In [154]:
```
import numpy as np
import pandas as pd
pd.set_option('display.max_rows', 800)
pd.set_option('display.max_columns', 500)

import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

# import all libraries and dependencies for machine learning
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
import statsmodels.api as sm
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression
from statsmodels.stats.outliers_influence import variance_inflation_fac
from sklearn.metrics import mean_absolute_error, mean_squared_error,r2_
import random
```

In [155]:
```
df=pd.read_csv('/Users/LuckyDog/Downloads/Life Expectancy Data (3) (1).
```

In [156]:
```
df.info()
```

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

In [157]:
```
1 df.head()
```

Out[157]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Me
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	

In [158]:
```
1 df.describe()
```

Out[158]:

	Year	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B
count	2938.000000	2928.000000	2928.000000	2938.000000	2744.000000	2938.000000	2385.000000
mean	2007.518720	69.224932	164.796448	30.303948	4.602861	738.251295	80.940461
std	4.613841	9.523867	124.292079	117.926501	4.052413	1987.914858	25.070016
min	2000.000000	36.300000	1.000000	0.000000	0.010000	0.000000	1.000000
25%	2004.000000	63.100000	74.000000	0.000000	0.877500	4.685343	77.000000
50%	2008.000000	72.100000	144.000000	3.000000	3.755000	64.912906	92.000000
75%	2012.000000	75.700000	228.000000	22.000000	7.702500	441.534144	97.000000
max	2015.000000	89.000000	723.000000	1800.000000	17.870000	19479.911610	99.000000

In [159]:
```
1 df.head(5)
```

Out[159]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Me
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0	
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0	
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0	
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0	
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0	

In [160]:
```
num_col = df.select_dtypes(include=np.number).columns
print(f'Numberical columns:\n{num_col}')
cat_col = df.select_dtypes(exclude=np.number).columns
print(f'Categotical columns:\n {cat_col}')
```

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

####Removing Extra Spaces int he columns names###

In [161]:
```
df = df.rename(columns=lambda x:x.strip())
```

In [162]:
```
#Removing spaces
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()

#Encoding the categorical columns
df['Status']= label_encoder.fit_transform(df['Status'])
df.head()
```

Out[162]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measle
0	Afghanistan	2015	1	65.0	263.0	62	0.01	71.279624	65.0	115
1	Afghanistan	2014	1	59.9	271.0	64	0.01	73.523582	62.0	49
2	Afghanistan	2013	1	59.9	268.0	66	0.01	73.219243	64.0	43
3	Afghanistan	2012	1	59.5	272.0	69	0.01	78.184215	67.0	278
4	Afghanistan	2011	1	59.2	275.0	71	0.01	7.097109	68.0	301

In [163]:
```
#Clean up the data
print(df.isna().sum())
print(df.shape)
```

```
Country                             0
Year                                0
Status                              0
Life expectancy                    10
Adult Mortality                    10
infant deaths                       0
Alcohol                           194
percentage expenditure              0
Hepatitis B                       553
Measles                             0
BMI                                34
under-five deaths                   0
Polio                              19
Total expenditure                 226
Diphtheria                         19
HIV/AIDS                            0
GDP                               448
Population                        652
thinness  1-19 years               34
thinness 5-9 years                 34
Income composition of resources   167
Schooling                         163
dtype: int64
(2938, 22)
```

In [164]:
```
#Treat the na's
#Replace using mean
for i in df.columns.drop('Country'):
    df[i].fillna(df[i].mean(),inplace =True)
```

In [165]:
```
df.head()
```

Out[165]:

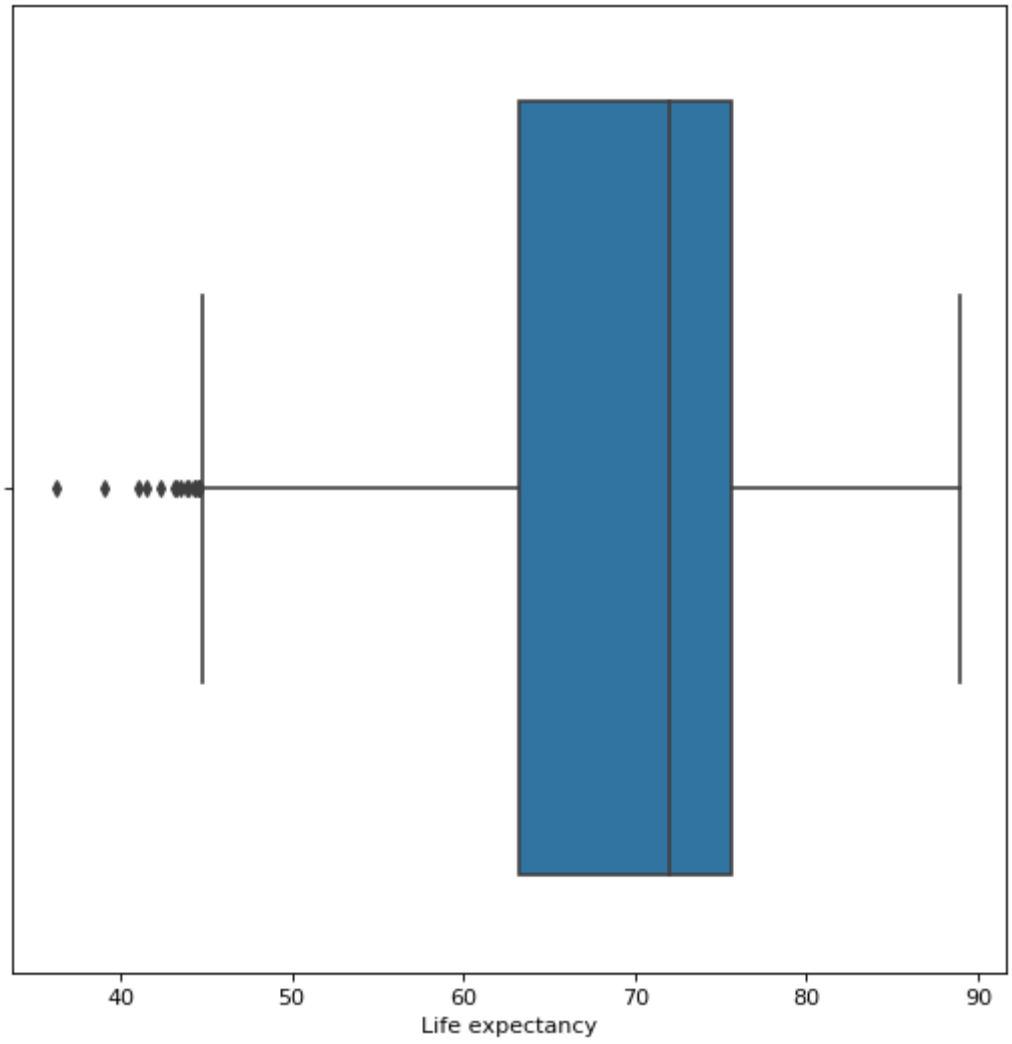
	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measle
0	Afghanistan	2015	1	65.0	263.0	62	0.01	71.279624	65.0	115
1	Afghanistan	2014	1	59.9	271.0	64	0.01	73.523582	62.0	49
2	Afghanistan	2013	1	59.9	268.0	66	0.01	73.219243	64.0	43
3	Afghanistan	2012	1	59.5	272.0	69	0.01	78.184215	67.0	278
4	Afghanistan	2011	1	59.2	275.0	71	0.01	7.097109	68.0	301

