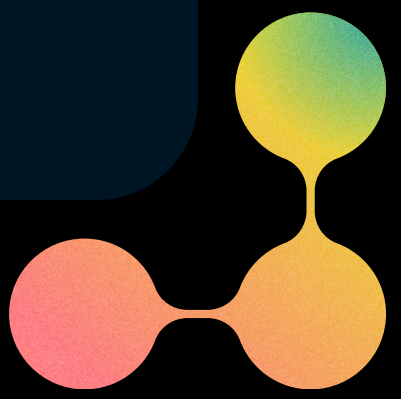
LIFE EXPECTENCY PREDICTION USING LINEAR REGRESSION AND XGBOOST

Yuaan Hussain Raheem

LIBRARIES

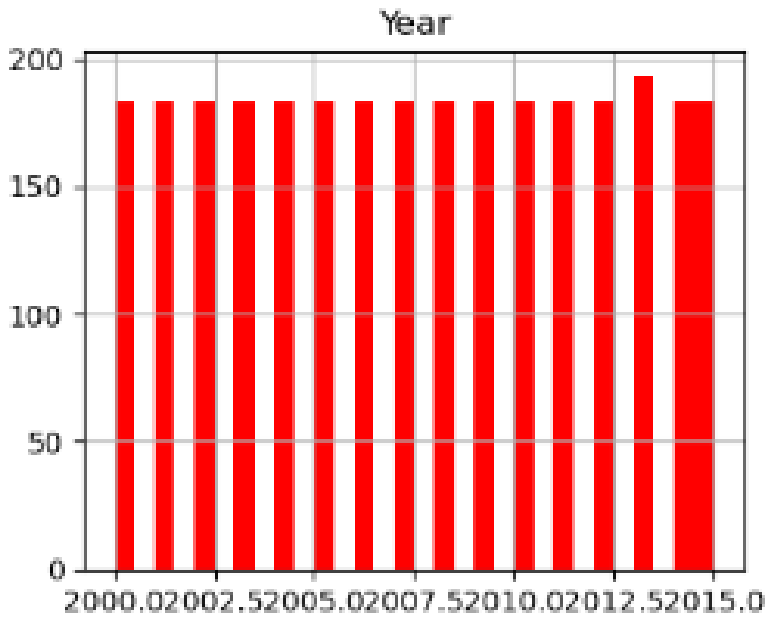
- pandas
- numpy
- seaborn
- matplotlib.pyplot
- train_test_split
- sklearn.preprocessing
- sklearn.linear_model
- sklearn.metrics
- xgboost
- sklearn.metric

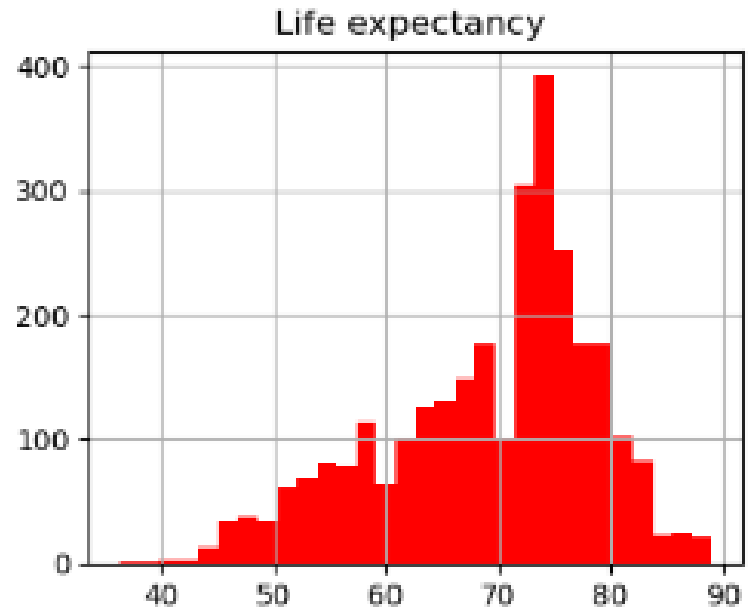


DATASET 2

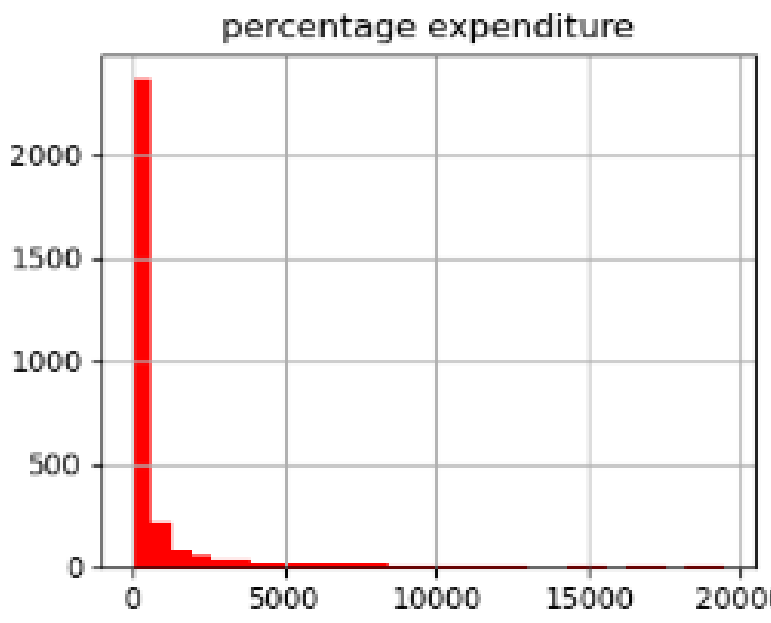
```
life_expectancy_df.info()
life_expectancy_df.hist(bins = 30, figsize =
(20, 20), color = 'r');
```

