Life Expectancy by Linear Regression

Prepared by Aylin Aydın S018183

1- *Introduction*

Data science has been quickly developing and progressing in recent years, with the recognition that the knowledge collected from enormous databases is significant. The areas in which it is used have also expanded over the years. It is also used in the field of health for purposes such as data science and diagnosis, medical imaging. The challenge of determining the life expectancy factor in various nations is handled in this article using the WHO's (world health organization) dataset with linear regression approach.

Linear regression seeks to model the connection between two variables, dependent and independent, by fitting a linear equation to observable data. It's a type of supervised regression in which we try to predict a continuous value for a given data point by generalizing the data we already have. The linear component refers to the linear data generalization approach. The goal is to forecast the dependent variable (Y) based on a given independent variable (X). This may be performed by fitting the data using a best fit line. The best fit line or regression line is the one with the lowest sum of residual errors.

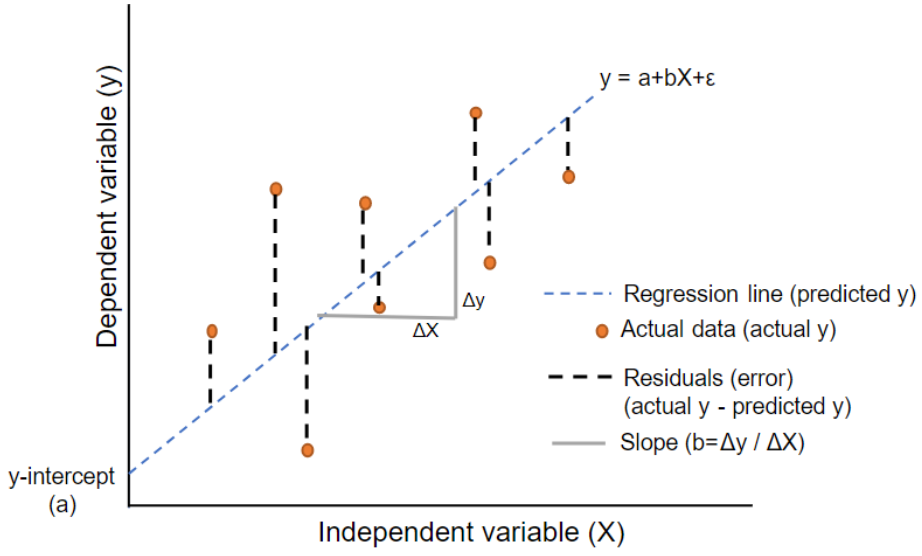


Figure 1. Linear Regression Visualization

2- *Methodology*

This study was prepared using numpy, pandas, matplotlib, seaborn and sklearn from python's existing libraries by using WHO's dataset in python software language and performing operations via Jupyter Notebook.

In this study, various dataset features were identified in order to recognize the dataset we had in the first place. The connections of these characteristics with life expectancy were investigated using 22 features from the supplied dataset, including the life expectancy factor. During this review, the features were divided according to whether they were categorical or numerical, and the distribution of each of them on the data was shown with the help of different graphs. The categorical data were then encoded and represented in numerical order to ensure a better result of our model and to prevent user errors. Then, in order to construct a usable data set for the challenge, it was determined whether the data set had any missing data. As a consequence of these tests, the properties with more than a hundred missing data points were filled using the property's average value to avoid reducing the data set we have. The remaining rows with empty data, which we did not fill with means, were removed from the data set in order for the models to work efficiently.

The Country column was removed from the dataset because it did not positively affect the functioning of the model during the processing of data, or even made operations more difficult, increasing its complexity After obtaining an available dataset with Feature engineering, the correlations of each of the given columns with life expectancy were examined and visualized. This process was done in order to create the most compact groups by selecting the most effective features at the modeling stage of the data we have.

These investigations resulted in the creation of sets containing at least eight characteristics. The data set was separated into two groups: train and test, with the train group performing five different models. With the test group, the performance of these models was also reported qualitatively and quantitatively. According to the score obtained by the models, the best model was determined, and the general formula was printed.

In the following chapters, it was also explained how the above-described methodology was implemented, the model coefficients of the five models were discussed, and how the use of these and similar ML algorithms may have benefits was discussed.

3- Implementation Details

Necessary libraries imported to Jupyter Notebook and, to standardize the randomization random seed was equalized to 147.

