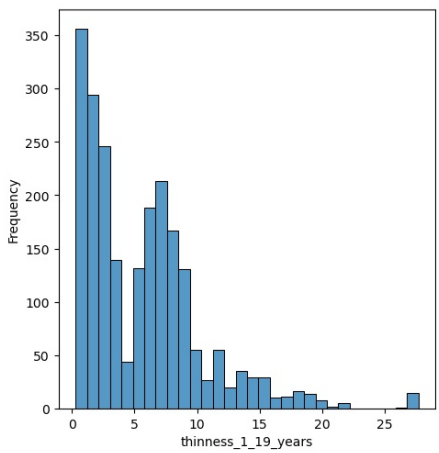

Frequency
Total_expenditure

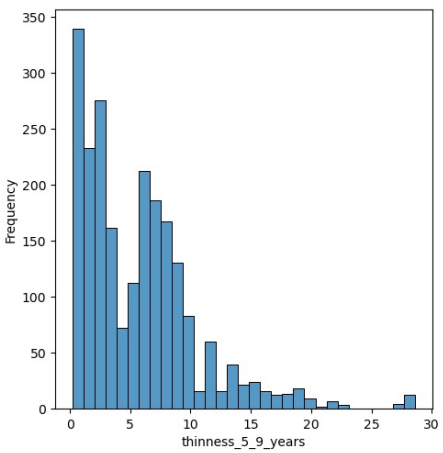

Frequency
Diphtheria

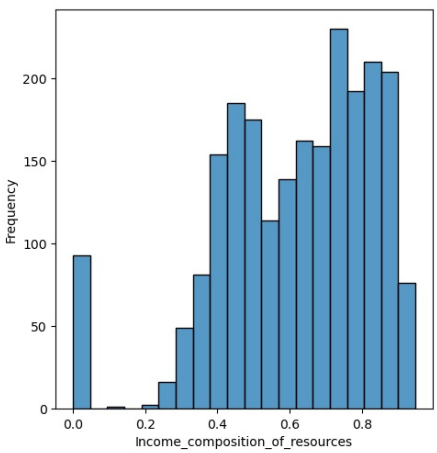

Frequency
HIV

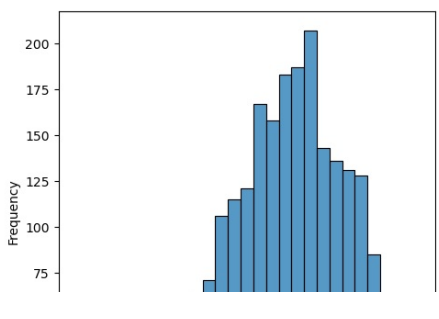

Frequency
GDP

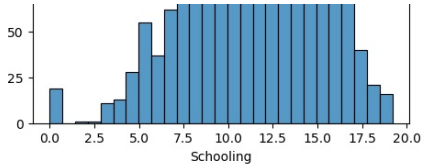

Frequency
Population
1e9


Frequency
thinness_1_19_years


Frequency
thinness_5_9_years

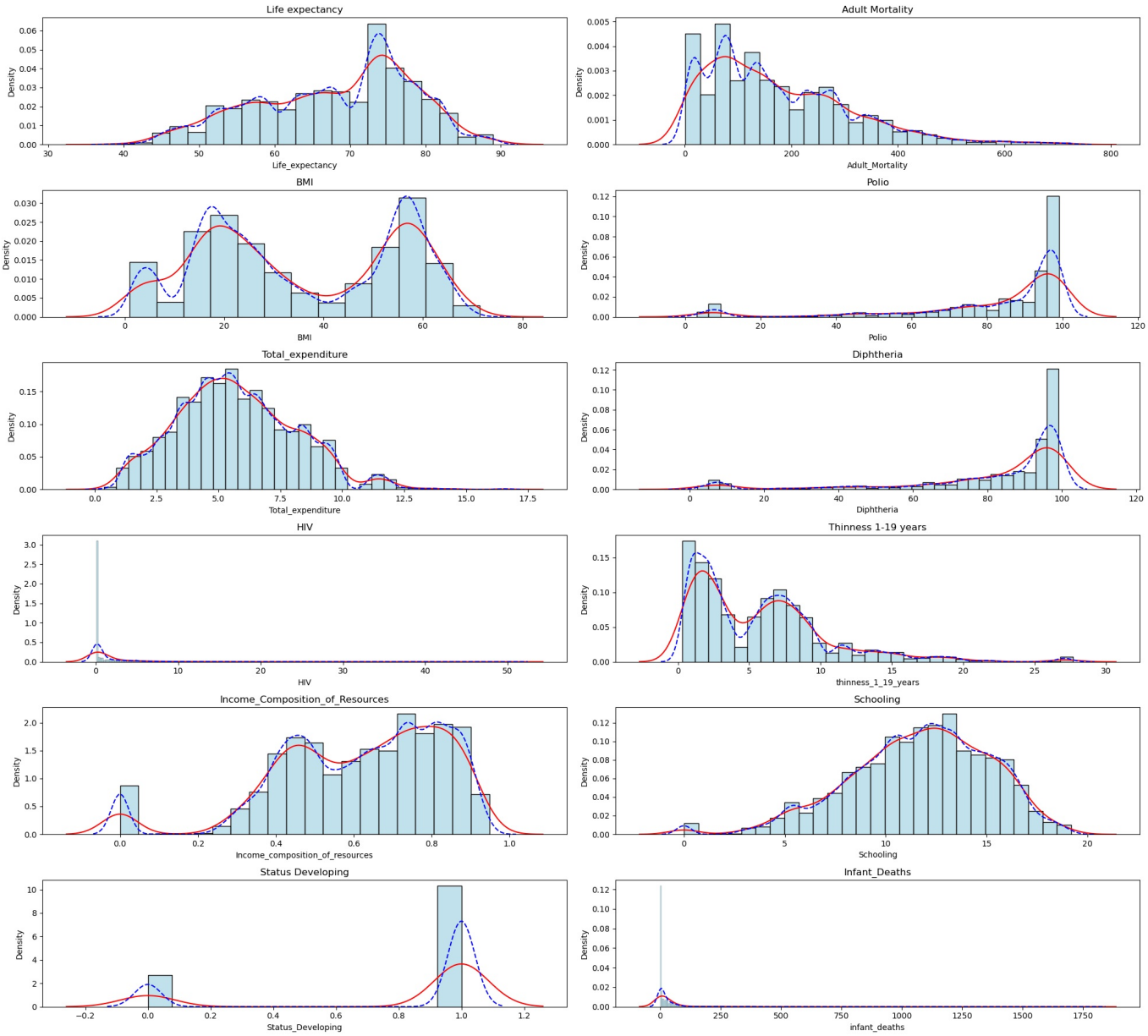

Frequency
Income_composition_of_resources


Frequency

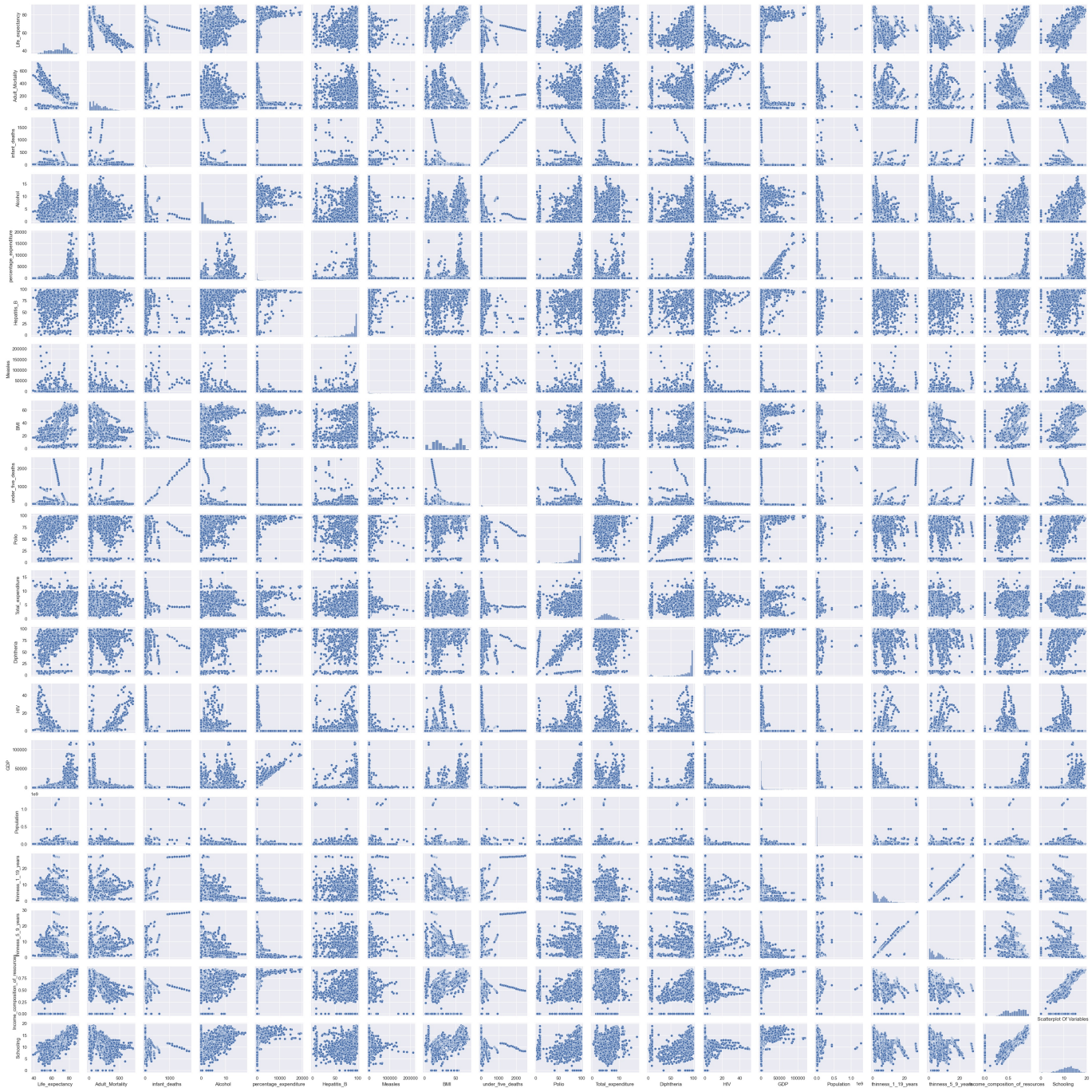


In [35]:

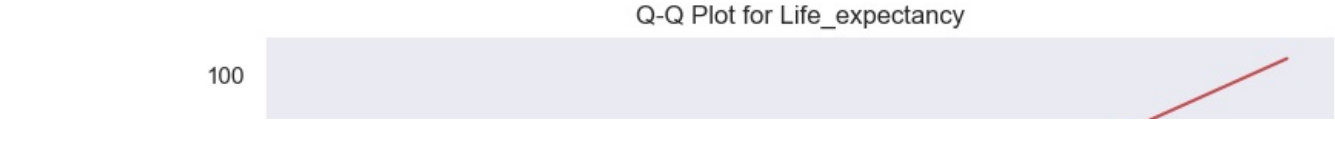
```
# KDE Plots
fig, ax = plt.subplots(6, 2, figsize=(20, 18))

sns.histplot(raw_data_knn_test_imputed['Life_expectancy'], kde=False, stat='density', color='lightblue', ax=ax[(
sns.kdeplot(raw_data_knn_test_imputed['Life_expectancy'], color='red', ax=ax[0, 0])
sns.kdeplot(raw_data_knn_test_imputed['Life_expectancy'], bw_adjust=0.5, color='blue', linestyle='--', ax=ax[0,
ax[0, 0].set_title('Life expectancy')

sns.histplot(raw_data_knn_test_imputed['Adult_Mortality'], kde=False, stat='density', color='lightblue', ax=ax[(
sns.kdeplot(raw_data_knn_test_imputed['Adult_Mortality'], color='red', ax=ax[0, 1])
sns.kdeplot(raw_data_knn_test_imputed['Adult_Mortality'], bw_adjust=0.5, color='blue', linestyle='--', ax=ax[0,
ax[0, 1].set_title('Adult Mortality')

sns.histplot(raw_data_knn_test_imputed['BMI'], kde=False, stat='density', color='lightblue', ax=ax[1, 0])
sns.kdeplot(raw_data_knn_test_imputed['BMI'], color='red', ax=ax[1, 0])
sns.kdeplot(raw_data_knn_test_imputed['BMI'], bw_adjust=0.5, color='blue', linestyle='--', ax=ax[1, 0])
ax[1, 0].set_title('BMI')

sns.histplot(raw_data_knn_test_imputed['Polio'], kde=False, stat='density', color='lightblue', ax=ax[1, 1])
sns.kdeplot(raw_data_knn_test_imputed['Polio'], color='red', ax=ax[1, 1])
sns.kdeplot(raw_data_knn_test_imputed['Polio'], bw_adjust=0.5, color='blue', linestyle='--', ax=ax[1, 1])
ax[1, 1].set_title('Polio')

sns.histplot(raw_data_knn_test_imputed['Total_expenditure'], kde=False, stat='density', color='lightblue', ax=ax
sns.kdeplot(raw_data_knn_test_imputed['Total_expenditure'], color='red', ax=ax[2, 0])
sns.kdeplot(raw_data_knn_test_imputed['Total_expenditure'], bw_adjust=0.5, color='blue', linestyle='--', ax=ax[2
ax[2, 0].set_title('Total_expenditure')

sns.histplot(raw_data_knn_test_imputed['Diphtheria'], kde=False, stat='density', color='lightblue', ax=ax[2, 1]
sns.kdeplot(raw_data_knn_test_imputed['Diphtheria'], color='red', ax=ax[2, 1])
sns.kdeplot(raw_data_knn_test_imputed['Diphtheria'], bw_adjust=0.5, color='blue', linestyle='--', ax=ax[2, 1])
ax[2, 1].set_title('Diphtheria')

sns.histplot(raw_data_knn_test_imputed['HIV'], kde=False, stat='density', color='lightblue', ax=ax[3, 0])
sns.kdeplot(raw_data_knn_test_imputed['HIV'], color='red', ax=ax[3, 0])
sns.kdeplot(raw_data_knn_test_imputed['HIV'], bw_adjust=0.5, color='blue', linestyle='--', ax=ax[3, 0])
ax[3, 0].set_title('HIV')

sns.histplot(raw_data_knn_test_imputed['thinness_1_19_years'], kde=False, stat='density', color='lightblue', ax
sns.kdeplot(raw_data_knn_test_imputed['thinness_1_19_years'], color='red', ax=ax[3, 1])
sns.kdeplot(raw_data_knn_test_imputed['thinness_1_19_years'], bw_adjust=0.5, color='blue', linestyle='--', ax=ax
ax[3, 1].set_title('Thinness 1-19 years')

sns.histplot(raw_data_knn_test_imputed['Income_composition_of_resources'], kde=False, stat='density', color='li
sns.kdeplot(raw_data_knn_test_imputed['Income_composition_of_resources'], color='red', ax=ax[4, 0])
sns.kdeplot(raw_data_knn_test_imputed['Income_composition_of_resources'], bw_adjust=0.5, color='blue', linestyl
ax[4, 0].set_title('Income_Composition_of_Resources')

sns.histplot(raw_data_knn_test_imputed['Schooling'], kde=False, stat='density', color='lightblue', ax=ax[4, 1])
sns.kdeplot(raw_data_knn_test_imputed['Schooling'], color='red', ax=ax[4, 1])
sns.kdeplot(raw_data_knn_test_imputed['Schooling'], bw_adjust=0.5, color='blue', linestyle='--', ax=ax[4, 1])
ax[4, 1].set_title('Schooling')

sns.histplot(raw_data_knn_test_imputed['Status_Developing'], kde=False, stat='density', color='lightblue', ax=ax
sns.kdeplot(raw_data_knn_test_imputed['Status_Developing'], color='red', ax=ax[5, 0])
sns.kdeplot(raw_data_knn_test_imputed['Status_Developing'], bw_adjust=0.5, color='blue', linestyle='--', ax=ax[5
ax[5, 0].set_title('Status Developing')

sns.histplot(raw_data_knn_test_imputed['infant_deaths'], kde=False, stat='density', color='lightblue', ax=ax[5,
sns.kdeplot(raw_data_knn_test_imputed['infant_deaths'], color='red', ax=ax[5, 1])
sns.kdeplot(raw_data_knn_test_imputed['infant_deaths'], bw_adjust=0.5, color='blue', linestyle='--', ax=ax[5, 1
ax[5, 1].set_title('Infant_Deaths')

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

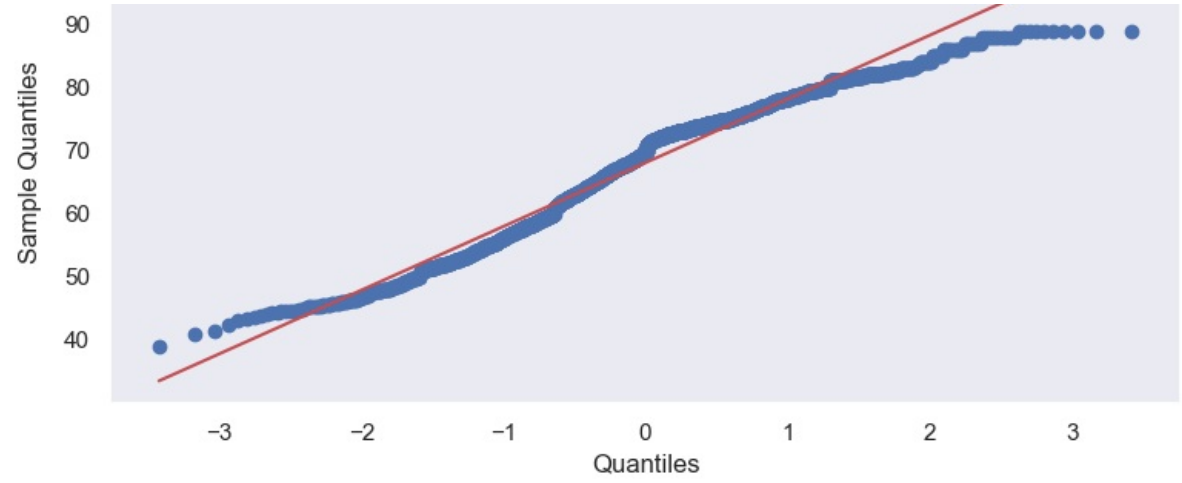


In [36]:
```
# Scatterplot

sns.set()
sns.pairplot(raw_data_knn_test_imputed.drop(['Year', 'Continent', 'Country', 'Status_Developing', 'Status_Develo

plt.title('Scatterplot Of Variables')
plt.show()
```

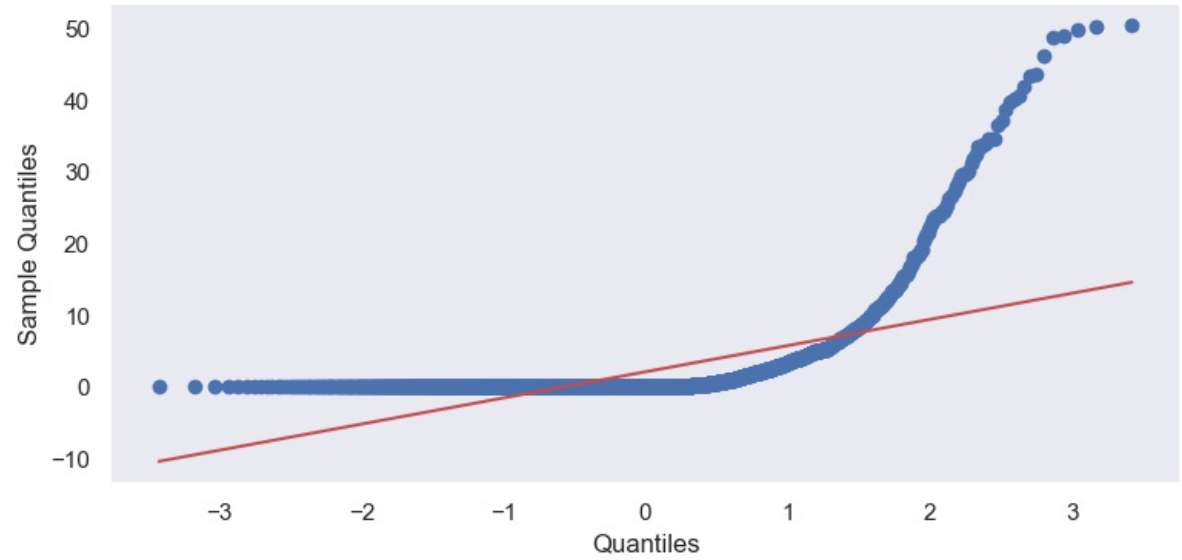


In [37]:
```
# Quantile Plots

col = raw_data_knn_test_imputed.columns.size


predictors = ['Life_expectancy', 'HIV', 'Income_composition_of_resources', 'Adult_Mortality',
              'Schooling', 'BMI', 'Total_expenditure', 'Diphtheria', 'Polio',
              'thinness_1_19_years', 'infant_deaths']

fig_height = 4 * len(predictors)  # Adjust figure height based on the number of predictors
plt.figure(figsize=(8, fig_height))

for i, column in enumerate(predictors):
    if column in raw_data_knn_test_imputed.columns:  # Check if the column exists
        plt.subplot(len(predictors), 1, i + 1)
        stats.probplot(raw_data_knn_test_imputed[column], dist='norm', plot=plt)
        plt.title(f'Q-Q Plot for {column}')
        plt.xlabel('Quantiles')
        plt.ylabel('Sample Quantiles')
        plt.grid()
    else:
        print(f"Column '{column}' not found in the DataFrame.")