In [166]:
```
1 df.isna().sum()
```

Out[166]:
```
Country                            0
Year                               0
Status                             0
Life expectancy                    0
Adult Mortality                    0
infant deaths                      0
Alcohol                            0
percentage expenditure             0
Hepatitis B                        0
Measles                            0
BMI                                0
under-five deaths                  0
Polio                              0
Total expenditure                  0
Diphtheria                         0
HIV/AIDS                           0
GDP                                0
Population                         0
thinness  1-19 years               0
thinness 5-9 years                 0
Income composition of resources    0
Schooling                          0
dtype: int64
```

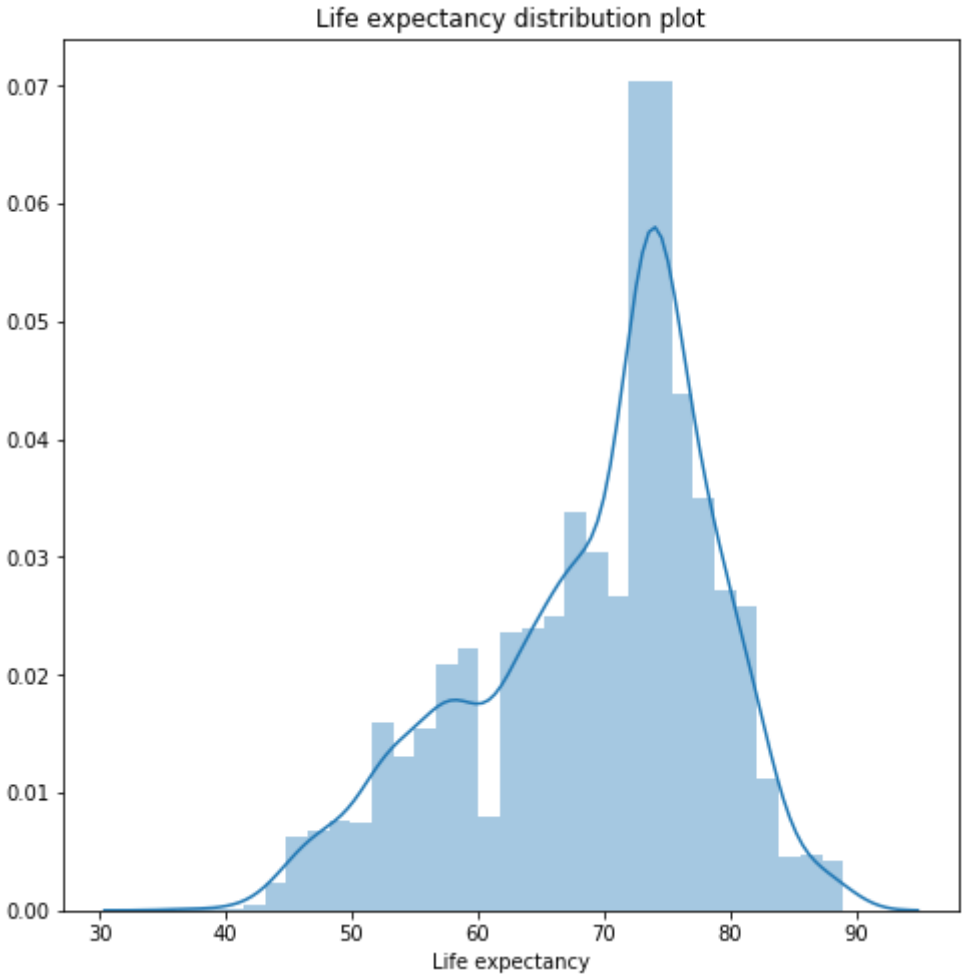
In [167]:
```
#Exploratory data analyses EDA
#Checking the distrubution of the 'Life expectancy'

plt.figure(figsize=(8,8), dpi=80)
sns.boxplot(df['Life expectancy'])
```

Out[167]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9d5c5813d0>



In [168]:
```
1 plt.figure(figsize=(8,8))
2 plt.title('Life expectancy distribution plot')
3 sns.distplot(df['Life expectancy'])
```

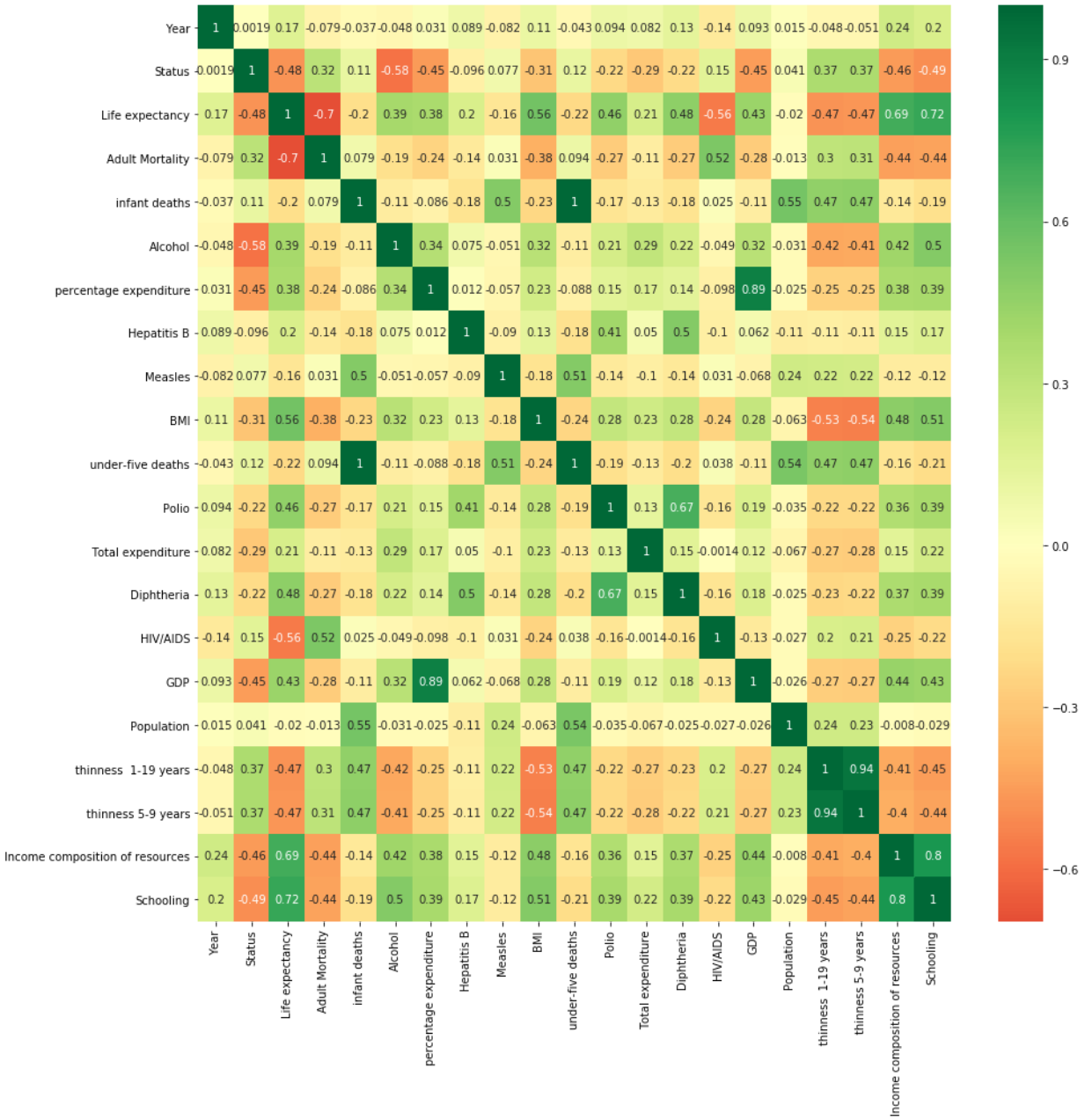
Out[168]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9d5c5fdad0>