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import random
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
```

In [2]:
```
#setting the seed number for code reproducibility
np.random.seed(147)
random.seed(147)
```

Data is fetched in the notebook, and display setting is adjusted to two decimals to see the data clearer.

Loading the Dataset

In [3]:
```
#reading data
data=pd.read_csv("assignment-1-data.csv")
```

In [4]:
```
pd.set_option("display.precision", 2)
```

Data shaped is examined, there were 22 features includes the life expectancy and there were 2938 rows of data in the file.

In [5]:
```
data.shape
#there are 22 features in the dataset and, for each feature 2938 data is avaliable
```

Out[5]:
```
(2938, 22)
```

Data's info is gotten with the help of Pandas. The names of all the features, how many pieces of data are there and, data type of the features are as follows:

In [6]:
```
data.info()
```

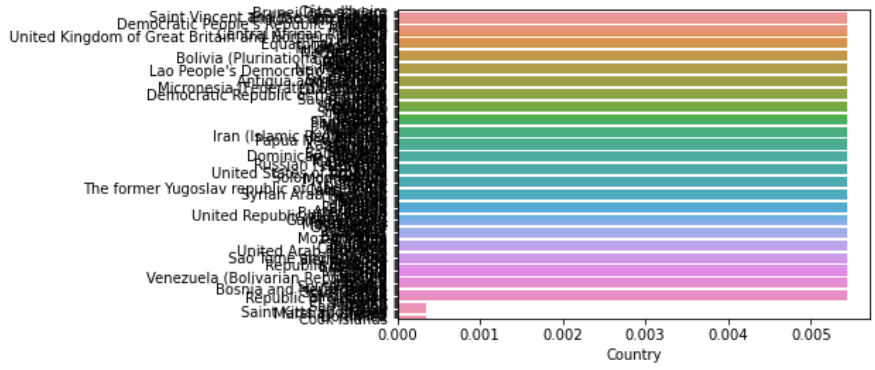
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 22 columns):
 #   Column                           Non-Null Count  Dtype
---  ------                           --------------  -----
 0   Country                          2938 non-null   object
 1   Year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

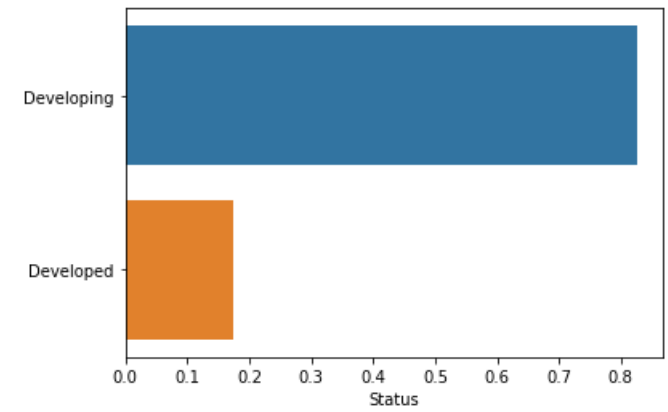
Categorical features are defined of the data set, because in order to work efficiently with categorical data, we need to encode it by assigning numeric values to the data.

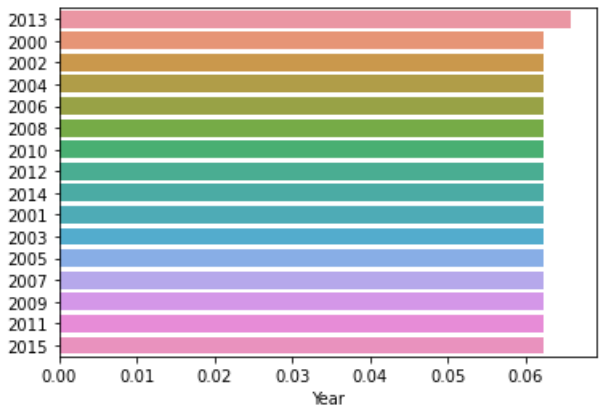
```
In [112]: #Selecting categorical columns
          categorical_columns = ['Country', 'Status','Year']
          data[categorical_columns].astype(object)
          data.info()
```

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

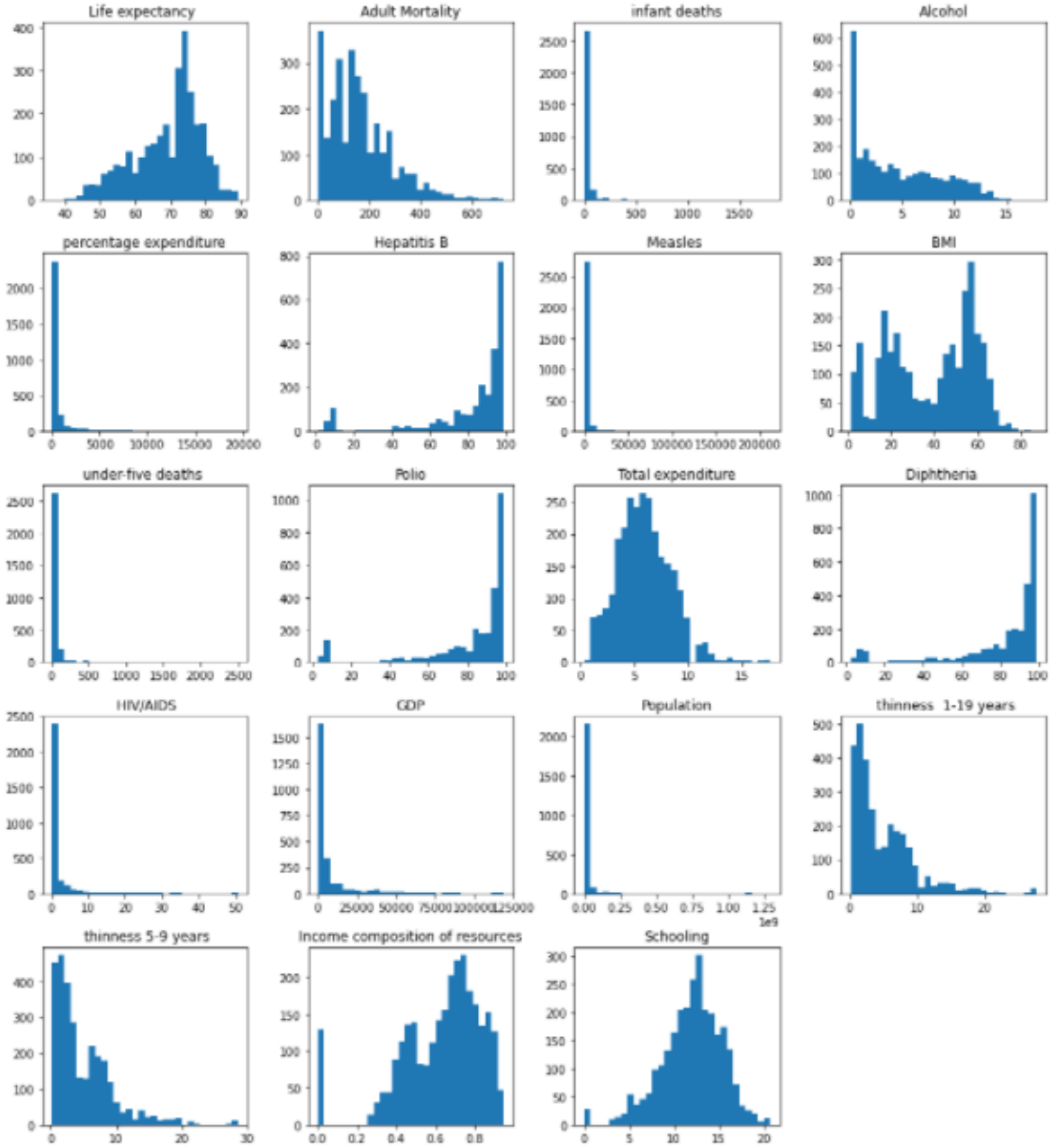
All categorical features are shown with barplots to visualize the data. After visualization, it was observed that the country column is very complex, so it was thought that the models that will be created in the future will have a bad effect on its performance. For this reason, it was decided to delete it when dealing with lost data.







After categorical characteristics, a single histogram of each numerical characteristic contained in the data set was printed in order to see the data as a visual whole.