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

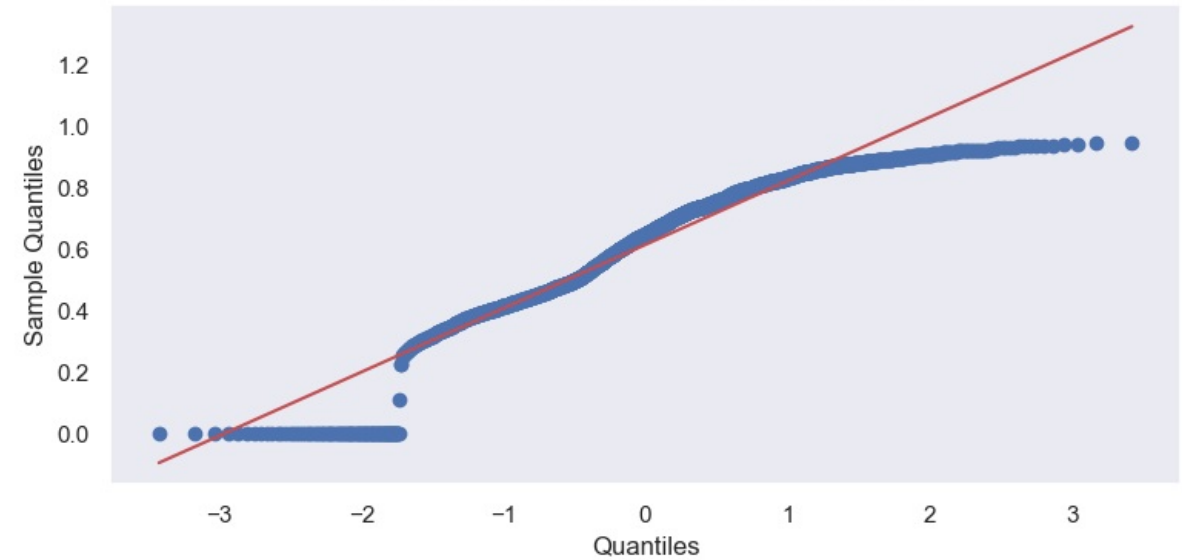




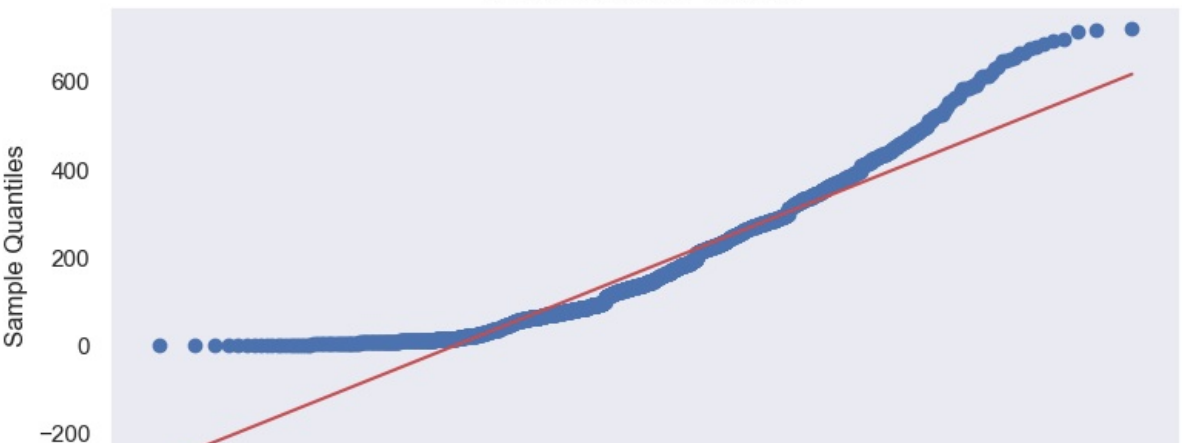
Q-Q Plot for HIV

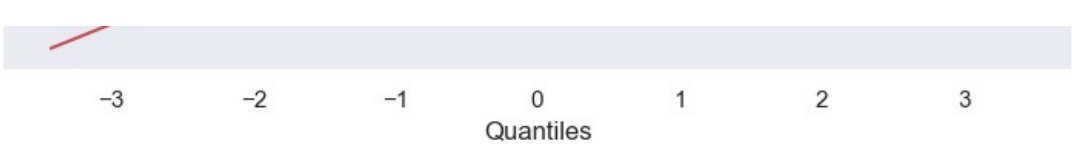


Q-Q Plot for Income_composition_of_resources

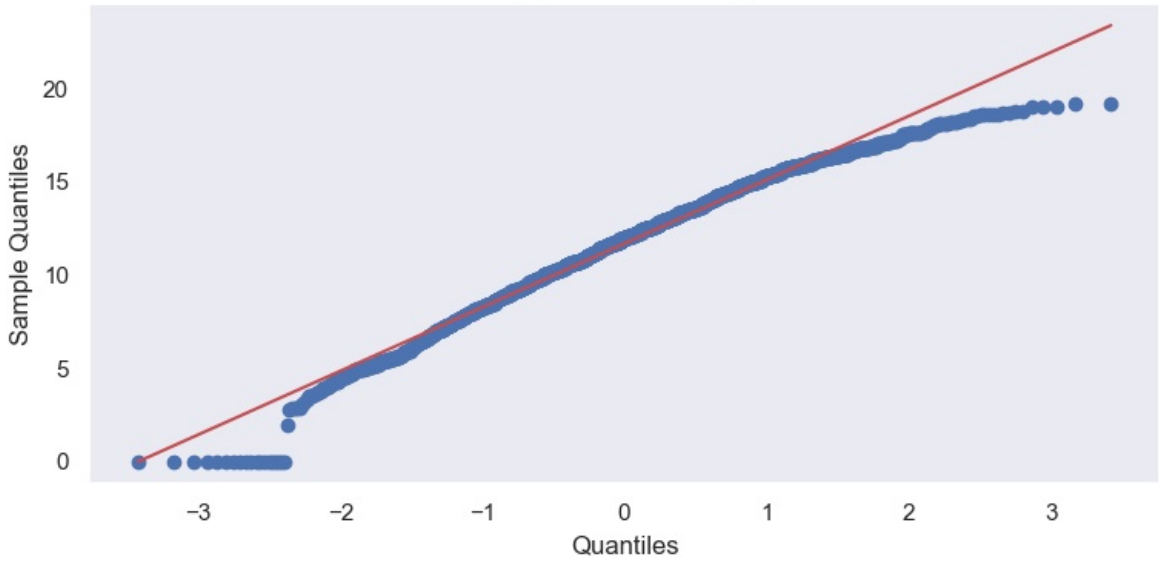


Q-Q Plot for Adult_Mortality

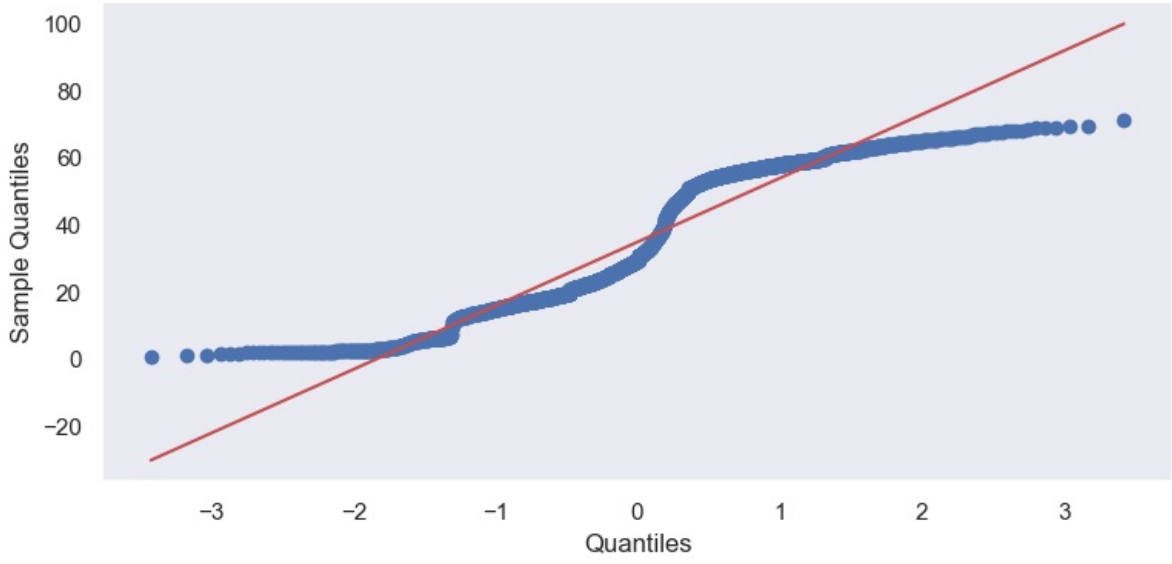




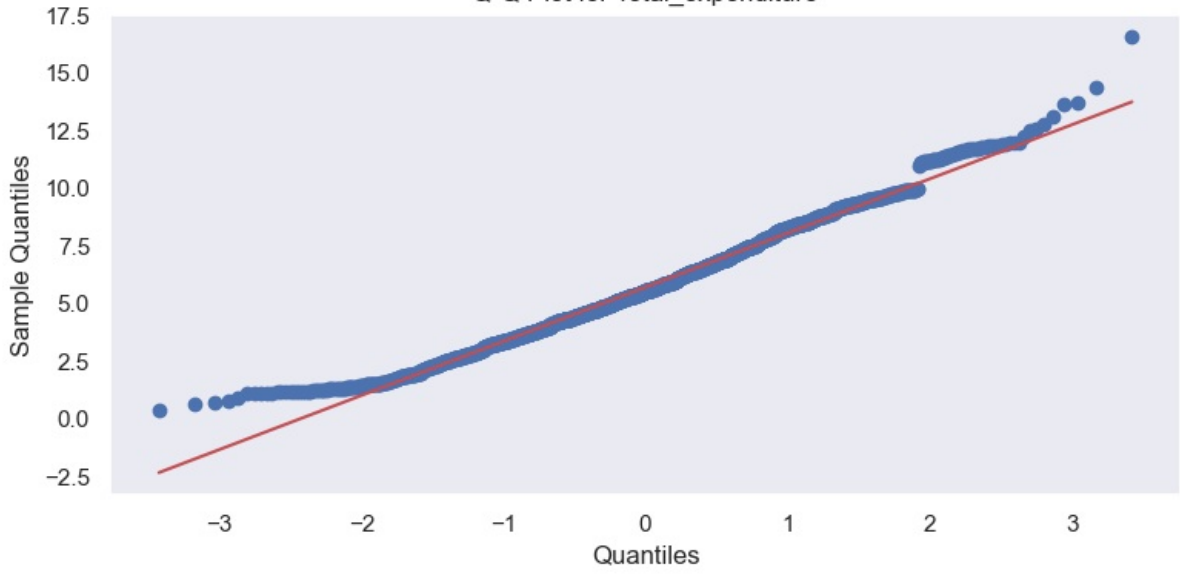
Q-Q Plot for Schooling



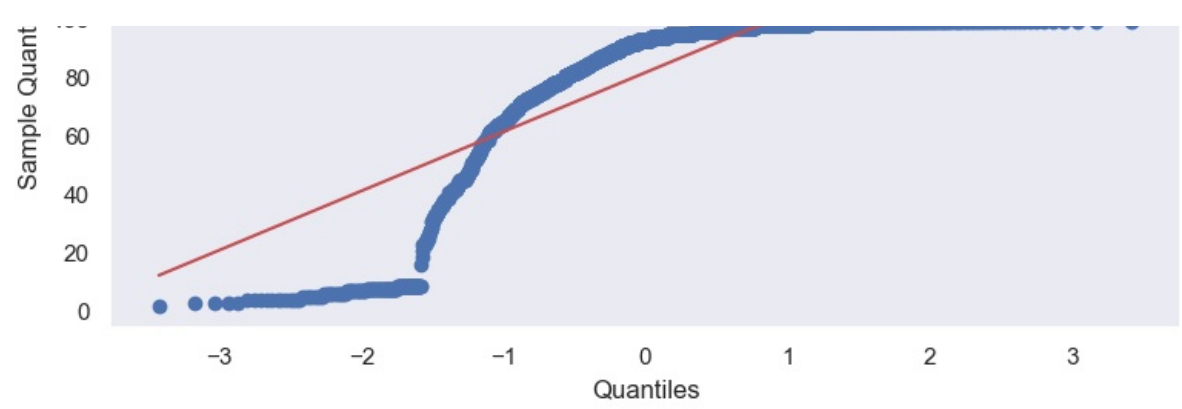
Q-Q Plot for BMI

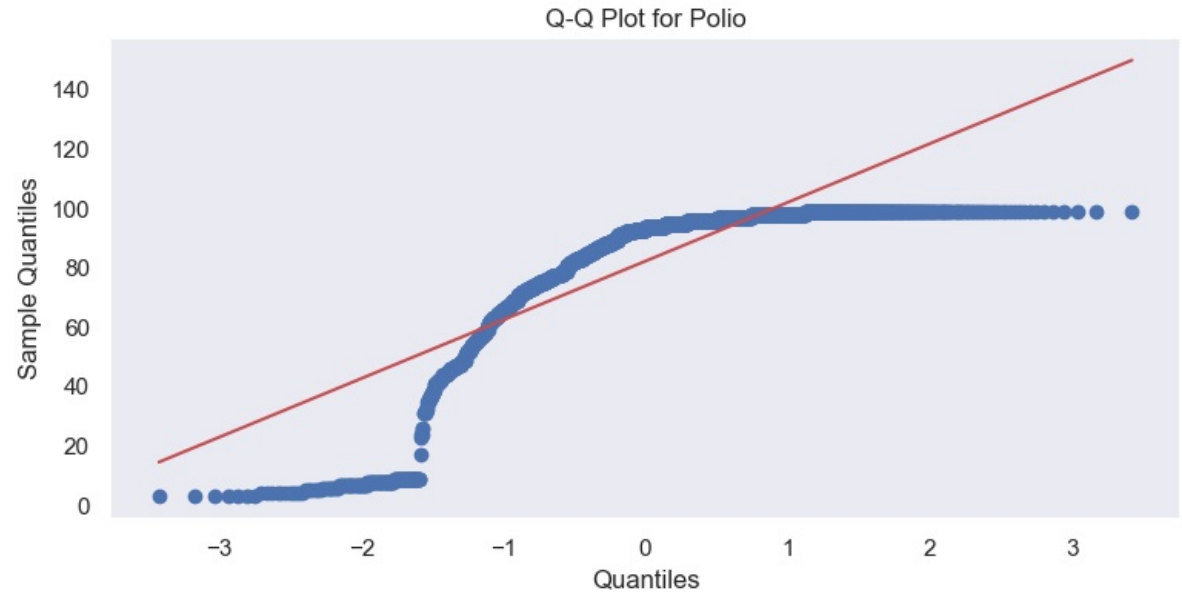


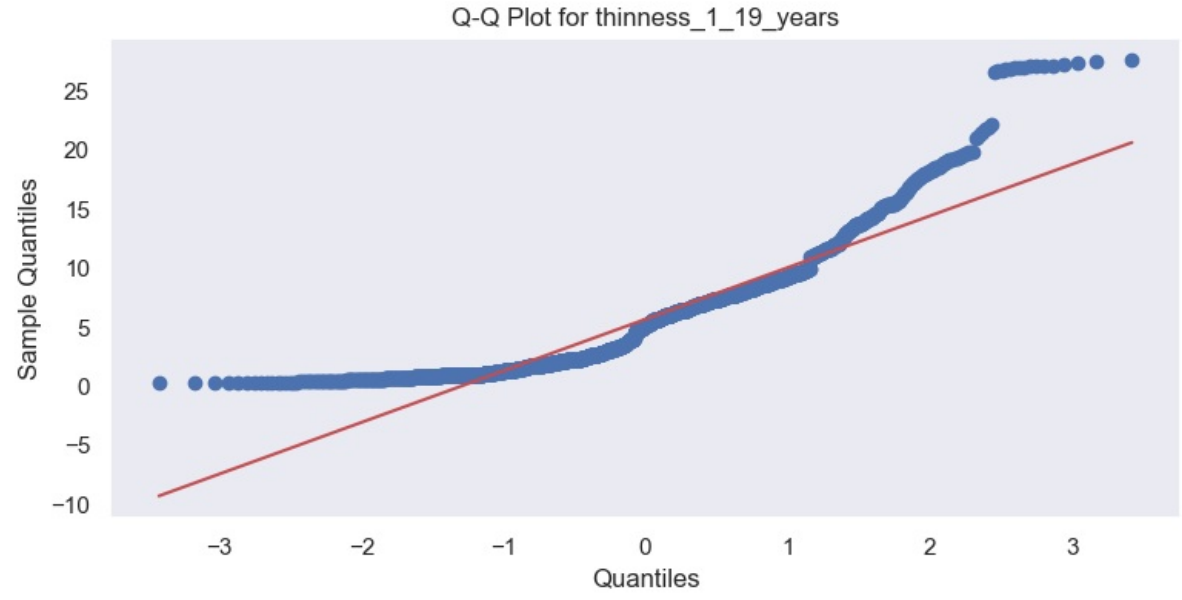
Q-Q Plot for Total_expenditure



Q-Q Plot for Diphtheria

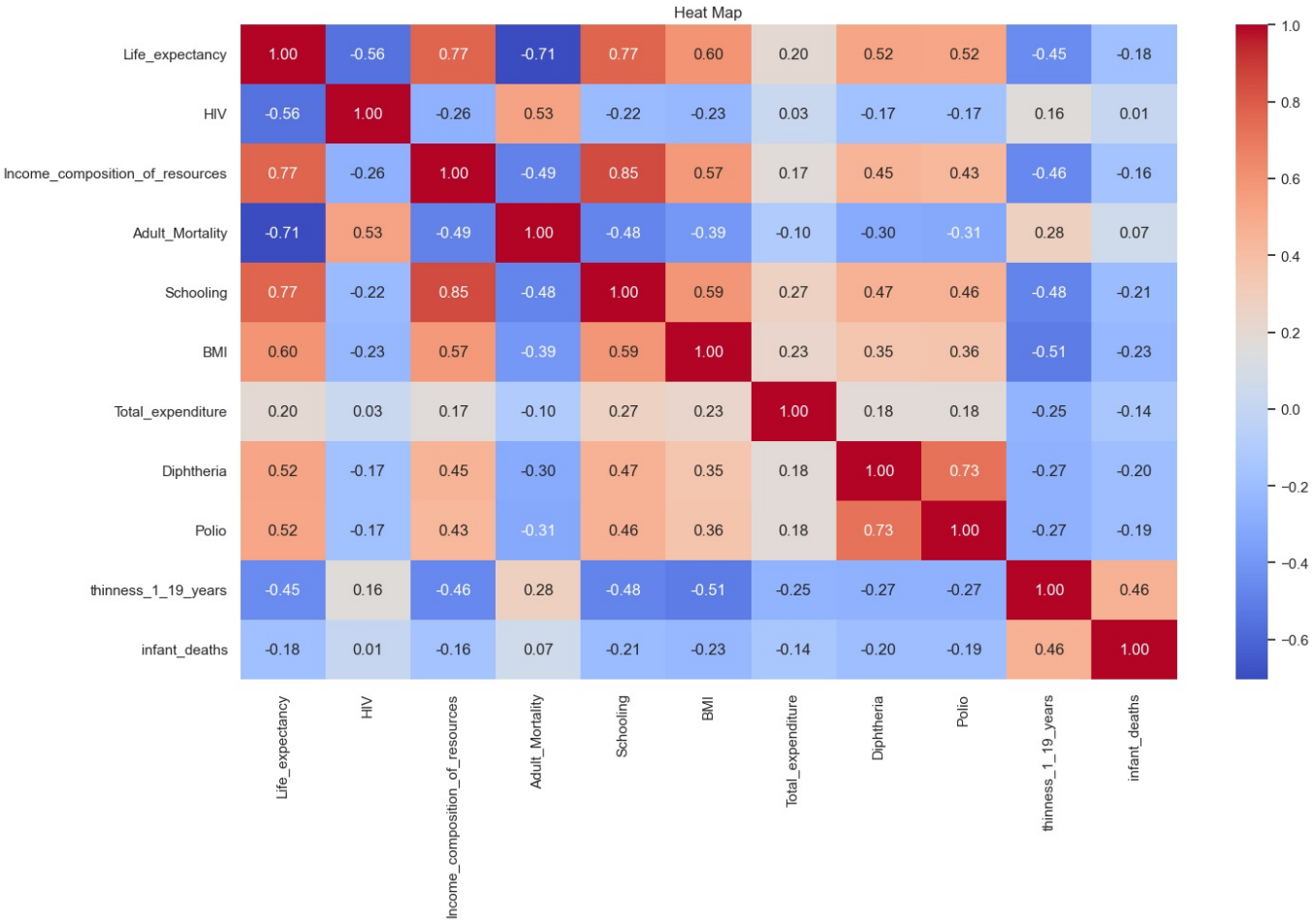

Sample Quant
0
20
40
60
80
−3
−2
−1
0
1
2
3
Quantiles


Q-Q Plot for Polio
Sample Quantiles
0
20
40
60
80
100
120
140
−3
−2
−1
0
1
2
3
Quantiles


Q-Q Plot for thinness_1_19_years
Sample Quantiles
−10
−5
0
5
10
15
20
25
−3
−2
−1
0
1
2
3
Quantiles


Q-Q Plot for infant_deaths
Sample Quantiles
−250
0
250
500
750
1000
1250
1500
1750
−3
−2
−1
0
1
2
3

In [38]:
```
data_quan = raw_data_knn_test_imputed[['Life_expectancy' ,'HIV','Income_composition_of_resources', 'Adult_Morta
heat_map = data_quan.corr()

plt.figure(figsize=(16, 9))
sns.heatmap(heat_map, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Heat Map')
plt.show()
```



Variables like HIV and infant_deaths showed extreme skewness and kurtosis, indicating outliers and heavy tails. Histograms visualized frequency distributions, capturing patterns like skewness and outliers. HIV and infant_deaths displayed highly skewed distributions, while Life_expectancy appeared more symmetrical but deviated from normality. Adaptive bin sizes (bins='fd') highlighted finer details. KDE plots, overlaid on histograms, provided smooth density curves to emphasize distribution shape and density peaks. Adjusted bandwidth highlighted variations in data spread, particularly for variables like Adult_Mortality and BMI. Scatterplots revealed pairwise relationships among variables, highlighting linear or nonlinear trends and outliers. Strong positive relationships were observed between Life_expectancy and both Income_composition_of_resources and Schooling, while a clear negative association was evident between HIV and Life_expectancy. The correlation heatmap summarized linear relationships, reinforcing patterns seen in scatterplots. For example, Life_expectancy positively correlated with Income_composition_of_resources (0.77) and negatively with HIV (-0.56). These visualizations highlighted key patterns and relationships, confirming the need for transformations (e.g., log, Box-Cox) to address skewness. This exploration provides a solid foundation for feature selection and modeling.

Q2) b)

Next, we implement the Jarque-Bera (JB) test to assess the normality of multiple variables in the dataset, including Life_expectancy, HIV, Income_composition_of_resources, Adult_Mortality, and others. The JB test evaluates whether the skewness and kurtosis of each variable's distribution align with those expected under a normal distribution. This is particularly useful, since many linear regression properties, depends on the normality of predictors. We also look at the ECDF's of different variables to get a better understanding of variable distribution

Jarque Bera Test

In [39]:
```
# Checking for normality using JB Test

variables = ['Life_expectancy' ,'HIV','Income_composition_of_resources', 'Adult_Mortality', 'Schooling', 'BMI',
# Loop through each variable and run the Jarque-Bera test
for var in variables:
    print(f'\n===== Jarque-Bera Test for {var} =====')
```

```
    # Drop missing values from the current variable in the happiness dataframe
    data = raw_data_knn_test_imputed[var]

    # Run Jarque-Bera test on the variable
    jb_test_stat, jb_test_pvalue, jb_test_skew, jb_test_kurtosis = sms.jarque_bera(data)

    # Print the test results
    print(f"Test Statistic: {jb_test_stat}\n"
          f"p-value: {jb_test_pvalue}\n"
          f"Skewness: {jb_test_skew}\n"
          f"Kurtosis: {jb_test_kurtosis}")
```

```
===== Jarque-Bera Test for Life_expectancy =====
Test Statistic: 106.8481199010845
p-value: 6.28384642670712e-24
Skewness: -0.39867124094690287
Kurtosis: 2.2872412616935502

===== Jarque-Bera Test for HIV =====
Test Statistic: 71534.5396422747
p-value: 0.0
Skewness: 4.686788931124857
Kurtosis: 29.036394180465024

===== Jarque-Bera Test for Income_composition_of_resources =====
Test Statistic: 311.1553224184798
p-value: 2.713190135542864e-68
Skewness: -0.8567196487441608
Kurtosis: 3.628457509811718

===== Jarque-Bera Test for Adult_Mortality =====
Test Statistic: 480.3723957180598
p-value: 4.8804474456533696e-105
Skewness: 1.0285158839587123
Kurtosis: 3.9543982936588025

===== Jarque-Bera Test for Schooling =====
Test Statistic: 97.06323170907862
p-value: 8.375042826664959e-22
Skewness: -0.4982388670347464
Kurtosis: 3.2146336696112643

===== Jarque-Bera Test for BMI =====
Test Statistic: 197.00524501424977
p-value: 1.662855809863331e-43
Skewness: 0.04777481977029914
Kurtosis: 1.5509458901168023

===== Jarque-Bera Test for Total_expenditure =====
Test Statistic: 49.76236234506024
p-value: 1.5640128378045812e-11
Skewness: 0.36478922733493735
Kurtosis: 2.9798052453241546

===== Jarque-Bera Test for Diphtheria =====
Test Statistic: 2289.1519016786333
p-value: 0.0
Skewness: -1.953453099738341
Kurtosis: 6.039873718913615

===== Jarque-Bera Test for Polio =====
Test Statistic: 2590.691666791939
p-value: 0.0
Skewness: -2.0050756702195334
Kurtosis: 6.413405286261263

===== Jarque-Bera Test for thinness_1_19_years =====
Test Statistic: 1542.5708941395178
p-value: 0.0
Skewness: 1.4327275634096404
Kurtosis: 5.881313699959023

===== Jarque-Bera Test for infant_deaths =====
Test Statistic: 762315.2048959864
p-value: 0.0
Skewness: 8.5861776091764
Kurtosis: 91.68758129009805
```

The results reveal that most variables have extremely small p-values, often near zero, indicating strong evidence against the null hypothesis of normality. For instance, variables like HIV and GDP exhibit high skewness (e.g., HIV with a skewness of 4.69) and kurtosis values (e.g., HIV with kurtosis of 29), pointing to significant deviations from normality. Similarly, variables like Adult_Mortality show moderate skewness (1.03) but still deviate from normality due to elevated kurtosis (3.95). These findings suggest that the dataset contains variables with heavy tails, asymmetry, or outliers, necessitating transformations or robust methods for further statistical analysis.

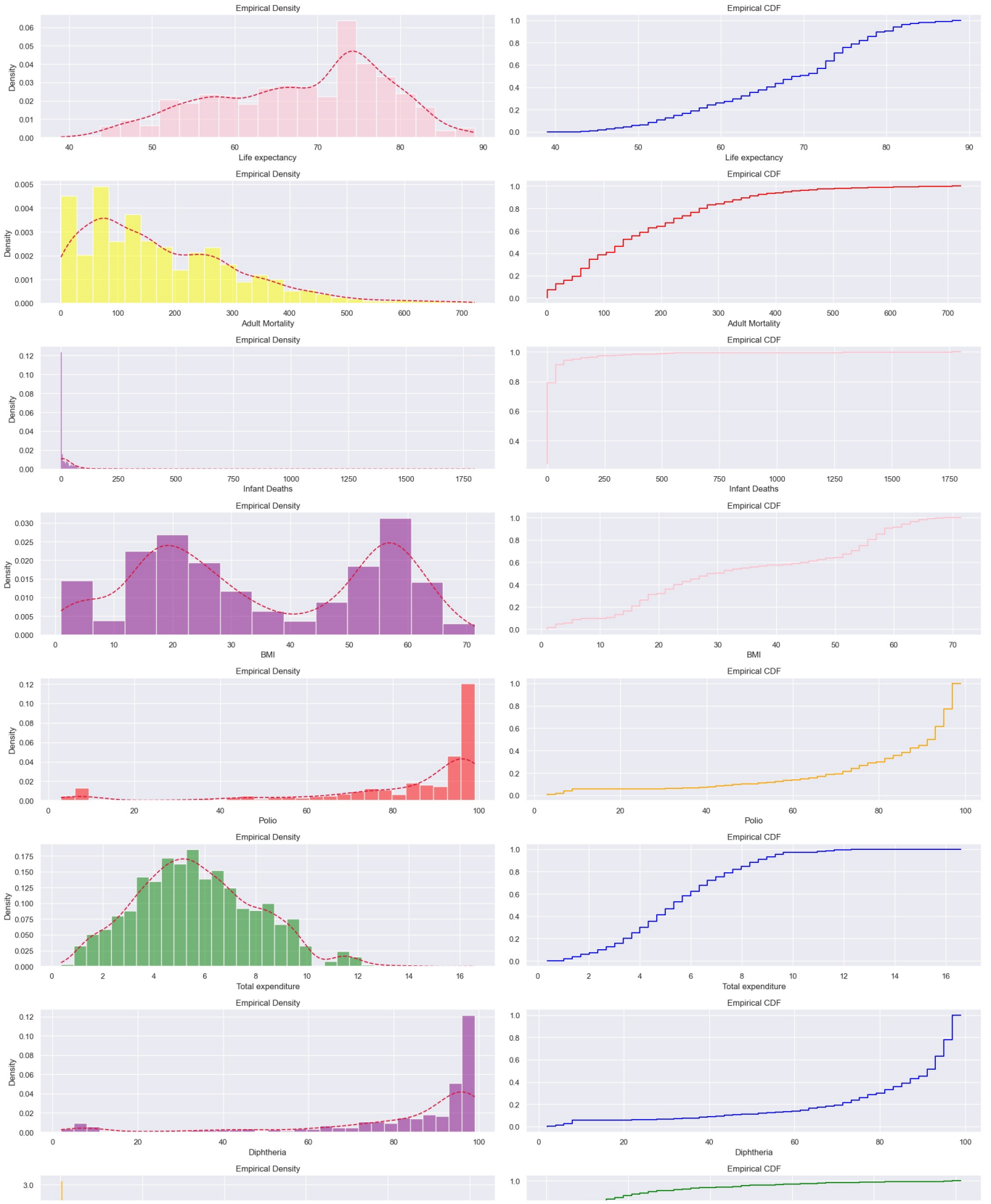
ECDF's And Density Plots

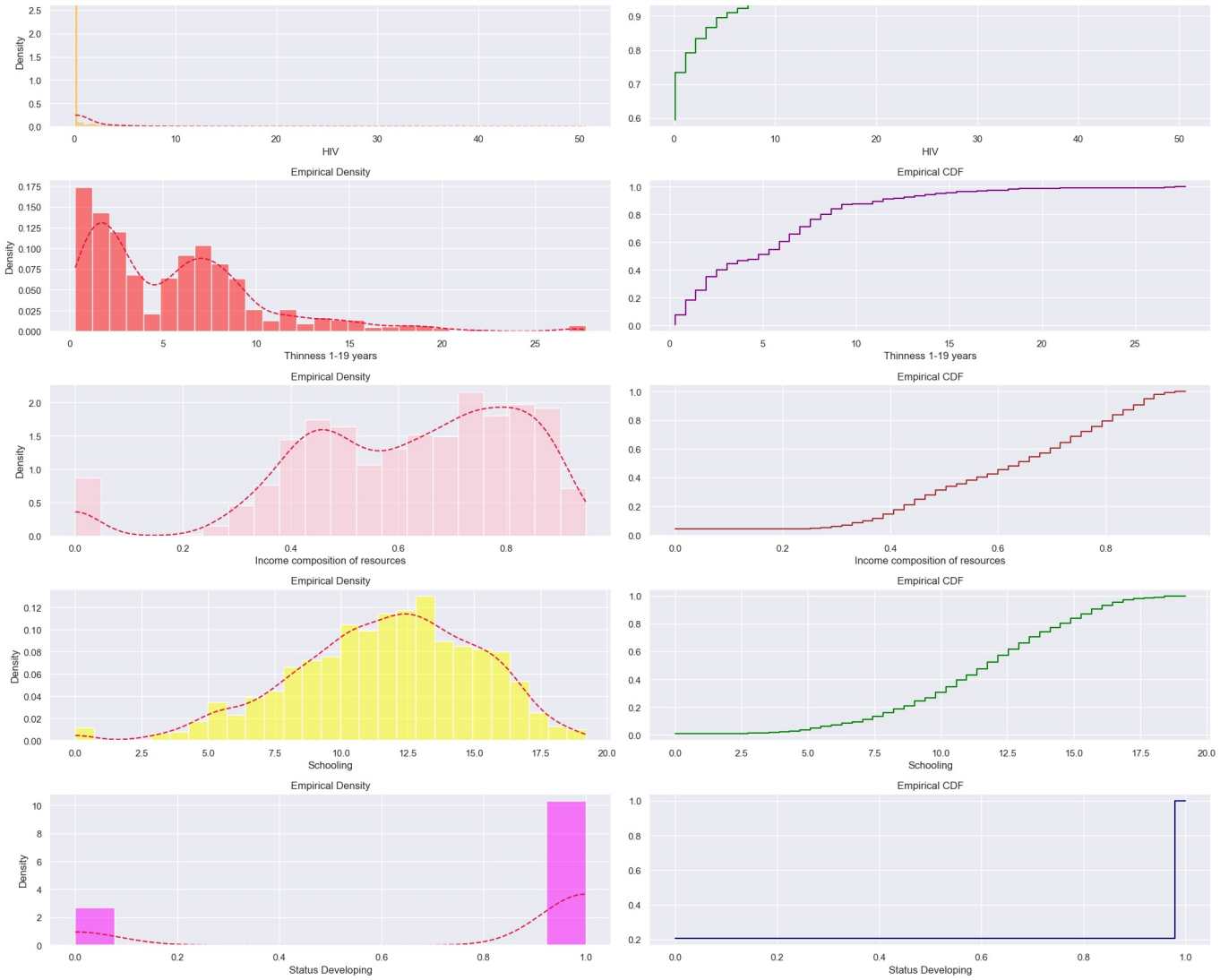
In this analysis, the code generates empirical density and cumulative distribution function (CDF) plots for multiple variables in the dataset to explore their distributions. Empirical density plots provide a visualization of how values are distributed, overlaid with a kernel density estimate (KDE) curve for a smoothed representation. Corresponding CDF plots showcase the cumulative probability for each value, offering insights into the probability structure and percentiles of the data.