In [169]:
```
num_col = df.select_dtypes(include=np.number).columns
print("Numerical columns: \n",num_col)

cat_col = df.select_dtypes(exclude=np.number).columns
print("Categorical columns: \n",cat_col)
```

```
Numerical columns:
 Index(['Year', 'Status', 'Life expectancy', 'Adult Mortality', 'infant d
eaths',
       'Alcohol', 'percentage expenditure', 'Hepatitis B', 'Measles', 'BM
I',
       'under-five deaths', 'Polio', 'Total expenditure', 'Diphtheria',
       'HIV/AIDS', 'GDP', 'Population', 'thinness  1-19 years',
       'thinness 5-9 years', 'Income composition of resources', 'Schoolin
g'],
      dtype='object')
Categorical columns:
 Index(['Country'], dtype='object')
```

In [170]:
```
#Check the multicollinearity
plt.figure(figsize=(15,15))
p=sns.heatmap(df[num_col].corr(), annot=True,cmap='RdYlGn',center=0)
```



In [171]:
```
#Model Building
x=df.drop(columns=['Life expectancy','Country'])
y=df[['Life expectancy']]
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.3,ra
```

In [172]:
```
#1) Adding 1 variable after 1
x_train1 = x_train['Income composition of resources']
```

In [173]:
```
#add a constant with stat model (sm)
x_train1 = sm.add_constant(x_train1)

model_1=sm.OLS(y_train,x_train1).fit()
```

```
/opt/anaconda3/lib/python3.7/site-packages/numpy/core/fromnumeric.py:249
5: FutureWarning: Method .ptp is deprecated and will be removed in a futu
re version. Use numpy.ptp instead.
  return ptp(axis=axis, out=out, **kwargs)
```

In [174]:
```
model_1.params
```

Out[174]:
```
const                              48.440947
Income composition of resources    33.059741
dtype: float64
```

In [175]:
```
1 print(model_1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Life expectancy   R-squared:                       0.490
Model:                            OLS   Adj. R-squared:                  0.490
Method:                 Least Squares   F-statistic:                     1974.
Date:                Sun, 06 Dec 2020   Prob (F-statistic):          1.09e-302
Time:                        01:03:43   Log-Likelihood:                -6894.3
No. Observations:                2056   AIC:                         1.379e+04
Df Residuals:                    2054   BIC:                         1.380e+04
Df Model:                           1
Covariance Type:            nonrobust
===============================================================================
                                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
const                           48.4409      0.492     98.412      0.000      47.476      49.406
Income composition of resources    33.0597      0.744     44.427      0.000      31.600      34.519
==============================================================================
Omnibus:                      138.959   Durbin-Watson:                   2.047
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              617.560
Skew:                           0.121   Prob(JB):                     7.92e-135
Kurtosis:                       5.674   Cond. No.                         6.86
==============================================================================

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

In [176]:
```
1 #The values on R-Squared and P-value are not enough for the model, lets
2 #it will change
3 #Variable2
4 x_train2=x_train[['Income composition of resources','Schooling']]
```

In [177]:
```
#add_constant
x_train2=sm.add_constant(x_train2)

#Create_second_model
model_2=sm.OLS(y_train,x_train2).fit()
```

In [178]:
```
model_2.params
```

Out[178]:
```
const                              43.145928
Income composition of resources    16.273079
Schooling                           1.320315
dtype: float64
```

In [179]:
```
1 print(model_2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Life expectancy   R-squared:                       0.562
Model:                            OLS   Adj. R-squared:                  0.561
Method:                 Least Squares   F-statistic:                     1316.
Date:                Sun, 06 Dec 2020   Prob (F-statistic):               0.00
Time:                        01:03:43   Log-Likelihood:                -6738.3
No. Observations:                2056   AIC:                         1.348e+04
Df Residuals:                    2053   BIC:                         1.350e+04
Df Model:                           2
Covariance Type:            nonrobust
===================================================================================
                                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------
const                              43.1459      0.540     79.895      0.000      42.087      44.205
Income composition of resources    16.2731      1.146     14.197      0.000      14.025      18.521
Schooling                           1.3203      0.072     18.340      0.000       1.179       1.461
==============================================================================
Omnibus:                      182.792   Durbin-Watson:                   2.037
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              596.381
Skew:                          -0.427   Prob(JB):                     3.14e-130
Kurtosis:                       5.497   Cond. No.                         101.
==============================================================================

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

In [180]:
```
1 #Model with 3 variables
2 x_train3=x_train[['Income composition of resources','Schooling','Adult
```

In [181]:

```
#add_constant
x_train3=sm.add_constant(x_train3)

#Create_second_model
model_3=sm.OLS(y_train,x_train3).fit()
```

In [182]:
```
1 print(model_3.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Life expectancy   R-squared:                       0.721
Model:                            OLS   Adj. R-squared:                  0.720
Method:                 Least Squares   F-statistic:                     1765.
Date:                Sun, 06 Dec 2020   Prob (F-statistic):               0.00
Time:                        01:03:43   Log-Likelihood:                -6275.3
No. Observations:                2056   AIC:                         1.256e+04
Df Residuals:                    2052   BIC:                         1.258e+04
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const                         56.2277      0.577     97.502      0.000      55.097      57.359
Income composition of resources    10.6375      0.930     11.438      0.000       8.814      12.461
Schooling                      1.0037      0.058     17.236      0.000       0.889       1.118
Adult Mortality               -0.0348      0.001    -34.168      0.000      -0.037      -0.033
==============================================================================
Omnibus:                      379.309   Durbin-Watson:                   1.962
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1628.478
Skew:                          -0.829   Prob(JB):                         0.00
Kurtosis:                       7.032   Cond. No.                     1.72e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.72e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [183]:
```
#RFE Approach
#Running RFE with important column count to be 15
lm=LinearRegression()
lm.fit(x_train,y_train)