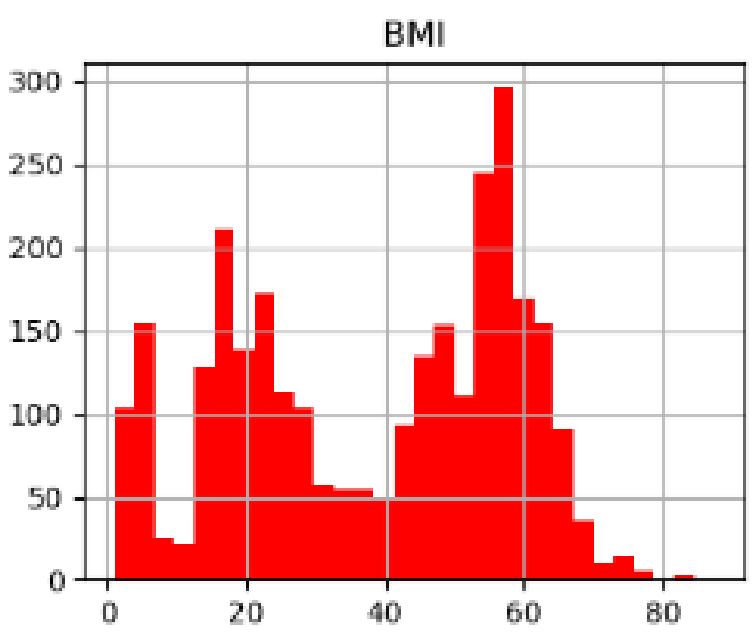
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 21 columns):
 #   Column                           Non-Null Count  Dtype  
---  ------                           --------------  -----  
 0   Year                             2938 non-null   int64  
 1   Status                           2938 non-null   object 
 2   Life expectancy                  2928 non-null   float64
 3   Adult Mortality                  2928 non-null   float64
 4   infant deaths                    2938 non-null   int64  
 5   Alcohol                          2744 non-null   float64
 6   percentage expenditure           2938 non-null   float64
 7   Hepatitis B                      2385 non-null   float64
 8   Measles                          2938 non-null   int64  
 9    BMI                             2904 non-null   float64