According to the data visualized above, if we try to predict the factors that most affect life expectancy without making any calculations yet, it can be considered that the data that are distributed in a similar or exactly opposite direction to life expectation are the data that most criticize it. In this context, there are similar “Income composition of resources”, ”BMI” in order, especially "Schooling" and “Total expenditure”, “Adult mortality” , “thinness 1-19 years” exactly opposite direction.

	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	under-five deaths	Polio	Total expenditure	Diphtheria	HIV/AIDS
count	2928.00	2928.00	2938.00	2744.00	2938.00	2385.00	2938.00	2904.00	2938.00	2919.00	2712.00	2919.00	2938.00
mean	69.22	164.80	30.30	4.60	738.25	80.94	2419.59	38.32	42.04	82.55	5.94	82.32	1.74
std	9.52	124.29	117.93	4.05	1987.91	25.07	11467.27	20.04	160.45	23.43	2.50	23.72	5.08
min	36.30	1.00	0.00	0.01	0.00	1.00	0.00	1.00	0.00	3.00	0.37	2.00	0.10
25%	63.10	74.00	0.00	0.88	4.69	77.00	0.00	19.30	0.00	78.00	4.26	78.00	0.10
50%	72.10	144.00	3.00	3.75	64.91	92.00	17.00	43.50	4.00	93.00	5.75	93.00	0.10
75%	75.70	228.00	22.00	7.70	441.53	97.00	360.25	56.20	28.00	97.00	7.49	97.00	0.80
max	89.00	723.00	1800.00	17.87	19479.91	99.00	212183.00	87.30	2500.00	99.00	17.60	99.00	50.60

GDP	Population	thinness 1-19 years	thinness 5-9 years	Income composition of resources	Schooling
2490.00	2.29e+03	2904.00	2904.00	2771.00	2775.00
7483.16	1.28e+07	4.84	4.87	0.63	11.99
14270.17	6.10e+07	4.42	4.51	0.21	3.36
1.68	3.40e+01	0.10	0.10	0.00	0.00
463.94	1.96e+05	1.60	1.50	0.49	10.10
1766.95	1.39e+06	3.30	3.30	0.68	12.30
5910.81	7.42e+06	7.20	7.20	0.78	14.30
119172.74	1.29e+09	27.70	28.60	0.95	20.70

All statistical information about the characteristics of the data is located above, and its continuation is also on the side. This information will then be able to be considered when choosing features for modeling.

After analyzing the data, visualizing it and getting more information about the data, preprocessing operations were performed to make the data suitable for the model before modeling.

Before this process, the data was copied to be cleaned up in order to avoid losing the data in the data during the process.