rfe = RFE(lm,15)
rfe = rfe.fit(x_train,y_train)
```

```
/opt/anaconda3/lib/python3.7/site-packages/sklearn/utils/validation.py:72
4: DataConversionWarning: A column-vector y was passed when a 1d array wa
s expected. Please change the shape of y to (n_samples, ), for example us
ing ravel().
  y = column_or_1d(y, warn=True)
```

In [184]:
```
list(zip(x_train.columns,rfe.support_,rfe.ranking_))
```

Out[184]:
```
[('Year', False, 2),
 ('Status', True, 1),
 ('Adult Mortality', True, 1),
 ('infant deaths', True, 1),
 ('Alcohol', True, 1),
 ('percentage expenditure', False, 3),
 ('Hepatitis B', True, 1),
 ('Measles', False, 5),
 ('BMI', True, 1),
 ('under-five deaths', True, 1),
 ('Polio', True, 1),
 ('Total expenditure', True, 1),
 ('Diphtheria', True, 1),
 ('HIV/AIDS', True, 1),
 ('GDP', False, 4),
 ('Population', False, 6),
 ('thinness  1-19 years', True, 1),
 ('thinness 5-9 years', True, 1),
 ('Income composition of resources', True, 1),
 ('Schooling', True, 1)]
```

In [185]:
```
imp_columns=x_train.columns[rfe.support_]
imp_columns
```

Out[185]:
```
Index(['Status', 'Adult Mortality', 'infant deaths', 'Alcohol', 'Hepatiti
s B',
       'BMI', 'under-five deaths', 'Polio', 'Total expenditure', 'Diphthe
ria',
       'HIV/AIDS', 'thinness  1-19 years', 'thinness 5-9 years',
       'Income composition of resources', 'Schooling'],
      dtype='object')
```

High p-value High VIF : Drop the variable High p-value Low VIF : Drop the variable with high p-value first Low p-value Low VIF : accept the variable

In [186]:

```
#VIF = Variance Inflation Factor == Basic quantitative idea about how m
#with each other
# Creating X_train dataframe with RFE selected variables
x_train_rfe = x_train[imp_columns]
```

In [187]:
```
random.seed(0)

# Add a constant
x_train_rfec = sm.add_constant(x_train_rfe)

# Build the model with RFE features
lm_rfe = sm.OLS(y_train,x_train_rfec).fit()

#Summary of linear model
print(lm_rfe.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Life expectancy   R-squared:                       0.820
Model:                            OLS   Adj. R-squared:                  0.819
Method:                 Least Squares   F-statistic:                     620.1
Date:                Sun, 06 Dec 2020   Prob (F-statistic):               0.00
Time:                        01:03:43   Log-Likelihood:                -5823.0
No. Observations:                2056   AIC:                         1.168e+04
Df Residuals:                    2040   BIC:                         1.177e+04
Df Model:                          15
Covariance Type:            nonrobust
============================================================================
                                 coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------
const                           56.3507      0.815     69.162      0.000      54.753      57.949
Status                          -2.0823      0.313     -6.663      0.000      -2.695      -1.469
Adult Mortality                 -0.0200      0.001    -20.474      0.000      -0.022      -0.018
infant deaths                    0.0944      0.010      9.677      0.000       0.075       0.114
Alcohol                          0.0403      0.031      1.289      0.198      -0.021       0.102
Hepatitis B                     -0.0209      0.005     -4.303      0.000      -0.030      -0.011
BMI                              0.0488      0.006      7.963      0.000       0.037       0.061
under-five deaths               -0.0714      0.007     -9.983      0.000      -0.085      -0.057
Polio                            0.0272      0.005      4.962      0.000       0.016       0.038
Total expenditure                0.0768      0.042      1.835      0.067      -0.005       0.159
Diphtheria                       0.0456      0.006      7.807      0.000
```

```
00       0.034       0.057
HIV/AIDS                         -0.4970      0.024    -20.592      0.0
00      -0.544      -0.450
thinness  1-19 years             -0.0739      0.061     -1.212      0.2
26      -0.193       0.046
thinness 5-9 years                0.0032      0.060      0.054      0.9
57      -0.114       0.120
Income composition of resources   6.4836      0.769      8.435      0.0
00       4.976       7.991
Schooling                         0.6908      0.051     13.544      0.0
00       0.591       0.791
==============================================================================
Omnibus:                      110.684   Durbin-Watson:                   1.979
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              321.018
Skew:                          -0.238   Prob(JB):                     1.96e-70
Kurtosis:                       4.876   Cond. No.                     2.45e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.45e+03. This might indicate that there are
strong multicollinearity or other numerical problems.


/opt/anaconda3/lib/python3.7/site-packages/numpy/core/fromnumeric.py:2495: FutureWarning: Method .ptp is deprecated and will be removed in a future version. Use numpy.ptp instead.
  return ptp(axis=axis, out=out, **kwargs)
```

In [188]:
```
#thinness 5-9 years is too high, but before dropping lets check the VIF
#Create a dataframe that will contain the names of all the feature vari
vif = pd.DataFrame()
vif['Features'] = x_train_rfe.columns
vif['VIF'] = [variance_inflation_factor(x_train_rfe.values,i) for i in
vif['VIF'] = round(vif['VIF'],2)
vif = vif.sort_values(by='VIF',ascending = False)
vif
```

Out[188]:

	Features	VIF
6	under-five deaths	178.16
2	infant deaths	177.70
14	Schooling	44.59
13	Income composition of resources	30.42
9	Diphtheria	30.31
7	Polio	26.28
11	thinness 1-19 years	19.47
12	thinness 5-9 years	19.31
4	Hepatitis B	19.00
5	BMI	8.28
8	Total expenditure	7.74
0	Status	7.13
1	Adult Mortality	4.42
3	Alcohol	4.35
10	HIV/AIDS	1.70

In [189]:
```
#11 thinness 5-9 years 19.47 bu the P-value=0.957 (prioritize the P-val
```

In [190]:
```
#Dropping insignificant variables
x_train_rfe1= x_train_rfe.drop(['thinness 5-9 years'],1)

#Adding a constant variable and build a second fitted model

x_train_rfe1c = sm.add_constant(x_train_rfe1)
lm_rfe1 = sm.OLS(y_train,x_train_rfe1c).fit()

print(lm_rfe1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Life expectancy   R-squared:                       0.820
Model:                            OLS   Adj. R-squared:                  0.819
Method:                 Least Squares   F-statistic:                     664.7
Date:                Sun, 06 Dec 2020   Prob (F-statistic):               0.00
Time:                        01:03:43   Log-Likelihood:                -5823.0
No. Observations:                2056   AIC:                         1.168e+04
Df Residuals:                    2041   BIC:                         1.176e+04
Df Model:                          14
Covariance Type:            nonrobust
===========================================================================
                              coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------
const                       56.3537      0.813     69.336      0.000      54.760      57.948
Status                      -2.0819      0.312     -6.665      0.000      -2.695      -1.469
Adult Mortality             -0.0200      0.001    -20.484      0.000      -0.022      -0.018
infant deaths                0.0945      0.010      9.703      0.000       0.075       0.114
Alcohol                      0.0403      0.031      1.288      0.198      -0.021       0.102
Hepatitis B                 -0.0209      0.005     -4.304      0.000      -0.030      -0.011
BMI                          0.0487      0.006      8.014      0.000       0.037       0.061
under-five deaths           -0.0715      0.007    -10.003      0.000      -0.085      -0.057
Polio                        0.0272      0.005      4.963      0.000       0.016       0.038
Total expenditure            0.0767      0.042      1.835      0.067      -0.005       0.159
Diphtheria                   0.0456      0.006      7.811      0.000       0.034       0.057
```