 10  under-five deaths                2938 non-null   int64  
 11  Polio                            2919 non-null   float64
 12  Total expenditure                2712 non-null   float64
 13  Diphtheria                       2919 non-null   float64
 14   HIV/AIDS                        2938 non-null   float64
 15  GDP                              2490 non-null   float64
 16  Population                       2286 non-null   float64
 17   thinness  1-19 years            2904 non-null   float64
 18   thinness 5-9 years              2904 non-null   float64
 19  Income composition of resources  2771 non-null   float64
 20  Schooling                        2775 non-null   float64
dtypes: float64(16), int64(4), object(1)
memory usage: 482.1+ KB
```

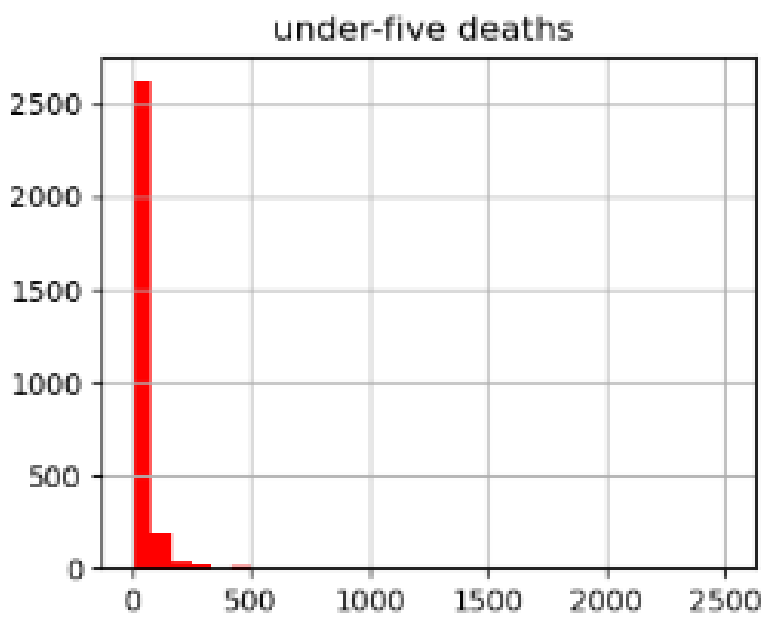

Year
200
150
100
50
0
2000.0
2002.5
2005.0