```
#Copy the original dataframe before changing operations
df = data.copy()
```

During the preprocessing stage, actions were carried out in order to improve the future analysis of categorical column data. The first of them, the 'Country' column, has been eliminated from the data set since it adds complexity and does not help efficiency at the same time. Second, 'Status' and 'Year' were combined into a single hot dummy vector to make processing easier and produce more efficient results.

```
In [21]: df.drop(["Country"], axis=1,inplace=True)
```

```
In [22]: df['Status'] = df['Status'].map({'Developed': 1, 'Developing': 0})
         df
```

Out[22]:

	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	...	Polio	Total expenditure	Diphtheria	HIV/AIDS	GDP	Populati
0	2015	0	65.0	263.0	62	0.01	71.28	65.0	1154	19.1	...	6.0	8.16	65.0	0.1	584.26	3.37e+
1	2014	0	59.9	271.0	64	0.01	73.52	62.0	492	18.6	...	58.0	8.18	62.0	0.1	612.70	3.28e+
2	2013	0	59.9	268.0	66	0.01	73.22	64.0	430	18.1	...	62.0	8.13	64.0	0.1	631.74	3.17e+
3	2012	0	59.5	272.0	69	0.01	78.18	67.0	2787	17.6	...	67.0	8.52	67.0	0.1	669.96	3.70e+
4	2011	0	59.2	275.0	71	0.01	7.10	68.0	3013	17.2	...	68.0	7.87	68.0	0.1	63.54	2.98e+
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2933	2004	0	44.3	723.0	27	4.36	0.00	68.0	31	27.1	...	67.0	7.13	65.0	33.6	454.37	1.28e+
2934	2003	0	44.5	715.0	26	4.06	0.00	7.0	998	26.7	...	7.0	6.52	68.0	36.7	453.35	1.26e+
2935	2002	0	44.8	73.0	25	4.43	0.00	73.0	304	26.3	...	73.0	6.53	71.0	39.8	57.35	1.26e+
2936	2001	0	45.3	686.0	25	1.72	0.00	76.0	529	25.9	...	76.0	6.16	75.0	42.1	548.59	1.24e+
2937	2000	0	46.0	665.0	24	1.68	0.00	79.0	1483	25.5	...	78.0	7.10	78.0	43.5	547.36	1.22e+

2938 rows × 21 columns

```
In [23]: df[["Year"]]=df[["Year"]].astype(object)
         encodedDf=pd.get_dummies(df[["Year"]])
         encodedDf
```

Out[23]:

	Year_2000	Year_2001	Year_2002	Year_2003	Year_2004	Year_2005	Year_2006	Year_2007	Year_2008	Year_2009	Year_2010	Year_2011	Year_2012	Year
0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	0	0	1	
4	0	0	0	0	0	0	0	0	0	0	0	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2933	0	0	0	0	1	0	0	0	0	0	0	0	0	
2934	0	0	0	1	0	0	0	0	0	0	0	0	0	
2935	0	0	1	0	0	0	0	0	0	0	0	0	0	
2936	0	1	0	0	0	0	0	0	0	0	0	0	0	
2937	1	0	0	0	0	0	0	0	0	0	0	0	0	

2938 rows × 16 columns

The missing data was then identified in order to go on to the sequence modeling step. The amount of missing data in several columns was found to be excessive. As a result, if there are over a hundred data losses, the empty data in this column is filled in using the column's average value.

Handling with Missing Values