```
HIV/AIDS                         -0.4969      0.024    -20.599      0.0
00      -0.544      -0.450
thinness  1-19 years             -0.0710      0.029     -2.427      0.0
15      -0.128      -0.014
Income composition of resources     6.4837      0.768      8.437      0.0
00       4.977       7.991
Schooling                        0.6909      0.051     13.548      0.0
00       0.591       0.791
==============================================================================
Omnibus:                      110.689   Durbin-Watson:                   1.979
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              320.883
Skew:                          -0.238   Prob(JB):                     2.09e-70
Kurtosis:                       4.876   Cond. No.                     2.45e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.45e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [191]:
```
#Total expenditur is not high but lets check the VIF
vif = pd.DataFrame()
vif['Features'] = x_train_rfe1.columns
vif['VIF'] = [variance_inflation_factor(x_train_rfe1.values,i) for i in
vif['VIF'] = round(vif['VIF'],2)
vif = vif.sort_values(by='VIF',ascending = False)
vif
```

Out[191]:

	Features	VIF
6	under-five deaths	177.82
2	infant deaths	177.16
13	Schooling	44.55
12	Income composition of resources	30.42
9	Diphtheria	30.30
7	Polio	26.28
4	Hepatitis B	18.99
5	BMI	8.19
8	Total expenditure	7.74
0	Status	7.10
1	Adult Mortality	4.41
3	Alcohol	4.35
11	thinness 1-19 years	4.07
10	HIV/AIDS	1.70

In [192]:
```
#P-values is not too much, so lets prioritizes the VIF in this case
#Lets drop the variable

# Dropping insignificant variables

x_train_rfe2 = x_train_rfe1.drop('under-five deaths', 1,)

# Adding a constant variable and Build a second fitted model

x_train_rfe2c = sm.add_constant(x_train_rfe2)
lm_rfe2 = sm.OLS(y_train, x_train_rfe2c).fit()

#Summary of linear model
print(lm_rfe2.summary())
```

```
                            OLS Regression Results
==============================================================================
=====
Dep. Variable:        Life expectancy   R-squared:
0.811
Model:                            OLS   Adj. R-squared:
0.810
Method:                 Least Squares   F-statistic:
675.4
Date:                Sun, 06 Dec 2020   Prob (F-statistic):
0.00
Time:                        01:03:44   Log-Likelihood:                -5
872.2
No. Observations:                2056   AIC:                         1.17
7e+04
Df Residuals:                    2042   BIC:                         1.18
5e+04
Df Model:                          13
Covariance Type:            nonrobust
==============================================================================
==========================
                                   coef    std err          t      P>|
t|      [0.025      0.975]
------------------------------------------------------------------------------
--------------------------
const                             55.1520      0.823     67.005      0.0
00      53.538      56.766
Status                            -2.0413      0.320     -6.382      0.0
00      -2.668      -1.414
Adult Mortality                   -0.0204      0.001    -20.368      0.0
00      -0.022      -0.018
infant deaths                     -0.0025      0.001     -2.797      0.0
05      -0.004      -0.001
Alcohol                           -0.0038      0.032     -0.121      0.9
04      -0.066       0.058
Hepatitis B                       -0.0243      0.005     -4.887      0.0
00      -0.034      -0.015
BMI                                0.0502      0.006      8.071      0.0
00       0.038       0.062
Polio                              0.0307      0.006      5.484      0.0
00       0.020       0.042
Total expenditure                  0.0825      0.043      1.928      0.0
```

```
54      -0.001       0.166
Diphtheria                          0.0536      0.006      9.039      0.0
00       0.042       0.065
HIV/AIDS                           -0.5147      0.025    -20.893      0.0
00      -0.563      -0.466
thinness  1-19 years               -0.0578      0.030     -1.932      0.0
53      -0.116       0.001
Income composition of resources     7.1638      0.784      9.140      0.0
00       5.627       8.701
Schooling                           0.7020      0.052     13.448      0.0
00       0.600       0.804
==============================================================================
Omnibus:                      110.170   Durbin-Watson:                   1.988
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              309.575
Skew:                          -0.250   Prob(JB):                     5.98e-68
Kurtosis:                       4.834   Cond. No.                     2.30e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.3e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [193]:

```
#now the alcoholl is to big for the P-value, so lets drop it too
#Lets drop the variable

#Dropping insignificant variables
x_train_rfe3= x_train_rfe2.drop(['Alcohol'],1)

#Adding a constant variable and build a second fitted model

x_train_rfe3c = sm.add_constant(x_train_rfe3)
lm_rfe3 = sm.OLS(y_train,x_train_rfe3c).fit()

print(lm_rfe3.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Life expectancy   R-squared:                       0.811
Model:                            OLS   Adj. R-squared:                  0.810
Method:                 Least Squares   F-statistic:                     732.0
Date:                Sun, 06 Dec 2020   Prob (F-statistic):               0.00
Time:                        01:03:44   Log-Likelihood:                -5872.2
No. Observations:                2056   AIC:                         1.177e+04
Df Residuals:                    2043   BIC:                         1.184e+04
Df Model:                          12
Covariance Type:            nonrobust
===========================================================================
                             coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------
const                     55.1373      0.814     67.751      0.000      53.541      56.733
Status                    -2.0259      0.293     -6.904      0.000      -2.601      -1.450
Adult Mortality           -0.0204      0.001    -20.439      0.000      -0.022      -0.018
infant deaths             -0.0025      0.001     -2.826      0.005      -0.004      -0.001
Hepatitis B               -0.0243      0.005     -4.887      0.000      -0.034      -0.015
BMI                        0.0502      0.006      8.073      0.000       0.038       0.062
Polio                      0.0307      0.006      5.484      0.000       0.020       0.042
Total expenditure          0.0819      0.042      1.928      0.054      -0.001       0.165
Diphtheria                 0.0536      0.006      9.043      0.000       0.042       0.065
HIV/AIDS                  -0.5149      0.025    -20.942      0.000
```

```
00      -0.563      -0.467
thinness  1-19 years               -0.0571      0.029     -1.949      0.0
51      -0.114       0.000
Income composition of resources     7.1646      0.784      9.144      0.0
00       5.628       8.701
Schooling                           0.7009      0.051     13.647      0.0
00       0.600       0.802
==============================================================================
=====
Omnibus:                      110.297   Durbin-Watson:
1.988
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              30
9.319
Skew:                          -0.251   Prob(JB):                     6.8
0e-68
Kurtosis:                       4.833   Cond. No.                     2.2
8e+03
==============================================================================
=====