2007.5
2010.0
2012.5
2015.0

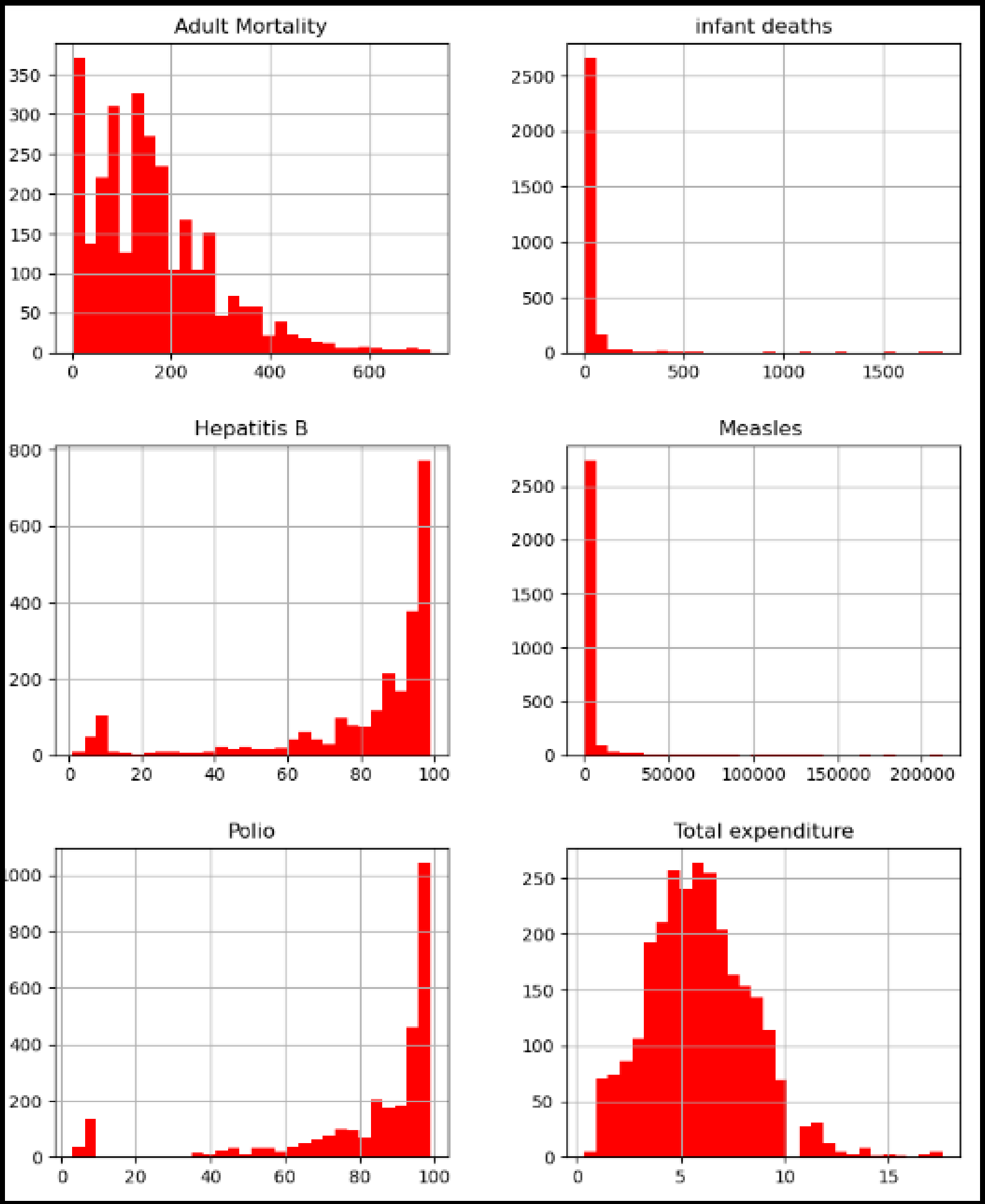

Life expectancy
400
300
200
100
0
40
50
60
70
80
90

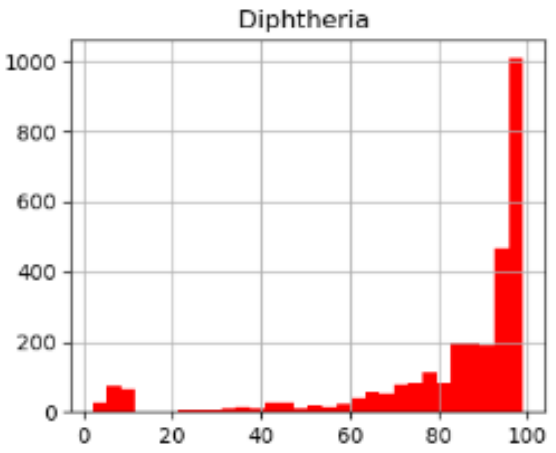

Alcohol
600
500
400
300
200
100
0
0
5
10
15

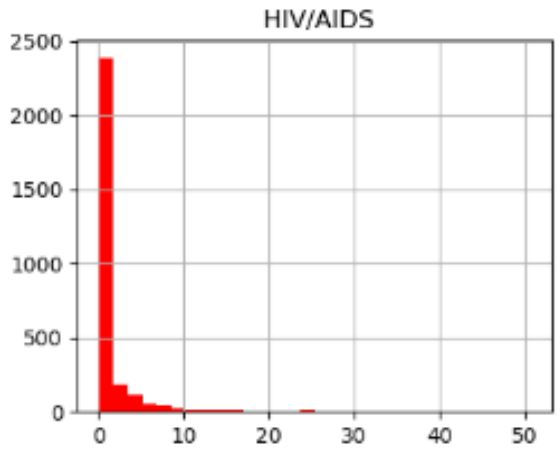

percentage expenditure
2000
1500
1000
500
0
0
5000
10000
15000
20000


BMI
300
250
200
150
100
50
0
0
20
40
60
80


under-five deaths
2500
2000
1500
1000
500
0
0
500
1000
1500
2000
2500

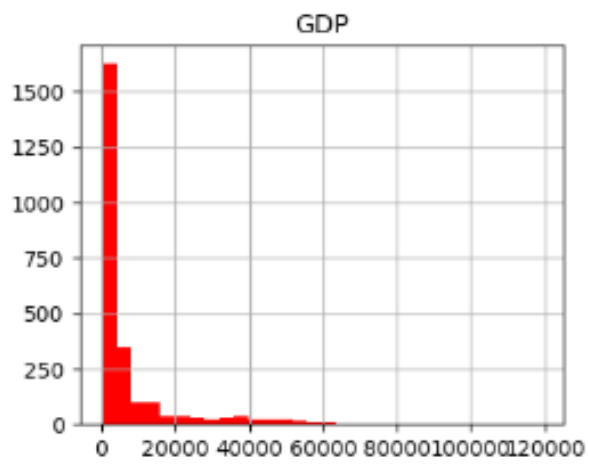

Adult Mortality
0
50
100
150
200
250
300
350
0
200
400
600
infant deaths
0
500
1000
1500
2000
2500
0
500
1000
1500
Hepatitis B
0
200
400
600
800
0
20
40
60
80
100
Measles
0
500
1000
1500
2000
2500
0
50000
100000
150000
200000
Polio
0
200
400
600
800
1000
0
20
40
60
80
100
Total expenditure
0
50
100
150
200
250
0
5
10
15

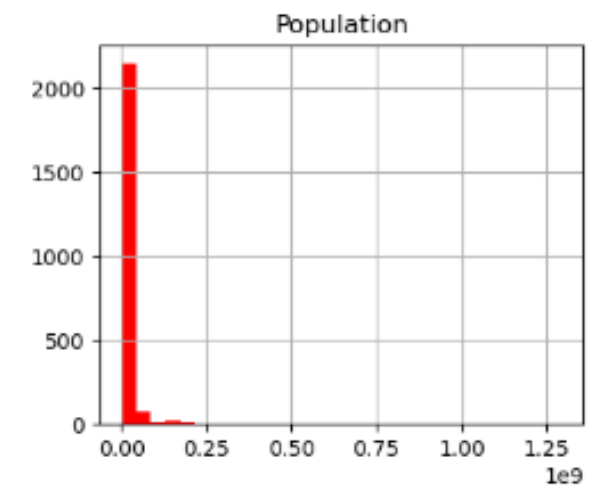