```
In [25]: df.isnull().sum().sort_values(ascending=False)
```

```
Out[25]: Population                       652
         Hepatitis B                      553
         GDP                              448
         Total expenditure                226
         Alcohol                          194
         Income composition of resources  167
         Schooling                        163
          thinness 5-9 years               34
          thinness  1-19 years              34
          BMI                              34
         Polio                             19
         Diphtheria                        19
         Life expectancy                   10
         Adult Mortality                   10
         Year                               0
          HIV/AIDS                          0
         Status                            0
         Measles                           0
         percentage expenditure            0
         infant deaths                     0
         under-five deaths                 0
         dtype: int64
```

dtype: int64

In [26]:
```
df["Population"]=df["Population"].fillna(df["Population"].mean())
df["Hepatitis B"]=df["Hepatitis B"].fillna(df["Hepatitis B"].mean())
df["GDP"]=df["GDP"].fillna(df["GDP"].mean())
df["Total expenditure"]=df["Total expenditure"].fillna(df["Total expenditure"].mean())
df["Alcohol"]=df["Alcohol"].fillna(df["Alcohol"].mean())
df["Income composition of resources"]=df["Income composition of resources"].fillna(df["Income composition of resources"].mean())
df["Schooling"]=df["Schooling"].fillna(df["Schooling"].mean())
df
```

Out[26]:

Life ectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	...	Polio	Total expenditure	Diphtheria	HIV/AIDS	GDP	Population	thinness 1-19 years	thinness 5-9 years	c
65.0	263.0	62	0.01	71.28	65.0	1154	19.1	...	6.0	8.16	65.0	0.1	584.26	3.37e+07	17.2	17.3	
59.9	271.0	64	0.01	73.52	62.0	492	18.6	...	58.0	8.18	62.0	0.1	612.70	3.28e+05	17.5	17.5	
59.9	268.0	66	0.01	73.22	64.0	430	18.1	...	62.0	8.13	64.0	0.1	631.74	3.17e+07	17.7	17.7	
59.5	272.0	69	0.01	78.18	67.0	2787	17.6	...	67.0	8.52	67.0	0.1	669.96	3.70e+06	17.9	18.0	
59.2	275.0	71	0.01	7.10	68.0	3013	17.2	...	68.0	7.87	68.0	0.1	63.54	2.98e+06	18.2	18.2	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
44.3	723.0	27	4.36	0.00	68.0	31	27.1	...	67.0	7.13	65.0	33.6	454.37	1.28e+07	9.4	9.4	
44.5	715.0	26	4.06	0.00	7.0	998	26.7	...	7.0	6.52	68.0	36.7	453.35	1.26e+07	9.8	9.9	
44.8	73.0	25	4.43	0.00	73.0	304	26.3	...	73.0	6.53	71.0	39.8	57.35	1.26e+05	1.2	1.3	
45.3	686.0	25	1.72	0.00	76.0	529	25.9	...	76.0	6.16	75.0	42.1	548.59	1.24e+07	1.6	1.7	
46.0	665.0	24	1.68	0.00	79.0	1483	25.5	...	78.0	7.10	78.0	43.5	547.36	1.22e+07	11.0	11.2	

Again, the empty data in the data set was checked and the columns with empty data were removed from the data set. As a result of this process, no empty data was found in the data set, and we have 2888 lines of data left.

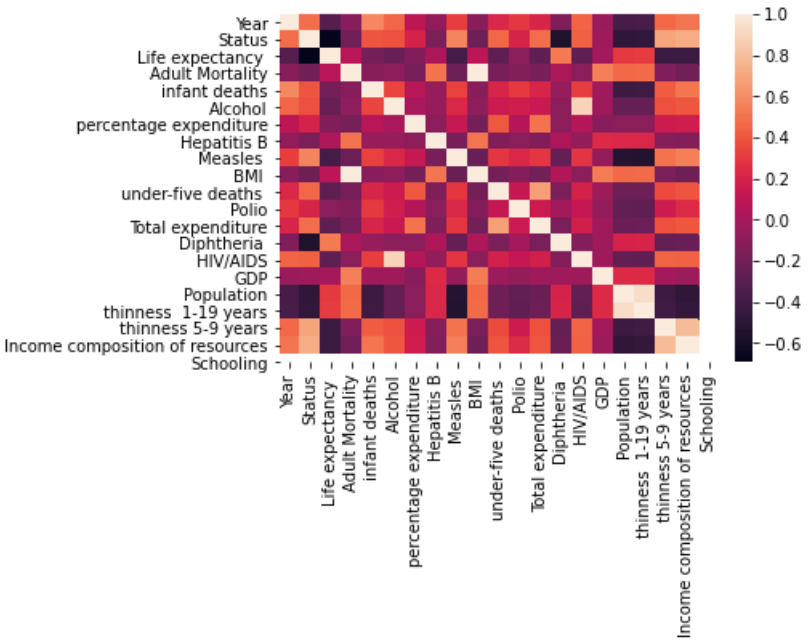
In [27]:
```
df.isnull().sum().sort_values(ascending=False)
```

Out[27]:
```
 thinness 5-9 years                34
 thinness  1-19 years              34
 BMI                               34
Diphtheria                         19
Polio                              19
Life expectancy                    10
Adult Mortality                    10
Year                                0
Income composition of resources     0
Population                          0
GDP                                 0
 HIV/AIDS                           0
under-five deaths                   0
Total expenditure                   0
Status                              0
Measles                             0
Hepatitis B                         0
percentage expenditure              0
Alcohol                             0
infant deaths                       0
Schooling                           0
dtype: int64
```

In [28]:
```
df.dropna(inplace=True) #remove the rows missing values
```

After this process, our data set was made ready for modeling. But in order for the models to have a high score, this correlation was examined, since the selected characteristics should have a high correlation between them and the life expectancy.

A scatter diagram was printed for each characteristic to evaluate the correlations between it and life expectancy. Then, to acquire more numerical data, the decorrelation between them was computed, and a heatmap was printed to display it.



The attributes listed below are listed in order of their link with life expectancy, in declining order. To be able to pick from this list, a threshold of 0.40 was set, and attributes that were higher than this number were listed. When constructing models, these tables were used.

```
Correlation with the target variable

: Schooling                          0.73
Adult Mortality                     0.69
Income composition of resources     0.69
 BMI                                0.57
 HIV/AIDS                           0.56
 thinness  1-19 years               0.48
Status                              0.48
Diphtheria                          0.48
 thinness 5-9 years                 0.47
Polio                               0.46
GDP                                 0.43
Alcohol                             0.39
percentage expenditure              0.38
under-five deaths                   0.22
Total expenditure                   0.21
Hepatitis B                         0.21
infant deaths                       0.20
Year                                0.17
Measles                             0.16
Population                          0.02
Name: Life expectancy , dtype: float64
```