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is co
rrectly specified.
[2] The condition number is large, 2.28e+03. This might indicate that the
re are
strong multicollinearity or other numerical problems.
```

In [194]:
```
vif = pd.DataFrame()
vif['Features'] = x_train_rfe3.columns
vif['VIF'] = [variance_inflation_factor(x_train_rfe3.values,i) for i in
vif['VIF'] = round(vif['VIF'],2)
vif = vif.sort_values(by='VIF',ascending = False)
vif
```

Out[194]:

	Features	VIF
11	Schooling	42.11
10	Income composition of resources	30.28
7	Diphtheria	29.80
5	Polio	26.16
3	Hepatitis B	18.73
4	BMI	8.18
6	Total expenditure	7.49
0	Status	6.05
1	Adult Mortality	4.30
9	thinness 1-19 years	3.96
8	HIV/AIDS	1.69
2	infant deaths	1.45

In [195]:

```
#Dropping insignificant variables
x_train_rfe4= x_train_rfe3.drop(['Schooling'],1)

#Adding a constant variable and build a second fitted model

x_train_rfe4c = sm.add_constant(x_train_rfe4)
lm_rfe4 = sm.OLS(y_train,x_train_rfe4c).fit()

print(lm_rfe4.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Life expectancy   R-squared:                       0.794
Model:                            OLS   Adj. R-squared:                  0.793
Method:                 Least Squares   F-statistic:                     716.7
Date:                Sun, 06 Dec 2020   Prob (F-statistic):               0.00
Time:                        01:03:44   Log-Likelihood:                -5961.9
No. Observations:                2056   AIC:                         1.195e+04
Df Residuals:                    2044   BIC:                         1.202e+04
Df Model:                          11
Covariance Type:            nonrobust
===========================================================================
                              coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------
const                      58.7499      0.804     73.101      0.000      57.174      60.326
Status                     -2.6868      0.302     -8.889      0.000      -3.280      -2.094
Adult Mortality            -0.0217      0.001    -20.948      0.000      -0.024      -0.020
infant deaths              -0.0029      0.001     -3.092      0.002      -0.005      -0.001
Hepatitis B                -0.0249      0.005     -4.801      0.000      -0.035      -0.015
BMI                         0.0626      0.006      9.740      0.000       0.050       0.075
Polio                       0.0360      0.006      6.188      0.000       0.025       0.047
Total expenditure           0.1155      0.044      2.608      0.009       0.029       0.202
Diphtheria                  0.0572      0.006      9.249      0.000       0.045       0.069
HIV/AIDS                   -0.4918      0.026    -19.200      0.000      -0.542      -0.442
thinness  1-19 years       -0.0830      0.031     -2.719      0.007
```

```
07      -0.143      -0.023
Income composition of resources    13.9929      0.630     22.220      0.0
00      12.758      15.228
==============================================================================
=====
Omnibus:                       88.663   Durbin-Watson:
2.009
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              26
2.468
Skew:                          -0.121   Prob(JB):                     1.0
1e-57
Kurtosis:                       4.734   Cond. No.                     2.2
7e+03
==============================================================================
=====

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is co
rrectly specified.
[2] The condition number is large, 2.27e+03. This might indicate that the
re are
strong multicollinearity or other numerical problems.
```

In [196]:

```
vif = pd.DataFrame()
vif['Features'] = x_train_rfe4.columns
vif['VIF'] = [variance_inflation_factor(x_train_rfe4.values,i) for i in
vif['VIF'] = round(vif['VIF'],2)
vif = vif.sort_values(by='VIF',ascending = False)
vif
```

Out[196]:

	Features	VIF
7	Diphtheria	29.69
5	Polio	25.80
3	Hepatitis B	18.59
10	Income composition of resources	13.90
4	BMI	7.83
6	Total expenditure	7.22
0	Status	6.05
1	Adult Mortality	4.30
9	thinness 1-19 years	3.95
8	HIV/AIDS	1.69
2	infant deaths	1.45

In [197]:
```
#Dropping insignificant variables
x_train_rfe5= x_train_rfe4.drop(['Diphtheria'],1)

#Adding a constant variable and build a second fitted model

x_train_rfe5c = sm.add_constant(x_train_rfe5)
lm_rfe5 = sm.OLS(y_train,x_train_rfe5c).fit()

print(lm_rfe5.summary())
```

```
                            OLS Regression Results
==============================================================================
=======
Dep. Variable:        Life expectancy   R-squared:
0.785
Model:                            OLS   Adj. R-squared:
0.784
Method:                 Least Squares   F-statistic:
748.8
Date:                Sun, 06 Dec 2020   Prob (F-statistic):
0.00
Time:                        01:03:44   Log-Likelihood:
-6004.1
No. Observations:                2056   AIC:                            1.
203e+04
Df Residuals:                    2045   BIC:                            1.
209e+04
Df Model:                          10
Covariance Type:            nonrobust
==============================================================================
============================
                                   coef    std err          t      P
>|t|      [0.025      0.975]
------------------------------------------------------------------------------
----------------------------
const                             59.1919      0.819     72.302
0.000      57.586      60.797
Status                            -2.6778      0.308     -8.682
0.000      -3.283      -2.073
Adult Mortality                   -0.0221      0.001    -20.928
0.000      -0.024      -0.020
infant deaths                     -0.0031      0.001     -3.256
0.001      -0.005      -0.001
Hepatitis B                       -0.0102      0.005     -2.018
0.044      -0.020      -0.000
BMI                                0.0645      0.007      9.838
0.000       0.052       0.077
Polio                              0.0649      0.005     12.954
0.000       0.055       0.075
Total expenditure                  0.1491      0.045      3.310
0.001       0.061       0.237
HIV/AIDS                          -0.4937      0.026    -18.889
0.000      -0.545      -0.442
thinness  1-19 years               -0.0823      0.031     -2.644
0.008      -0.143      -0.021
Income composition of resources   14.7697      0.637     23.190
```

```
0.000      13.521      16.019
==============================================================================
=======
Omnibus:                       92.191   Durbin-Watson:
2.014
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
273.609
Skew:                          -0.141   Prob(JB):
3.86e-60
Kurtosis:                       4.765   Cond. No.
2.17e+03
==============================================================================
=======

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
[2] The condition number is large, 2.17e+03. This might indicate that t
here are
strong multicollinearity or other numerical problems.
```

In [198]:

```
vif = pd.DataFrame()
vif['Features'] = x_train_rfe5.columns
vif['VIF'] = [variance_inflation_factor(x_train_rfe5.values,i) for i in
vif['VIF'] = round(vif['VIF'],2)
vif = vif.sort_values(by='VIF',ascending = False)
vif
```

Out[198]:

	Features	VIF
5	Polio	17.90
3	Hepatitis B	16.46
9	Income composition of resources	13.42
4	BMI	7.82
6	Total expenditure	7.12
0	Status	6.04
1	Adult Mortality	4.29
8	thinness 1-19 years	3.95
7	HIV/AIDS	1.69
2	infant deaths	1.45

In [201]:
```
#Dropping insignificant variables
x_train_rfe6= x_train_rfe5.drop(['Polio'],1)