Diphtheria
1000
800
600
400
200
0
0
20
40
60
80
100

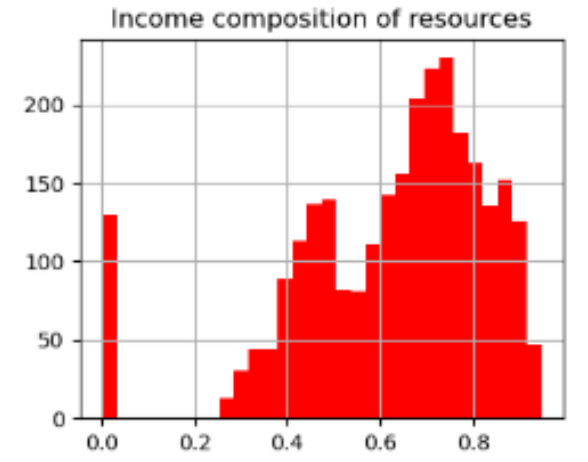

HIV/AIDS
2500
2000
1500
1000
500
0
0
10
20
30
40
50

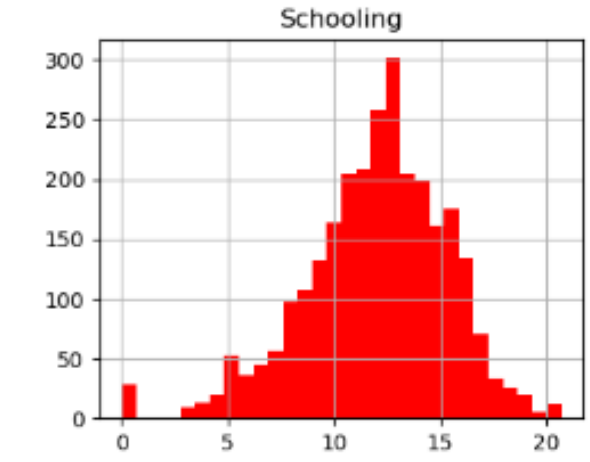

thinness 1-19 years
500
400
300
200
100
0
0
5
10
15
20
25


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


GDP
1500
1250
1000
750
500
250
0
0
20000
40000
60000
80000
100000
120000

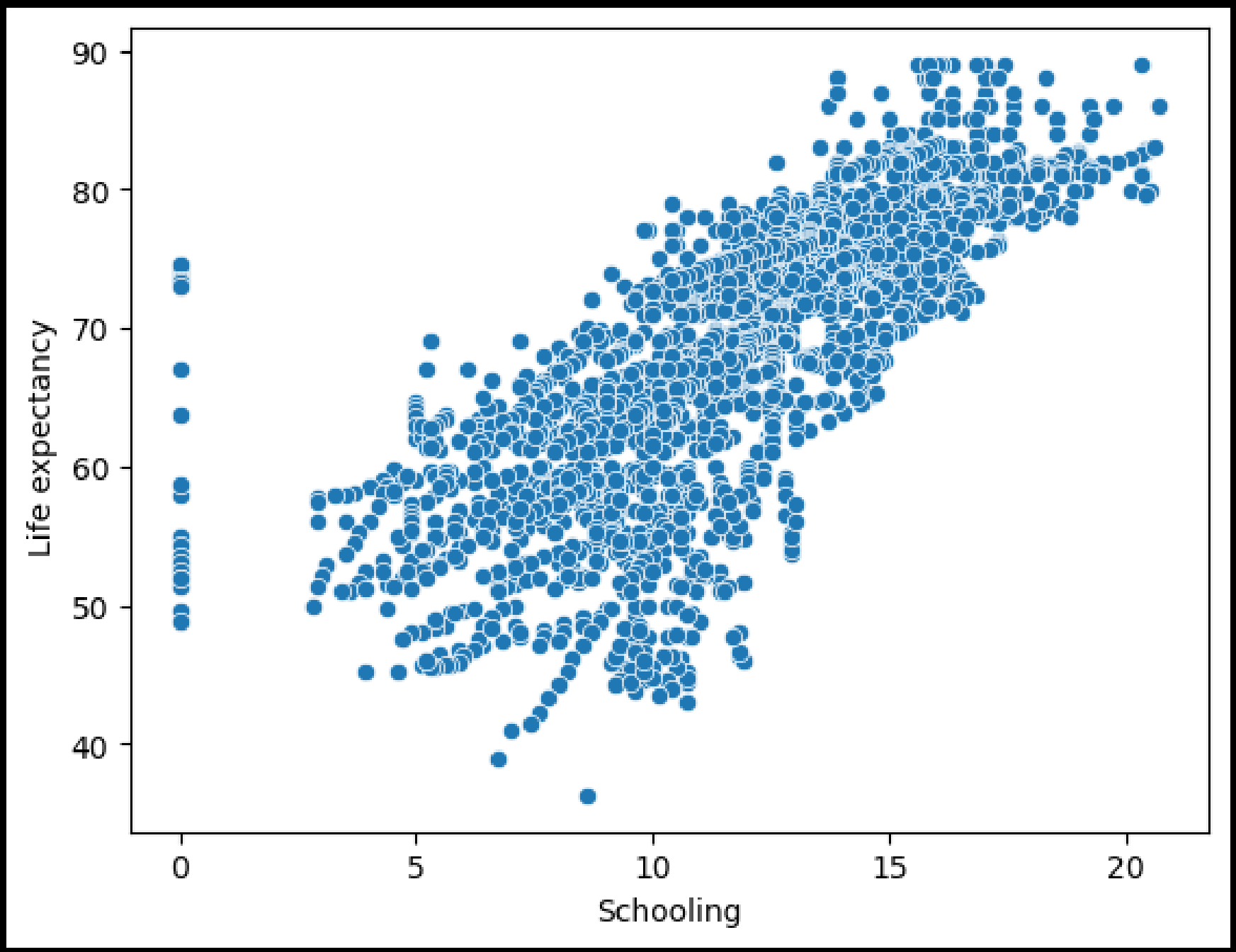

Population
2000
1500
1000
500
0
0.00
0.25
0.50
0.75
1.00
1.25
1e9

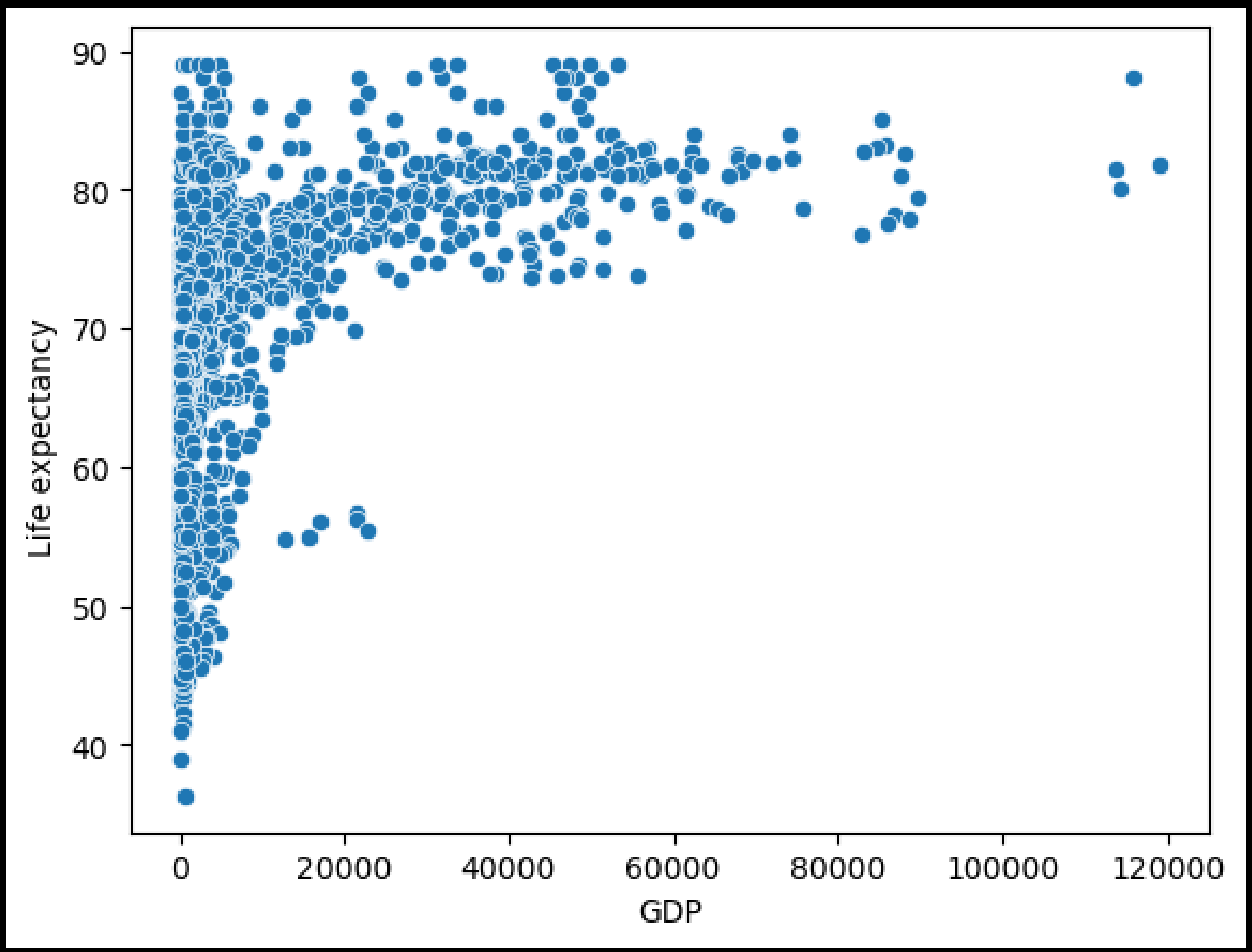

Income composition of resources
200
150
100
50
0
0.0
0.2
0.4
0.6
0.8

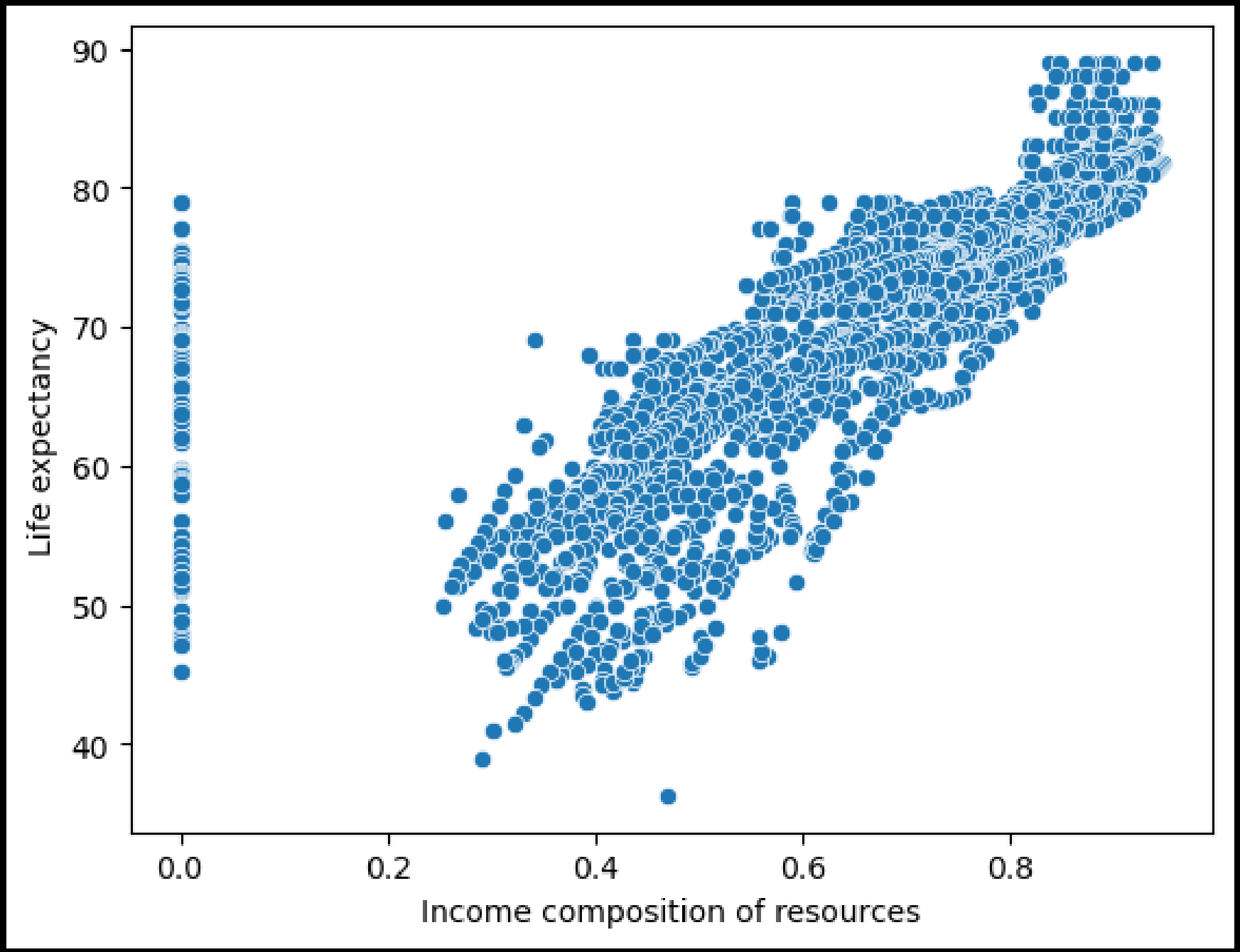

Schooling
300
250
200
150
100
50
0
0
5
10
15
20

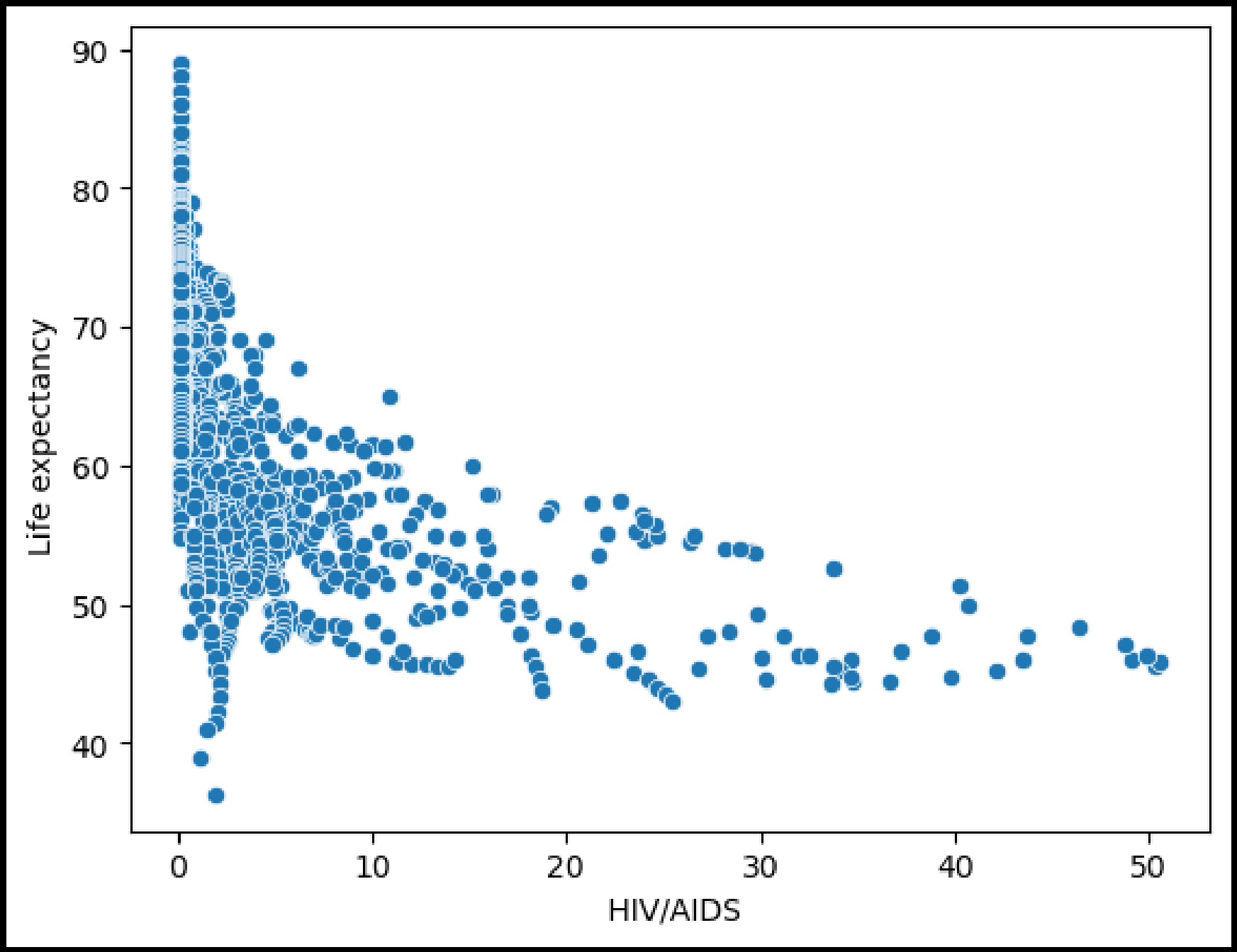
EDA

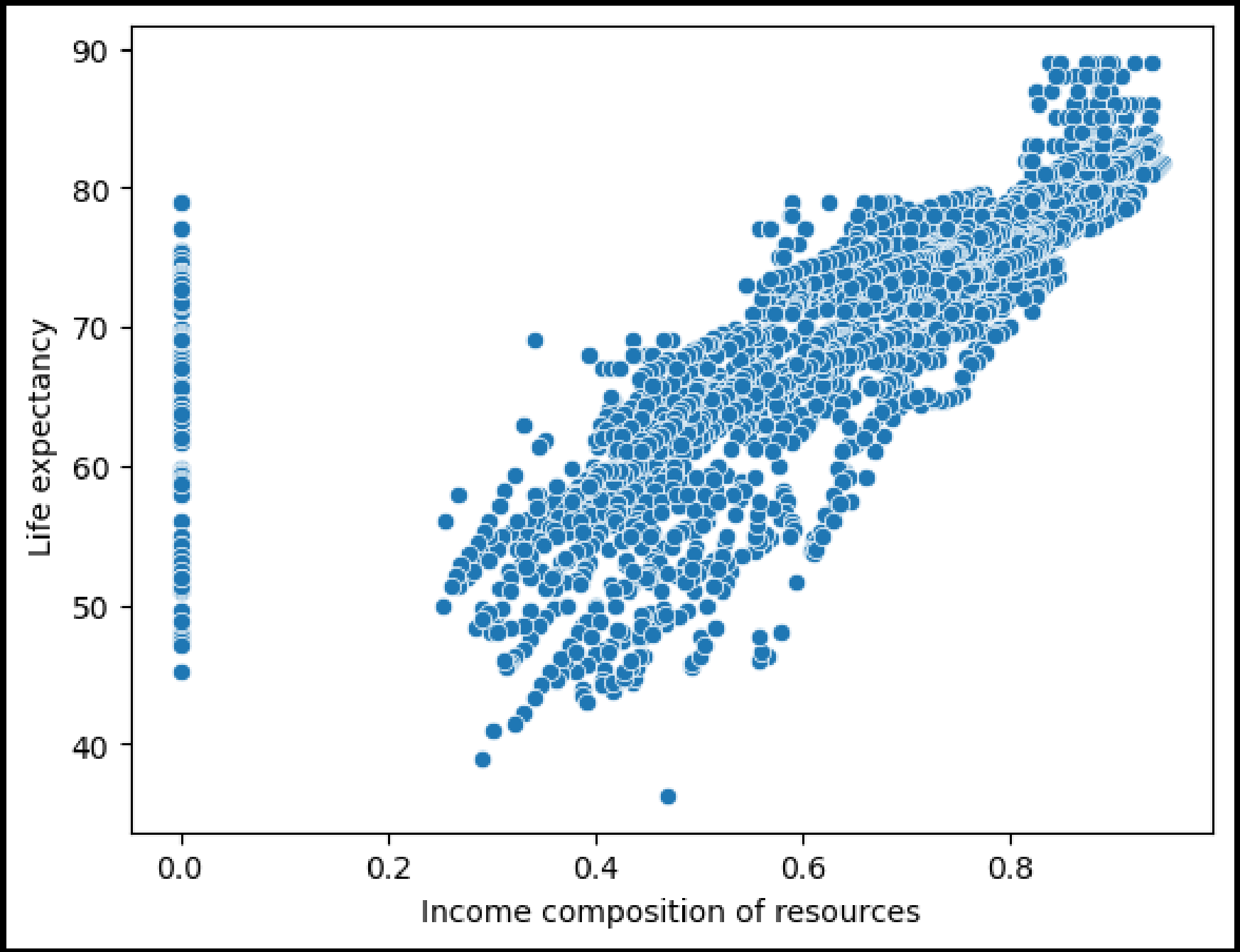
```
1.sns.scatterplot(data = life_expectancy_df, x =