```
Schooling                          0.73
Adult Mortality                    0.69
Income composition of resources    0.69
 BMI                               0.57
 HIV/AIDS                          0.56
 thinness  1-19 years              0.48
Status                             0.48
Diphtheria                         0.48
 thinness 5-9 years                0.47
Polio                              0.46
GDP                                0.43
Name: Life expectancy , dtype: float64
```

Figure 2. Correlations between Life expectancy and other features

Finally, 5 models were developed to predict with linear regression, and at least 8 characteristics were chosen for each model. Following that, the models were trained by splitting the data sets by 80 percent to 20 percent into train and test data sets. As a consequence, the scores and errors of several types of models were calculated. Graphs were created to depict the projected and actual data. The model formula's were extracted. As a consequence of all of the procedures, 5 models were compared to one another and explained in depth in the following section.

To develop Model 1, all values whose association with life expectancy is larger than a threshold value were selected, and train and test data sets were produced

Model 1 => with higher correlations than 0.40

In [39]:
```
X1=df[["Schooling","Adult Mortality","Income composition of resources"," BMI "," HIV/AIDS"," thinness  1-19 years","Status","Diph
X1
```

Out[39]:

	Schooling	Adult Mortality	Income composition of resources	BMI	HIV/AIDS	thinness 1-19 years	Status	Diphtheria	thinness 5-9 years	Polio	GDP
0	10.1	263.0	0.48	19.1	0.1	17.2	0	65.0	17.3	6.0	584.26
1	10.0	271.0	0.48	18.6	0.1	17.5	0	62.0	17.5	58.0	612.70
2	9.9	268.0	0.47	18.1	0.1	17.7	0	64.0	17.7	62.0	631.74
3	9.8	272.0	0.46	17.6	0.1	17.9	0	67.0	18.0	67.0	669.96
4	9.5	275.0	0.45	17.2	0.1	18.2	0	68.0	18.2	68.0	63.54
...	...	...	...	...	...	...	...	...	...	...	...
2933	9.2	723.0	0.41	27.1	33.6	9.4	0	65.0	9.4	67.0	454.37
2934	9.5	715.0	0.42	26.7	36.7	9.8	0	68.0	9.9	7.0	453.35
2935	10.0	73.0	0.43	26.3	39.8	1.2	0	71.0	1.3	73.0	57.35
2936	9.8	686.0	0.43	25.9	42.1	1.6	0	75.0	1.7	76.0	548.59
2937	9.8	665.0	0.43	25.5	43.5	11.0	0	78.0	11.2	78.0	547.36

2888 rows × 11 columns

For Model 2, In addition to the attributes that we filled up using the average of the column, "Adult mortality" was included in place of missing data due to its substantial association with life expectancy.

Model 2

In [54]:
```
X2=df[["Schooling","Population","Hepatitis B","Income composition of resources","Total expenditure","Alcohol","GDP","Adult Mortal
X2
```

Out[54]:

	Schooling	Population	Hepatitis B	Income composition of resources	Total expenditure	Alcohol	GDP	Adult Mortality
0	10.1	3.37e+07	65.0	0.48	8.16	0.01	584.26	263.0
1	10.0	3.28e+05	62.0	0.48	8.18	0.01	612.70	271.0
2	9.9	3.17e+07	64.0	0.47	8.13	0.01	631.74	268.0
3	9.8	3.70e+06	67.0	0.46	8.52	0.01	669.96	272.0
4	9.5	2.98e+06	68.0	0.45	7.87	0.01	63.54	275.0
...	...	...	...	...	...	...	...	...
2933	9.2	1.28e+07	68.0	0.41	7.13	4.36	454.37	723.0
2934	9.5	1.26e+07	7.0	0.42	6.52	4.06	453.35	715.0
2935	10.0	1.26e+05	73.0	0.43	6.53	4.43	57.35	73.0
2936	9.8	1.24e+07	76.0	0.43	6.16	1.72	548.59	686.0
2937	9.8	1.22e+07	79.0	0.43	7.10	1.68	547.36	665.0

2888 rows × 8 columns

For Model 3 features with the highest correlation value of 8 were selected.

Model 3 with highest 8 correlation

In [115]:
```
X3=df[["Schooling","Adult Mortality","Income composition of resources"," BMI "," HIV/AIDS"," thinness  1-19 years","Status","Diph
X3
```

Out[115]:

	Schooling	Adult Mortality	Income composition of resources	BMI	HIV/AIDS	thinness 1-19 years	Status	Diphtheria
0	10.1	263.0	0.48	19.1	0.1	17.2	0	65.0
1	10.0	271.0	0.48	18.6	0.1	17.5	0	62.0
2	9.9	268.0	0.47	18.1	0.1	17.7	0	64.0
3	9.8	272.0	0.46	17.6	0.1	17.9	0	67.0
4	9.5	275.0	0.45	17.2	0.1	18.2	0	68.0
...	...	...	...	...	...	...	...	...
2933	9.2	723.0	0.41	27.1	33.6	9.4	0	65.0
2934	9.5	715.0	0.42	26.7	36.7	9.8	0	68.0
2935	10.0	73.0	0.43	26.3	39.8	1.2	0	71.0
2936	9.8	686.0	0.43	25.9	42.1	1.6	0	75.0
2937	9.8	665.0	0.43	25.5	43.5	11.0	0	78.0

2888 rows × 8 columns

The model was tested to see how it would have altered if we hadn't gotten the three most associated characteristics with Model 4 and instead obtained the other eight high correlated features in order.

Model 4

In [131]:
```
X4=df[[" BMI "," HIV/AIDS"," thinness  1-19 years","Status","Diphtheria "," thinness 5-9 years","Polio","GDP"]]
X4
```

Out[131]:

	BMI	HIV/AIDS	thinness 1-19 years	Status	Diphtheria	thinness 5-9 years	Polio	GDP
0	19.1	0.1	17.2	0	65.0	17.3	6.0	584.26
1	18.6	0.1	17.5	0	62.0	17.5	58.0	612.70
2	18.1	0.1	17.7	0	64.0	17.7	62.0	631.74
3	17.6	0.1	17.9	0	67.0	18.0	67.0	669.96
4	17.2	0.1	18.2	0	68.0	18.2	68.0	63.54
...	...	...	...	...	...	...	...	...
2933	27.1	33.6	9.4	0	65.0	9.4	67.0	454.37
2934	26.7	36.7	9.8	0	68.0	9.9	7.0	453.35
2935	26.3	39.8	1.2	0	71.0	1.3	73.0	57.35
2936	25.9	42.1	1.6	0	75.0	1.7	76.0	548.59
2937	25.5	43.5	11.0	0	78.0	11.2	78.0	547.36

2888 rows × 8 columns

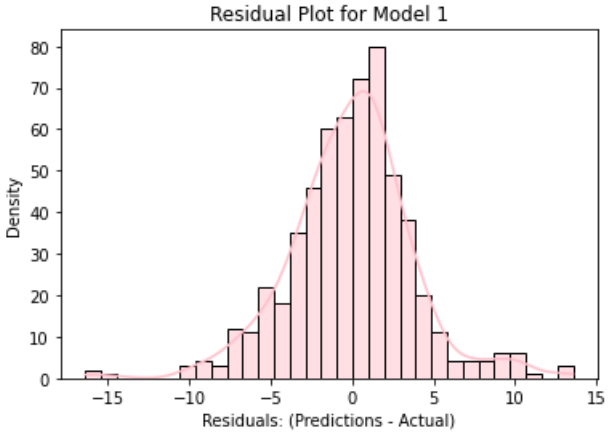
In our latest model, Model 5, we examined how to get a result if all the features were taken