In [40]:
```
# Estimating Density Plots

life_expectancy_ecdf = sm.distributions.ECDF(raw_data_knn_test_imputed['Life_expectancy'])
life_expectancy_range = np.linspace(min(raw_data_knn_test_imputed['Life_expectancy']), max(raw_data_knn_test_imp
life_expectancy_cdf = life_expectancy_ecdf(life_expectancy_range)

adult_mortality_ecdf = sm.distributions.ECDF(raw_data_knn_test_imputed['Adult_Mortality'])
adult_mortality_range = np.linspace(min(raw_data_knn_test_imputed['Adult_Mortality']), max(raw_data_knn_test_imp
adult_mortality_cdf = adult_mortality_ecdf(adult_mortality_range)

infant_deaths_ecdf = sm.distributions.ECDF(raw_data_knn_test_imputed['infant_deaths'])
infant_deaths_range = np.linspace(min(raw_data_knn_test_imputed['infant_deaths']), max(raw_data_knn_test_imputed
infant_deaths_cdf = infant_deaths_ecdf(infant_deaths_range)

bmi_ecdf = sm.distributions.ECDF(raw_data_knn_test_imputed['BMI'])
bmi_range = np.linspace(min(raw_data_knn_test_imputed['BMI']), max(raw_data_knn_test_imputed['BMI']))
bmi_cdf = bmi_ecdf(bmi_range)

polio_ecdf = sm.distributions.ECDF(raw_data_knn_test_imputed['Polio'])
polio_range = np.linspace(min(raw_data_knn_test_imputed['Polio']), max(raw_data_knn_test_imputed['Polio']))
polio_cdf = polio_ecdf(polio_range)

total_expenditure_ecdf = sm.distributions.ECDF(raw_data_knn_test_imputed['Total_expenditure'])
total_expenditure_range = np.linspace(min(raw_data_knn_test_imputed['Total_expenditure']), max(raw_data_knn_tes
total_expenditure_cdf = total_expenditure_ecdf(total_expenditure_range)

diphtheria_ecdf = sm.distributions.ECDF(raw_data_knn_test_imputed['Diphtheria'])
diphtheria_range = np.linspace(min(raw_data_knn_test_imputed['Diphtheria']), max(raw_data_knn_test_imputed['Dipl
diphtheria_cdf = diphtheria_ecdf(diphtheria_range)

hiv_aids_ecdf = sm.distributions.ECDF(raw_data_knn_test_imputed['HIV'])
hiv_aids_range = np.linspace(min(raw_data_knn_test_imputed['HIV']), max(raw_data_knn_test_imputed['HIV']))
hiv_aids_cdf = hiv_aids_ecdf(hiv_aids_range)

thinness_1_19_years_ecdf = sm.distributions.ECDF(raw_data_knn_test_imputed['thinness_1_19_years'])
thinness_1_19_years_range = np.linspace(min(raw_data_knn_test_imputed['thinness_1_19_years']), max(raw_data_knn_
thinness_1_19_years_cdf = thinness_1_19_years_ecdf(thinness_1_19_years_range)

income_comp_ecdf = sm.distributions.ECDF(raw_data_knn_test_imputed['Income_composition_of_resources'])
income_comp_range = np.linspace(min(raw_data_knn_test_imputed['Income_composition_of_resources']), max(raw_data_
income_comp_cdf = income_comp_ecdf(income_comp_range)

schooling_ecdf = sm.distributions.ECDF(raw_data_knn_test_imputed['Schooling'])
schooling_range = np.linspace(min(raw_data_knn_test_imputed['Schooling']), max(raw_data_knn_test_imputed['Schoo
schooling_cdf = schooling_ecdf(schooling_range)

status_developing_ecdf = sm.distributions.ECDF(raw_data_knn_test_imputed['Status_Developing'])
status_developing_range = np.linspace(min(raw_data_knn_test_imputed['Status_Developing']), max(raw_data_knn_tes
status_developing_cdf = status_developing_ecdf(status_developing_range)

fig, ax3 = plt.subplots(12, 2, figsize=(20, 40))

# Life Expectancy
sns.histplot(raw_data_knn_test_imputed['Life_expectancy'], stat='density', kde=True, ax=ax3[0, 0], color='pink'
ax3[0, 0].lines[0].set_color('crimson')
ax3[0, 0].lines[0].set_linestyle('--')
ax3[0, 0].set_xlabel('Life expectancy')
ax3[0, 0].set_title('Empirical Density')
ax3[0, 1].step(life_expectancy_range, life_expectancy_cdf, color='blue')
ax3[0, 1].set_title('Empirical CDF')
ax3[0, 1].set_xlabel('Life expectancy')

# Adult Mortality
sns.histplot(raw_data_knn_test_imputed['Adult_Mortality'], stat='density', kde=True, ax=ax3[1, 0], color='yellow
ax3[1, 0].lines[0].set_color('crimson')
ax3[1, 0].lines[0].set_linestyle('--')
ax3[1, 0].set_xlabel('Adult Mortality')
ax3[1, 0].set_title('Empirical Density')
ax3[1, 1].step(adult_mortality_range, adult_mortality_cdf, color='red')
ax3[1, 1].set_title('Empirical CDF')
ax3[1, 1].set_xlabel('Adult Mortality')

# Infant Deaths
```

```
sns.histplot(raw_data_knn_test_imputed['infant_deaths'], stat='density', kde=True, ax=ax3[2, 0], color='purple'
ax3[2, 0].lines[0].set_color('crimson')
ax3[2, 0].lines[0].set_linestyle('--')
ax3[2, 0].set_xlabel('Infant Deaths')
ax3[2, 0].set_title('Empirical Density')
ax3[2, 1].step(infant_deaths_range, infant_deaths_cdf, color='pink')
ax3[2, 1].set_title('Empirical CDF')
ax3[2, 1].set_xlabel('Infant Deaths')

# BMI
sns.histplot(raw_data_knn_test_imputed['BMI'], stat='density', kde=True, ax=ax3[3, 0], color='purple')
ax3[3, 0].lines[0].set_color('crimson')
ax3[3, 0].lines[0].set_linestyle('--')
ax3[3, 0].set_xlabel('BMI')
ax3[3, 0].set_title('Empirical Density')
ax3[3, 1].step(bmi_range, bmi_cdf, color='pink')
ax3[3, 1].set_title('Empirical CDF')
ax3[3, 1].set_xlabel('BMI')

# Polio
sns.histplot(raw_data_knn_test_imputed['Polio'], stat='density', kde=True, ax=ax3[4, 0], color='red')
ax3[4, 0].lines[0].set_color('crimson')
ax3[4, 0].lines[0].set_linestyle('--')
ax3[4, 0].set_xlabel('Polio')
ax3[4, 0].set_title('Empirical Density')
ax3[4, 1].step(polio_range, polio_cdf, color='orange')
ax3[4, 1].set_title('Empirical CDF')
ax3[4, 1].set_xlabel('Polio')

# Total Expenditure
sns.histplot(raw_data_knn_test_imputed['Total_expenditure'], stat='density', kde=True, ax=ax3[5, 0], color='gre
ax3[5, 0].lines[0].set_color('crimson')
ax3[5, 0].lines[0].set_linestyle('--')
ax3[5, 0].set_xlabel('Total expenditure')
ax3[5, 0].set_title('Empirical Density')
ax3[5, 1].step(total_expenditure_range, total_expenditure_cdf, color='blue')
ax3[5, 1].set_title('Empirical CDF')
ax3[5, 1].set_xlabel('Total expenditure')

# Diphtheria
sns.histplot(raw_data_knn_test_imputed['Diphtheria'], stat='density', kde=True, ax=ax3[6, 0], color='purple')
ax3[6, 0].lines[0].set_color('crimson')
ax3[6, 0].lines[0].set_linestyle('--')
ax3[6, 0].set_xlabel('Diphtheria')
ax3[6, 0].set_title('Empirical Density')
ax3[6, 1].step(diphtheria_range, diphtheria_cdf, color='blue')
ax3[6, 1].set_title('Empirical CDF')
ax3[6, 1].set_xlabel('Diphtheria')

# HIV/AIDS
sns.histplot(raw_data_knn_test_imputed['HIV'], stat='density', kde=True, ax=ax3[7, 0], color='orange')
ax3[7, 0].lines[0].set_color('crimson')
ax3[7, 0].lines[0].set_linestyle('--')
ax3[7, 0].set_xlabel('HIV')
ax3[7, 0].set_title('Empirical Density')
ax3[7, 1].step(hiv_aids_range, hiv_aids_cdf, color='green')
ax3[7, 1].set_title('Empirical CDF')
ax3[7, 1].set_xlabel('HIV')

# Thinness 1-19 years
sns.histplot(raw_data_knn_test_imputed['thinness_1_19_years'], stat='density', kde=True, ax=ax3[8, 0], color='r
ax3[8, 0].lines[0].set_color('crimson')
ax3[8, 0].lines[0].set_linestyle('--')
ax3[8, 0].set_xlabel('Thinness 1-19 years')
ax3[8, 0].set_title('Empirical Density')
ax3[8, 1].step(thinness_1_19_years_range, thinness_1_19_years_cdf, color='purple')
ax3[8, 1].set_title('Empirical CDF')
ax3[8, 1].set_xlabel('Thinness 1-19 years')

# Income Composition of Resources
sns.histplot(raw_data_knn_test_imputed['Income_composition_of_resources'], stat='density', kde=True, ax=ax3[9, (
ax3[9, 0].lines[0].set_color('crimson')
ax3[9, 0].lines[0].set_linestyle('--')
ax3[9, 0].set_xlabel('Income composition of resources')
ax3[9, 0].set_title('Empirical Density')
ax3[9, 1].step(income_comp_range, income_comp_cdf, color='brown')
ax3[9, 1].set_title('Empirical CDF')
ax3[9, 1].set_xlabel('Income composition of resources')

# Schooling
sns.histplot(raw_data_knn_test_imputed['Schooling'], stat='density', kde=True, ax=ax3[10, 0], color='yellow')
ax3[10, 0].lines[0].set_color('crimson')
ax3[10, 0].lines[0].set_linestyle('--')
```

```
ax3[10, 0].set_xlabel('Schooling')
ax3[10, 0].set_title('Empirical Density')
ax3[10, 1].step(schooling_range, schooling_cdf, color='green')
ax3[10, 1].set_title('Empirical CDF')
ax3[10, 1].set_xlabel('Schooling')

# Status Developing
sns.histplot(raw_data_knn_test_imputed['Status_Developing'], stat='density', kde=True, ax=ax3[11, 0], color='mag
ax3[11, 0].lines[0].set_color('crimson')
ax3[11, 0].lines[0].set_linestyle('--')
ax3[11, 0].set_xlabel('Status Developing')
ax3[11, 0].set_title('Empirical Density')
ax3[11, 1].step(status_developing_range, status_developing_cdf, color='navy')
ax3[11, 1].set_title('Empirical CDF')
ax3[11, 1].set_xlabel('Status Developing')



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





Variables such as Life_expectancy, Adult_Mortality, Infant_Deaths, and others exhibit varied distributions, with some showing normal-like patterns (e.g., BMI) and others being highly skewed or with heavy tails (e.g., HIV and Infant_Deaths). These visualizations highlight potential transformations required for variables with significant deviations from normality, aiding in preparing the data for further modeling or statistical analysis. This approach complements prior normality assessments, reinforcing the importance of variable-specific adjustments for robust analyses.

Cullen Frey Graphs

The code utilizes the R library fitdistrplus to perform a detailed exploratory analysis of variable distributions using bootstrapping for increased robustness. Summary statistics such as mean, median, standard deviation, skewness, and kurtosis are computed for each variable. The Cullen and Frey graphs visually map the skewness and kurtosis of the empirical data alongside theoretical distributions like normal, exponential, and gamma.

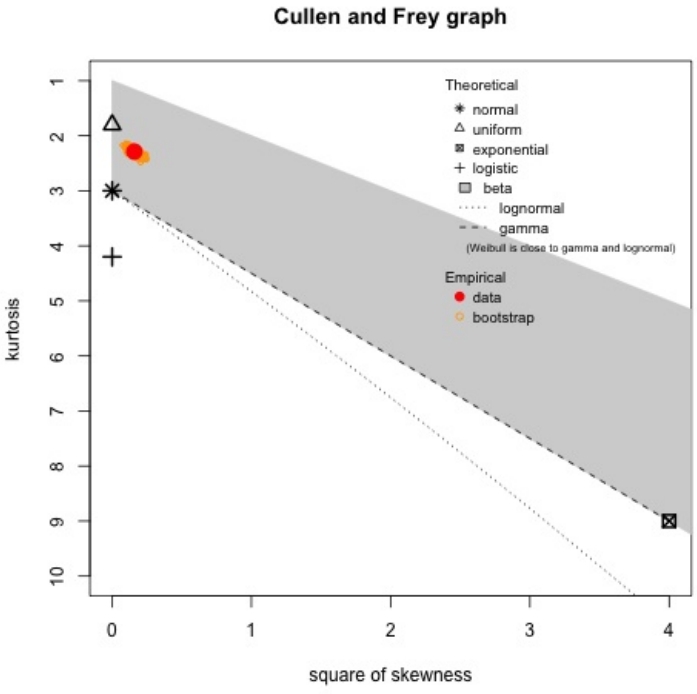
In [44]:
```
%R -i raw_data_knn_test_imputed
```

In [45]:
```
%%R
ls()  # List objects in the environment
str(raw_data_knn_test_imputed)  # Replace with your dataset name to inspect its structure
```

```
'data.frame':   2242 obs. of  24 variables:
 $ Year                           : chr  "2015.0" "2014.0" "2013.0" "2012.0" ...
 $ Life_expectancy                : num  65 59.9 59.9 59.5 59.2 58.8 58.6 58.1 57.5 57.3 ...
 $ Adult_Mortality                : num  263 271 268 272 275 279 281 287 295 295 ...
 $ infant_deaths                  : num  62 64 66 69 71 74 77 80 82 84 ...
 $ Alcohol                        : num  0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.03 0.02 0.03 ...
 $ percentage_expenditure         : num  71.3 73.5 73.2 78.2 7.1 ...
 $ Hepatitis_B                    : num  65 62 64 67 68 66 63 64 63 64 ...
 $ Measles                        : num  1154 492 430 2787 3013 ...
 $ BMI                            : num  19.1 18.6 18.1 17.6 17.2 16.7 16.2 15.7 15.2 14.7 ...
 $ under_five_deaths              : num  83 86 89 93 97 102 106 110 113 116 ...
 $ Polio                          : num  6 58 62 67 68 66 63 64 63 58 ...
 $ Total_expenditure              : num  8.16 8.18 8.13 8.52 7.87 9.2 9.42 8.33 6.73 7.43 ...
 $ Diphtheria                     : num  65 62 64 67 68 66 63 64 63 58 ...
 $ HIV                            : num  0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 ...
 $ GDP                            : num  584.3 612.7 631.7 670 63.5 ...
 $ Population                     : num  33736494 327582 31731688 3696958 2978599 ...
 $ thinness_1_19_years            : num  17.2 17.5 17.7 17.9 18.2 18.4 18.6 18.8 19 19.2 ...
 $ thinness_5_9_years             : num  17.3 17.5 17.7 18 18.2 18.4 18.7 18.9 19.1 19.3 ...
 $ Income_composition_of_resources: num  0.479 0.476 0.47 0.463 0.454 0.448 0.434 0.433 0.415 0.405 ...
 $ Schooling                      : num  10.1 10 9.9 9.8 9.5 9.2 8.9 8.7 8.4 8.1 ...
 $ Status_Developed               : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Status_Developing              : num  1 1 1 1 1 1 1 1 1 1 ...
 $ Country                        : chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
 $ Continent                      : chr  "Asia" "Asia" "Asia" "Asia" ...
```

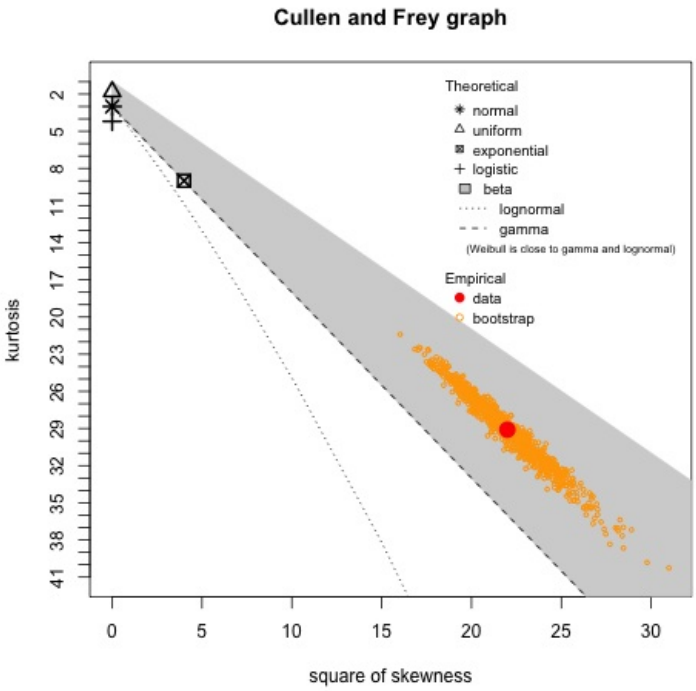
In [46]:
```
%%R
set.seed(10) # for bootstrap
library("fitdistrplus")
descdist(raw_data_knn_test_imputed$ Life_expectancy, boot = 1000)
```

```
summary statistics
------
min:  39   max:  89
median:  69.9
mean:  68.08397
estimated sd:  10.25364
estimated skewness:  -0.3989382
estimated kurtosis:  2.28833
Loading required package: MASS
Loading required package: survival
```



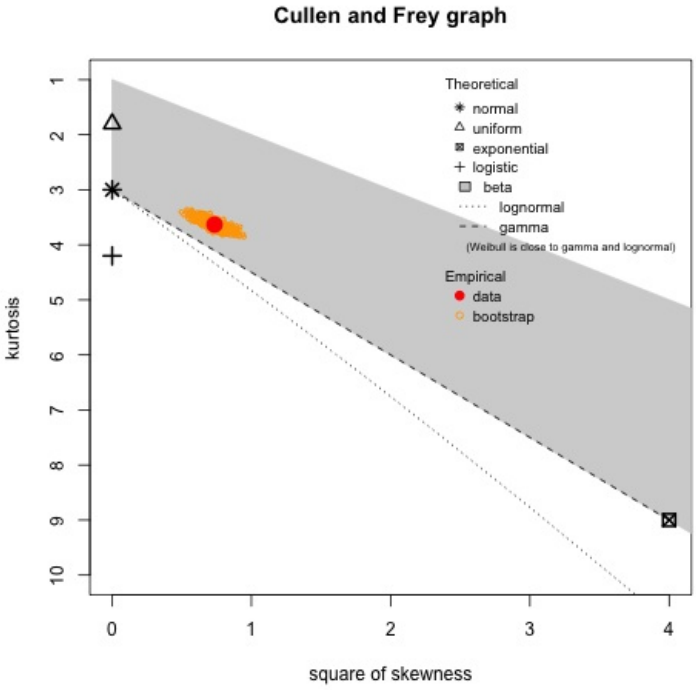
In [47]:
```
%%R
set.seed(10) # for bootstrap
library("fitdistrplus")
descdist(raw_data_knn_test_imputed$ HIV, boot = 1000)
```

```
summary statistics
------
min:  0.1   max:  50.6
median:  0.1
mean:  2.174888
estimated sd:  5.734309
estimated skewness:  4.689927
estimated kurtosis:  29.09723
```



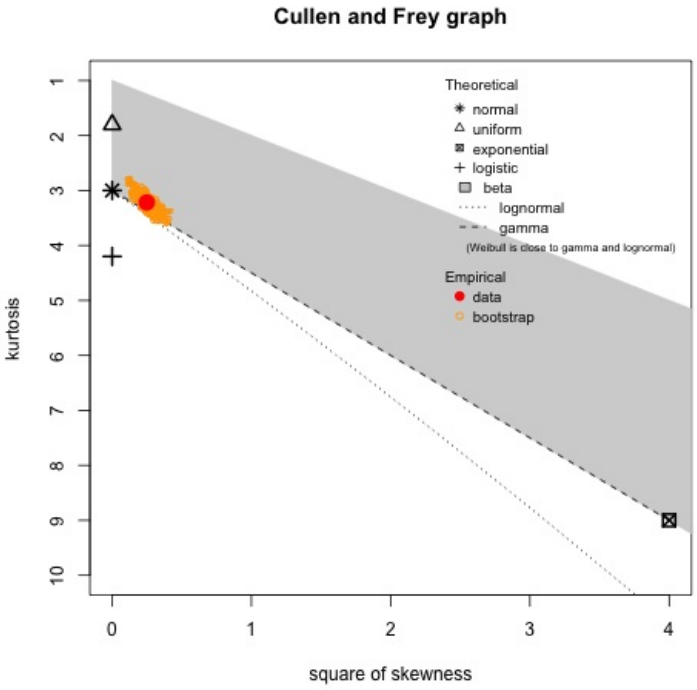
In [48]:
```
%%R
set.seed(10) # for bootstrap
library("fitdistrplus")
descdist(raw_data_knn_test_imputed$ Income_composition_of_resources, boot = 1000)
```

```
summary statistics
------
min:  0   max:  0.948
median:  0.649
mean:  0.6156598
estimated sd:  0.2147092
estimated skewness:  -0.8572933
estimated kurtosis:  3.632542
```



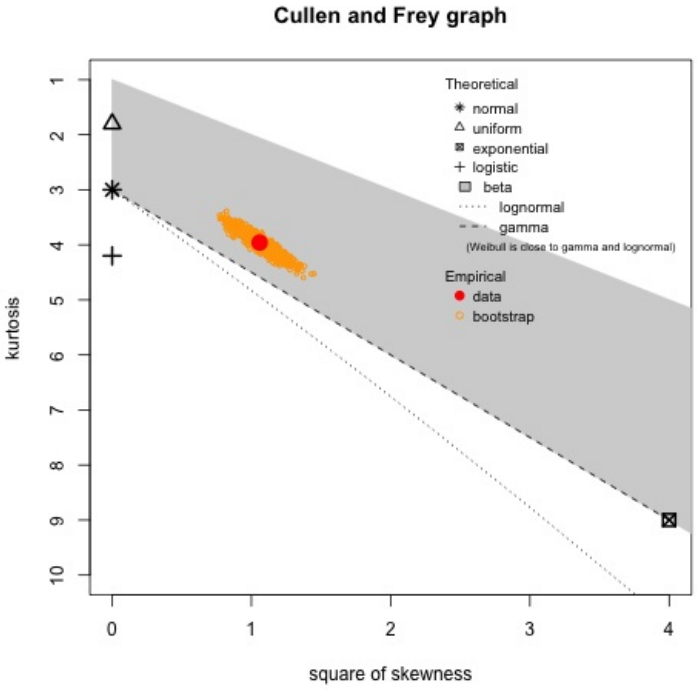
In [49]:
```
%%R
set.seed(10) # for bootstrap
library("fitdistrplus")
descdist(raw_data_knn_test_imputed$ Schooling, boot = 1000)
```

```
summary statistics
------
min:  0   max:  19.2
median:  12
mean:  11.68708
estimated sd:  3.441876
estimated skewness:  -0.4985725
estimated kurtosis:  3.217794
```



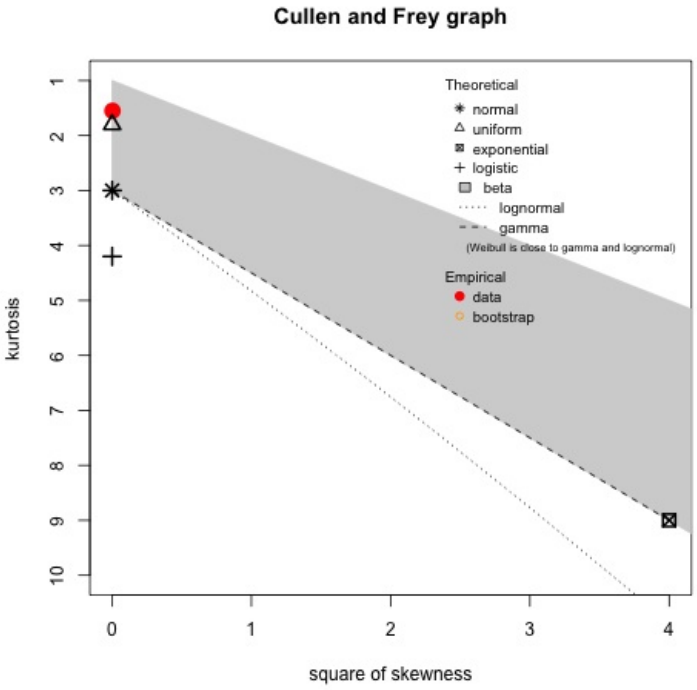
In [50]:
```
%%R
set.seed(10) # for bootstrap
library("fitdistrplus")
descdist(raw_data_knn_test_imputed$ Adult_Mortality, boot = 1000)
```

```
summary statistics
------
min:  1   max:  723
median:  142
mean:  174.8833
estimated sd:  134.7135
estimated skewness:  1.029205
estimated kurtosis:  3.959211
```



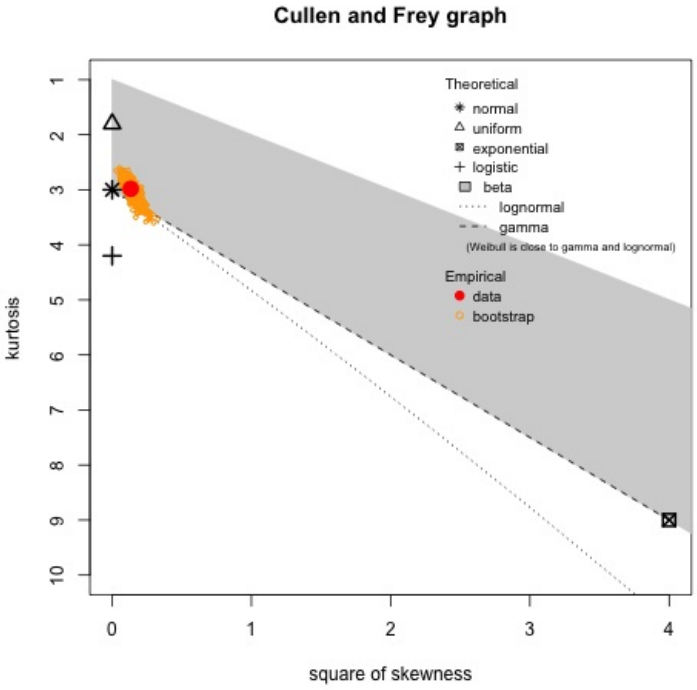
In [51]:
```
%%R
set.seed(10) # for bootstrap
library("fitdistrplus")
descdist(raw_data_knn_test_imputed$ BMI, boot = 1000)
```

```
summary statistics
------
min:  1   max:  71.4
median:  29.7
mean:  35.09841
estimated sd:  19.85304
estimated skewness:  0.04780681
estimated kurtosis:  1.55039
```



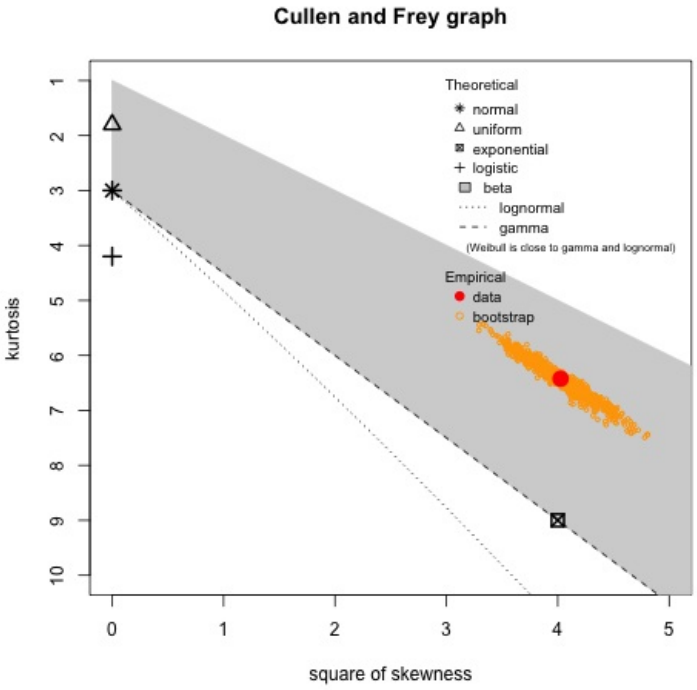
In [52]:
```
%%R
set.seed(10) # for bootstrap
library("fitdistrplus")
descdist(raw_data_knn_test_imputed$ Total_expenditure, boot = 1000)
```

```
summary statistics
------
min:  0.37   max:  16.59
median:  5.54
mean:  5.707266
estimated sd:  2.359766
estimated skewness:  0.3650335
estimated kurtosis:  2.982441
```



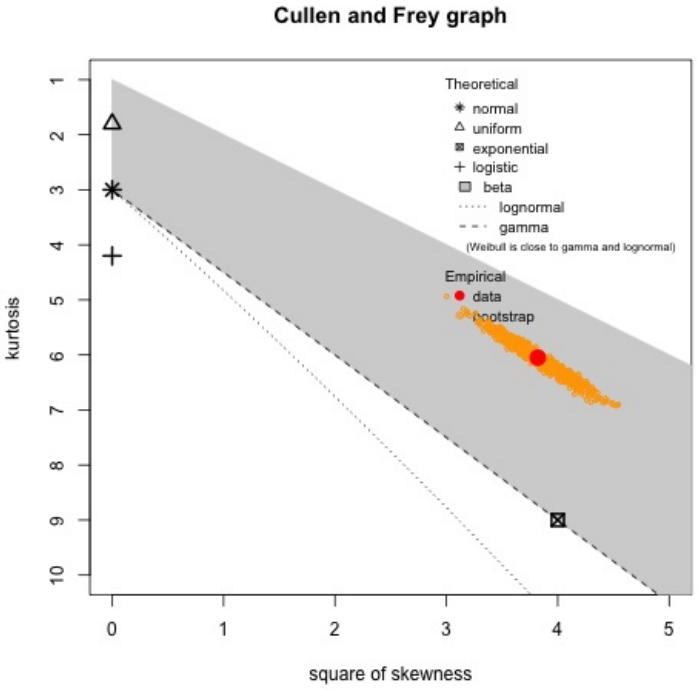
In [53]:
```
%%R
set.seed(10) # for bootstrap
library("fitdistrplus")
descdist(raw_data_knn_test_imputed$ Polio, boot = 1000)
```

```
summary statistics
------
min:  3   max:  99
median:  93
mean:  82.29304
estimated sd:  23.60052
estimated skewness:  -2.006418
estimated kurtosis:  6.423711
```



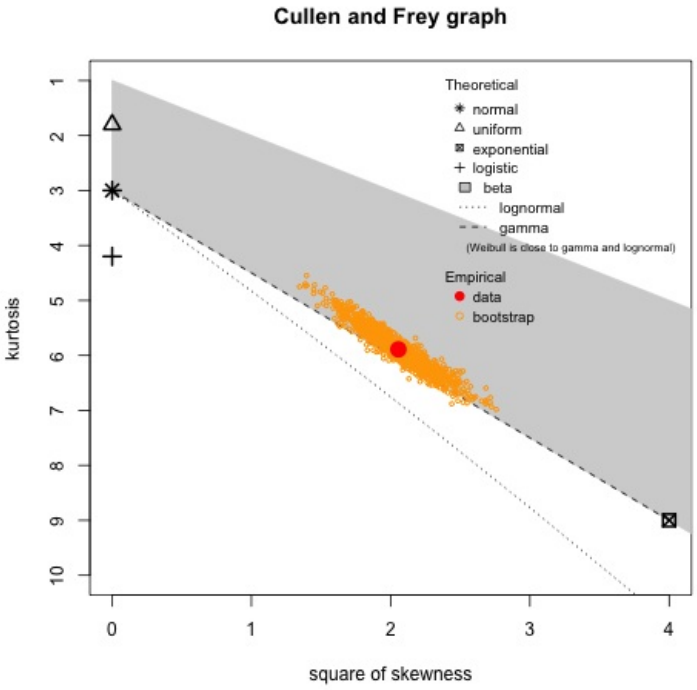
In [54]:
```
%%R
set.seed(10) # for bootstrap
library("fitdistrplus")
descdist(raw_data_knn_test_imputed$ Diphtheria, boot = 1000)
```

```
summary statistics
------
min:  2   max:  99
median:  93
mean:  81.84255
estimated sd:  24.17475
estimated skewness:  -1.954761
estimated kurtosis:  6.049345
```



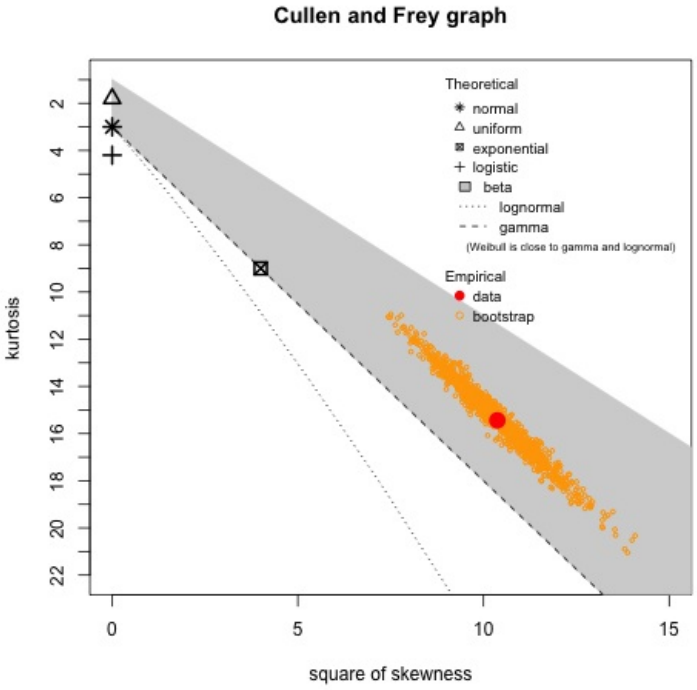
In [55]:
```
%%R
set.seed(10) # for bootstrap
library("fitdistrplus")
descdist(raw_data_knn_test_imputed$ thinness_1_19_years, boot = 1000)
```

```
summary statistics
------
min:  0.3   max:  27.7
median:  5.1
mean:  5.685995
estimated sd:  4.678248
estimated skewness:  1.433687
estimated kurtosis:  5.890431
```



In [56]:
```
%%R
set.seed(10) # for bootstrap
library("fitdistrplus")
descdist(raw_data_knn_test_imputed$ GDP, boot = 1000)
```

```
summary statistics
------
min:  1.68135   max:  119172.7
median:  1335.054
mean:  7257.415
estimated sd:  14491.2
estimated skewness:  3.220751
estimated kurtosis:  15.44049
```



Variables such as Adult_Mortality and Polio exhibit high skewness and kurtosis, aligning closer to lognormal or gamma distributions rather than normal. This suggests significant deviations from symmetry and potential long tails in these distributions. The inclusion of bootstrapped estimates enhances the reliability of the analysis by providing a resampling-based view of the distribution characteristics.

The insights obtained help identify the most suitable transformations or modeling approaches for each variable to meet distributional assumptions in subsequent analyses.

Q2) c)

Box-Cox and Yeo Johnson

Next, we examine non-linearity in our variables and apply the Box Cox And Yeo Johnson transformations to our data to correct for them.

The Box-Cox transformation is particularly useful for handling skewed or non-normally distributed variables by identifying the optimal power transformation parameter (λ) to stabilize variance and improve linearity.

Firstly, we ran Boruta Algorithm, Mallow's CP, forward and backward testing on all the variables and decided on top 10 quantitative variables like HIV, Schooling, Adult Mortality etc. Additionally, we considered Continent and Status_Developing as our factor variables.

Next, we found out Income composition of resources and Schooling has non-positive numbers and thereby conducted Yeo Johnson for them and Box-Cox for the rest of the variables to transformithem closer to normality, which improved their fit in the linear regression model. This transformation is crucial because using non-linear variables in regression without adjusting them can lead to misleading results, biased estimates, and inefficiencies. The transformation helped meet the assumptions of ordinary least squares (OLS) regression, ensuring the model's validity. Finally, we reassessed the model to confirm improvements in the linear relationship between the dependent and independent variables.

In [57]:
```
# Box Cox

#variables which reflected 'False' for including non-positive numbers as Box-Cox can be applied only to positiv
variables = ['Life_expectancy' , 'Adult_Mortality' , 'Alcohol' , 'BMI' , 'Polio' , 'Total_expenditure' , 'Dipht
# Loop through each variable and perform Box-Cox transformation
for var in variables:
    print(f'\n===== Results for {var} =====')

    # Drop missing values from the current variables
    data = raw_data_knn_test_imputed[var].dropna()

    # Perform Box-Cox transformation and get optimal lambda
    _, lambda_opt, ci = stats.boxcox(data, alpha=0.05)

    # Print the optimal results
    print(f'estimated power = {lambda_opt:.4}\n' +
          f'rounded power = {np.round(lambda_opt,1)}\n' +
          f'Confidence interval = {np.round(ci, 4)}')