  'Schooling', y = 'Life expectancy ')
2.sns.scatterplot(data = life_expectancy_df, x =
  'GDP', y = 'Life expectancy ')
3.sns.scatterplot(data = life_expectancy_df, x =
  'Income composition of resources', y = 'Life
  expectancy ')
4.sns.scatterplot(data = life_expectancy_df, x =
  ' HIV/AIDS', y = 'Life expectancy ')
5.sns.scatterplot(data = life_expectancy_df, x =
  'Income composition of resources', y = 'Life
  expectancy ')
```


90
80
70
60
50
40
Life expectancy
0
5
10
15
20
Schooling


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

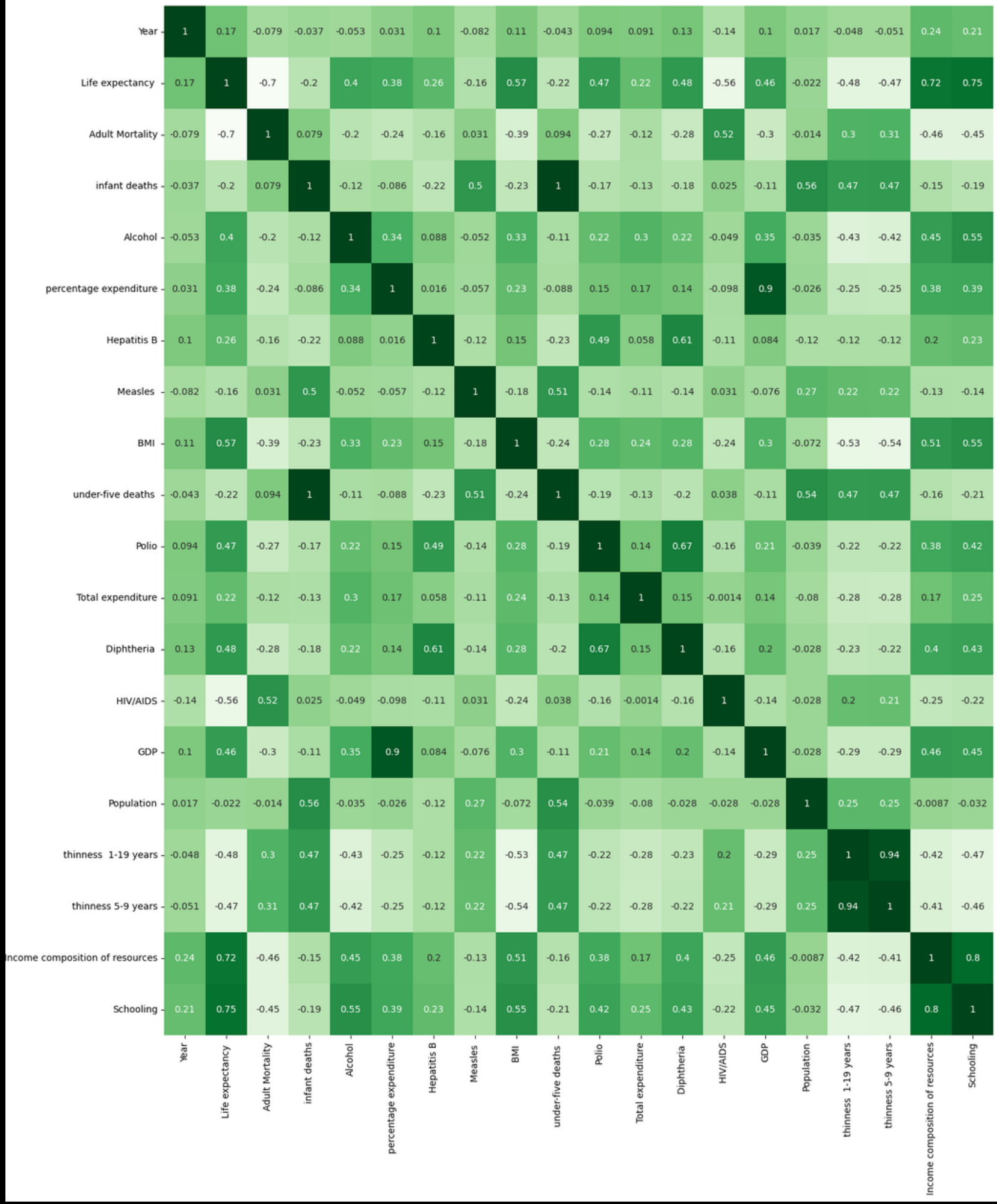

90
80
70
60
50
40
Life expectancy
0.0
0.2
0.4
0.6
0.8
Income composition of resources


90
80
70
60
50
40
Life expectancy
0
10
20
30
40
50
HIV/AIDS


90
80
70
60
50
40
Life expectancy
0.0
0.2
0.4
0.6
0.8
Income composition of resources

Correlation Heatmap