#Adding a constant variable and build a second fitted model

x_train_rfe6c = sm.add_constant(x_train_rfe6)
lm_rfe6 = sm.OLS(y_train,x_train_rfe6c).fit()

print(lm_rfe6.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Life expectancy   R-squared:                       0.768
Model:                            OLS   Adj. R-squared:                  0.767
Method:                 Least Squares   F-statistic:                     752.1
Date:                Sun, 06 Dec 2020   Prob (F-statistic):               0.00
Time:                        01:04:17   Log-Likelihood:                -6085.2
No. Observations:                2056   AIC:                             1.219e+04
Df Residuals:                    2046   BIC:                             1.225e+04
Df Model:                           9
Covariance Type:            nonrobust
===========================================================================
                                  coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------
const                            61.5258      0.831     74.081      0.000      59.897      63.155
Status                           -2.8026      0.321     -8.742      0.000      -3.431      -2.174
Adult Mortality                  -0.0232      0.001    -21.164      0.000      -0.025      -0.021
infant deaths                    -0.0038      0.001     -3.849      0.000      -0.006      -0.002
Hepatitis B                       0.0135      0.005      2.766      0.006       0.004       0.023
BMI                               0.0703      0.007     10.336      0.000       0.057       0.084
Total expenditure                 0.1839      0.047      3.932      0.000       0.092       0.276
HIV/AIDS                         -0.4923      0.027    -18.113      0.000      -0.546      -0.439
thinness  1-19 years             -0.0744      0.032     -2.297      0.022      -0.138      -0.011
Income composition of resources  16.2560      0.652     24.951      0.000      14.978      17.534
==============================================================================
```

```
Omnibus:                      124.779   Durbin-Watson:
1.992
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              40
1.157
Skew:                          -0.247   Prob(JB):                     7.7
6e-88
Kurtosis:                       5.107   Cond. No.                     2.0
6e+03
==============================================================================
=====

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is co
rrectly specified.
[2] The condition number is large, 2.06e+03. This might indicate that the
re are
strong multicollinearity or other numerical problems.
```

In [202]:

```
vif = pd.DataFrame()
vif['Features'] = x_train_rfe6.columns
vif['VIF'] = [variance_inflation_factor(x_train_rfe6.values,i) for i in
vif['VIF'] = round(vif['VIF'],2)
vif = vif.sort_values(by='VIF',ascending = False)
vif
```

Out[202]:

	Features	VIF
3	Hepatitis B	12.86
8	Income composition of resources	11.81
4	BMI	7.70
5	Total expenditure	6.94
0	Status	6.00
1	Adult Mortality	4.29
7	thinness 1-19 years	3.92
6	HIV/AIDS	1.69
2	infant deaths	1.45

In [204]:
```
#Dropping insignificant variables
x_train_rfe7= x_train_rfe6.drop(['Hepatitis B'],1)

#Adding a constant variable and build a second fitted model

x_train_rfe7c = sm.add_constant(x_train_rfe7)
lm_rfe7 = sm.OLS(y_train,x_train_rfe7c).fit()

print(lm_rfe7.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        Life expectancy   R-squared:                       0.767
Model:                            OLS   Adj. R-squared:                  0.766
Method:                 Least Squares   F-statistic:                     842.4
Date:                Sun, 06 Dec 2020   Prob (F-statistic):               0.00
Time:                        01:05:56   Log-Likelihood:                -6089.0
No. Observations:                2056   AIC:                         1.220e+04
Df Residuals:                    2047   BIC:                         1.225e+04
Df Model:                           8
Covariance Type:            nonrobust
============================================================================
                                 coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------
const                           62.5312      0.748     83.601      0.000      61.064      63.998
Status                          -2.8044      0.321     -8.733      0.000      -3.434      -2.175
Adult Mortality                 -0.0233      0.001    -21.330      0.000      -0.025      -0.021
infant deaths                   -0.0043      0.001     -4.378      0.000      -0.006      -0.002
BMI                              0.0712      0.007     10.456      0.000       0.058       0.085
Total expenditure                0.1859      0.047      3.971      0.000       0.094       0.278
HIV/AIDS                        -0.4957      0.027    -18.225      0.000      -0.549      -0.442
thinness  1-19 years            -0.0701      0.032     -2.164      0.031      -0.134      -0.007
Income composition of resources 16.3814      0.651     25.164      0.000      15.105      17.658
==============================================================================
Omnibus:                      123.339   Durbin-Watson:                   1.995
```

```
Prob(Omnibus):                  0.000   Jarque-Bera (JB):                40
0.688
Skew:                          -0.237   Prob(JB):                      9.8
1e-88
Kurtosis:                       5.110   Cond. No.                      1.9
0e+03
==============================================================================
=====

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is co
rrectly specified.
[2] The condition number is large, 1.9e+03. This might indicate that ther
e are
strong multicollinearity or other numerical problems.
```

In [205]:

```
vif = pd.DataFrame()
vif['Features'] = x_train_rfe7.columns
vif['VIF'] = [variance_inflation_factor(x_train_rfe7.values,i) for i in
vif['VIF'] = round(vif['VIF'],2)
vif = vif.sort_values(by='VIF',ascending = False)
vif
```

Out[205]:

	Features	VIF
7	Income composition of resources	9.62
3	BMI	7.49
4	Total expenditure	6.53
0	Status	5.56
1	Adult Mortality	4.25
6	thinness 1-19 years	3.75
5	HIV/AIDS	1.68
2	infant deaths	1.41

In [208]:

```
#VIF under 10 is good enough
#Stepwise Regression
def stepwise_selection(x, y,
                       initial_list=[],
                       threshold_in=0.01,
                       threshold_out = 0.05,
                       verbose=True):
    """ Perform a forward-backward feature selection
    based on p-value from statsmodels.api.OLS
    Arguments:
        X - pandas.DataFrame with candidate features
        y - list-like with the target
        initial_list - list of features to start with (column names of
        threshold_in - include a feature if its p-value < threshold_in
        threshold_out - exclude a feature if its p-value > threshold_ou
        verbose - whether to print the sequence of inclusions and exclu
    Returns: list of selected features
    Always set threshold_in < threshold_out to avoid infinite looping.
    See https://en.wikipedia.org/wiki/Stepwise_regression for the detai
    """
    included = list(initial_list)
    while True:
        changed=False
        # forward step
        excluded = list(set(x.columns)-set(included))
        new_pval = pd.Series(index=excluded)
        for new_column in excluded:
            model = sm.OLS(y, sm.add_constant(pd.DataFrame(x[included+[
            new_pval[new_column] = model.pvalues[new_column]
        best_pval = new_pval.min()
        if best_pval < threshold_in:
            best_feature = new_pval.argmin()
            included.append(best_feature)
            changed=True
            if verbose:
                print('Add  {:30} with p-value {:.6}'.format(best_featu

        # backward step
        model = sm.OLS(y, sm.add_constant(pd.DataFrame(x[included]))).f
        # use all coefs except intercept
        pvalues = model.pvalues.iloc[1:]
        worst_pval = pvalues.max() # null if pvalues is empty
        if worst_pval > threshold_out:
            changed=True
            worst_feature = pvalues.argmax()
            included.remove(worst_feature)
            if verbose:
                print('Drop {:30} with p-value {:.6}'.format(worst_feat
        if not changed:
            break
    return included

result = stepwise_selection(x_train, y_train)

print('resulting features:')
print(result)
```

```
/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:32: Futu
reWarning:
The current behaviour of 'Series.argmin' is deprecated, use 'idxmin'
instead.
The behavior of 'argmin' will be corrected to return the positional
minimum in the future. For now, use 'series.values.argmin' or
'np.argmin(np.array(values))' to get the position of the minimum
row.