    # Compute log-likelihoods for different lambda values
    Lopt_ll = stats.boxcox_llf(data=data, lmb=lambda_opt)  # optimal lambda
    L0_ll = stats.boxcox_llf(data=data, lmb=0)  # lambda = 0
    L05_ll = stats.boxcox_llf(data=data, lmb=0.5)  # lambda = 0.5
    L1_ll = stats.boxcox_llf(data=data, lmb=1)  # lambda = 1
    test_lambda = [0, 0.5, 1]  # combine into one list

    # Compute likelihood-ratio test statistics: T = 2 * (LogLik_optimal - LogLik_)
    L0_stat = 2 * (Lopt_ll - L0_ll)
    L05_stat = 2 * (Lopt_ll - L05_ll)
    L1_stat = 2 * (Lopt_ll - L1_ll)
    test_stat = [L0_stat, L05_stat, L1_stat]  # combine into one list

    # Compute p-values
    L0_p = 1 - stats.chi2.cdf(L0_stat, df=1)
    L05_p = 1 - stats.chi2.cdf(L05_stat, df=1)
    L1_p = 1 - stats.chi2.cdf(L1_stat, df=1)
    test_p = [L0_p, L05_p, L1_p]  # combine into one list

    # Print the results
    print('\nLambda \t Test Stat \t p-value')
    for p in zip(test_lambda, test_stat, test_p):
        print(f'{p[0]} \t {p[1]:.6} \t {p[2]:.4}')
```

```
===== Results for Life_expectancy =====
estimated power = 2.124
rounded power = 2.1
Confidence interval = [1.8534 2.3959]

Lambda   Test Stat       p-value
0        248.382         0.0
0.5      143.452         0.0
1        67.857          2.22e-16

===== Results for Adult_Mortality =====
estimated power = 0.3996
```

```
rounded power = 0.4
Confidence interval = [0.3629 0.4368]

Lambda   Test Stat       p-value
0        526.278         0.0
0.5      27.2829         1.758e-07
1        824.061         0.0

===== Results for Alcohol =====
estimated power = 0.3089
rounded power = 0.3
Confidence interval = [0.2865 0.3315]

Lambda   Test Stat       p-value
0        842.749         0.0
0.5      245.766         0.0
1        2385.23         0.0

===== Results for BMI =====
estimated power = 0.6608
rounded power = 0.7
Confidence interval = [0.6044 0.7185]

Lambda   Test Stat       p-value
0        666.999         0.0
0.5      32.5453         1.164e-08
1        119.706         0.0

===== Results for Polio =====
estimated power = 2.637
rounded power = 2.6
Confidence interval = [2.5069 2.771 ]

Lambda   Test Stat       p-value
0        3994.31         0.0
0.5      2133.92         0.0
1        1014.72         0.0

===== Results for Total_expenditure =====
estimated power = 0.6591
rounded power = 0.7
Confidence interval = [0.5839 0.7351]

Lambda   Test Stat       p-value
0        324.695         0.0
0.5      17.4291         2.982e-05
1        74.1804         0.0

===== Results for Diphtheria =====
estimated power = 2.616
rounded power = 2.6
Confidence interval = [2.4894 2.7472]

Lambda   Test Stat       p-value
0        3910.96         0.0
0.5      2095.54         0.0
1        1001.48         0.0

===== Results for HIV =====
estimated power = -0.5534
rounded power = -0.6
Confidence interval = [-0.5923 -0.5151]

Lambda   Test Stat       p-value
0        1029.78         0.0
0.5      4635.57         0.0
1        11173.8         0.0

===== Results for GDP =====
estimated power = -0.0008351
rounded power = -0.0
Confidence interval = [-0.02    0.0185]

Lambda   Test Stat       p-value
0        0.00723834      0.9322
0.5      2195.34         0.0
1        7277.91         0.0

===== Results for thinness_1_19_years =====
estimated power = 0.2143
rounded power = 0.2
Confidence interval = [0.1704 0.2584]
```

```
Lambda   Test Stat       p-value
0        92.2617         0.0
0.5      158.851         0.0
1        1154.46         0.0
```

In [58]:
```python
# Yeo Johnson

# Variables which may have non-positive numbers
variables1 = ['Income_composition_of_resources', 'Schooling']

# Loop through each variable and perform Box-Cox transformation
for var in variables1:
    print(f'\n===== Results for {var} =====')

    # Drop missing values from the current variable
    data = raw_data_knn_test_imputed[var].dropna()

    # Add 1 to the data wherever there is a zero to make sure the data is positive
    data = data.apply(lambda x: x + 1 )

    # Perform Box-Cox transformation and get optimal lambda
    ##transformed_data, lambda_opt = stats.boxcox(data)
    _, lambda_opt, ci = stats.boxcox(data, alpha=0.05)

    # Print the optimal results
    print(f'Estimated power (lambda) = {lambda_opt:.4}\n' +
          f'Rounded power (lambda) = {np.round(lambda_opt, 1)}\n' +
          f'Confidence interval = {np.round(ci, 4)}')

    # Calculate the log-likelihood of the transformed data at the optimal lambda
    log_likelihood_opt = stats.boxcox_llf(data=data, lmb=lambda_opt)

    # Compute the likelihood for the transformed data at different lambdas (0, 0.5, and 1)
    log_likelihood_0 = stats.boxcox_llf(data=data, lmb=0)
    log_likelihood_05 = stats.boxcox_llf(data=data, lmb=0.5)
    log_likelihood_1 = stats.boxcox_llf(data=data, lmb=1)
    test_lambda = [0, 0.5, 1]  # combine into one list

    L0_stat = 2 * (Lopt_ll - L0_ll)
    L05_stat = 2 * (Lopt_ll - L05_ll)
    L1_stat = 2 * (Lopt_ll - L1_ll)
    test_stat = [L0_stat, L05_stat, L1_stat]  # combine into one list

    # Compute likelihood ratio statistics for 0, 0.5, and 1 lambdas
    L0_stat = 2 * (log_likelihood_opt - log_likelihood_0)
    L05_stat = 2 * (log_likelihood_opt - log_likelihood_05)
    L1_stat = 2 * (log_likelihood_opt - log_likelihood_1)

    # Compare each test statistic to the chi-squared critical value to check for significance
    L0_p = 1 - stats.chi2.cdf(L0_stat, df=1)
    L05_p = 1 - stats.chi2.cdf(L05_stat, df=1)
    L1_p = 1 - stats.chi2.cdf(L1_stat, df=1)
    test_p = [L0_p, L05_p, L1_p]  # combine into one list


    # Print the results
    print('\nLambda \t Test Stat \t p-value')
    for p in zip([0, 0.5, 1], [L0_stat, L05_stat, L1_stat], [L0_p, L05_p, L1_p]):
        print(f'{p[0]} \t {p[1]:.6} \t {p[2]:.4}')
```

```
===== Results for Income_composition_of_resources =====
Estimated power (lambda) = 3.241
Rounded power (lambda) = 3.2
Confidence interval = [2.9551 3.5342]

Lambda   Test Stat       p-value
0        623.555         0.0
0.5      428.838         0.0
1        275.377         0.0

===== Results for Schooling =====
Estimated power (lambda) = 1.447
Rounded power (lambda) = 1.4
Confidence interval = [1.3375 1.5604]

Lambda   Test Stat       p-value
0        1237.54         0.0
0.5      399.51          0.0
1        71.5892         0.0
```

Key Insights:

1. Variables like Life expectancy, Polio, Diphtheria, and Alcohol suggest that substantial transformations like quadratic transformation may be needed.

2. Variables like Adult Mortality, BMI, and Total expenditure suggest that a mild transformation like square-root transformation can be done.

3. HIV has a negative estimated power, an inverse transformation may be appropriate.

4. GDP has a very low estimated power, indicating that no transformation is needed.

By applying these transformations, we aim to reduce non-linearity and better approximate a linear relationship between the predictors and the dependent variable, life expectancy.

What would happen if you included nonlinear variables in your regression models without transforming them first?

If we included these variables without addressing their non-linearities, the model could become misspecified, leading to biased estimates of coefficients and potentially inaccurate interpretations of the relationships between variables.

Q2) d)

Next, we look at unusual features of our model using the following tools:

Leverage plot

The leverage vs. studentized residuals plot identifies influential observations. Points with high leverage are far from the center of the predictor variable space, and those with high studentized residuals indicate a poor fit for the regression model.

Cook's distance plot

This plot measures the influence of each observation on the overall regression coefficients. Observations with high Cook's distance have a disproportionate effect on the model.

QQ plot

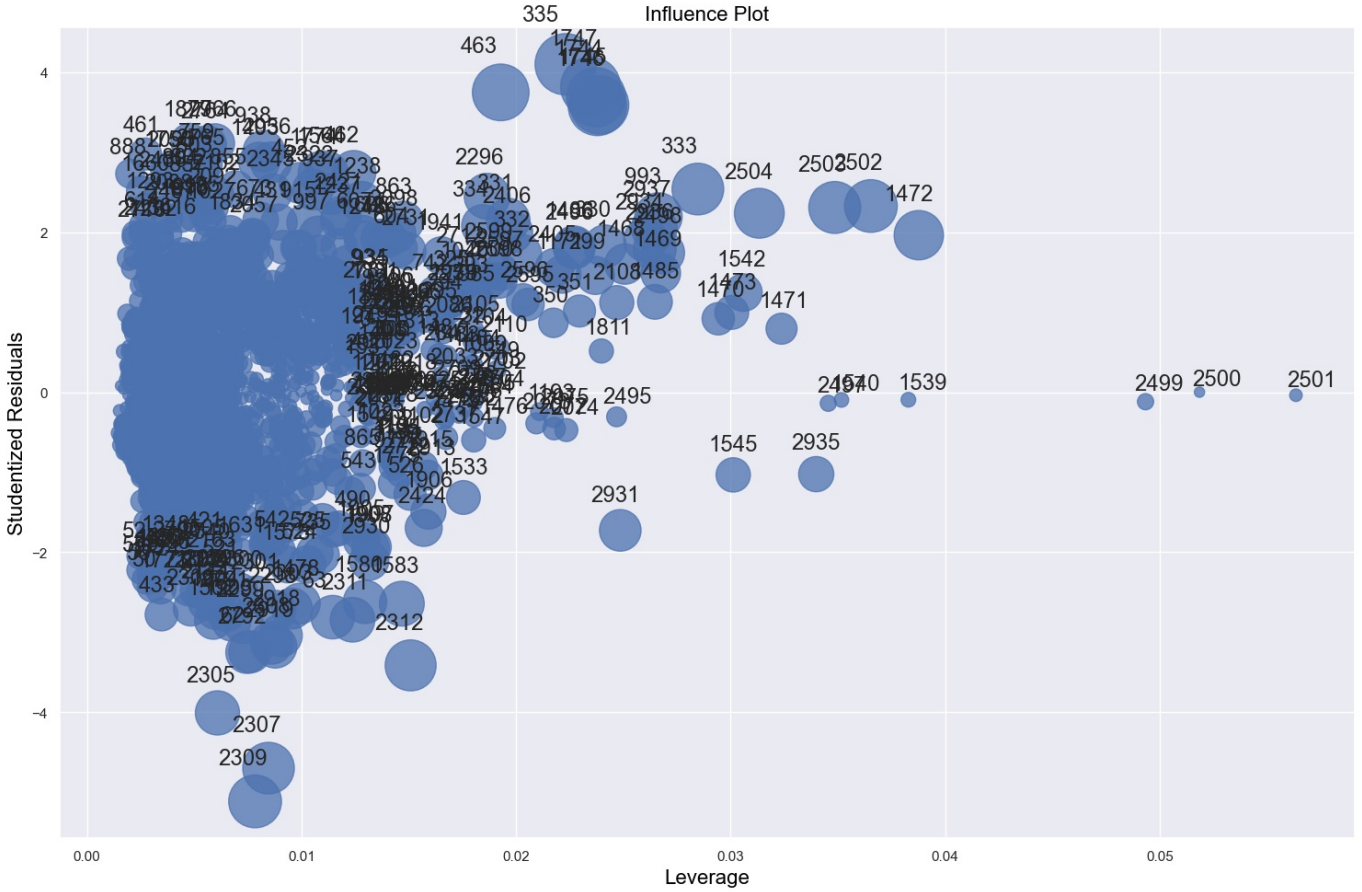
Looking at residual normality

We run a simple linear regression to examine these factors

In [59]:
```
model_1_lev = smf.ols('Life_expectancy ~ HIV + Adult_Mortality + BMI +  Total_expenditure + Polio + Diphtheria +
print(model_1_lev.summary())
```

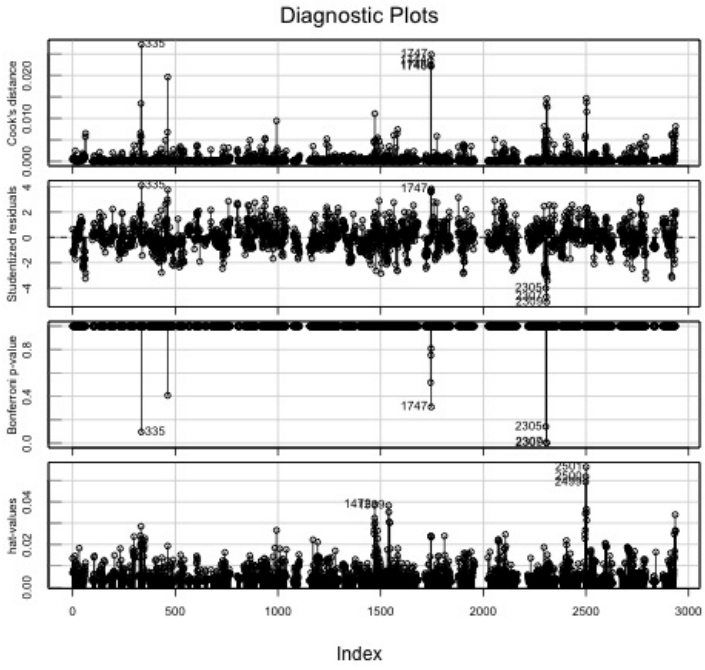
```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Life_expectancy   R-squared:                       0.863
Model:                            OLS   Adj. R-squared:                  0.862
Method:                 Least Squares   F-statistic:                     1076.
Date:                Wed, 20 Nov 2024   Prob (F-statistic):               0.00
Time:                        14:14:43   Log-Likelihood:                -6174.1
No. Observations:                2242   AIC:                         1.238e+04
Df Residuals:                    2228   BIC:                         1.246e+04
Df Model:                          13
Covariance Type:            nonrobust
=====================================================================================================
                                    coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------------------
Intercept                        51.5092      0.648     79.456      0.000      50.238      52.780
Continent[T.Asia]                 3.7678      0.240     15.701      0.000       3.297       4.238
Continent[T.Europe]               3.6934      0.355     10.415      0.000       2.998       4.389
HIV                              -0.3928      0.017    -22.557      0.000      -0.427      -0.359
Adult_Mortality                  -0.0151      0.001    -18.537      0.000      -0.017      -0.013
BMI                               0.0303      0.006      5.371      0.000       0.019       0.041
Total_expenditure                 0.0298      0.038      0.784      0.433      -0.045       0.104
Polio                             0.0322      0.005      6.319      0.000       0.022       0.042
Diphtheria                        0.0331      0.005      6.628      0.000       0.023       0.043
thinness_1_19_years              -0.0292      0.022     -1.306      0.192      -0.073       0.015
Schooling                         0.6412      0.049     12.989      0.000       0.544       0.738
Income_composition_of_resources   7.6323      0.757     10.077      0.000       6.147       9.118
GDP                           4.446e-05   6.65e-06      6.685      0.000    3.14e-05    5.75e-05
Status_Developing                -1.4702      0.312     -4.714      0.000      -2.082      -0.859
==============================================================================
Omnibus:                       63.027   Durbin-Watson:                   0.689
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              153.908
Skew:                           0.028   Prob(JB):                     3.80e-34
Kurtosis:                       4.282   Cond. No.                     1.59e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.59e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [60]:
```
# Leverage Influence Plot
fig, ax = plt.subplots(figsize=(15,10))
fig = sm.graphics.influence_plot(model_1_lev, ax = ax, criterion="DFFITS")
fig.tight_layout()
```

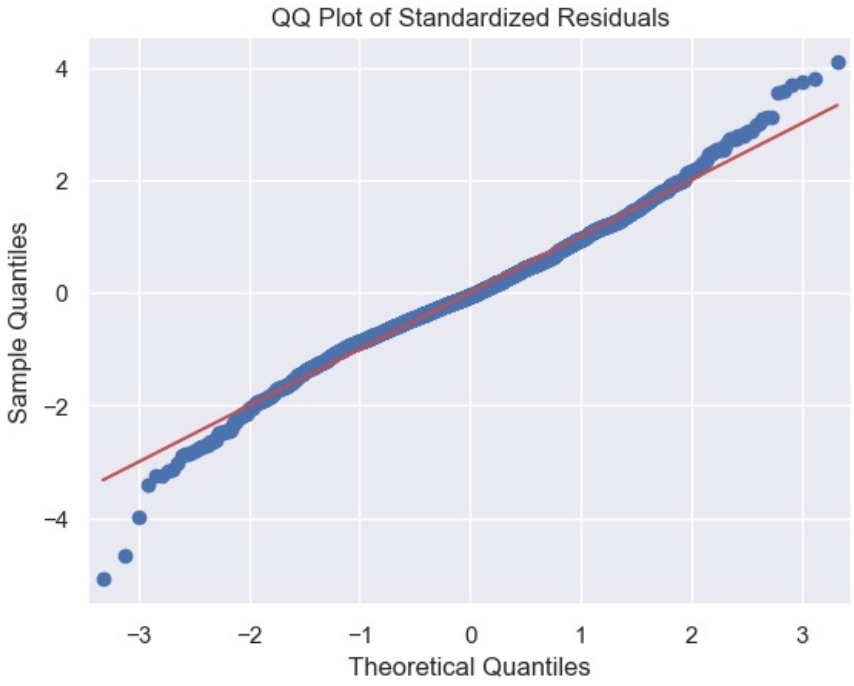


In [65]:
```
%%R
# Load necessary library
library(car)

# Fit the linear regression model
model <- lm(
  Life_expectancy ~ HIV + Adult_Mortality + BMI + Total_expenditure + Polio +
  Diphtheria + thinness_1_19_years + Schooling + Income_composition_of_resources +
  GDP + Status_Developing + Continent,
  data = raw_data_knn_test_imputed
)

# Plot Cook's distance
influenceIndexPlot(model, id = list(n = 5))
```

```
Loading required package: carData
```



In [66]:
```
# Of OLS residuals
model_1_lev_influence = model_1_lev.get_influence()
model_1_lev_adj_std_resid = model_1_lev_influence.resid_studentized_internal # standardized residual
sm.qqplot(model_1_lev_adj_std_resid, line='s')
plt.title('QQ Plot of Standardized Residuals')
plt.show()
```



Interpretation

Leverage plot

Points on the far right of the plot, such as 2499, 2500 and 2501, have high leverage, meaning they are far from the mean of the predictor variables. Points high up or far down on the Y-axis are outliers in the dependent variable. These points deviate significantly from the model's prediction. Points with both high leverage and large residuals are influential. These are typically large bubbles far from the origin, such as points 1542, 1545, and 1472. They need to be looked at as influential points can disproportionately affect the regression coefficients and predictions. The dense cluster of points near the origin (small bubble sizes) are not influential and conform well to the regression model.

Cook's distance plot

The observations, such as 335 and 17476, have unusually high values indicating that these points exert a significant influence on the regression model. Most points have a very low Cook's Distance, indicating that they have little to no influence on the model.

QQ plot

The points closely follow the red line, indicating that the assumption of normally distributed residuals is met for the majority of the data. There are a few points that deviate significantly from the linear trend, suggesting potential outliers in the data.

In [67]:
```
## Checking Influential Points

raw_data_knn_test_imputed.iloc[["463", "335", "1747", "1744", "1746"]]
```

Out[67]:

	Year	Life_expectancy	Adult_Mortality	infant_deaths	Alcohol	percentage_expenditure	Hepatitis_B	Measles	BMI	under_f
656	2000.0	74.7	127.0	0.0	12.73	649.390987	93.333333	9.0	54.7	
463	2000.0	69.9	155.0	0.0	3.49	122.574470	52.666667	2.0	21.5	
2314	2014.0	82.9	56.0	0.0	1.83	7971.646361	96.000000	0.0	32.9	
2311	2001.0	41.0	519.0	30.0	4.21	33.346915	55.333333	649.0	17.5	
2313	2015.0	83.1	55.0	0.0	1.79	0.000000	96.000000	0.0	33.2	

5 rows × 24 columns

In [68]:
```
# Dropping Highly Influential Points

raw_data_knn_test_imputed = raw_data_knn_test_imputed.drop([463, 335, 1747, 1744, 1746])
```

Q2) e)

We used KNN to impute NA's and the missing data. Refer to the start of the code for the same.

Question 3

Check Untransformed Model

We first start our model building with a basic linear model with non-robust OLS errors

In [69]:
```
# Model 1 Untransformed based on Boruta

model_1_untrans = smf.ols('Life_expectancy ~ HIV + Income_composition_of_resources + Schooling + Adult_Mortality
print(model_1_untrans.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Life_expectancy   R-squared:                       0.867
Model:                            OLS   Adj. R-squared:                  0.866
Method:                 Least Squares   F-statistic:                     1117.
Date:                Wed, 20 Nov 2024   Prob (F-statistic):               0.00
Time:                        14:18:43   Log-Likelihood:                -6124.4
No. Observations:                2237   AIC:                         1.228e+04
Df Residuals:                    2223   BIC:                         1.236e+04
Df Model:                          13
Covariance Type:            nonrobust
=====================================================================================================
                                    coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------------------
Intercept                        50.9434      0.641     79.455      0.000      49.686      52.201
Continent[T.Asia]                 3.6225      0.237     15.283      0.000       3.158       4.087
Continent[T.Europe]               3.2561      0.354      9.199      0.000       2.562       3.950
HIV                              -0.3966      0.017    -23.127      0.000      -0.430      -0.363
Income_composition_of_resources   8.1781      0.750     10.901      0.000       6.707       9.649
Schooling                         0.6961      0.050     14.009      0.000       0.599       0.794
Adult_Mortality                  -0.0147      0.001    -18.353      0.000      -0.016      -0.013
BMI                               0.0274      0.006      4.928      0.000       0.017       0.038
Total_expenditure                 0.0232      0.037      0.620      0.536      -0.050       0.097
Polio                             0.0326      0.005      6.493      0.000       0.023       0.042
Diphtheria                        0.0314      0.005      6.377      0.000       0.022       0.041
thinness_1_19_years              -0.0234      0.022     -1.063      0.288      -0.067       0.020
GDP                            4.233e-05   6.55e-06      6.463      0.000    2.95e-05    5.52e-05
Status_Developing                -1.5966      0.307     -5.194      0.000      -2.199      -0.994
==============================================================================
Omnibus:                       52.325   Durbin-Watson:                   0.692
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              115.550
Skew:                          -0.036   Prob(JB):                     8.10e-26
Kurtosis:                       4.111   Cond. No.                     1.60e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.6e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Model Summary:

R-squared: 0.867 indicates the model explains 86.7% of the variance in life expectancy. Adj. R-squared: 0.866 confirms strong model performance, adjusted for the number of predictors. F-statistic: 1117 with a p-value of 0.00 suggests the overall model is statistically significant.

Significant Predictors:

Predictors with p-values < 0.05 are statistically significant:

1. Continents (Asia, Europe): Both significantly affect life expectancy positively.
2. HIV: A negative impact on life expectancy.
3. Income_composition_of_resources: Strong positive effect.
4. Schooling: Positive impact.
5. Adult_Mortality: Significant negative effect.
6. BMI: Positive effect.
7. Diphtheria: Positive effect.
8. GDP: Positive but small effect.
9. Status_Developing: Negative effect compared to developed status.

Non-Significant Predictors:

Total_expenditure, Polio, thinness_1_19_years: These predictors are not statistically significant (p > 0.05). Key Observations:

- Adult_Mortality has the strongest negative impact on life expectancy.
- Income_composition_of_resources and Schooling are the strongest positive contributors.

However, we still need to check for multicollinearity, non-linearity and heteroskedasticity etc to determine the best model

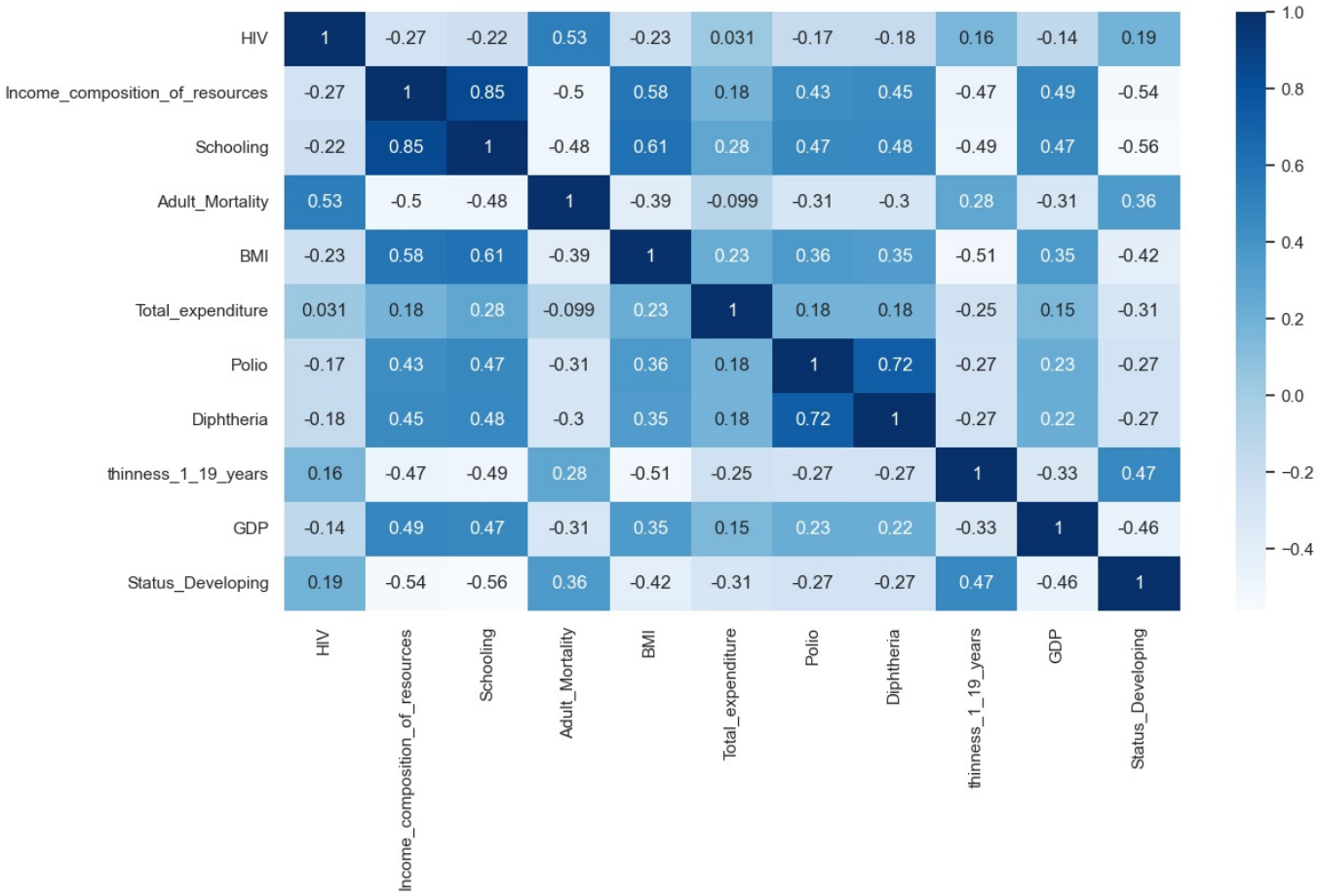
Check For Multicollinearity

First, we check for multicollinearity, first with a pairwise correlation plot and then look at VIF to determine multi-collinearity and high correlations between predictors

1. Pairwise Correlation

In [70]:
```
#Correlation Matrix
plt.figure(figsize=(13,7))
c= raw_data_knn_test_imputed[['HIV', 'Income_composition_of_resources', 'Schooling', 'Adult_Mortality', 'BMI',
print(c)
sns.heatmap(c,annot=True, cmap="Blues")
plt.show()
```

```
                                     HIV  Income_composition_of_resources  \
HIV                                1.000000                         -0.269203
Income_composition_of_resources   -0.269203                          1.000000
Schooling                         -0.223052                          0.846194
Adult_Mortality                    0.533299                         -0.496151
BMI                               -0.228437                          0.580210
Total_expenditure                  0.030504                          0.179131
Polio                             -0.174253                          0.434031
Diphtheria                        -0.175443                          0.451839
thinness_1_19_years                0.161692                         -0.467004
GDP                               -0.144986                          0.486176
Status_Developing                  0.185349                         -0.540280

                                  Schooling  Adult_Mortality       BMI  \
HIV                               -0.223052         0.533299 -0.228437
Income_composition_of_resources    0.846194        -0.496151  0.580210
Schooling                          1.000000        -0.483057  0.605056
Adult_Mortality                   -0.483057         1.000000 -0.394376
BMI                                0.605056        -0.394376  1.000000
Total_expenditure                  0.275659        -0.099230  0.229534
Polio                              0.467903        -0.308889  0.363713
Diphtheria                         0.475143        -0.300267  0.346755
thinness_1_19_years               -0.493221         0.279744 -0.513418
GDP                                0.465442        -0.313634  0.353390
Status_Developing                 -0.562276         0.355501 -0.417207

                                  Total_expenditure     Polio  Diphtheria  \
HIV                                        0.030504 -0.174253   -0.175443
Income_composition_of_resources            0.179131  0.434031    0.451839
Schooling                                  0.275659  0.467903    0.475143
Adult_Mortality                           -0.099230 -0.308889   -0.300267
BMI                                        0.229534  0.363713    0.346755
Total_expenditure                          1.000000  0.179371    0.176357
Polio                                      0.179371  1.000000    0.724245
Diphtheria                                 0.176357  0.724245    1.000000
thinness_1_19_years                       -0.252603 -0.270488   -0.265884
GDP                                        0.149512  0.225049    0.219715
Status_Developing                         -0.308644 -0.265257   -0.266562

                                  thinness_1_19_years       GDP  \
HIV                                          0.161692 -0.144986
Income_composition_of_resources             -0.467004  0.486176
Schooling                                   -0.493221  0.465442
Adult_Mortality                              0.279744 -0.313634
BMI                                         -0.513418  0.353390
Total_expenditure                           -0.252603  0.149512
Polio                                       -0.270488  0.225049
Diphtheria                                  -0.265884  0.219715
thinness_1_19_years                          1.000000 -0.332536
GDP                                         -0.332536  1.000000
Status_Developing                            0.469085 -0.460420