```
plt.figure(figsize = (20,20))
corr_matrix = life_expectancy_df.drop('Status',
axis=1).corr()
sns.heatmap(corr_matrix, annot = True,
cmap='Greens')
plt.show()
```


Year 1 0.17 -0.079 -0.037 -0.053 0.031 0.1 -0.082 0.11 -0.043 0.094 0.091 0.13 -0.14 0.1 0.017 -0.048 -0.051 0.24 0.21
Life expectancy 0.17 1 -0.7 -0.2 0.4 0.38 0.26 -0.16 0.57 -0.22 0.47 0.22 0.48 -0.56 0.46 -0.022 -0.48 -0.47 0.72 0.75
Adult Mortality -0.079 -0.7 1 0.079 -0.2 -0.24 -0.16 0.031 -0.39 0.094 -0.27 -0.12 -0.28 0.52 -0.3 -0.014 0.3 0.31 -0.46 -0.45
infant deaths -0.037 -0.2 0.079 1 -0.12 -0.086 -0.22 0.5 -0.23 1 -0.17 -0.13 -0.18 0.025 -0.11 0.56 0.47 0.47 -0.15 -0.19
Alcohol -0.053 0.4 -0.2 -0.12 1 0.34 0.088 -0.052 0.33 -0.11 0.22 0.3 0.22 -0.049 0.35 -0.035 -0.43 -0.42 0.45 0.55
percentage expenditure 0.031 0.38 -0.24 -0.086 0.34 1 0.016 -0.057 0.23 -0.088 0.15 0.17 0.14 -0.098 0.9 -0.026 -0.25 -0.25 0.38 0.39
Hepatitis B 0.1 0.26 -0.16 -0.22 0.088 0.016 1 -0.12 0.15 -0.23 0.49 0.058 0.61 -0.11 0.084 -0.12 -0.12 -0.12 0.2 0.23
Measles -0.082 -0.16 0.031 0.5 -0.052 -0.057 -0.12 1 -0.18 0.51 -0.14 -0.11 -0.14 0.031 -0.076 0.27 0.22 0.22 -0.13 -0.14
BMI 0.11 0.57 -0.39 -0.23 0.33 0.23 0.15 -0.18 1 -0.24 0.28 0.24 0.28 -0.24 0.3 -0.072 -0.53 -0.54 0.51 0.55
under-five deaths -0.043 -0.22 0.094 1 -0.11 -0.088 -0.23 0.51 -0.24 1 -0.19 -0.13 -0.2 0.038 -0.11 0.54 0.47 0.47 -0.16 -0.21
Polio 0.094 0.47 -0.27 -0.17 0.22 0.15 0.49 -0.14 0.28 -0.19 1 0.14 0.67 -0.16 0.21 -0.039 -0.22 -0.22 0.38 0.42
Total expenditure 0.091 0.22 -0.12 -0.13 0.3 0.17 0.058 -0.11 0.24 -0.13 0.14 1 0.15 -0.0014 0.14 -0.08 -0.28 -0.28 0.17 0.25
Diphtheria 0.13 0.48 -0.28 -0.18 0.22 0.14 0.61 -0.14 0.28 -0.2 0.67 0.15 1 -0.16 0.2 -0.028 -0.23 -0.22 0.4 0.43
HIV/AIDS -0.14 -0.56 0.52 0.025 -0.049 -0.098 -0.11 0.031 -0.24 0.038 -0.16 -0.0014 -0.16 1 -0.14 -0.028 0.2 0.21 -0.25 -0.22
GDP 0.1 0.46 -0.3 -0.11 0.35 0.9 0.084 -0.076 0.3 -0.11 0.21 0.14 0.2 -0.14 1 -0.028 -0.29 -0.29 0.46 0.45
Population 0.017 -0.022 -0.014 0.56 -0.035 -0.026 -0.12 0.27 -0.072 0.54 -0.039 -0.08 -0.028 -0.028 -0.028 1 0.25 0.25 -0.0087 -0.032
thinness 1-19 years -0.048 -0.48 0.3 0.47 -0.43 -0.25 -0.12 0.22 -0.53 0.47 -0.22 -0.28 -0.23 0.2 -0.29 0.25 1 0.94 -0.42 -0.47
thinness 5-9 years -0.051 -0.47 0.31 0.47 -0.42 -0.25 -0.12 0.22 -0.54 0.47 -0.22 -0.28 -0.22 0.21 -0.29 0.25 0.94 1 -0.41 -0.46
Income composition of resources 0.24 0.72 -0.46 -0.15 0.45 0.38 0.2 -0.13 0.51 -0.16 0.38 0.17 0.4 -0.25 0.46 -0.0087 -0.42 -0.41 1 0.8
Schooling 0.21 0.75 -0.45 -0.19 0.55 0.39 0.23 -0.14 0.55 -0.21 0.42 0.25 0.43 -0.22 0.45 -0.032 -0.47 -0.46 0.8 1
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
Schooling

ONE-HOT ENCODING

Since the status column has two categorical values, i.e., Developing and Developed, we will convert it into nominal for our ML model to understand it properly.