```
(['Year', 'Status', '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', 'Schoo
ling',dtype='object')
```

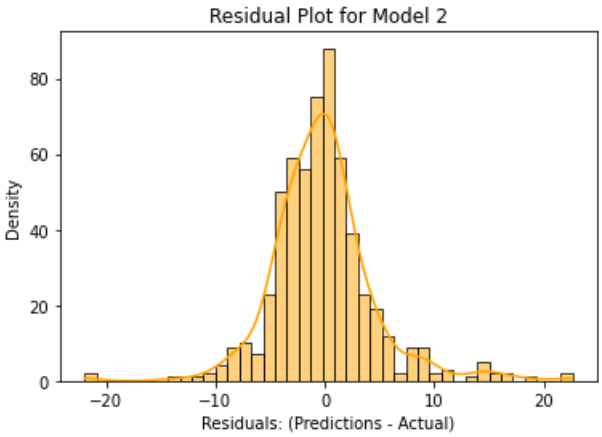
After the test and train data were created, independent variables were scaled separately for each model to get a more accurate model. Thanks to this process, the dec between dependent variable and independent variable became more understandable for the model. It was merged with the train data sets prepared in order to do linear regression, and the model was asked to estimate the independent variable value of the test data sets, which it had never seen before. The score, which represents the model's accuracy rate, was generated based on these findings.

4- Results

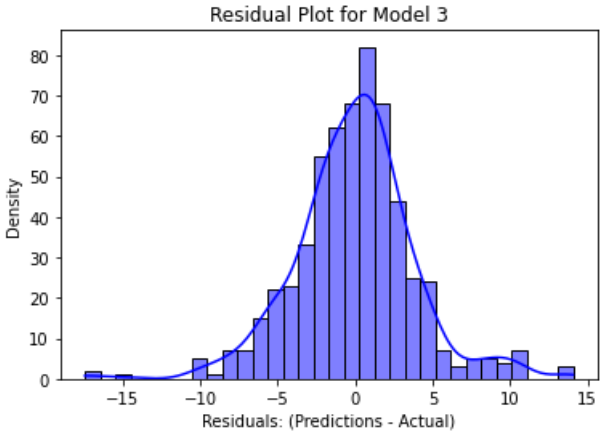
The individual residual graphs for each model are shown below. For the linear regression model, the residual graph depicts the distribution of discrepancies between the estimated and real values. The fact that the histogram is high around zero and its horizontal axis is positioned in a shorter range, according to this graph, indicates how accurate the model's predictions are.



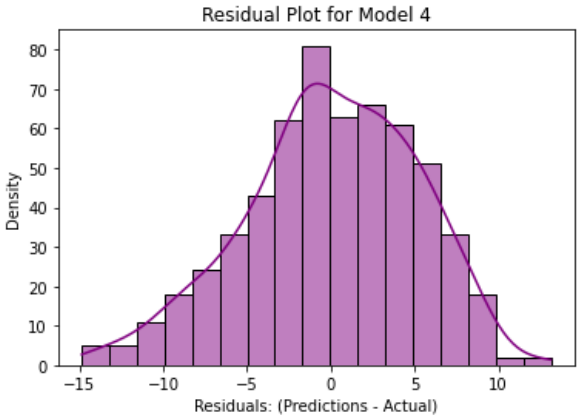
When we look at the residual graph for Model 1, we can see that the residual values are centered around zero and peak; in this situation, we can state that our model performs well when viewed separately.



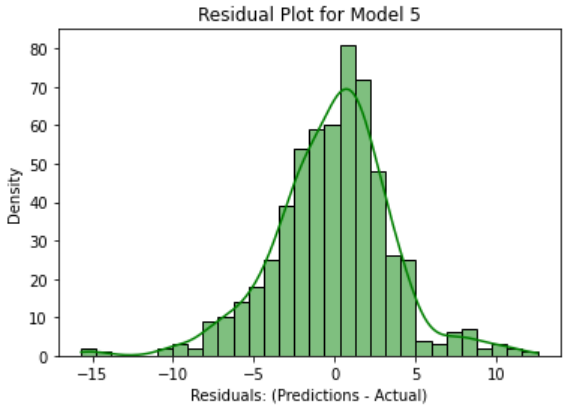
When we look at Model 2, we can see that it has similar features to Model 1, but the horizontal plane range is greater, and the disparity between the histogram and the distribution line is greater.



There is essentially no difference between Model 1 and Model 3 when it comes to Model 3. At the model1 level, the Model 3 performed admirably.



We see that the model4 graph is distributed much more equally than other graphs, in other words it hasn't peaked at zero and isn't centered on it. At this point, it's safe to assume that the discrepancy between estimated and real values might be anywhere in the range, and the model isn't very good.



Finally, while looking at the graph of Model 5, it can be seen that it has a similar distribution to Model 1 and Model 3, with the exception that the peak point is somewhat more specific.

After the models are plotted and examined one by one, it can be argued which model is better than the other with the help of numerical value. As a result, the intersection, score, mean square error, root mean square error, r square errors of all the models were listed below one by one.

	MODEL 1	MODEL 2	MODEL 3	MODEL 4	MODEL 5
SCORE	0.8358	0.7496	0.83	0.7154	0.8499
INTERCEPT	69.478	69.478	69.478	69.291	69.478
MeanSquaredError	14.451	22.037	14.96	26.099	13.209
RootMeanSquaredError	3.8014	4.6944	3.8678	5.1087	3.6344
MeanAbsoluteError	2.8025	3.2364	2.829	4.1113	2.7034
r2_score	0.8358	0.7496	0.83	0.7154	0.8499

The intersection values given in the table above give the predicted mean value of Y for all X = 0. When we look at the intercept values of the models, we notice that the majority of them are quite near to the average value of life expectancy in the data set, which is 69.22.

The mean squared error indicates the proximity of a regression line to a number of points, so we can say that the model with the lowest mean squared error is much better than the models. At this point, although the errors of the model 1 and model 3 are low, the lowest belongs to the model 5. For this reason, we can say that the model 5 is once again the best model.

The standard deviation of the residuals may alternatively be defined as the root mean squared error, which illustrates how much of the data is clustered around the best fit line (prediction errors). Again, it is observed that Model 5 has a lower value.

The variance ratio of an independent or dependent variable defined by variables in a regression model is represented by the R squared score. Model 5, Model 3, and Model 1 had the greatest r squared scores, correspondingly.

According to the table and the discussion above, Model 5 is the best model that produced the highest score, done the best forecast.

In [117]:
```
#y= x1*..+x2*...+ intercept (general formula for linear regression model)
_str="y="
for i,m in enumerate((model5.coef_).round(2)):
    _str+= "x_{}*{}+".format(i, m)
_str+=str((model5.intercept_).round(2))
print(_str)
```