                                  Status_Developing
HIV                                        0.185349
Income_composition_of_resources           -0.540280
Schooling                                 -0.562276
Adult_Mortality                            0.355501
BMI                                       -0.417207
Total_expenditure                         -0.308644
Polio                                     -0.265257
Diphtheria                                -0.266562
thinness_1_19_years                        0.469085
GDP                                       -0.460420
Status_Developing                          1.000000
```



The heatmap shows that some predictors in the regression model are highly correlated, which could lead to multicollinearity issues.

For example, Income_composition_of_resources and Schooling have a very strong correlation (0.85), suggesting they overlap significantly in the information they provide. Similarly, Polio and Diphtheria are strongly correlated (0.73), which might make one of them redundant. There are also moderate correlations, like between GDP and Schooling (0.48) and Diphtheria and Income_composition_of_resources (0.45), which are less severe but still worth noting. On the other hand, variables like HIV, thinness_1_19_years, and Total_expenditure don't show strong correlations with most others, which helps balance the model. However, the strong overlaps, especially between Income_composition_of_resources & Schooling and Polio & Diphtheria, could cause inflated standard errors and unreliable coefficient estimates. To handle this, we next look at VIF to confirm multicollinearity. Depending on the results, we could consider dropping or combining highly correlated variables.

2. VIF Calculation

In [71]:
```
# Check for multicollinearity using VIF in the untransformed model

from statsmodels.stats.outliers_influence import variance_inflation_factor
# Extract the design matrix from the model
X = model_1_untrans.model.exog

# Get the names of the predictors
predictors = model_1_untrans.model.exog_names

# Create a DataFrame for easier handling
X_df = pd.DataFrame(X, columns=predictors)

# Initialize a DataFrame to store VIF results
vif_data = pd.DataFrame()
vif_data["Feature"] = X_df.columns
vif_data["VIF"] = [variance_inflation_factor(X_df.values, i) for i in range(X_df.shape[1])]

# Display the VIF results
print(vif_data)
```

	Feature	VIF
0	Intercept	65.366910
1	Continent[T.Asia]	1.972132
2	Continent[T.Europe]	4.061548
3	HIV	1.540064
4	Income_composition_of_resources	4.056850
5	Schooling	4.562904
6	Adult_Mortality	1.858583
7	BMI	1.941118
8	Total_expenditure	1.242905
9	Polio	2.219911
10	Diphtheria	2.245641
11	thinness_1_19_years	1.691625
12	GDP	1.434273
13	Status_Developing	2.469995

Based on the VIF, we can see that Income composition of resources and Schooling have VIF > 4 indicating potential multicollinearity. Hence, we will drop these variables from our model

In [72]:
```
# Model 1 Untransformed Post VIF

model_1_untrans_post_vif = smf.ols('Life_expectancy ~ HIV + Adult_Mortality + BMI +  Total_expenditure + Polio +
print(model_1_untrans.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Life_expectancy   R-squared:                       0.867
Model:                            OLS   Adj. R-squared:                  0.866
Method:                 Least Squares   F-statistic:                     1117.
Date:                Wed, 20 Nov 2024   Prob (F-statistic):               0.00
Time:                        14:19:04   Log-Likelihood:                -6124.4
No. Observations:                2237   AIC:                         1.228e+04
Df Residuals:                    2223   BIC:                         1.236e+04
Df Model:                          13
Covariance Type:            nonrobust
=================================================================================================
                                    coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------------------
Intercept                        50.9434      0.641     79.455      0.000      49.686      52.201
Continent[T.Asia]                 3.6225      0.237     15.283      0.000       3.158       4.087
Continent[T.Europe]               3.2561      0.354      9.199      0.000       2.562       3.950
HIV                              -0.3966      0.017    -23.127      0.000      -0.430      -0.363
Income_composition_of_resources   8.1781      0.750     10.901      0.000       6.707       9.649
Schooling                         0.6961      0.050     14.009      0.000       0.599       0.794
Adult_Mortality                  -0.0147      0.001    -18.353      0.000      -0.016      -0.013
BMI                               0.0274      0.006      4.928      0.000       0.017       0.038
Total_expenditure                 0.0232      0.037      0.620      0.536      -0.050       0.097
Polio                             0.0326      0.005      6.493      0.000       0.023       0.042
Diphtheria                        0.0314      0.005      6.377      0.000       0.022       0.041
thinness_1_19_years              -0.0234      0.022     -1.063      0.288      -0.067       0.020
GDP                            4.233e-05   6.55e-06      6.463      0.000    2.95e-05    5.52e-05
Status_Developing                -1.5966      0.307     -5.194      0.000      -2.199      -0.994
==============================================================================
Omnibus:                       52.325   Durbin-Watson:                   0.692
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              115.550
Skew:                          -0.036   Prob(JB):                     8.10e-26
Kurtosis:                       4.111   Cond. No.                     1.60e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.6e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Misspecification

Next we check for misspecification/Non-linearity in our variables. We will use the RAMSEY Reset Test to test for the need for transforming our variables or adding indicator variables to fit our model better

Check for misspecification Using RAMSEY Test

In [73]:
```
# Testing model 1

print(linear_reset(model_1_untrans_post_vif, power=2, test_type='fitted', use_f= True))
print(linear_reset(model_1_untrans_post_vif, power=3, test_type='fitted', use_f= True))
```

```
<F test: F=25.966171892291296, p=3.7673921979941895e-07, df_denom=2.22e+03, df_num=1>
<F test: F=100.85475548314162, p=1.1856007440558616e-42, df_denom=2.22e+03, df_num=2>
```

Here, we performed the RAMSEY RESET test specifically using fitted values and test for the need for squaring/cubing our predictor

variables. We find the p-values for both to be < 0.05 indicating that the model may require transformation.

We first attempt to transform using Box-Cox and Yeo-Johnson paramters from Q2

In [74]:
```
# Transforming model using box cox and Yeo Johnson from earlier

model_1_transformed = smf.ols('I(Life_expectancy**2.1) ~ I(HIV**(-0.6)) + I(Adult_Mortality**0.4) + I(BMI**0.7)
print(model_1_transformed.summary())
```

```
                            OLS Regression Results
==============================================================================================
Dep. Variable:     I(Life_expectancy ** 2.1)   R-squared:                       0.836
Model:                                   OLS   Adj. R-squared:                  0.835
Method:                        Least Squares   F-statistic:                     1028.
Date:                       Wed, 20 Nov 2024   Prob (F-statistic):               0.00
Time:                               14:19:07   Log-Likelihood:                -18345.
No. Observations:                       2237   AIC:                         3.671e+04
Df Residuals:                           2225   BIC:                         3.678e+04
Df Model:                                 11
Covariance Type:                  nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
Intercept                     5788.7883    233.514     24.790      0.000    5330.861    6246.715
Continent[T.Asia]              383.6642     67.298      5.701      0.000     251.691     515.637
Continent[T.Europe]            561.2927     88.509      6.342      0.000     387.725     734.861
I(HIV ** (-0.6))               527.5376     23.150     22.788      0.000     482.140     572.935
I(Adult_Mortality ** 0.4)     -135.1697      8.715    -15.509      0.000    -152.261    -118.079
I(BMI ** 0.7)                    6.5912      4.993      1.320      0.187      -3.200      16.382
I(Total_expenditure ** 0.7)     -4.2191     20.634     -0.204      0.838     -44.683      36.244
I(Polio ** 2.6)                  0.0041      0.001      4.897      0.000       0.002       0.006
I(Diphtheria ** 2.6)             0.0043      0.001      5.224      0.000       0.003       0.006
I(thinness_1_19_years ** 0.2)  -779.8222    112.712     -6.919      0.000   -1000.854    -558.790
I(np.log(GDP))                 173.8140     12.319     14.109      0.000     149.656     197.972
Status_Developing             -623.5647     71.238     -8.753      0.000    -763.266    -483.864
==============================================================================
Omnibus:                       41.586   Durbin-Watson:                   0.639
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               54.582
Skew:                           0.235   Prob(JB):                     1.40e-12
Kurtosis:                       3.605   Cond. No.                     2.25e+06
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.25e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [75]:
```
# Testing model 1 transformed

print(linear_reset(model_1_transformed, power=2, test_type='fitted', use_f= True))
print(linear_reset(model_1_transformed, power=3, test_type='fitted', use_f= True))
```

```
<F test: F=7.743622399767395, p=0.005435783370111695, df_denom=2.22e+03, df_num=1>
<F test: F=3.9891893438097568, p=0.018647414086073233, df_denom=2.22e+03, df_num=2>
```

While Box Cox and Yeo Johnson improved p value and make it less statistically significant, it is still statistically significant at the 5% level. We may thus, need to add indicator terms or more squared and cubed terms to our model.

Interaction Terms and Quadratic

In this section, we first run a OLS regression with all our variables and squared terms along with interaction terms from combining all quantitative and factor variables. We will look at the statistical significance of our squared and interaction terms and then, remove those which are not statistically significant

In [76]:
```
VIF_passed_interaction_reg1 = smf.ols(
    'Life_expectancy ~ Adult_Mortality  + BMI  + Polio + \
    Total_expenditure + Diphtheria + HIV + GDP + thinness_1_19_years + \
    Status_Developing + Continent + I(Adult_Mortality**2) + I(BMI**2) + I(Polio**2) + \
    I(Total_expenditure**2) + I(Diphtheria**2) + I(HIV**2) + I(GDP**2) + I(thinness_1_19_years**2) + \
    HIV*Continent + HIV*Status_Developing + \
    Adult_Mortality*Continent + \
    Adult_Mortality*Status_Developing + BMI*Continent + BMI*Status_Developing + Total_expenditure*Continent + T
    Polio*Continent + Polio*Status_Developing + Diphtheria*Continent + Diphtheria*Status_Developing + thinness_
    thinness_1_19_years*Status_Developing + GDP*Continent + GDP*Status_Developing',
    data=raw_data_knn_test_imputed
).fit()

print(VIF_passed_interaction_reg1.summary())
```

OLS Regression Results

Dep. Variable:	Life_expectancy	R-squared:	0.888
Model:	OLS	Adj. R-squared:	0.886
Method:	Least Squares	F-statistic:	415.2
Date:	Wed, 20 Nov 2024	Prob (F-statistic):	0.00
Time:	14:19:10	Log-Likelihood:	-5931.3
No. Observations:	2237	AIC:	1.195e+04
Df Residuals:	2194	BIC:	1.219e+04
Df Model:	42		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	61.1175	2.952	20.702	0.000	55.328	66.907
Continent[T.Asia]	18.1146	1.151	15.737	0.000	15.857	20.372
Continent[T.Europe]	27.9271	2.241	12.464	0.000	23.533	32.321
Adult_Mortality	0.0133	0.005	2.435	0.015	0.003	0.024
Adult_Mortality:Continent[T.Asia]	-0.0350	0.002	-15.152	0.000	-0.040	-0.030
Adult_Mortality:Continent[T.Europe]	-0.0215	0.004	-5.900	0.000	-0.029	-0.014
BMI	-0.0063	0.027	-0.231	0.818	-0.060	0.048
BMI:Continent[T.Asia]	-0.1744	0.014	-12.691	0.000	-0.201	-0.147
BMI:Continent[T.Europe]	-0.1439	0.019	-7.687	0.000	-0.181	-0.107
Polio	-0.0604	0.040	-1.516	0.130	-0.139	0.018
Polio:Continent[T.Asia]	-0.0152	0.010	-1.471	0.141	-0.035	0.005
Polio:Continent[T.Europe]	-0.0548	0.020	-2.759	0.006	-0.094	-0.016
Total_expenditure	-0.6434	0.183	-3.514	0.000	-1.002	-0.284
Total_expenditure:Continent[T.Asia]	0.6082	0.094	6.501	0.000	0.425	0.792
Total_expenditure:Continent[T.Europe]	0.4166	0.112	3.703	0.000	0.196	0.637
Diphtheria	-0.0820	0.038	-2.178	0.030	-0.156	-0.008
Diphtheria:Continent[T.Asia]	-0.0335	0.011	-3.056	0.002	-0.055	-0.012
Diphtheria:Continent[T.Europe]	-0.0579	0.016	-3.738	0.000	-0.088	-0.028
HIV	3.0381	0.286	10.614	0.000	2.477	3.599
HIV:Continent[T.Asia]	-1.4500	0.702	-2.066	0.039	-2.826	-0.074
HIV:Continent[T.Europe]	-3.9054	1.791	-2.181	0.029	-7.417	-0.394
GDP	0.0004	5.44e-05	7.514	0.000	0.000	0.001
GDP:Continent[T.Asia]	-0.0003	5.22e-05	-6.028	0.000	-0.000	-0.000
GDP:Continent[T.Europe]	-0.0003	5.47e-05	-6.335	0.000	-0.000	-0.000
thinness_1_19_years	0.2131	0.448	0.476	0.634	-0.665	1.091
thinness_1_19_years:Continent[T.Asia]	-0.1827	0.058	-3.175	0.002	-0.296	-0.070
thinness_1_19_years:Continent[T.Europe]	-2.6178	0.375	-6.979	0.000	-3.353	-1.882
Status_Developing	-4.3602	2.829	-1.541	0.123	-9.908	1.187
I(Adult_Mortality ** 2)	-5.404e-05	4.77e-06	-11.338	0.000	-6.34e-05	-4.47e-05
I(BMI ** 2)	0.0020	0.000	6.597	0.000	0.001	0.003
I(Polio ** 2)	0.0010	0.000	4.026	0.000	0.001	0.001
I(Total_expenditure ** 2)	0.0059	0.011	0.549	0.583	-0.015	0.027
I(Diphtheria ** 2)	0.0013	0.000	5.230	0.000	0.001	0.002
I(HIV ** 2)	0.0070	0.001	6.355	0.000	0.005	0.009
I(GDP ** 2)	-4.196e-10	2e-10	-2.097	0.036	-8.12e-10	-2.72e-11
I(thinness_1_19_years ** 2)	-0.0002	0.003	-0.060	0.952	-0.006	0.006
HIV:Status_Developing	-3.5097	0.287	-12.225	0.000	-4.073	-2.947
Adult_Mortality:Status_Developing	0.0048	0.005	0.950	0.342	-0.005	0.015
BMI:Status_Developing	0.0512	0.017	3.007	0.003	0.018	0.085
Total_expenditure:Status_Developing	0.3393	0.109	3.109	0.002	0.125	0.553
Polio:Status_Developing	-0.0116	0.032	-0.362	0.717	-0.075	0.051
Diphtheria:Status_Developing	-0.0087	0.028	-0.308	0.758	-0.064	0.047
thinness_1_19_years:Status_Developing	-0.1259	0.443	-0.284	0.776	-0.995	0.743
GDP:Status_Developing	-7.628e-06	2.08e-05	-0.367	0.713	-4.84e-05	3.31e-05

Omnibus:	72.786	Durbin-Watson:	0.780
Prob(Omnibus):	0.000	Jarque-Bera (JB):	167.531
Skew:	0.154	Prob(JB):	4.18e-37
Kurtosis:	4.305	Cond. No.	2.26e+15

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.26e+15. This might indicate that there are
strong multicollinearity or other numerical problems.

Based on the p-values >= 0.05, we will now remove insignificant squared and interaction terms.

In [77]:
```
VIF_passed_interaction_reg2 = smf.ols(
    'Life_expectancy ~ Adult_Mortality  + BMI  + Polio + \
    Total_expenditure + Diphtheria + HIV + GDP + thinness_1_19_years + \
    Status_Developing + Continent + I(Adult_Mortality**2) + I(BMI**2) + I(Polio**2) + \
    I(Diphtheria**2) + I(HIV**2) + I(GDP**2) + \
    HIV*Continent + HIV*Status_Developing + \
    Adult_Mortality*Continent + \
    BMI*Continent + BMI*Status_Developing + Total_expenditure*Continent + Total_expenditure*Status_Developing +
    Polio*Continent + Diphtheria*Continent + thinness_1_19_years*Continent + \
    GDP*Continent',
data=raw_data_knn_test_imputed
```

```
).fit()

print(VIF_passed_interaction_reg2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Life_expectancy   R-squared:                       0.888
Model:                            OLS   Adj. R-squared:                  0.886
Method:                 Least Squares   F-statistic:                     499.3
Date:                Wed, 20 Nov 2024   Prob (F-statistic):               0.00
Time:                        14:19:11   Log-Likelihood:                -5932.5
No. Observations:                2237   AIC:                         1.194e+04
Df Residuals:                    2201   BIC:                         1.214e+04
Df Model:                          35
Covariance Type:            nonrobust
=================================================================================================================
                                                    coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------------------------------
Intercept                                        62.8588      1.309     48.007      0.000      60.291      65.427
Continent[T.Asia]                                18.2390      1.133     16.104      0.000      16.018      20.460
Continent[T.Europe]                              27.0258      1.732     15.604      0.000      23.629      30.422
Adult_Mortality                                   0.0178      0.002      7.181      0.000       0.013       0.023
Adult_Mortality:Continent[T.Asia]                -0.0351      0.002    -15.261      0.000      -0.040      -0.031
Adult_Mortality:Continent[T.Europe]              -0.0235      0.003     -7.822      0.000      -0.029      -0.018
BMI                                              -0.0075      0.027     -0.275      0.784      -0.061       0.046
BMI:Continent[T.Asia]                            -0.1750      0.014    -12.784      0.000      -0.202      -0.148
BMI:Continent[T.Europe]                          -0.1411      0.018     -7.660      0.000      -0.177      -0.105
Polio                                            -0.0706      0.023     -3.135      0.002      -0.115      -0.026
Polio:Continent[T.Asia]                          -0.0151      0.010     -1.461      0.144      -0.035       0.005
Polio:Continent[T.Europe]                        -0.0480      0.017     -2.888      0.004      -0.081      -0.015
Total_expenditure                                -0.5712      0.123     -4.662      0.000      -0.811      -0.331
Total_expenditure:Continent[T.Asia]               0.5971      0.089      6.678      0.000       0.422       0.772
Total_expenditure:Continent[T.Europe]             0.4176      0.111      3.746      0.000       0.199       0.636
Diphtheria                                       -0.0922      0.023     -4.064      0.000      -0.137      -0.048
Diphtheria:Continent[T.Asia]                     -0.0336      0.011     -3.076      0.002      -0.055      -0.012
Diphtheria:Continent[T.Europe]                   -0.0567      0.014     -4.132      0.000      -0.084      -0.030
HIV                                               3.2209      0.113     28.506      0.000       2.999       3.442
HIV:Continent[T.Asia]                            -1.4577      0.692     -2.107      0.035      -2.814      -0.101
HIV:Continent[T.Europe]                          -3.4665      1.703     -2.035      0.042      -6.806      -0.127
GDP                                               0.0004   5.02e-05      8.033      0.000       0.000       0.001
GDP:Continent[T.Asia]                            -0.0003   5.21e-05     -6.058      0.000      -0.000      -0.000
GDP:Continent[T.Europe]                          -0.0003   5.21e-05     -6.579      0.000      -0.000      -0.000
thinness_1_19_years                               0.0855      0.045      1.901      0.057      -0.003       0.174
thinness_1_19_years:Continent[T.Asia]            -0.1867      0.051     -3.658      0.000      -0.287      -0.087
thinness_1_19_years:Continent[T.Europe]          -2.5461      0.222    -11.477      0.000      -2.981      -2.111
Status_Developing                                -6.3060      1.040     -6.066      0.000      -8.345      -4.267
I(Adult_Mortality ** 2)                      -5.348e-05   4.74e-06    -11.284      0.000   -6.28e-05   -4.42e-05
I(BMI ** 2)                                       0.0020      0.000      6.581      0.000       0.001       0.003
I(Polio ** 2)                                     0.0010      0.000      3.993      0.000       0.001       0.001
I(Diphtheria ** 2)                                0.0013      0.000      5.332      0.000       0.001       0.002
I(HIV ** 2)                                       0.0070      0.001      6.374      0.000       0.005       0.009
I(GDP ** 2)                                  -3.952e-10   1.95e-10     -2.023      0.043   -7.78e-10   -1.21e-11
HIV:Status_Developing                            -3.6956      0.115    -32.088      0.000      -3.921      -3.470
BMI:Status_Developing                             0.0542      0.017      3.276      0.001       0.022       0.087
Total_expenditure:Status_Developing               0.3450      0.107      3.229      0.001       0.136       0.555
==============================================================================
Omnibus:                       71.491   Durbin-Watson:                   0.778
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              161.923
Skew:                           0.156   Prob(JB):                     6.90e-36
Kurtosis:                       4.281   Cond. No.                     2.26e+15
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.26e+15. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [78]:
```
# Testing Final VIF passed model transformed for non-linearity

print(linear_reset(VIF_passed_interaction_reg2, power=2, test_type='fitted', use_f= True))
print(linear_reset(VIF_passed_interaction_reg2, power=3, test_type='fitted', use_f= True))
```

```
<F test: F=77.2632152956621, p=2.9535835318619504e-18, df_denom=2.2e+03, df_num=1>
<F test: F=73.15075579223071, p=1.750601144516229e-31, df_denom=2.2e+03, df_num=2>
```

After obtaining our final model with squared and interaction terms, we ran the RESET test to check for mis-specification. However the p values are still very low and much lower than 0.05. Thus, we will now attempt to transform our variables to a cubed form and test this model for mis-specification.

In [79]:
```
VIF_passed_interaction_reg3 = smf.ols(
    'Life_expectancy ~ Adult_Mortality  + BMI  + Polio + \
    Total_expenditure + Diphtheria + HIV + GDP + thinness_1_19_years + \
    Status_Developing + Continent + I(Adult_Mortality**2) + I(BMI**2) + I(Polio**2) + \
```

```
    I(Diphtheria**2) + I(HIV**2) + I(GDP**2) + \
    HIV*Continent + HIV*Status_Developing + \
    Adult_Mortality*Continent + \
    BMI*Continent + BMI*Status_Developing + Total_expenditure*Continent + Total_expenditure*Status_Developing +
    Polio*Continent + Diphtheria*Continent + thinness_1_19_years*Continent + \
    GDP*Continent + I(Adult_Mortality**3) + I(BMI**3) + I(Polio**3) + I(Diphtheria**3) + I(HIV**3) + I(GDP**3)
data=raw_data_knn_test_imputed
).fit()

print(VIF_passed_interaction_reg3.summary())
```

OLS Regression Results

Dep. Variable:	Life_expectancy	R-squared:	0.780
Model:	OLS	Adj. R-squared:	0.777
Method:	Least Squares	F-statistic:	244.2
Date:	Wed, 20 Nov 2024	Prob (F-statistic):	0.00
Time:	14:19:13	Log-Likelihood:	-6688.9
No. Observations:	2237	AIC:	1.344e+04
Df Residuals:	2204	BIC:	1.363e+04
Df Model:	32		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	22.4301	0.419	53.479	0.000	21.608	23.253
Continent[T.Asia]	11.1301	0.252	44.110	0.000	10.635	11.625
Continent[T.Europe]	5.4416	0.111	49.101	0.000	5.224	5.659
Adult_Mortality	0.0881	0.007	13.523	0.000	0.075	0.101
Adult_Mortality:Continent[T.Asia]	-0.0433	0.003	-14.511	0.000	-0.049	-0.037
Adult_Mortality:Continent[T.Europe]	-0.0380	0.004	-8.745	0.000	-0.047	-0.029
BMI	0.4637	0.063	7.417	0.000	0.341	0.586
BMI:Continent[T.Asia]	-0.1589	0.018	-8.724	0.000	-0.195	-0.123
BMI:Continent[T.Europe]	-0.2813	0.023	-12.317	0.000	-0.326	-0.236
Polio	0.2532	0.101	2.509	0.012	0.055	0.451
Polio:Continent[T.Asia]	0.0097	0.014	0.682	0.495	-0.018	0.038
Polio:Continent[T.Europe]	0.1971	0.020	9.642	0.000	0.157	0.237
Total_expenditure	1.4769	0.146	10.125	0.000	1.191	1.763
Total_expenditure:Continent[T.Asia]	0.7616	0.119	6.419	0.000	0.529	0.994
Total_expenditure:Continent[T.Europe]	-0.3909	0.137	-2.848	0.004	-0.660	-0.122
Diphtheria	-0.0521	0.097	-0.537	0.592	-0.243	0.138
Diphtheria:Continent[T.Asia]	0.0024	0.014	0.167	0.868	-0.026	0.031
Diphtheria:Continent[T.Europe]	0.0354	0.019	1.846	0.065	-0.002	0.073
HIV	-1.8359	0.067	-27.286	0.000	-1.968	-1.704
HIV:Continent[T.Asia]	-2.1919	0.946	-2.317	0.021	-4.047	-0.336
HIV:Continent[T.Europe]	-3.7223	0.070	-52.986	0.000	-3.860	-3.584
GDP	0.0005	7.3e-05	6.574	0.000	0.000	0.001
GDP:Continent[T.Asia]	-7.236e-05	7.73e-05	-0.937	0.349	-0.000	7.91e-05
GDP:Continent[T.Europe]	-0.0001	7.67e-05	-1.854	0.064	-0.000	8.2e-06
thinness_1_19_years	-0.0371	0.059	-0.627	0.530	-0.153	0.079
thinness_1_19_years:Continent[T.Asia]	-0.0865	0.065	-1.339	0.181	-0.213	0.040
thinness_1_19_years:Continent[T.Europe]	0.0283	0.299	0.095	0.924	-0.557	0.614
Status_Developing	28.7305	0.540	53.241	0.000	27.672	29.789
I(Adult_Mortality ** 2)	-0.0004	2.6e-05	-13.835	0.000	-0.000	-0.000
I(BMI ** 2)	0.0003	0.002	0.128	0.898	-0.004	0.005
I(Polio ** 2)	-0.0047	0.002	-2.039	0.042	-0.009	-0.000
I(Diphtheria ** 2)	0.0011	0.002	0.519	0.604	-0.003	0.005
I(HIV ** 2)	0.0366	0.008	4.773	0.000	0.022	0.052
I(GDP ** 2)	-6.343e-09	1.15e-09	-5.536	0.000	-8.59e-09	-4.1e-09
HIV:Status_Developing	1.0593	0.060	17.533	0.000	0.941	1.178
BMI:Status_Developing	-0.3205	0.018	-17.728	0.000	-0.356	-0.285
Total_expenditure:Status_Developing	-1.7400	0.125	-13.926	0.000	-1.985	-1.495
I(Adult_Mortality ** 3)	3.218e-07	2.84e-08	11.348	0.000	2.66e-07	3.77e-07
I(BMI ** 3)	-7.391e-06	2.13e-05	-0.347	0.728	-4.91e-05	3.43e-05
I(Polio ** 3)	2.565e-05	1.44e-05	1.782	0.075	-2.58e-06	5.39e-05
I(Diphtheria ** 3)	-1.139e-06	1.38e-05	-0.082	0.934	-2.83e-05	2.6e-05
I(HIV ** 3)	-0.0005	0.000	-4.539	0.000	-0.001	-0.000
I(GDP ** 3)	3.399e-14	8.05e-15	4.220	0.000	1.82e-14	4.98e-14

Omnibus:	674.516	Durbin-Watson:	0.934
Prob(Omnibus):	0.000	Jarque-Bera (JB):	3111.083
Skew:	1.372	Prob(JB):	0.00
Kurtosis:	8.084	Cond. No.	8.03e+16

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 8.03e+16. This might indicate that there are
strong multicollinearity or other numerical problems.

In [80]:
```
# Testing VIF passed model 3 transformed

print(linear_reset(VIF_passed_interaction_reg3, power=2, test_type='fitted', use_f= True))
print(linear_reset(VIF_passed_interaction_reg3, power=3, test_type='fitted', use_f= True))
```

```
<F test: F=549.0091186179081, p=1.367005577035467e-108, df_denom=2.2e+03, df_num=1>
<F test: F=2453.2255658352624, p=0.0, df_denom=2.2e+03, df_num=2>
```

In the cubed model, as well, using the RAMSEY RESET Test, we can see that the p-value is still << 0.05. Thus, this indicates that our model still requires further transformation

Based on all the models, we have considered so far, all the models show statistical significant results at the 5% level with the RAMSEY test. However, the Box-Cox and Yeo Johnson transformed model shows the best p-value stat for the RAMSEY Test. Even though, among the models, considered, statistically it may be the best model, in terms of economic interpretability, this may not be a good model.

Thus, to navigate the tradeoff between statistical validity and economic interpretability, we will choose the model with quadratic terms and interaction terms combined as it does show a squared transformation, accounts for interaction terms which are significant and is easy to interpret.

We will now take this model forward and examine heteroskedasticity

Test For Heteroskedasticity

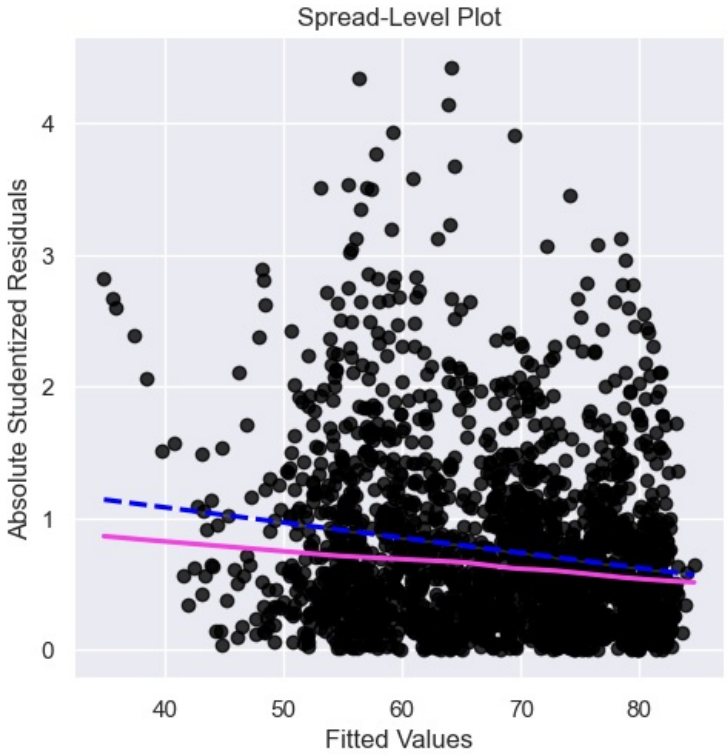
To check for heteroskedasticity, we first look at a basic spread level plot of residuals vs fitted values to visually check for heteroskedasticity. We also look at what suggest power transform may be needed to account for heteroskedasticity.

Next, we look at the formal Breusch Pagan test to test for heteroskedasticity

1. Detecting Heteroskedasticity

a) Spread Level Plots

In [81]:
```
# Extract studentized residuals
VIF_passed_interaction_reg2_std_resid = VIF_passed_interaction_reg2.get_influence().resid_studentized_external #
VIF_passed_interaction_reg2_spreadlevel = pd.DataFrame({
    'Fitted Values':VIF_passed_interaction_reg2.fittedvalues,
    'Absolute Studentized Residuals':np.abs(VIF_passed_interaction_reg2_std_resid)
})

# Plot spread-level plot
sns.lmplot(VIF_passed_interaction_reg2_spreadlevel, x='Fitted Values', y='Absolute Studentized Residuals',
           line_kws={'color':'blue', 'linestyle':'--'}, ci=False, scatter_kws={'color':'black'})
sns.regplot(VIF_passed_interaction_reg2_spreadlevel, x='Fitted Values', y='Absolute Studentized Residuals',
            color='#f542e3', lowess=True, scatter=False)
plt.title('Spread-Level Plot')
plt.show()

# Power transform
## Formula: lambda = 1 - slope(RLM), where RLM is Robust Linear Model of (log(abs studentized residuals) ~ log(
VIF_passed_interaction_reg2_rlm = RLM(endog=np.log(np.abs(VIF_passed_interaction_reg2_std_resid)),            #
                   exog=sm.add_constant(np.log(VIF_passed_interaction_reg2.fittedvalues))  # x = log fitted valu
                   ).fit()

print(f'Suggested power transform: {1 - VIF_passed_interaction_reg2_rlm.params[0]:.4f}')
# Index is 0 because x was unlabeled; the other (0-place) estimate has index 'const'
```




```
Suggested power transform: 1.9421
```

The spread-level plot visualizes the absolute studentized residuals against the fitted values, which helps assess heteroskedasticity in the regression model. Ideally, the residuals should exhibit a random scatter without a discernible pattern, indicating constant variance (homoskedasticity). However, in this plot, the residuals show a slight downward trend, as indicated by the pink fitted line. This suggests the presence of mild heteroskedasticity, where variance decreases as the fitted values increase.