Add  Schooling                       with p-value 0.0
Add  Adult Mortality                 with p-value 2.94814e-217
Add  HIV/AIDS                        with p-value 8.85176e-80
Add  Diphtheria                      with p-value 6.63636e-50
Add  BMI                             with p-value 6.41342e-29
Add  Income composition of resources with p-value 6.43838e-22
Add  Status                          with p-value 1.13954e-15
Add  percentage expenditure          with p-value 9.00493e-08
Add  Polio                           with p-value 5.68587e-07
Add  Measles                         with p-value 8.01425e-06
Add  Hepatitis B                     with p-value 9.17377e-06
Add  under-five deaths               with p-value 0.00233237
Add  infant deaths                   with p-value 5.69409e-21
Add  thinness  1-19 years            with p-value 0.00227501
resulting features:
['Schooling', 'Adult Mortality', 'HIV/AIDS', 'Diphtheria', 'BMI', 'Income
composition of resources', 'Status', 'percentage expenditure', 'Polio',
'Measles', 'Hepatitis B', 'under-five deaths', 'infant deaths', 'thinness
1-19 years']
```

In [209]:
```
x_train_stepwise = x_train[['Schooling', 'Adult Mortality', 'HIV/AIDS',

# Adding a constant variable and Build a second fitted model

x_train_stepwise = sm.add_constant(x_train_stepwise)
lm_stepwise = sm.OLS(y_train, x_train_stepwise).fit()

#Summary of linear model
print(lm_stepwise.summary())
```

```
                            OLS Regression Results
==============================================================================
=====
Dep. Variable:        Life expectancy   R-squared:
0.823
Model:                            OLS   Adj. R-squared:
0.822
Method:                 Least Squares   F-statistic:
677.3
Date:                Sun, 06 Dec 2020   Prob (F-statistic):
0.00
Time:                        01:17:32   Log-Likelihood:                -5
807.2
No. Observations:                2056   AIC:                          1.16
4e+04
Df Residuals:                    2041   BIC:                          1.17
3e+04
Df Model:                          14
Covariance Type:            nonrobust
==============================================================================
==========================
                                   coef    std err          t      P>|
t|      [0.025      0.975]
------------------------------------------------------------------------------
--------------------------
const                             56.8039      0.756     75.122      0.0
00      55.321      58.287
Schooling                          0.6985      0.050     14.021      0.0
00       0.601       0.796
Adult Mortality                   -0.0197      0.001    -20.365      0.0
00      -0.022      -0.018
HIV/AIDS                          -0.4943      0.024    -20.731      0.0
00      -0.541      -0.448
Diphtheria                         0.0457      0.006      7.901      0.0
00       0.034       0.057
BMI                                0.0491      0.006      8.134      0.0
00       0.037       0.061
Income composition of resources    5.8765      0.763      7.704      0.0
00       4.380       7.372
Status                            -1.8727      0.291     -6.433      0.0
00      -2.444      -1.302
percentage expenditure             0.0003   5.07e-05      5.555      0.0
00       0.000       0.000
Polio                              0.0272      0.005      5.004      0.0
00       0.017       0.038
Measles                        -2.367e-05   9.35e-06     -2.530      0.0
```

```
11    -4.2e-05   -5.32e-06
Hepatitis B                      -0.0194      0.005     -4.010      0.0
00      -0.029      -0.010
under-five deaths                -0.0701      0.007     -9.950      0.0
00      -0.084      -0.056
infant deaths                     0.0940      0.010      9.797      0.0
00       0.075       0.113
thinness  1-19 years             -0.0868      0.028     -3.056      0.0
02      -0.143      -0.031
==============================================================================
=====
Omnibus:                      106.745   Durbin-Watson:
1.971
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              30
8.003
Skew:                          -0.225   Prob(JB):                     1.3
1e-67
Kurtosis:                       4.842   Cond. No.                     1.0
5e+05
==============================================================================
=====

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is co
rrectly specified.
[2] The condition number is large, 1.05e+05. This might indicate that the
re are
strong multicollinearity or other numerical problems.
```

In [212]:
```
x_test_stepwise = x_test[['Schooling', 'Adult Mortality', 'HIV/AIDS', '
x_test_stepwise = sm.add_constant(x_test_stepwise)
actual     = y_test["Life expectancy"]
prediction = lm_stepwise.predict(x_test_stepwise)
```

In [213]:
```
#Evaluation : MSE
model_mse = mean_squared_error(prediction,actual)
print(model_mse)
```

```
15.972714682410807
```

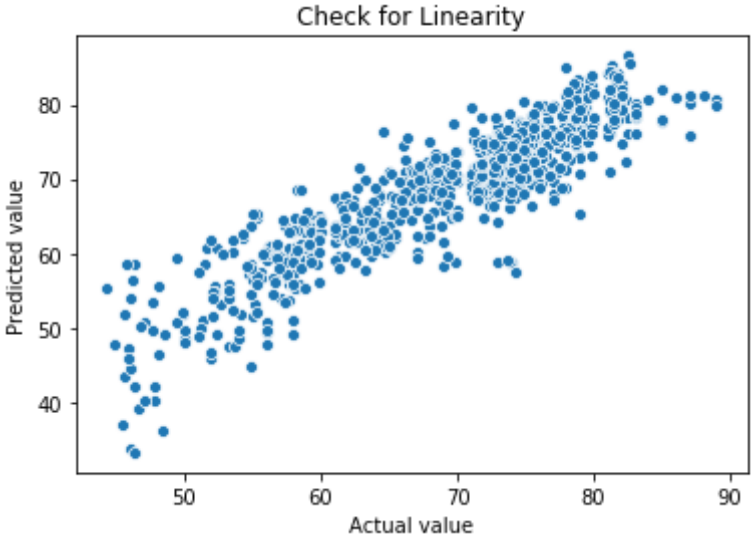
In [216]:
```
def mean_absolute_percentage_error(y_true,y_pred):
    y_true,y_pred = np.array(y_true),np.array(y_pred)
    return np.mean(np.abs((y_true-y_pred)/y_true))*100
```

In [218]:
```
mean_absolute_percentage_error(actual, prediction)
```

Out[218]:
```
4.558248666207347
```

In [219]:
```
# Check for Linearity
sns.scatterplot(y_test['Life expectancy'],prediction)
plt.title('Check for Linearity')
plt.xlabel('Actual value')
plt.ylabel('Predicted value')
```

Out[219]: Text(0, 0.5, 'Predicted value')



In [220]:
```
# Plot the histogram of the error terms
fig = plt.figure()
sns.distplot((y_test['Life expectancy'] - prediction), bins = 20)
fig.suptitle('Error Terms Analysis', fontsize = 20)
plt.xlabel('Errors', fontsize = 18)
```

Out[220]: Text(0.5, 0, 'Errors')