```
y=x_0*-0.08+x_1*0.6+x_2*-2.41+x_3*11.88+x_4*0.16+x_5*0.1+x_6*-0.3+x_7*-0.23+x_8*0.75+x_9*-12.06+x_10*0.65+x_11*0.08+x_12*0.96+x
_13*-2.39+x_14*0.48+x_15*0.01+x_16*-0.34+x_17*0.02+x_18*1.18+x_19*2.34+69.48
```

The formula of our best model, Model 5, is given above. In this formula, x0,x1 and the rest denote the multiples of the independent parameters that we use to train each instance model. These are actually all the columns in our dataset, respectively:

```
(['Year', 'Status', 'Adult Mortality', 'infant deaths','Alcohol’
, ‘percentage expenditure', 'Hepatitis B', 'Measles ', ' BMI ','under-f
ive deaths ', 'Polio', 'Total expenditure', 'Diphtheria,' HIV/AIDS', 'G
DP', 'Population', ' thinness 1-19 years’, ‘thinness 5-9 years', 'Incom
e composition of resources', 'Schooling’)
```

As a result, we got the best result when we used all the features in our data set, but our formula was not compact enough. When using fewer features that have the highest correlations with life expectancy, we get very close to the score of Model 5. For example, as in Model 1 and Model 3.

5- *Conclusion*

Life expectancy is influenced by many different factors. Although the impact of these factors is not always the same, there may also be environmental factors that we have to change. As can be seen from ***Figure 2***, as a result of studying all this data set, the factor that most affects the life expectancy is schooling. It has been observed that the body mass index also has a very big impact on life expectancy, in the ***Figure 2***. In addition, a country's standing, its portion of the income distribution, and its level of development are all important determinants of life expectancy. Again, it can be observed that the vaccination rates of various diseases included in the list have a great impact on life expectancy. For example, Diphtheria, polio, etc.

Apart from making a life expectancy estimate, the factors affecting Life expectancy can be inferred from this study. The importance that countries' health organizations, such as the Turkish Ministry of Health or the World Organization of Health (WHO), will place on the features we just mentioned, the recommendations they will make, or the programs they will implement, can significantly increase the life expectancy factor. These might include research on improving immunization rates, raising countries' wealth levels, or lowering the body mass index by introducing habits like regular diet and athletics into society.

To summarize, all of the data columns in the WHO dataset must be processed and corrected utilizing the work done in this study in order to achieve the most accurate life expectancy estimate using linear regression. However, utilizing 8 to 11 data columns having the strongest connection with life expectancy yields a result that is extremely close to the value obtained when all data columns are included. Furthermore, it was discovered that predicted life expectancy is affected by environmental, changeable elements, and that life expectancy may be increased via research on these aspects throughout this study.

6- *References*

Figure 1 - https://www.enjoyalgorthms.com/blog/lfe-expectancy-predcton-usng-lnear-regresson.com