The suggested power transform (1.9421) implies that applying a transformation to the dependent variable, such as raising it to this power or using a similar adjustment, could help stabilize the variance and improve model fit. Addressing heteroskedasticity is crucial for ensuring the reliability of standard errors and hypothesis tests in the regression analysis.

However, applying the power transform may reduce interpretability of the model. We will thus, look at other methods.

b) Breusch Pagan Test

We now conduct the Breusch Pagan test to check for heteroskedasticity

In [82]:
```
VIF_passed_interaction_reg2_bp = het_breuschpagan(resid=VIF_passed_interaction_reg2.resid, exog_het = VIF_passe
print(f'Test statistic: \t{VIF_passed_interaction_reg2_bp[0]:.5f}\np-value: \t\t{VIF_passed_interaction_reg2_bp
```

```
Test statistic:         224.74797
p-value:                0.00000
```

Since the p-value is well below the common significance level (e.g., 0.05), we reject the null hypothesis of homoskedasticity. This indicates strong evidence of heteroskedasticity in the model, meaning the variance of the residuals is not constant.

We will now look to 2 methods to correct for this issue:

1. Using White Errors robust to heteroskedasticity
2. FGLS Model

Correcting for Heteroskedasticity

In the previous section, we concluded that our model has heteroskedasticity. In this section, we look for 2 ways to correct for it and then compare the 2 models:

1. Using White Errors robust to heteroskedasticity
2. FGLS Model

1. HC0 Errors

In [84]:
```
model_3_with_int_het_adj = smf.ols(
    'Life_expectancy ~ Adult_Mortality  + BMI  + Polio + \
    Total_expenditure + Diphtheria + HIV + GDP + thinness_1_19_years + \
    Status_Developing + Continent + I(Adult_Mortality**2) + I(BMI**2) + I(Polio**2) + \
    I(Diphtheria**2) + I(HIV**2) + I(GDP**2) + \
    HIV*Continent + HIV*Status_Developing + \
```

```
    Adult_Mortality*Continent + \
    BMI*Continent + BMI*Status_Developing + Total_expenditure*Continent + Total_expenditure*Status_Developing +
    Polio*Continent + Diphtheria*Continent + thinness_1_19_years*Continent + \
    GDP*Continent',
data=raw_data_knn_test_imputed
).fit(cov_type = 'HC0')

print(model_3_with_int_het_adj.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Life_expectancy   R-squared:                       0.888
Model:                            OLS   Adj. R-squared:                  0.886
Method:                 Least Squares   F-statistic:                 3.048e+04
Date:                Wed, 20 Nov 2024   Prob (F-statistic):               0.00
Time:                        14:19:26   Log-Likelihood:                -5932.5
No. Observations:                2237   AIC:                         1.194e+04
Df Residuals:                    2201   BIC:                         1.214e+04
Df Model:                          35
Covariance Type:                  HC0
==============================================================================
```

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	62.8588	1.510	41.636	0.000	59.900	65.818
Continent[T.Asia]	18.2390	1.255	14.531	0.000	15.779	20.699
Continent[T.Europe]	27.0258	1.814	14.898	0.000	23.470	30.581
Adult_Mortality	0.0178	0.003	5.166	0.000	0.011	0.025
Adult_Mortality:Continent[T.Asia]	-0.0351	0.003	-13.240	0.000	-0.040	-0.030
Adult_Mortality:Continent[T.Europe]	-0.0235	0.003	-7.388	0.000	-0.030	-0.017
BMI	-0.0075	0.029	-0.254	0.800	-0.065	0.050
BMI:Continent[T.Asia]	-0.1750	0.015	-11.544	0.000	-0.205	-0.145
BMI:Continent[T.Europe]	-0.1411	0.019	-7.247	0.000	-0.179	-0.103
Polio	-0.0706	0.025	-2.864	0.004	-0.119	-0.022
Polio:Continent[T.Asia]	-0.0151	0.011	-1.385	0.166	-0.036	0.006
Polio:Continent[T.Europe]	-0.0480	0.015	-3.287	0.001	-0.077	-0.019
Total_expenditure	-0.5712	0.136	-4.190	0.000	-0.838	-0.304
Total_expenditure:Continent[T.Asia]	0.5971	0.103	5.820	0.000	0.396	0.798
Total_expenditure:Continent[T.Europe]	0.4176	0.129	3.246	0.001	0.165	0.670
Diphtheria	-0.0922	0.023	-4.078	0.000	-0.137	-0.048
Diphtheria:Continent[T.Asia]	-0.0336	0.011	-3.198	0.001	-0.054	-0.013
Diphtheria:Continent[T.Europe]	-0.0567	0.011	-5.158	0.000	-0.078	-0.035
HIV	3.2209	0.121	26.639	0.000	2.984	3.458
HIV:Continent[T.Asia]	-1.4577	0.470	-3.098	0.002	-2.380	-0.536
HIV:Continent[T.Europe]	-3.4665	0.994	-3.487	0.000	-5.415	-1.518
GDP	0.0004	6.43e-05	6.281	0.000	0.000	0.001
GDP:Continent[T.Asia]	-0.0003	6.57e-05	-4.800	0.000	-0.000	-0.000
GDP:Continent[T.Europe]	-0.0003	6.56e-05	-5.223	0.000	-0.000	-0.000
thinness_1_19_years	0.0855	0.058	1.463	0.143	-0.029	0.200
thinness_1_19_years:Continent[T.Asia]	-0.1867	0.062	-3.001	0.003	-0.309	-0.065
thinness_1_19_years:Continent[T.Europe]	-2.5461	0.253	-10.048	0.000	-3.043	-2.049
Status_Developing	-6.3060	1.050	-6.008	0.000	-8.363	-4.249
I(Adult_Mortality ** 2)	-5.348e-05	7.25e-06	-7.374	0.000	-6.77e-05	-3.93e-05
I(BMI ** 2)	0.0020	0.000	6.746	0.000	0.001	0.003
I(Polio ** 2)	0.0010	0.000	3.816	0.000	0.000	0.001
I(Diphtheria ** 2)	0.0013	0.000	5.411	0.000	0.001	0.002
I(HIV ** 2)	0.0070	0.002	3.869	0.000	0.003	0.011
I(GDP ** 2)	-3.952e-10	1.72e-10	-2.299	0.021	-7.32e-10	-5.83e-11
HIV:Status_Developing	-3.6956	0.126	-29.351	0.000	-3.942	-3.449
BMI:Status_Developing	0.0542	0.015	3.520	0.000	0.024	0.084
Total_expenditure:Status_Developing	0.3450	0.106	3.240	0.001	0.136	0.554

```
==============================================================================
Omnibus:                       71.491   Durbin-Watson:                   0.778
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              161.923
Skew:                           0.156   Prob(JB):                     6.90e-36
Kurtosis:                       4.281   Cond. No.                     2.26e+15
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC0)
[2] The condition number is large, 2.26e+15. This might indicate that there are
strong multicollinearity or other numerical problems.
```

2. FLGS Model

In [85]:
```
# Extract residuals from the model

residuals = model_3_with_int_het_adj.resid
raw_data_het = raw_data_knn_test_imputed.copy()

# Calculate squared residuals
residuals_squared = residuals ** 2

raw_data_het['residuals_squared'] = residuals_squared
```

```
model_log_resid_squared = smf.ols(
    'residuals_squared ~ Adult_Mortality  + BMI  + Polio + \
    Total_expenditure + Diphtheria + HIV + GDP + thinness_1_19_years + \
    Status_Developing + Continent + I(Adult_Mortality**2) + I(BMI**2) + I(Polio**2) + \
    I(Diphtheria**2) + I(HIV**2) + I(GDP**2) + \
    HIV*Continent + HIV*Status_Developing + \
    Adult_Mortality*Continent + \
    BMI*Continent + BMI*Status_Developing + Total_expenditure*Continent + Total_expenditure*Status_Developing +
    Polio*Continent + Diphtheria*Continent + thinness_1_19_years*Continent + \
    GDP*Continent',
data=raw_data_het
).fit()

print(model_log_resid_squared.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:      residuals_squared   R-squared:                       0.100
Model:                            OLS   Adj. R-squared:                  0.086
Method:                 Least Squares   F-statistic:                     7.024
Date:                Wed, 20 Nov 2024   Prob (F-statistic):           2.00e-31
Time:                        14:19:30   Log-Likelihood:                -9901.2
No. Observations:                2237   AIC:                         1.987e+04
Df Residuals:                    2201   BIC:                         2.008e+04
Df Model:                          35
Covariance Type:            nonrobust
==============================================================================================================
                                                 coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------------------------------
Intercept                                    28.0863      7.719      3.639      0.000      12.949      43.224
Continent[T.Asia]                           -10.7466      6.677     -1.610      0.108     -23.840       2.346
Continent[T.Europe]                         -14.2219     10.210     -1.393      0.164     -34.245       5.801
Adult_Mortality                              -0.0696      0.015     -4.752      0.000      -0.098      -0.041
Adult_Mortality:Continent[T.Asia]            -0.0067      0.014     -0.494      0.621      -0.033       0.020
Adult_Mortality:Continent[T.Europe]           0.0234      0.018      1.325      0.185      -0.011       0.058
BMI                                          -0.2465      0.160     -1.541      0.123      -0.560       0.067
BMI:Continent[T.Asia]                         0.3143      0.081      3.894      0.000       0.156       0.473
BMI:Continent[T.Europe]                       0.3001      0.109      2.764      0.006       0.087       0.513
Polio                                        -0.2511      0.133     -1.890      0.059      -0.512       0.009
Polio:Continent[T.Asia]                      -0.0405      0.061     -0.667      0.505      -0.160       0.079
Polio:Continent[T.Europe]                    -0.0434      0.098     -0.443      0.658      -0.236       0.149
Total_expenditure                             1.7853      0.722      2.472      0.014       0.369       3.202
Total_expenditure:Continent[T.Asia]          -0.3938      0.527     -0.747      0.455      -1.427       0.640
Total_expenditure:Continent[T.Europe]        -1.7798      0.657     -2.708      0.007      -3.069      -0.491
Diphtheria                                    0.1978      0.134      1.479      0.139      -0.065       0.460
Diphtheria:Continent[T.Asia]                 -0.0085      0.064     -0.132      0.895      -0.135       0.118
Diphtheria:Continent[T.Europe]                0.0683      0.081      0.844      0.399      -0.090       0.227
HIV                                           0.4663      0.666      0.700      0.484      -0.840       1.773
HIV:Continent[T.Asia]                        -1.2658      4.078     -0.310      0.756      -9.263       6.732
HIV:Continent[T.Europe]                      -2.0938     10.040     -0.209      0.835     -21.783      17.595
GDP                                           0.0006      0.000      2.154      0.031    5.72e-05       0.001
GDP:Continent[T.Asia]                        -0.0008      0.000     -2.496      0.013      -0.001      -0.000
GDP:Continent[T.Europe]                      -0.0007      0.000     -2.296      0.022      -0.001      -0.000
thinness_1_19_years                          -0.2405      0.265     -0.907      0.364      -0.760       0.279
thinness_1_19_years:Continent[T.Asia]         0.1115      0.301      0.370      0.711      -0.479       0.702
thinness_1_19_years:Continent[T.Europe]       1.3279      1.308      1.015      0.310      -1.237       3.893
Status_Developing                             9.3996      6.129      1.534      0.125      -2.619      21.418
I(Adult_Mortality ** 2)                  8.137e-05   2.79e-05      2.912      0.004    2.66e-05       0.000
I(BMI ** 2)                                  -0.0013      0.002     -0.735      0.462      -0.005       0.002
I(Polio ** 2)                                 0.0024      0.001      1.659      0.097      -0.000       0.005
I(Diphtheria ** 2)                           -0.0023      0.001     -1.586      0.113      -0.005       0.001
I(HIV ** 2)                                   0.0337      0.006      5.231      0.000       0.021       0.046
I(GDP ** 2)                              7.752e-10   1.15e-09      0.673      0.501   -1.48e-09    3.03e-09
HIV:Status_Developing                        -1.4024      0.679     -2.066      0.039      -2.734      -0.071
BMI:Status_Developing                        -0.0516      0.098     -0.529      0.597      -0.243       0.140
Total_expenditure:Status_Developing          -1.0359      0.630     -1.645      0.100      -2.271       0.199
==============================================================================
Omnibus:                     1821.515   Durbin-Watson:                   1.223
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            49295.468
Skew:                           3.718   Prob(JB):                         0.00
Kurtosis:                      24.762   Cond. No.                     2.26e+15
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.26e+15. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [86]:
```
# Calculate predicted log variance
predicted_log_variance = model_log_resid_squared.fittedvalues

# Convert log variance to variance
predicted_variance = np.exp(predicted_log_variance)
```

```
# Calculate weights as inverse of variance
weights = 1/predicted_variance

model_4_het_adj_flgp = smf.wls(
    '(Life_expectancy) ~ Adult_Mortality  + BMI  + Polio + \
    Total_expenditure + Diphtheria + HIV + GDP + thinness_1_19_years + \
    Status_Developing + Continent + I(Adult_Mortality**2) + I(BMI**2) + I(Polio**2) + \
    I(Diphtheria**2) + I(HIV**2) + I(GDP**2) + \
    HIV*Continent + HIV*Status_Developing + \
    Adult_Mortality*Continent + \
    BMI*Continent + BMI*Status_Developing + Total_expenditure*Continent + Total_expenditure*Status_Developing +
    Polio*Continent + Diphtheria*Continent + thinness_1_19_years*Continent + \
    GDP*Continent',
data=raw_data_knn_test_imputed, weights = weights
).fit()

print(model_4_het_adj_flgp.summary())
```

```
                            WLS Regression Results
==============================================================================
Dep. Variable:        Life_expectancy   R-squared:                       0.965
Model:                            WLS   Adj. R-squared:                  0.965
Method:                 Least Squares   F-statistic:                     1752.
Date:                Wed, 20 Nov 2024   Prob (F-statistic):               0.00
Time:                        14:19:37   Log-Likelihood:                -11662.
No. Observations:                2237   AIC:                         2.340e+04
Df Residuals:                    2201   BIC:                         2.360e+04
Df Model:                          35
Covariance Type:            nonrobust
================================================================================================================
                                                   coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------------------------------
Intercept                                       84.5975      4.316     19.599      0.000      76.133      93.062
Continent[T.Asia]                                2.3643      4.041      0.585      0.559      -5.561      10.290
Continent[T.Europe]                             15.5825      4.349      3.583      0.000       7.053      24.112
Adult_Mortality                                 -0.0255      0.012     -2.207      0.027      -0.048      -0.003
Adult_Mortality:Continent[T.Asia]               -0.0030      0.006     -0.461      0.645      -0.016       0.010
Adult_Mortality:Continent[T.Europe]             -0.0281      0.008     -3.376      0.001      -0.044      -0.012
BMI                                              0.0196      0.054      0.362      0.717      -0.086       0.125
BMI:Continent[T.Asia]                           -0.1315      0.048     -2.740      0.006      -0.226      -0.037
BMI:Continent[T.Europe]                         -0.1135      0.049     -2.329      0.020      -0.209      -0.018
Polio                                           -0.1668      0.041     -4.090      0.000      -0.247      -0.087
Polio:Continent[T.Asia]                         -0.0404      0.022     -1.814      0.070      -0.084       0.003
Polio:Continent[T.Europe]                       -0.0515      0.023     -2.261      0.024      -0.096      -0.007
Total_expenditure                               -0.3946      0.176     -2.239      0.025      -0.740      -0.049
Total_expenditure:Continent[T.Asia]              0.2867      0.146      1.962      0.050       0.000       0.573
Total_expenditure:Continent[T.Europe]           -0.0876      0.152     -0.576      0.565      -0.386       0.211
Diphtheria                                      -0.0581      0.029     -2.020      0.043      -0.114      -0.002
Diphtheria:Continent[T.Asia]                     0.0461      0.024      1.946      0.052      -0.000       0.092
Diphtheria:Continent[T.Europe]                   0.0234      0.023      1.006      0.315      -0.022       0.069
HIV                                              5.2594      0.267     19.734      0.000       4.737       5.782
HIV:Continent[T.Asia]                           -0.9055      0.302     -3.002      0.003      -1.497      -0.314
HIV:Continent[T.Europe]                         -1.6338      0.837     -1.952      0.051      -3.275       0.008
GDP                                          -3.134e-05      0.000     -0.230      0.818      -0.000       0.000
GDP:Continent[T.Asia]                         5.706e-05      0.000      0.411      0.681      -0.000       0.000
GDP:Continent[T.Europe]                       5.123e-05      0.000      0.374      0.709      -0.000       0.000
thinness_1_19_years                             -0.4257      0.208     -2.049      0.041      -0.833      -0.018
thinness_1_19_years:Continent[T.Asia]            0.5125      0.208      2.463      0.014       0.104       0.921
thinness_1_19_years:Continent[T.Europe]          0.5156      0.317      1.629      0.104      -0.105       1.136
Status_Developing                              -14.6582      0.905    -16.195      0.000     -16.433     -12.883
I(Adult_Mortality ** 2)                      -3.825e-05   1.34e-05     -2.847      0.004   -6.46e-05   -1.19e-05
I(BMI ** 2)                                     -0.0006      0.000     -2.091      0.037      -0.001   -3.55e-05
I(Polio ** 2)                                    0.0016      0.000      4.147      0.000       0.001       0.002
I(Diphtheria ** 2)                               0.0004      0.000      2.014      0.044    1.14e-05       0.001
I(HIV ** 2)                                     -0.0161      0.011     -1.525      0.127      -0.037       0.005
I(GDP ** 2)                                   2.002e-10   1.34e-10      1.490      0.136   -6.33e-11    4.64e-10
HIV:Status_Developing                           -4.6662      0.250    -18.676      0.000      -5.156      -4.176
BMI:Status_Developing                            0.2179      0.019     11.548      0.000       0.181       0.255
Total_expenditure:Status_Developing              0.4672      0.099      4.719      0.000       0.273       0.661
==============================================================================
Omnibus:                     1455.561   Durbin-Watson:                   1.094
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           165513.744
Skew:                           2.167   Prob(JB):                         0.00
Kurtosis:                      44.916   Cond. No.                     2.26e+15
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 5.41e-12. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Now that we have computed both heteroskedasticity adjusted models using HC0 errors and FLGS, we will compare both models based on performance to determine the best model based on R Squared, AIC and BIC

In [87]:
```
# Comparing Model Summaries

model_compare = summary_col([VIF_passed_interaction_reg2, model_3_with_int_het_adj, model_4_het_adj_flgp],
                            float_format='%0.3f',
                            stars = False,
                            model_names=['OLS', 'HC0', 'FGLS'])

print(model_compare)
```

```
================================================================
                                        OLS     HC0     FGLS
----------------------------------------------------------------
Intercept                              62.859  62.859  84.597
                                       (1.309) (1.510) (4.316)
Continent[T.Asia]                      18.239  18.239  2.364
                                       (1.133) (1.255) (4.041)
Continent[T.Europe]                    27.026  27.026  15.582
                                       (1.732) (1.814) (4.349)
Adult_Mortality                        0.018   0.018   -0.025
                                       (0.002) (0.003) (0.012)
Adult_Mortality:Continent[T.Asia]      -0.035  -0.035  -0.003
                                       (0.002) (0.003) (0.006)
Adult_Mortality:Continent[T.Europe]    -0.023  -0.023  -0.028
                                       (0.003) (0.003) (0.008)
BMI                                    -0.007  -0.007  0.020
                                       (0.027) (0.029) (0.054)
BMI:Continent[T.Asia]                  -0.175  -0.175  -0.132
                                       (0.014) (0.015) (0.048)
BMI:Continent[T.Europe]                -0.141  -0.141  -0.114
                                       (0.018) (0.019) (0.049)
Polio                                  -0.071  -0.071  -0.167
                                       (0.023) (0.025) (0.041)
Polio:Continent[T.Asia]                -0.015  -0.015  -0.040
                                       (0.010) (0.011) (0.022)
Polio:Continent[T.Europe]              -0.048  -0.048  -0.051
                                       (0.017) (0.015) (0.023)
Total_expenditure                      -0.571  -0.571  -0.395
                                       (0.123) (0.136) (0.176)
Total_expenditure:Continent[T.Asia]    0.597   0.597   0.287
                                       (0.089) (0.103) (0.146)
Total_expenditure:Continent[T.Europe]  0.418   0.418   -0.088
                                       (0.111) (0.129) (0.152)
Diphtheria                             -0.092  -0.092  -0.058
                                       (0.023) (0.023) (0.029)
Diphtheria:Continent[T.Asia]           -0.034  -0.034  0.046
                                       (0.011) (0.011) (0.024)
Diphtheria:Continent[T.Europe]         -0.057  -0.057  0.023
                                       (0.014) (0.011) (0.023)
HIV                                    3.221   3.221   5.259
                                       (0.113) (0.121) (0.267)
HIV:Continent[T.Asia]                  -1.458  -1.458  -0.906
                                       (0.692) (0.470) (0.302)
HIV:Continent[T.Europe]                -3.466  -3.466  -1.634
                                       (1.703) (0.994) (0.837)
GDP                                    0.000   0.000   -0.000
                                       (0.000) (0.000) (0.000)
GDP:Continent[T.Asia]                  -0.000  -0.000  0.000
                                       (0.000) (0.000) (0.000)
GDP:Continent[T.Europe]                -0.000  -0.000  0.000
                                       (0.000) (0.000) (0.000)
thinness_1_19_years                    0.085   0.085   -0.426
                                       (0.045) (0.058) (0.208)
thinness_1_19_years:Continent[T.Asia]  -0.187  -0.187  0.513
                                       (0.051) (0.062) (0.208)
thinness_1_19_years:Continent[T.Europe] -2.546 -2.546  0.516
                                       (0.222) (0.253) (0.317)
Status_Developing                      -6.306  -6.306  -14.658
                                       (1.040) (1.050) (0.905)
I(Adult_Mortality ** 2)                -0.000  -0.000  -0.000
                                       (0.000) (0.000) (0.000)
I(BMI ** 2)                            0.002   0.002   -0.001
                                       (0.000) (0.000) (0.000)
I(Polio ** 2)                          0.001   0.001   0.002
                                       (0.000) (0.000) (0.000)
I(Diphtheria ** 2)                     0.001   0.001   0.000
                                       (0.000) (0.000) (0.000)
I(HIV ** 2)                            0.007   0.007   -0.016
                                       (0.001) (0.002) (0.011)
I(GDP ** 2)                            -0.000  -0.000  0.000
                                       (0.000) (0.000) (0.000)
HIV:Status_Developing                  -3.696  -3.696  -4.666
                                       (0.115) (0.126) (0.250)
BMI:Status_Developing                  0.054   0.054   0.218
                                       (0.017) (0.015) (0.019)
Total_expenditure:Status_Developing    0.345   0.345   0.467
                                       (0.107) (0.106) (0.099)
R-squared                              0.888   0.888   0.965
R-squared Adj.                         0.886   0.886   0.965
================================================================
Standard errors in parentheses.
```

In [88]:
```
# Comparing Model AIC's
```

```
print(model_3_with_int_het_adj.aic)
print(model_4_het_adj_flgp.aic)
```

```
11936.966803259642
23395.929844728125
```

In [89]:
```
# Comparing Model BIC's
print(model_3_with_int_het_adj.bic)
print(model_4_het_adj_flgp.bic)
```

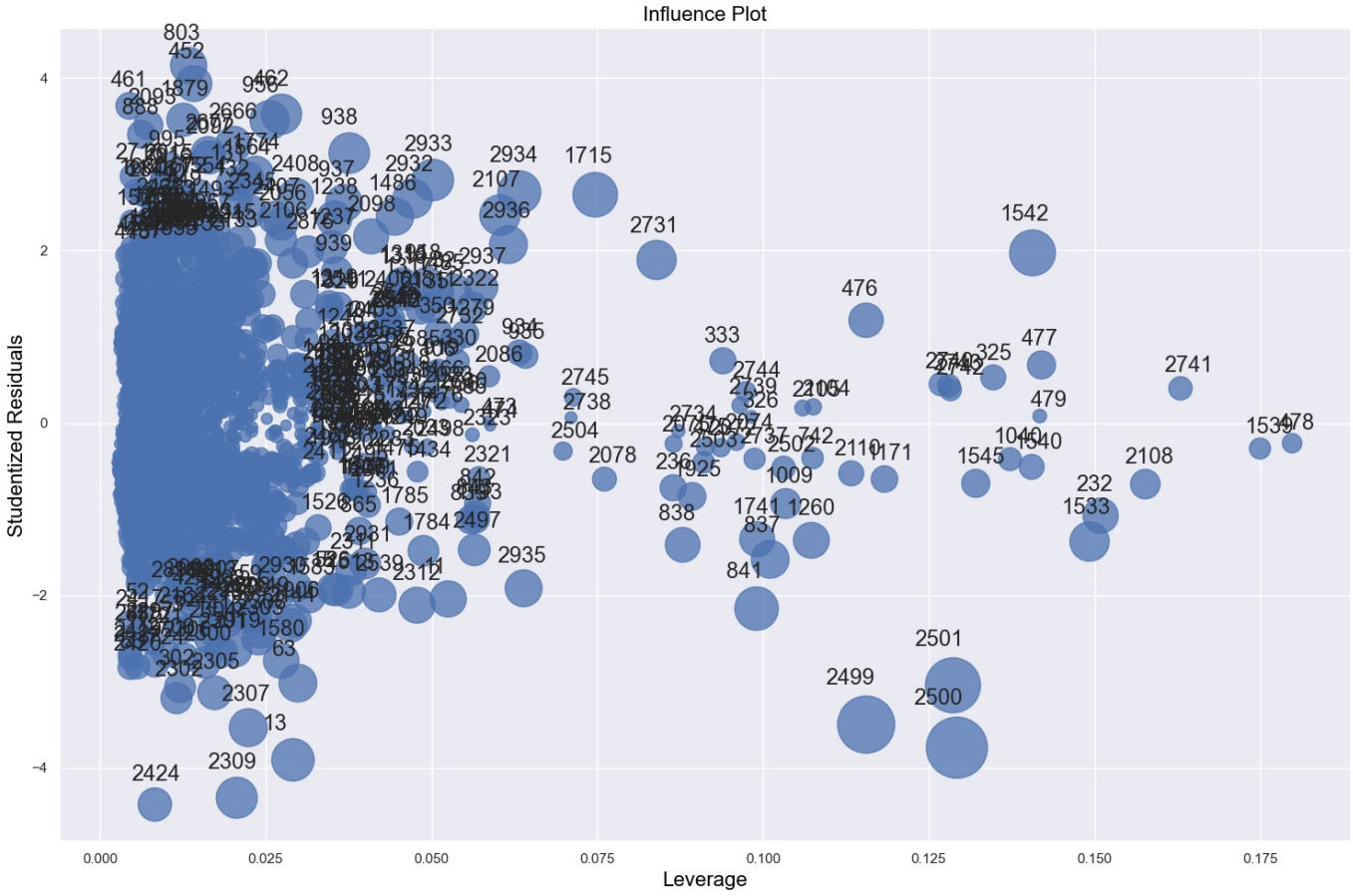
```
12142.630877873287
23601.59391934177
```

While R squared for HC0 is lower than the FGLS model, the Rsquared for HC0 is still 0.88 which is high enough. Additionally, the HC0 model has a lower AIC, BIC compared to the FGLS model. Thus, we will choose our HC0 model over the FGLS.

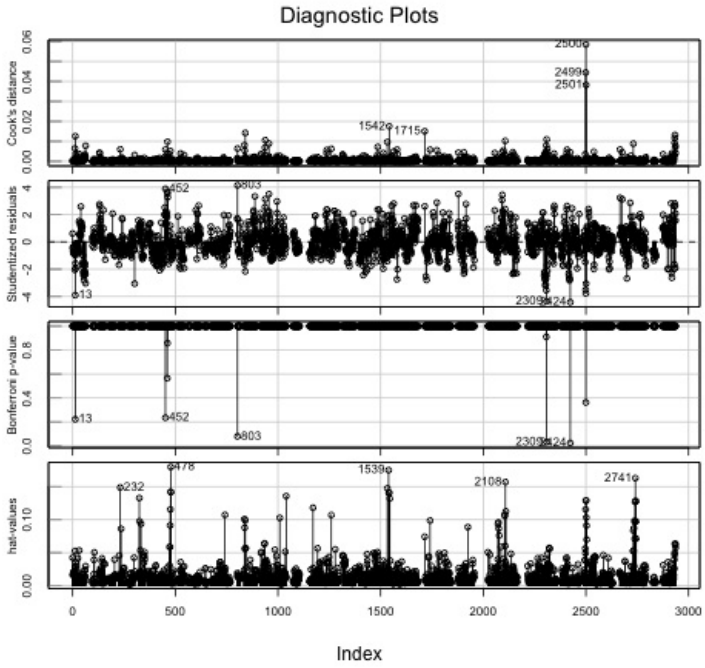
Add leverage and residuals and remove point

Here, we again, look at the leverage and residual plots to detemine influential points in our new model. This is to further fine tune our model.

In [90]:
```
fig, ax = plt.subplots(figsize=(15,10))
fig = sm.graphics.influence_plot(model_3_with_int_het_adj, ax = ax, criterion="DFFITS")
fig.tight_layout()
```



In [92]:
```
%%R
# Load necessary library
library(car)

# Fit the linear regression model
model <- lm(
  Life_expectancy ~ Adult_Mortality + BMI + Polio +
    Total_expenditure + Diphtheria + HIV + GDP + thinness_1_19_years +
    Status_Developing + Continent + I(Adult_Mortality^2) + I(BMI^2) + I(Polio^2) +
    I(Diphtheria^2) + I(HIV^2) + I(GDP^2) +
    HIV*Continent + HIV*Status_Developing +
    Adult_Mortality*Continent +
    BMI*Continent + BMI*Status_Developing +
    Total_expenditure*Continent + Total_expenditure*Status_Developing +
    Polio*Continent + Diphtheria*Continent + thinness_1_19_years*Continent +
    GDP*Continent,
  data = raw_data_knn_test_imputed
)

# Plot Cook's distance
influenceIndexPlot(model, id = list(n = 5))
```



In [93]:
```
# Of OLS residuals
model_3_with_int_het_adj_influence = model_3_with_int_het_adj.get_influence()
model_3_with_int_het_adj_std_resid = model_3_with_int_het_adj_influence.resid_studentized_internal # standardiz
sm.qqplot(model_3_with_int_het_adj_std_resid, line='s')
plt.title('QQ Plot of Standardized Residuals')
plt.show()
```

Leverage Plot-

The points 2424, 2309, 1542, 2741, 1539, 478, 2501 deviate significantly from the model's prediction. They have both high leverage and large residuals are influential apart from some others. These are typically large bubbles far from the origin. The dense cluster of points near the origin (small bubble sizes) are not influential and conform well to the regression model.

Thus, we will look to remove these 7 points to fine-tune our model

QQ Plot Of Standardized Residuals:

The points are following the straight line, which indicates that the residuals are normally distributed. This is a good sign as it means that the assumptions of the linear regression model are likely met. The points in the tail are a little bit off the straight line. This suggests some outliers in the data.

In [94]:
```
## Dropping Influential Points

raw_data_knn_test_imputed= raw_data_knn_test_imputed.drop([2424, 2309, 1542, 2741, 1539, 478, 2501])
```

Final Model

In [95]:
```
final_model_het_adj = smf.ols(
    'Life_expectancy ~ Adult_Mortality  + BMI  + Polio + \
    Total_expenditure + Diphtheria + HIV + GDP + thinness_1_19_years + \
    Status_Developing + Continent + I(Adult_Mortality**2) + I(BMI**2) + I(Polio**2) + \
    I(Diphtheria**2) + I(HIV**2) + I(GDP**2) + \
    HIV*Continent + HIV*Status_Developing + \
    Adult_Mortality*Continent + \
    BMI*Continent + BMI*Status_Developing + Total_expenditure*Continent + Total_expenditure*Status_Developing +
    Polio*Continent + Diphtheria*Continent + thinness_1_19_years*Continent + \
    GDP*Continent',
data=raw_data_knn_test_imputed
).fit(cov_type = 'HC0')

print(final_model_het_adj.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Life_expectancy   R-squared:                       0.890
Model:                            OLS   Adj. R-squared:                  0.888
Method:                 Least Squares   F-statistic:                 3.164e+04
Date:                Wed, 20 Nov 2024   Prob (F-statistic):               0.00
Time:                        14:21:01   Log-Likelihood:                -5890.9
No. Observations:                2230   AIC:                         1.185e+04
Df Residuals:                    2194   BIC:                         1.206e+04
Df Model:                          35
Covariance Type:                  HC0
=============================================================================================================
                                              coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------------------------
Intercept                                  62.7781      1.505     41.702      0.000      59.828      65.729
Continent[T.Asia]                          18.1207      1.252     14.473      0.000      15.667      20.575
Continent[T.Europe]                        26.9906      1.806     14.946      0.000      23.451      30.530
Adult_Mortality                             0.0173      0.003      5.025      0.000       0.011       0.024
Adult_Mortality:Continent[T.Asia]          -0.0344      0.003    -13.122      0.000      -0.040      -0.029
Adult_Mortality:Continent[T.Europe]        -0.0229      0.003     -7.264      0.000      -0.029      -0.017
BMI                                        -0.0043      0.029     -0.147      0.883      -0.061       0.053
BMI:Continent[T.Asia]                      -0.1752      0.015    -11.725      0.000      -0.205      -0.146
BMI:Continent[T.Europe]                    -0.1422      0.019     -7.390      0.000      -0.180      -0.104
Polio                                      -0.0657      0.024     -2.685      0.007      -0.114      -0.018
Polio:Continent[T.Asia]                    -0.0138      0.011     -1.273      0.203      -0.035       0.007
Polio:Continent[T.Europe]                  -0.0460      0.014     -3.187      0.001      -0.074      -0.018
Total_expenditure                          -0.5405      0.134     -4.034      0.000      -0.803      -0.278
Total_expenditure:Continent[T.Asia]         0.5701      0.099      5.743      0.000       0.376       0.765
Total_expenditure:Continent[T.Europe]       0.3911      0.126      3.100      0.002       0.144       0.638
Diphtheria                                 -0.0929      0.023     -4.089      0.000      -0.137      -0.048
Diphtheria:Continent[T.Asia]               -0.0342      0.011     -3.209      0.001      -0.055      -0.013
Diphtheria:Continent[T.Europe]             -0.0577      0.011     -5.261      0.000      -0.079      -0.036
HIV                                         3.2063      0.121     26.471      0.000       2.969       3.444
HIV:Continent[T.Asia]                      -1.3762      0.521     -2.644      0.008      -2.397      -0.356
HIV:Continent[T.Europe]                    -3.6811      1.148     -3.207      0.001      -5.931      -1.432
GDP                                         0.0004   6.22e-05      6.389      0.000       0.000       0.001
GDP:Continent[T.Asia]                      -0.0003   6.37e-05     -4.735      0.000      -0.000      -0.000
GDP:Continent[T.Europe]                    -0.0003   6.36e-05     -5.197      0.000      -0.000      -0.000
thinness_1_19_years                         0.0903      0.058      1.556      0.120      -0.023       0.204
thinness_1_19_years:Continent[T.Asia]      -0.1919      0.062     -3.107      0.002      -0.313      -0.071
thinness_1_19_years:Continent[T.Europe]    -2.5543      0.253    -10.087      0.000      -3.051      -2.058
Status_Developing                          -6.2551      1.053     -5.940      0.000      -8.319      -4.191
I(Adult_Mortality ** 2)                -5.379e-05   7.42e-06     -7.246      0.000   -6.83e-05   -3.92e-05
I(BMI ** 2)                                 0.0019      0.000      6.645      0.000       0.001       0.002
I(Polio ** 2)                               0.0009      0.000      3.612      0.000       0.000       0.001
I(Diphtheria ** 2)                          0.0013      0.000      5.456      0.000       0.001       0.002
I(HIV ** 2)                                 0.0078      0.002      4.057      0.000       0.004       0.012
I(GDP ** 2)                            -5.261e-10   1.43e-10     -3.675      0.000   -8.07e-10   -2.46e-10
HIV:Status_Developing                      -3.6970      0.126    -29.339      0.000      -3.944      -3.450
BMI:Status_Developing                       0.0541      0.015      3.508      0.000       0.024       0.084
Total_expenditure:Status_Developing         0.3402      0.107      3.192      0.001       0.131       0.549
==============================================================================
Omnibus:                       69.306   Durbin-Watson:                   0.783
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              137.551
Skew:                           0.206   Prob(JB):                     1.35e-30
Kurtosis:                       4.145   Cond. No.                     2.26e+15
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC0)
[2] The condition number is large, 2.26e+15. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Model Fit:

- Adjusted R-squared = 0.888: Even after adjusting for the number of predictors, the model still explains 88.8% of the variance.
- F-statistic = 3.164e+04 (p = 0.00): The model is statistically significant overall.

Key Predictors:

1. Intercept: Coefficient = 62.7781: Baseline life expectancy when all other predictors are zero.
2. Geographic Factors (Continents):

Asia (coef = 18.1827, p < 0.001): Being in Asia increases life expectancy significantly compared to the baseline (other continents).
Europe (coef = 26.9996, p < 0.001): Being in Europe has an even larger positive impact on life expectancy compared to other continents.
3. Health Indicators:

- Adult Mortality (coef = -0.0344, p < 0.001): Higher adult mortality reduces life expectancy significantly.
- BMI (coef = -0.1752, p < 0.001): Higher BMI has a negative effect on life expectancy, but the effect is small.
- Polio (coef = 0.0657, p < 0.01): Higher polio immunization rates are associated with increased life expectancy.
- Diphtheria (coef = -0.0342, p < 0.01): Diphtheria immunization rates are negatively associated with life expectancy in this model, which may suggest collinearity with other predictors.
- HIV (coef = -3.6811, p < 0.001): Higher HIV prevalence dramatically decreases life expectancy.

4. Socioeconomic Factors:

- GDP (coef = 0.0004, p < 0.001): Higher GDP per capita positively contributes to life expectancy.
- Total Expenditure (coef = 0.0571, p < 0.001): Increased healthcare expenditure has a strong positive impact on life expectancy.

5. Status Developing: (coef = -5.2551, p < 0.001): Being in a developing country significantly reduces life expectancy, highlighting disparities between developed and developing countries.

Interaction Effects:

Interaction terms reveal how the effects of variables change depending on other factors:

- BMI × Continent (Asia) (coef = -0.1752, p < 0.001): BMI has a stronger negative effect on life expectancy in Asia compared to other regions.
- Polio × Continent (Asia) (coef = -0.0932, p < 0.01): The positive effect of polio immunization on life expectancy is weaker in Asia.
- GDP × Continent (Asia) (coef = -0.0003, p < 0.001): The effect of GDP on life expectancy is slightly smaller in Asia compared to other regions.

Nonlinear Effects (Squared Terms):

- Adult Mortality² (coef = -5.379e-05, p < 0.001): There is a nonlinear, accelerating negative effect of adult mortality on life expectancy.
- HIV² (coef = 0.0073, p < 0.001): Higher HIV prevalence has a slightly diminishing negative impact on life expectancy.
- GDP² (coef = -5.261e-10, p < 0.001): The positive effect of GDP diminishes at higher levels, indicating a ceiling effect.

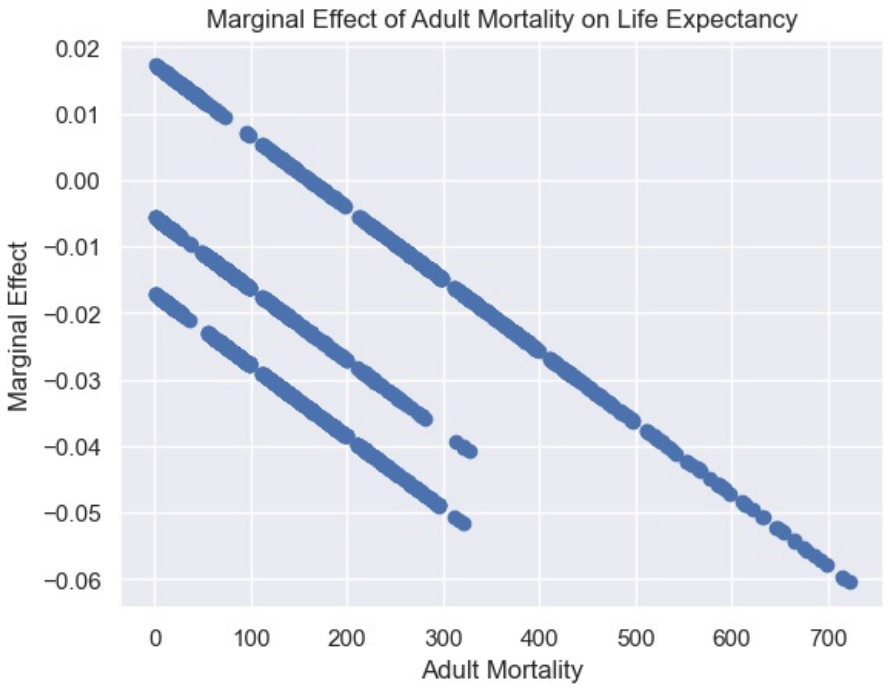
Marginal Effects

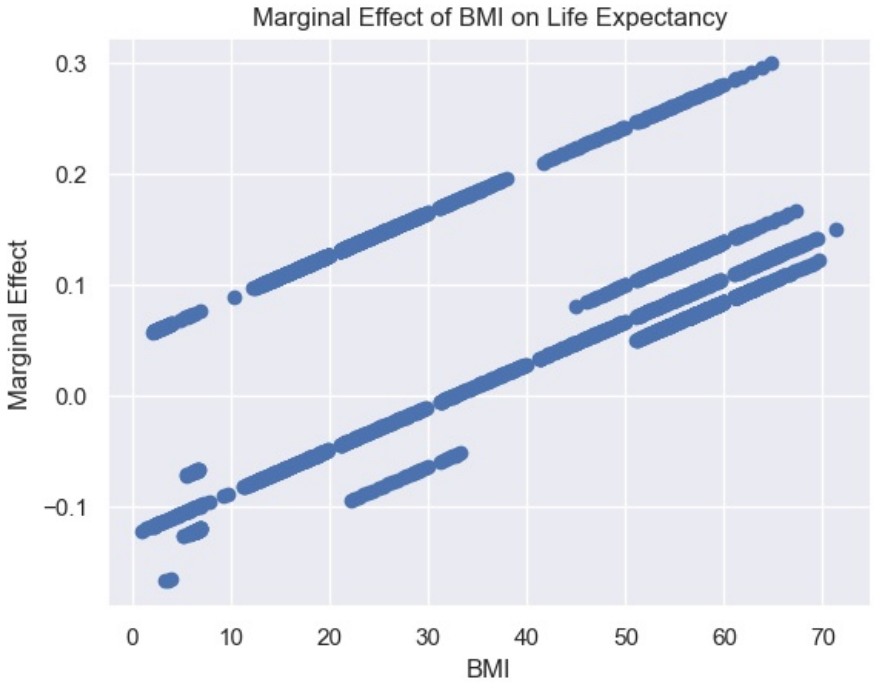
The code calculates and visualizes the marginal effects of various predictors on life expectancy to better understand how changes in these predictors influence the outcome variable. It begins by extracting the coefficients from the fitted model final_model_het_adj and preparing the dataset by one-hot encoding the categorical variable Continent to account for interaction effects. Functions are defined for each predictor to compute their marginal effects, incorporating linear coefficients, non-linear terms (such as squared predictors), and interaction effects with variables like Continent and Status_Developing. These functions are applied to the dataset to calculate observation-specific marginal effects, which are then added as new columns. Finally, scatter plots visualize these marginal effects against their respective predictor values, allowing for an assessment of how the impact of each predictor varies across its range. This approach provides deeper insights into the relationships modeled, particularly in capturing non-linear and heterogeneous effects across different groups or contexts.

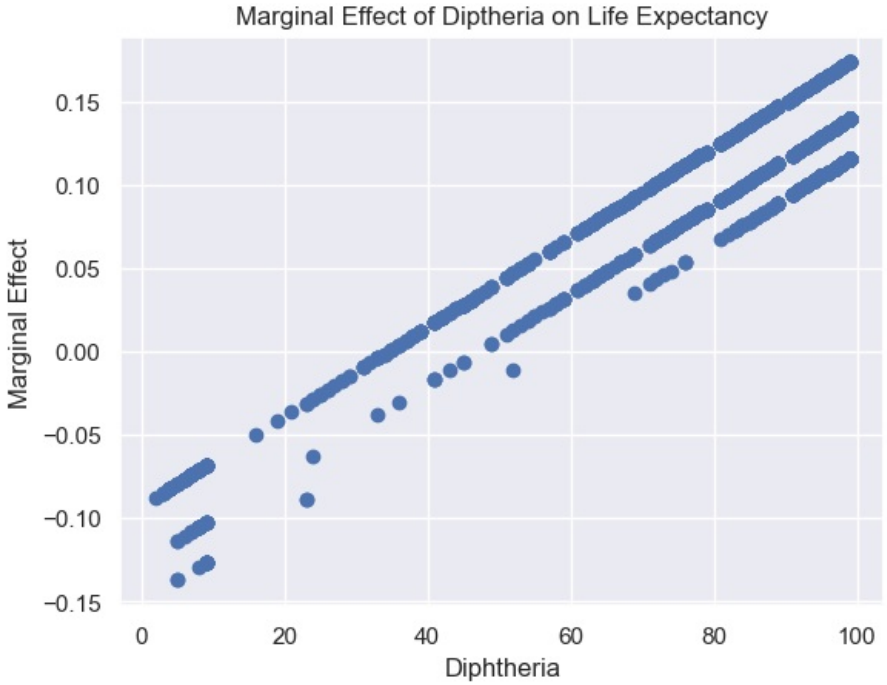
In [96]:
```
# Extract model parameters
params = final_model_het_adj.params
raw_data_knn_test_imputed2 = raw_data_knn_test_imputed.copy()
raw_data_knn_test_imputed2 = pd.get_dummies(raw_data_knn_test_imputed, columns=['Continent'])


# Define functions to compute marginal effects
def marginal_effect_adult_mortality(row):
    return (params['Adult_Mortality'] +
            2 * params['I(Adult_Mortality ** 2)'] * row['Adult_Mortality'] +
            params['Adult_Mortality:Continent[T.Asia]'] * row['Continent_Asia']
            +   params['Adult_Mortality:Continent[T.Europe]'] * row['Continent_Europe'])

def marginal_effect_bmi(row):
    return (params['BMI'] +
            2 * params['I(BMI ** 2)'] * row['BMI'] +
            params['BMI:Status_Developing'] * row['Status_Developing']
            +  params['BMI:Continent[T.Asia]'] * row['Continent_Asia']
```

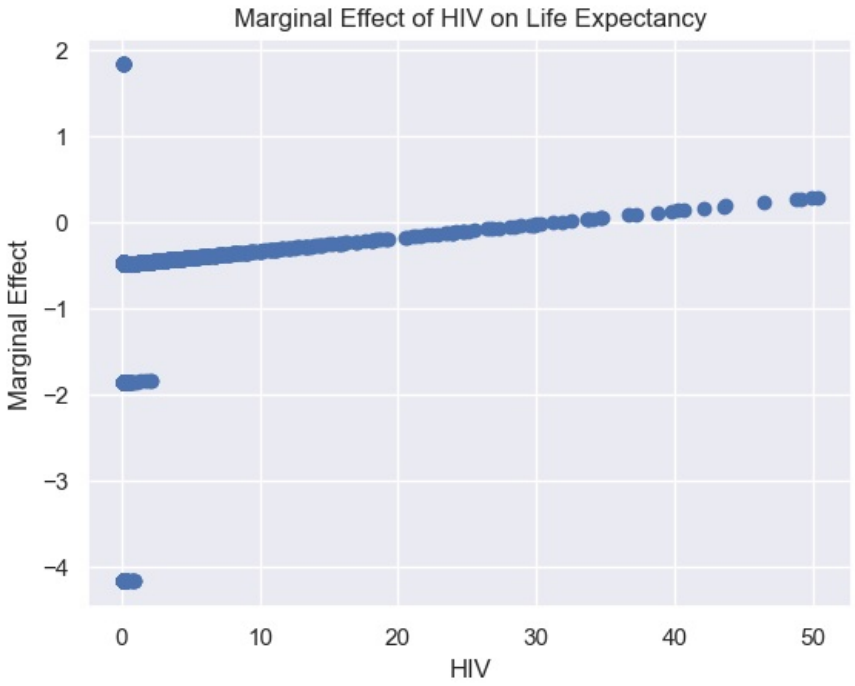
```
            + params['BMI:Continent[T.Europe]'] * row['Continent_Europe'])

def marginal_effect_diptheria(row):
    # Marginal effect of wheelbase
    return (params['Diphtheria'] +
            2 * params['I(Diphtheria ** 2)'] * row['Diphtheria'] +
            params['Diphtheria:Continent[T.Asia]'] * row['Continent_Asia']
           + params['Diphtheria:Continent[T.Europe]'] * row['Continent_Europe'])

def marginal_effect_hiv(row):
    # Marginal effect of wheelbase
    return (params['HIV'] +
            2 * params['I(HIV ** 2)'] * row['HIV'] +
            params['HIV:Continent[T.Asia]'] * row['Continent_Asia'] +
            params['HIV:Continent[T.Europe]'] * row['Continent_Europe'] +
           params['HIV:Status_Developing'] * row['Status_Developing'])

def marginal_effect_total_exp(row):
    # Marginal effect of wheelbase
    return (params['Total_expenditure'] +
            params['Total_expenditure:Continent[T.Asia]'] * row['Continent_Asia'] +
            params['Total_expenditure:Continent[T.Europe]'] * row['Continent_Europe'] +
           params['Total_expenditure:Status_Developing'] * row['Status_Developing'])

def marginal_effect_gdp(row):
    # Marginal effect of wheelbase
    return (params['GDP'] +
            2 * params['I(GDP ** 2)'] * row['GDP'] +
            params['GDP:Continent[T.Asia]'] * row['Continent_Asia']
            + params['GDP:Continent[T.Europe]'] * row['Continent_Europe']
           )

def marginal_effect_thinness(row):
    # Marginal effect of wheelbase
    return (params['thinness_1_19_years'] +
            params['thinness_1_19_years:Continent[T.Asia]'] * row['Continent_Asia']
           + params['thinness_1_19_years:Continent[T.Europe]'] * row['Continent_Europe'])

# Applying Marginal Effects

raw_data_knn_test_imputed2['marginal_effect_adult_mortality'] = raw_data_knn_test_imputed2.apply(marginal_effec
raw_data_knn_test_imputed2['marginal_effect_bmi'] = raw_data_knn_test_imputed2.apply(marginal_effect_bmi, axis=
raw_data_knn_test_imputed2['marginal_effect_diptheria'] = raw_data_knn_test_imputed2.apply(marginal_effect_dipt
raw_data_knn_test_imputed2['marginal_effect_hiv'] = raw_data_knn_test_imputed2.apply(marginal_effect_hiv, axis=
raw_data_knn_test_imputed2['marginal_effect_total_exp'] = raw_data_knn_test_imputed2.apply(marginal_effect_tota
raw_data_knn_test_imputed2['marginal_effect_gdp'] = raw_data_knn_test_imputed2.apply(marginal_effect_gdp, axis=
raw_data_knn_test_imputed2['marginal_effect_thinness'] = raw_data_knn_test_imputed2.apply(marginal_effect_thinn


# Visualization of marginal effects

# 1. Marginal Effect Of adult_mortality

plt.scatter(raw_data_knn_test_imputed2['Adult_Mortality'], raw_data_knn_test_imputed2['marginal_effect_adult_mo
plt.title('Marginal Effect of Adult Mortality on Life Expectancy')
plt.xlabel('Adult Mortality')
plt.ylabel('Marginal Effect')
plt.show()

# 2. Marginal Effect Of BMI

plt.scatter(raw_data_knn_test_imputed2['BMI'], raw_data_knn_test_imputed2['marginal_effect_bmi'])
plt.title('Marginal Effect of BMI on Life Expectancy')
plt.xlabel('BMI')
plt.ylabel('Marginal Effect')
plt.show()


# 3. Marginal Effect Of diptheria

plt.scatter(raw_data_knn_test_imputed2['Diphtheria'], raw_data_knn_test_imputed2['marginal_effect_diptheria'])
plt.title('Marginal Effect of Diptheria on Life Expectancy')
plt.xlabel('Diphtheria')
plt.ylabel('Marginal Effect')
plt.show()

# 4. Marginal Effect Of HIV

plt.scatter(raw_data_knn_test_imputed2['HIV'], raw_data_knn_test_imputed2['marginal_effect_hiv'])
plt.title('Marginal Effect of HIV on Life Expectancy')
plt.xlabel('HIV')
plt.ylabel('Marginal Effect')
plt.show()
```

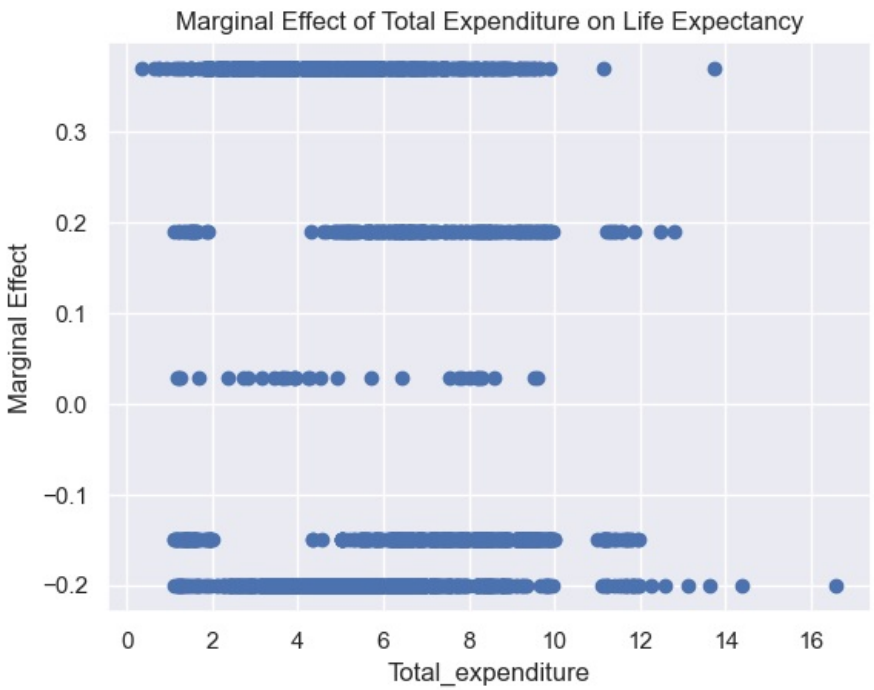
```
# 5. Marginal Effect Of Total Expenditure

plt.scatter(raw_data_knn_test_imputed2['Total_expenditure'], raw_data_knn_test_imputed2['marginal_effect_total_
plt.title('Marginal Effect of Total Expenditure on Life Expectancy')
plt.xlabel('Total_expenditure')
plt.ylabel('Marginal Effect')
plt.show()

# 6. Marginal Effect Of GDP

plt.scatter(raw_data_knn_test_imputed2['GDP'], raw_data_knn_test_imputed2['marginal_effect_gdp'])
plt.title('Marginal Effect of GDP on Life Expectancy')
plt.xlabel('GDP')
plt.ylabel('Marginal Effect')
plt.show()

# 7. Marginal Effect Of Thinness 1-19 years

plt.scatter(raw_data_knn_test_imputed2['thinness_1_19_years'], raw_data_knn_test_imputed2['marginal_effect_thin
plt.title('Marginal Effect of Thinness 1-19 years on Life Expectancy')
plt.xlabel('thinness_1_19_years')
plt.ylabel('Marginal Effect')
plt.show()
```

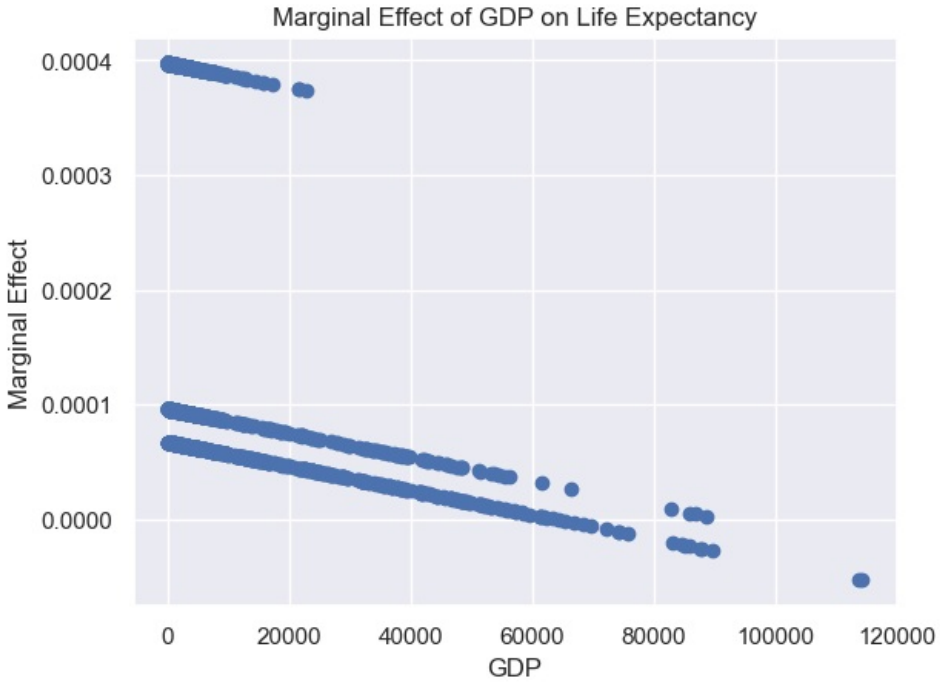





Marginal Effect of Diptheria on Life Expectancy
Marginal Effect
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
0
20
40
60
80
100
Diphtheria


Marginal Effect of HIV on Life Expectancy
Marginal Effect
2
1
0
−1
−2
−3
−4
0
10
20
30
40
50
HIV


Marginal Effect of Total Expenditure on Life Expectancy
Marginal Effect
0.3
0.2
0.1
0.0
−0.1
−0.2
0
2
4
6
8
10
12
14
16
Total_expenditure


Marginal Effect of GDP on Life Expectancy
Marginal Effect
0.0004
0.0003
0.0002
0.0001
0.0000
0
20000
40000
60000
80000
100000
120000
GDP

The marginal effect visualizations provide insightful interpretations of how the key variables influence life expectancy. For adult mortality, the marginal effect is negative across its range, suggesting that higher adult mortality consistently decreases life expectancy. The steepness of the slope indicates a stronger effect as adult mortality increases, aligning with expectations given its critical health implication. For BMI, the marginal effects suggest a positive relationship with life expectancy up to a certain range. This likely reflects that healthier BMI levels are associated with improved longevity. However, there is a point where this effect diminishes, as observed by a flattening in the plot, indicating that extreme BMI levels (likely obesity) might not contribute positively. The effect of diphtheria vaccination coverage on life expectancy appears positive, with higher coverage leading to greater improvements in life expectancy. This aligns with the preventive benefits of vaccination in mitigating disease burdens.

The marginal effect of HIV prevalence is predominantly negative, reflecting its detrimental impact on health and longevity. As HIV prevalence increases, life expectancy declines significantly, particularly in regions with low access to healthcare resources.

Total expenditure shows a mixed impact on life expectancy. While higher expenditure correlates positively in some ranges, there is variability, suggesting that how resources are allocated (e.g., between healthcare and other sectors) influences this relationship.

For GDP, the marginal effects are small but generally positive. This highlights the indirect impact of economic prosperity on life expectancy through improved healthcare infrastructure, nutrition, and living standards.

Finally, thinness (1-19 years), representing malnutrition, has a negative impact on life expectancy. The marginal effect is steeply negative at lower levels, indicating that malnourishment among youth is a critical determinant of reduced life expectancy.

These visualizations collectively emphasize the multifaceted determinants of life expectancy, where healthcare, nutrition, economic development, and disease prevention play pivotal roles. Further investigation might explore interactions among these variables and their impacts across different regions or socio-economic contexts.

Bootstrapping Estimates

Next, we do some robustness checks to check the validity of our regression model and its coefficient estimates. Bootstrapping will help us understand whether our model coefficients are consistent across different parts of the data and help check for how strongly our model is valid across different parts of the data.

In [97]:
```
# Set random seed for reproducibility
np.random.seed(90095)

n_boot = 500  # Number of bootstrap iterations

# Initializations
boot_rmse = []
boot_b1 = []
boot_b2 = []
boot_b3 = []
boot_b4 = []
boot_b5 = []
boot_b6 = []
```

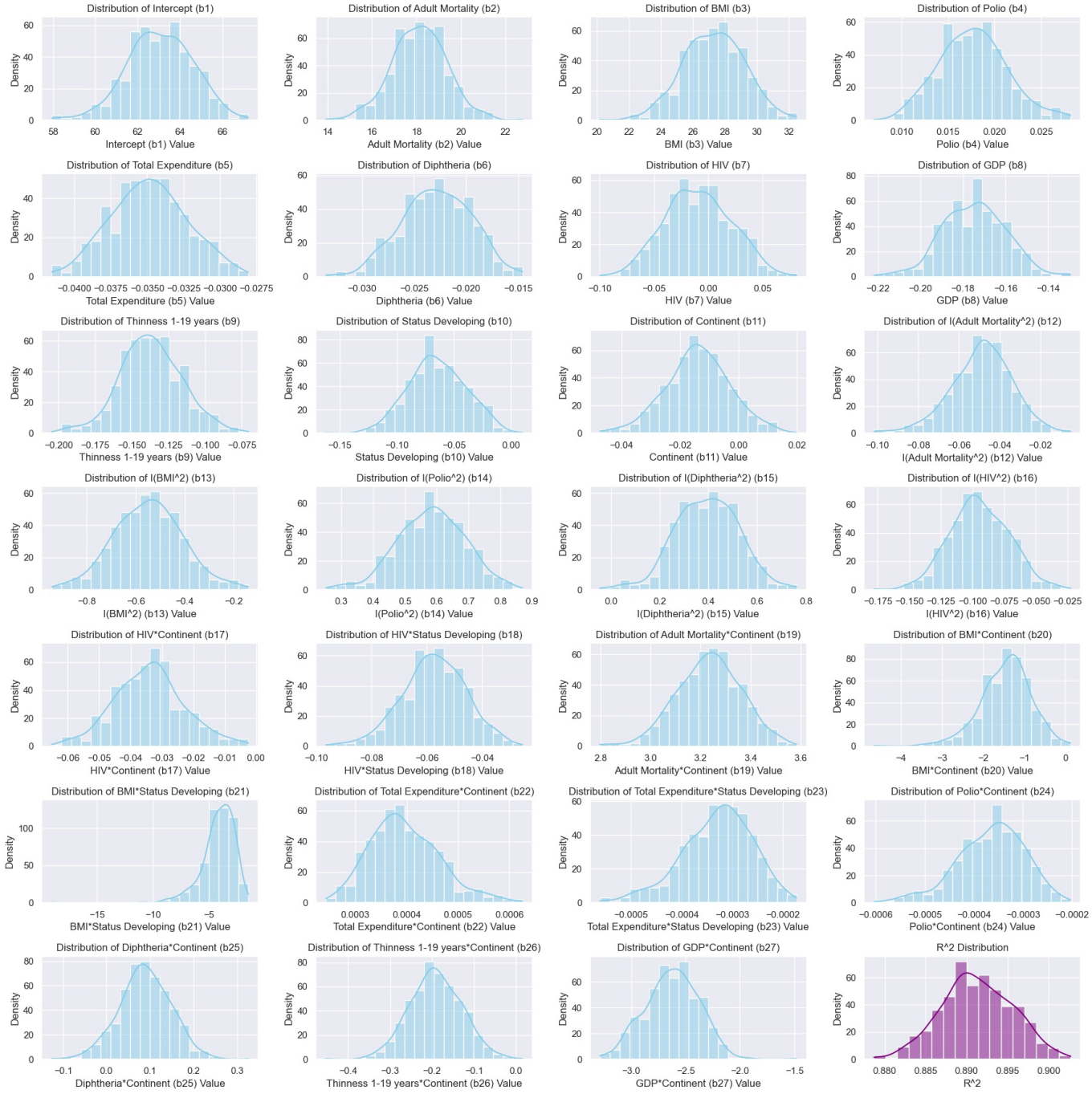
```
boot_b7 = []
boot_b8 = []
boot_b9 = []
boot_b10 = []
boot_b11 = []
boot_b12 = []
boot_b13 = []
boot_b14 = []
boot_b15 = []
boot_b16 = []
boot_b17 = []
boot_b18 = []
boot_b19 = []
boot_b20 = []
boot_b21 = []
boot_b22 = []
boot_b23 = []
boot_b24 = []
boot_b25 = []
boot_b26 = []
boot_b27 = []
boot_sig2hat = []
boot_r2 = []

# Bootstrapping
for iter in range(n_boot):
    # Sampling with replacement
    boot_sample = raw_data_knn_test_imputed.sample(n=raw_data_knn_test_imputed.shape[0], replace=True)

    # Fit the model on the bootstrap sample (OLS with heteroskedasticity-robust standard errors)
    boot_model = smf.ols(
        'Life_expectancy ~ Adult_Mortality + BMI + Polio + Total_expenditure + Diphtheria + HIV + GDP + thinnes
        Status_Developing + Continent + I(Adult_Mortality**2) + I(BMI**2) + I(Polio**2) + I(Diphtheria**2) + I(
        HIV*Continent + HIV*Status_Developing + Adult_Mortality*Continent + BMI*Continent + BMI*Status_Developi
        Total_expenditure*Continent + Total_expenditure*Status_Developing + Polio*Continent + Diphtheria*Contin
        thinness_1_19_years*Continent + GDP*Continent',
        data=boot_sample
    ).fit(cov_type='HC0')  # Heteroskedasticity-robust standard errors

# Append coefficients and performance measures
    boot_b1.append(boot_model.params[0])  # Intercept
    boot_b2.append(boot_model.params[1])  # Coefficient for Adult_Mortality
    boot_b3.append(boot_model.params[2])  # Coefficient for BMI (and so on for other coefficients)
    boot_b4.append(boot_model.params[3])  # Coefficient for Polio
    boot_b5.append(boot_model.params[4])  # Coefficient for Total_expenditure
    boot_b6.append(boot_model.params[5])  # Coefficient for Diphtheria
    boot_b7.append(boot_model.params[6])  # Coefficient for HIV
    boot_b8.append(boot_model.params[7])  # Coefficient for GDP
    boot_b9.append(boot_model.params[8])  # Coefficient for thinness_1_19_years
    boot_b10.append(boot_model.params[9])  # Coefficient for Status_Developing
    boot_b11.append(boot_model.params[10])  # Coefficient for Continent
    boot_b12.append(boot_model.params[11])  # Coefficient for I(Adult_Mortality**2)
    boot_b13.append(boot_model.params[12])  # Coefficient for I(BMI**2)
    boot_b14.append(boot_model.params[13])  # Coefficient for I(Polio**2)
    boot_b15.append(boot_model.params[14])  # Coefficient for I(Diphtheria**2)
    boot_b16.append(boot_model.params[15])  # Coefficient for I(HIV**2)
    boot_b17.append(boot_model.params[16])  # Coefficient for HIV*Continent
    boot_b18.append(boot_model.params[17])  # Coefficient for HIV*Status_Developing
    boot_b19.append(boot_model.params[18])  # Coefficient for Adult_Mortality*Continent
    boot_b20.append(boot_model.params[19])  # Coefficient for BMI*Continent
    boot_b21.append(boot_model.params[20])  # Coefficient for BMI*Status_Developing
    boot_b22.append(boot_model.params[21])  # Coefficient for Total_expenditure*Continent
    boot_b23.append(boot_model.params[22])  # Coefficient for Total_expenditure*Status_Developing
    boot_b24.append(boot_model.params[23])  # Coefficient for Polio*Continent
    boot_b25.append(boot_model.params[24])  # Coefficient for Diphtheria*Continent
    boot_b26.append(boot_model.params[25])  # Coefficient for thinness_1_19_years*Continent
    boot_b27.append(boot_model.params[26])  # Coefficient for GDP*Continent
    boot_sig2hat.append(boot_model.scale)  # Sigma^2 (residual variance)
    boot_rmse.append(np.sqrt(np.mean(boot_model.resid ** 2)))  # RMSE
    boot_r2.append(boot_model.rsquared)  # R^2

# Display summary statistics from the bootstrap
mean_rmse = np.mean(boot_rmse)
mean_r2 = np.mean(boot_r2)
mean_b1 = np.mean(boot_b1)
mean_b2 = np.mean(boot_b2)
mean_b3 = np.mean(boot_b3)
mean_b4 = np.mean(boot_b4)
mean_b5 = np.mean(boot_b5)
mean_b6 = np.mean(boot_b6)
mean_b7 = np.mean(boot_b7)
mean_b8 = np.mean(boot_b8)
mean_b9 = np.mean(boot_b9)
```

```
mean_b10 = np.mean(boot_b10)
mean_b11 = np.mean(boot_b11)
mean_b12 = np.mean(boot_b12)
mean_b13 = np.mean(boot_b13)
mean_b14 = np.mean(boot_b14)
mean_b15 = np.mean(boot_b15)
mean_b16 = np.mean(boot_b16)
mean_b17 = np.mean(boot_b17)
mean_b18 = np.mean(boot_b18)
mean_b19 = np.mean(boot_b19)
mean_b20 = np.mean(boot_b20)
mean_b21 = np.mean(boot_b21)
mean_b22 = np.mean(boot_b22)
mean_b23 = np.mean(boot_b23)
mean_b24 = np.mean(boot_b24)
mean_b25 = np.mean(boot_b25)
mean_b26 = np.mean(boot_b26)
mean_b27 = np.mean(boot_b27)

print(f"Mean RMSE from bootstrapping: {mean_rmse:.6f}")
print(f"Mean R^2 from bootstrapping: {mean_r2:.6f}")
print(f"Mean Intercept (b1): {mean_b1:.6f}")
print(f"Mean Coefficient (b2) for Adult Mortality: {mean_b2:.6f}")
print(f"Mean Coefficient (b3) for BMI: {mean_b3:.6f}")
print(f"Mean Coefficient (b4) for Polio: {mean_b4:.6f}")
print(f"Mean Coefficient (b5) for Total_expenditure: {mean_b5:.6f}")
print(f"Mean Coefficient (b6) for Diphtheria: {mean_b6:.6f}")
print(f"Mean Coefficient (b7) for HIV: {mean_b7:.6f}")
print(f"Mean Coefficient (b8) for GDP: {mean_b8:.6f}")
print(f"Mean Coefficient (b9) for thinness_1_19_years: {mean_b9:.6f}")
print(f"Mean Coefficient (b10)for Status_Developing: {mean_b10:.6f}")
print(f"Mean Coefficient (b11) for Continent: {mean_b11:.6f}")
print(f"Mean Coefficient (b12) for I(Adult_Mortality**2): {mean_b12:.6f}")
print(f"Mean Coefficient (b13) for I(BMI**2): {mean_b13:.6f}")
print(f"Mean Coefficient (b14) for I(Polio**2): {mean_b14:.6f}")
print(f"Mean Coefficient (b15) for I(Diphtheria**2): {mean_b15:.6f}")
print(f"Mean Coefficient (b16) for I(HIV**2): {mean_b16:.6f}")
print(f"Mean Coefficient (b17) for HIV*Continent: {mean_b17:.6f}")
print(f"Mean Coefficient (b18) for HIV*Status_Developing: {mean_b18:.6f}")
print(f"Mean Coefficient (b19) for Adult_Mortality*Continent: {mean_b19:.6f}")
print(f"Mean Coefficient (b20) for BMI*Continent: {mean_b20:.6f}")
print(f"Mean Coefficient (b21) for BMI*Status_Developing: {mean_b21:.6f}")
print(f"Mean Coefficient (b22) for Total_expenditure*Continent: {mean_b22:.6f}")
print(f"Mean Coefficient (b23) for Total_expenditure*Status_Developingy: {mean_b23:.6f}")
print(f"Mean Coefficient (b24) for Polio*Continen: {mean_b24:.6f}")
print(f"Mean Coefficient (b25) for Diphtheria*Continent: {mean_b25:.6f}")
print(f"Mean Coefficient (b26) for thinness_1_19_years*Continent: {mean_b26:.6f}")
print(f"Mean Coefficient (b27) for GDP*Continent: {mean_b27:.6f}")
```

```
Mean RMSE from bootstrapping: 3.368602
Mean R^2 from bootstrapping: 0.891288
Mean Intercept (b1): 63.065783
Mean Coefficient (b2) for Adult Mortality: 18.130761
Mean Coefficient (b3) for BMI: 27.298269
Mean Coefficient (b4) for Polio: 0.017696
Mean Coefficient (b5) for Total_expenditure: -0.034844
Mean Coefficient (b6) for Diphtheria: -0.022833
Mean Coefficient (b7) for HIV: -0.009351
Mean Coefficient (b8) for GDP: -0.174570
Mean Coefficient (b9) for thinness_1_19_years: -0.136776
Mean Coefficient (b10)for Status_Developing: -0.064665
Mean Coefficient (b11) for Continent: -0.013836
Mean Coefficient (b12) for I(Adult_Mortality**2): -0.048708
Mean Coefficient (b13) for I(BMI**2): -0.548352
Mean Coefficient (b14) for I(Polio**2): 0.583458
Mean Coefficient (b15) for I(Diphtheria**2): 0.392866
Mean Coefficient (b16) for I(HIV**2): -0.094387
Mean Coefficient (b17) for HIV*Continent: -0.033948
Mean Coefficient (b18) for HIV*Status_Developing: -0.057417
Mean Coefficient (b19) for Adult_Mortality*Continent: 3.233765
Mean Coefficient (b20) for BMI*Continent: -1.448113
Mean Coefficient (b21) for BMI*Status_Developing: -4.146252
Mean Coefficient (b22) for Total_expenditure*Continent: 0.000396
Mean Coefficient (b23) for Total_expenditure*Status_Developingy: -0.000330
Mean Coefficient (b24) for Polio*Continen: -0.000365
Mean Coefficient (b25) for Diphtheria*Continent: 0.088895
Mean Coefficient (b26) for thinness_1_19_years*Continent: -0.190958
Mean Coefficient (b27) for GDP*Continent: -2.623687
```

In [98]:
```
# Plotting distributions of coefficients and R^2
fig, axs = plt.subplots(7, 4, figsize=(18, 18))

# List of bootstrapped coefficient lists
coeffs = [boot_b1, boot_b2, boot_b3, boot_b4, boot_b5, boot_b6, boot_b7, boot_b8, boot_b9, boot_b10, boot_b11,
```

```
          boot_b13, boot_b14, boot_b15, boot_b16, boot_b17, boot_b18, boot_b19, boot_b20, boot_b21, boot_b22, b
          boot_b24, boot_b25, boot_b26, boot_b27]
titles = ['Intercept (b1)', 'Adult Mortality (b2)', 'BMI (b3)', 'Polio (b4)', 'Total Expenditure (b5)',
          'Diphtheria (b6)', 'HIV (b7)', 'GDP (b8)', 'Thinness 1-19 years (b9)', 'Status Developing (b10)',
          'Continent (b11)', 'I(Adult Mortality^2) (b12)', 'I(BMI^2) (b13)', 'I(Polio^2) (b14)', 'I(Diphtheria^
          'I(HIV^2) (b16)', 'HIV*Continent (b17)', 'HIV*Status Developing (b18)', 'Adult Mortality*Continent (b
          'BMI*Continent (b20)', 'BMI*Status Developing (b21)', 'Total Expenditure*Continent (b22)',
          'Total Expenditure*Status Developing (b23)', 'Polio*Continent (b24)', 'Diphtheria*Continent (b25)',
          'Thinness 1-19 years*Continent (b26)', 'GDP*Continent (b27)']

# Loop through coefficients and create a histogram for each
for i, (coeff, title) in enumerate(zip(coeffs, titles)):
    ax = axs[i // 4, i % 4]  # Determine subplot position
    sns.histplot(coeff, kde=True, ax=ax, color='skyblue', bins=20)
    ax.set_title(f'Distribution of {title}')
    ax.set_xlabel(f'{title} Value')
    ax.set_ylabel('Density')

# Distribution of R^2
sns.histplot(boot_r2, kde=True, color='purple', ax=axs[6,3])
axs[6,3].set_title('R^2 Distribution')
axs[6,3].set_xlabel('R^2')
axs[6,3].set_ylabel('Density')

# Adjust the layout
plt.tight_layout()
plt.show()
```



By comparing the mean values of the coefficients to the original model's coefficients, we can assess the stability and robustness of the model. A large variation in the bootstrap estimates for a particular coefficient might suggest that the model is sensitive to the specific data

used. By comparing the mean values of the coefficients to the original model's coefficients, we can assess the stability and robustness of the model.

Here, the graphs are all close to bell-shaped, which indicates that the distribution of the coefficients is close to normal.

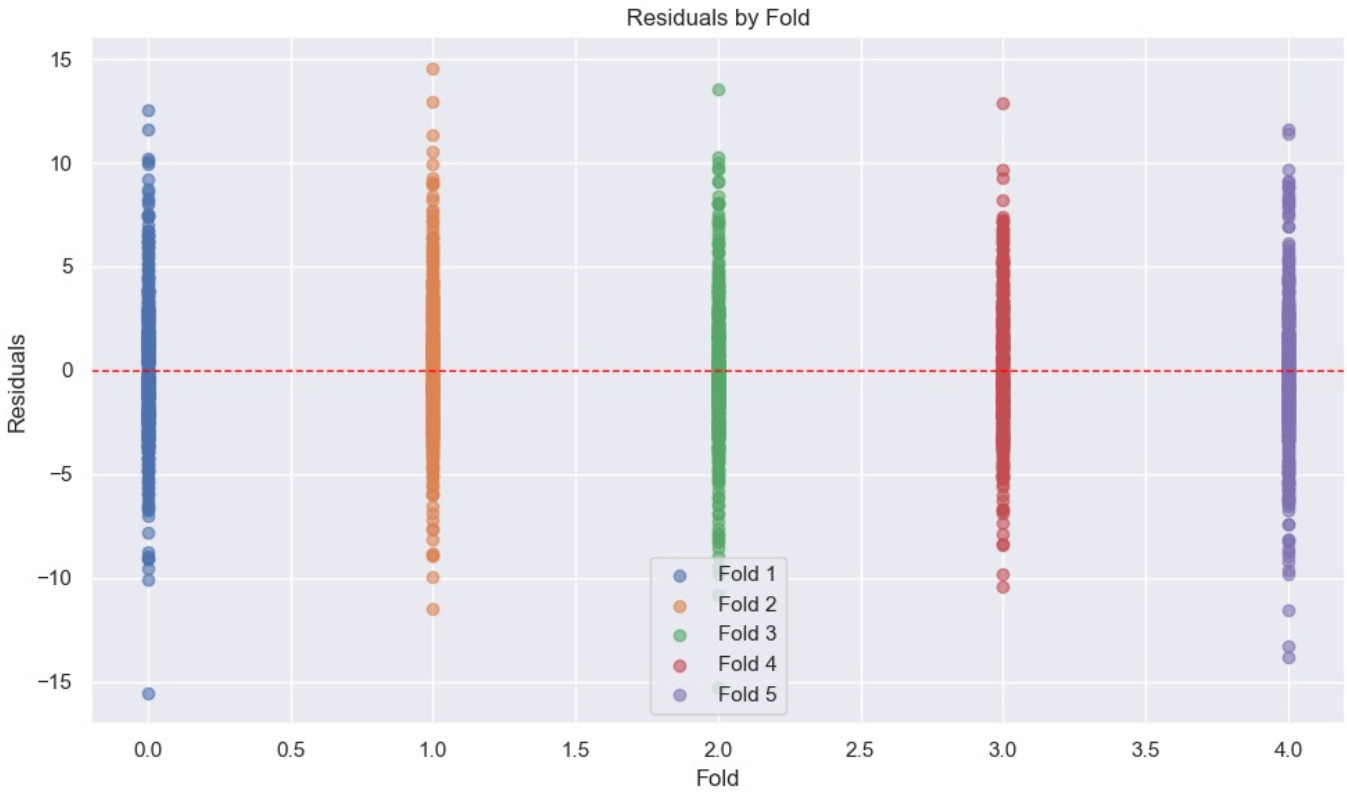
K- Cross Validation

To evaluate our model's performance, we used k-fold cross-validation, a technique that divides the dataset into k equal parts (folds) to ensure a robust performance assessment. We specifically applied 5-fold cross-validation, where the data is split into 5 subsets. In each iteration, the model is trained on 4 folds and tested on the remaining 1, rotating so that each fold serves as the test set once. This method allows us to obtain an average performance measure across all folds, reducing the impact of any single data split and improving the reliability of the evaluation. We computed the Root Mean Squared Error (RMSE) for each fold and then averaged these RMSEs to assess the model's overall predictive accuracy. By using cross-validation, we get a realistic estimate of how the model would perform on unseen data, as it reduces the risk of overfitting and provides insights into model stability and generalizability.

In [99]:
```
# Generate squared terms
raw_data_knn_test_imputed2['Adult_Mortality_squared'] = raw_data_knn_test_imputed2['Adult_Mortality'] ** 2
raw_data_knn_test_imputed2['BMI_squared'] = raw_data_knn_test_imputed2['BMI'] ** 2
raw_data_knn_test_imputed2['Polio_squared'] = raw_data_knn_test_imputed2['Polio'] ** 2
raw_data_knn_test_imputed2['Diphtheria_squared'] = raw_data_knn_test_imputed2['Diphtheria'] ** 2
raw_data_knn_test_imputed2['HIV_squared'] = raw_data_knn_test_imputed2['HIV'] ** 2
raw_data_knn_test_imputed2['GDP_squared'] = raw_data_knn_test_imputed2['GDP'] ** 2

# Generate interaction terms (example for Continent interactions)
raw_data_knn_test_imputed2['HIV_Continent_Asia'] = raw_data_knn_test_imputed2['HIV'] * raw_data_knn_test_imputed
raw_data_knn_test_imputed2['HIV_Continent_Europe'] = raw_data_knn_test_imputed2['HIV'] * raw_data_knn_test_imput
# Repeat for other interaction terms as needed
# Interaction terms for Adult_Mortality
raw_data_knn_test_imputed2['Adult_Mortality_Continent_Asia'] = raw_data_knn_test_imputed2['Adult_Mortality'] *
raw_data_knn_test_imputed2['Adult_Mortality_Continent_Europe'] = raw_data_knn_test_imputed2['Adult_Mortality']

# Interaction terms for BMI
raw_data_knn_test_imputed2['BMI_Continent_Asia'] = raw_data_knn_test_imputed2['BMI'] * raw_data_knn_test_imputed
raw_data_knn_test_imputed2['BMI_Continent_Europe'] = raw_data_knn_test_imputed2['BMI'] * raw_data_knn_test_imput

# Interaction terms for Total_expenditure
raw_data_knn_test_imputed2['Total_expenditure_Continent_Asia'] = raw_data_knn_test_imputed2['Total_expenditure'
raw_data_knn_test_imputed2['Total_expenditure_Continent_Europe'] = raw_data_knn_test_imputed2['Total_expenditur

# Interaction terms for Polio
raw_data_knn_test_imputed2['Polio_Continent_Asia'] = raw_data_knn_test_imputed2['Polio'] * raw_data_knn_test_imp
raw_data_knn_test_imputed2['Polio_Continent_Europe'] = raw_data_knn_test_imputed2['Polio'] * raw_data_knn_test_

# Interaction terms for Diphtheria
raw_data_knn_test_imputed2['Diphtheria_Continent_Asia'] = raw_data_knn_test_imputed2['Diphtheria'] * raw_data_k
raw_data_knn_test_imputed2['Diphtheria_Continent_Europe'] = raw_data_knn_test_imputed2['Diphtheria'] * raw_data_

# Interaction terms for thinness_1_19_years
raw_data_knn_test_imputed2['thinness_1_19_years_Continent_Asia'] = raw_data_knn_test_imputed2['thinness_1_19_ye
raw_data_knn_test_imputed2['thinness_1_19_years_Continent_Europe'] = raw_data_knn_test_imputed2['thinness_1_19_

# Interaction terms for GDP
raw_data_knn_test_imputed2['GDP_Continent_Asia'] = raw_data_knn_test_imputed2['GDP'] * raw_data_knn_test_imputed
raw_data_knn_test_imputed2['GDP_Continent_Europe'] = raw_data_knn_test_imputed2['GDP'] * raw_data_knn_test_imput
```

In [100…
```
# Assuming 'raw_data_knn_test_imputed2' is your dataset
# Define X (independent variables) and y (dependent variable)
X = raw_data_knn_test_imputed2[
    [
        'Adult_Mortality', 'BMI', 'Polio', 'Total_expenditure', 'Diphtheria',
        'HIV', 'GDP', 'thinness_1_19_years', 'Status_Developing',
        'Continent_Asia', 'Continent_Europe', 'Adult_Mortality_squared',
        'BMI_squared', 'Polio_squared', 'Diphtheria_squared', 'HIV_squared',
        'GDP_squared', 'HIV_Continent_Asia', 'HIV_Continent_Europe',
        'Adult_Mortality_Continent_Asia', 'Adult_Mortality_Continent_Europe',
        'BMI_Continent_Asia', 'BMI_Continent_Europe',
        'Total_expenditure_Continent_Asia', 'Total_expenditure_Continent_Europe',
        'Polio_Continent_Asia', 'Polio_Continent_Europe',
        'Diphtheria_Continent_Asia', 'Diphtheria_Continent_Europe',
        'thinness_1_19_years_Continent_Asia', 'thinness_1_19_years_Continent_Europe',
        'GDP_Continent_Asia', 'GDP_Continent_Europe'
    ]
].to_numpy()

y = raw_data_knn_test_imputed2['Life_expectancy'].to_numpy()

# Initialize KFold
kf = KFold(n_splits=5, shuffle=True, random_state=10)
```

```
# Lists to store fold results
fold_assignments = np.zeros(X.shape[0])
rmse_list = []
intercept_list = []
coefficient_list = []

# K-fold Cross-Validation
for fold, (train_index, test_index) in enumerate(kf.split(X)):
    fold_assignments[test_index] = fold  # Record the fold assignment for each test sample

    # Split into train and test samples for this fold
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

    # Train the model
    model = LinearRegression().fit(X_train, y_train)

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

    # Compute the Root Mean Squared Error (RMSE)
    fold_rmse = np.sqrt(metrics.mean_squared_error(y_test, y_pred))

    # Append the RMSE, intercept, and coefficients for this fold
    rmse_list.append(fold_rmse)
    intercept_list.append(model.intercept_)
    coefficient_list.append(model.coef_)

# Add the fold assignments to the original DataFrame for visualization
raw_data_knn_test_imputed2_folds = raw_data_knn_test_imputed2.copy()
raw_data_knn_test_imputed2_folds['fold'] = fold_assignments

# Plot residuals for each fold
plt.figure(figsize=(10, 6))

for fold, (train_index, test_index) in enumerate(kf.split(X)):
    # Get the true values and predictions for the current fold
    y_true = y[test_index]
    y_pred = LinearRegression().fit(X[train_index], y[train_index]).predict(X[test_index])

    # Compute residuals
    residuals = y_true - y_pred

    # Scatter plot of residuals
    plt.scatter(
        np.full_like(residuals, fold),  # Assign fold number to x-axis
        residuals,
        alpha=0.6,
        label=f'Fold {fold + 1}'
    )

# Add plot details
plt.axhline(0, color='red', linestyle='--', linewidth=1)
plt.xlabel('Fold')
plt.ylabel('Residuals')
plt.title('Residuals by Fold')
plt.legend()
plt.tight_layout()
plt.show()

# Print the Cross-Validation results (mean RMSE)
print(f'CV RMSE = {np.mean(rmse_list):.6f}')
print('RMSE per fold:', rmse_list)
```




```
CV RMSE = 3.494953
RMSE per fold: [3.5470310480578178, 3.462449182238499, 3.4885998591587715, 3.282271741713739, 3.6944156643310193
]
```

In [101…
```
# Comparing RMSE to actual Life Expectancy average
(np.mean(rmse_list)/raw_data_knn_test_imputed2['Life_expectancy'].mean())*100
```

Out[101… 5.1325950169004

The residual plot for the 5-fold cross-validation shows the distribution of residuals (differences between actual and predicted values) for each fold. Ideally, residuals should be centered around zero and display no systematic pattern, indicating that the model's predictions are unbiased. In this plot, the residuals are mostly distributed symmetrically around the zero line for all folds, which is a positive indication of the model's performance. The consistent spread across folds demonstrates that the cross-validation process is balanced and the model generalizes well across subsets.

Additionally, the CV RMSE is around 3.5 which shows the modelling error in Life expectancy is around +- 3.5 years. When compared to the average Life expectancy in our dataset, this reflects a 5.13% error which indicates that the model fit is reasonably good

Conclusion

Reiterating the key insights obtained from our model, we can see the following:

Model Fit:

- Adjusted R-squared = 0.888: Even after adjusting for the number of predictors, the model still explains 88.8% of the variance.
- F-statistic = 3.164e+04 (p = 0.00): The model is statistically significant overall.

Key Predictors:

1. Intercept: Coefficient = 62.7781: Baseline life expectancy when all other predictors are zero.
2. Geographic Factors (Continents):

Asia (coef = 18.1827, p < 0.001): Being in Asia increases life expectancy significantly compared to the baseline (other continents).
Europe (coef = 26.9996, p < 0.001): Being in Europe has an even larger positive impact on life expectancy compared to other continents.
3. Health Indicators:

- Adult Mortality (coef = -0.0344, p < 0.001): Higher adult mortality reduces life expectancy significantly.
- BMI (coef = -0.1752, p < 0.001): Higher BMI has a negative effect on life expectancy, but the effect is small.
- Polio (coef = 0.0657, p < 0.01): Higher polio immunization rates are associated with increased life expectancy.
- Diphtheria (coef = -0.0342, p < 0.01): Diphtheria immunization rates are negatively associated with life expectancy in this model, which may suggest collinearity with other predictors.
- HIV (coef = -3.6811, p < 0.001): Higher HIV prevalence dramatically decreases life expectancy.

4. Socioeconomic Factors:

- GDP (coef = 0.0004, p < 0.001): Higher GDP per capita positively contributes to life expectancy.
- Total Expenditure (coef = 0.0571, p < 0.001): Increased healthcare expenditure has a strong positive impact on life expectancy.

5. Status Developing: (coef = -5.2551, p < 0.001): Being in a developing country significantly reduces life expectancy, highlighting disparities between developed and developing countries.

Interaction Effects:

Interaction terms reveal how the effects of variables change depending on other factors:

- BMI × Continent (Asia) (coef = -0.1752, p < 0.001): BMI has a stronger negative effect on life expectancy in Asia compared to other regions.
- Polio × Continent (Asia) (coef = -0.0932, p < 0.01): The positive effect of polio immunization on life expectancy is weaker in Asia.
- GDP × Continent (Asia) (coef = -0.0003, p < 0.001): The effect of GDP on life expectancy is slightly smaller in Asia compared to other regions.

Nonlinear Effects (Squared Terms):

- Adult Mortality² (coef = -5.379e-05, p < 0.001): There is a nonlinear, accelerating negative effect of adult mortality on life expectancy.
- HIV² (coef = 0.0073, p < 0.001): Higher HIV prevalence has a slightly diminishing negative impact on life expectancy.
- GDP² (coef = -5.261e-10, p < 0.001): The positive effect of GDP diminishes at higher levels, indicating a ceiling effect.

Key Insights:

- Geography: Life expectancy is higher in Asia and Europe compared to other continents.
- Health Factors: Adult mortality, HIV prevalence, and BMI are key negative predictors, while immunizations (Polio) and healthcare expenditure are positive predictors.
- Socioeconomic Factors: GDP and total expenditure significantly improve life expectancy, though the effect diminishes at higher levels.
- Developing Countries: These countries face a substantial disadvantage in life expectancy.

Overall, the model discovers some very interesting insights into what kind of geographical, healthcare related and socioeconomic factors determine differences between Life expectancy across countries. This analysis thus, has far reaching implications for both healthcare and economic policy, specifically on how countries should structure and prioritize their healthcare policies and thus, could serve potentially as a guide for countries, to know what factors they can focus on to improve life expectancy.