```
life_expectancy_df =
pd.get_dummies(life_expectancy_df, columns =
['Status'])
```

5 NULL VALUES

```
life_expectancy_df.isnull().sum()
[np.where(life_expectancy_df.isnull().sum() != 0)
[0]]
```

```
Life expectancy                  10
Adult Mortality                  10
Alcohol                         194
Hepatitis B                     553
 BMI                             34
Polio                            19
Total expenditure               226
Diphtheria                       19
GDP                             448
Population                      652
 thinness  1-19 years            34
 thinness 5-9 years              34
Income composition of resources 167
Schooling                       163
dtype: int64
```

After checking for null values we can see that most of them are continuous values hence we will fill those null values with their mean respectively.

```
life_expectancy_df = life_expectancy_df.apply(lambda
x: x.fillna(x.mean()),axis=0)
```

TRAIN TEST SPLIT


```
X = life_expectancy_df.drop(columns = ['Life expectancy '])
y = life_expectancy_df[['Life expectancy ']]

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size = 0.2)

scaler_X = StandardScaler()
X_train = scaler_X.fit_transform(X_train)
X_test = scaler_X.transform(X_test)

scaler_y = StandardScaler()
y_train = scaler_y.fit_transform(y_train)
y_test = scaler_y.transform(y_test)
```

TRAINING MODEL

```
regresssion_model_sklearn = LinearRegression(fit_intercept =
True)
regresssion_model_sklearn.fit(X_train, y_train)
```

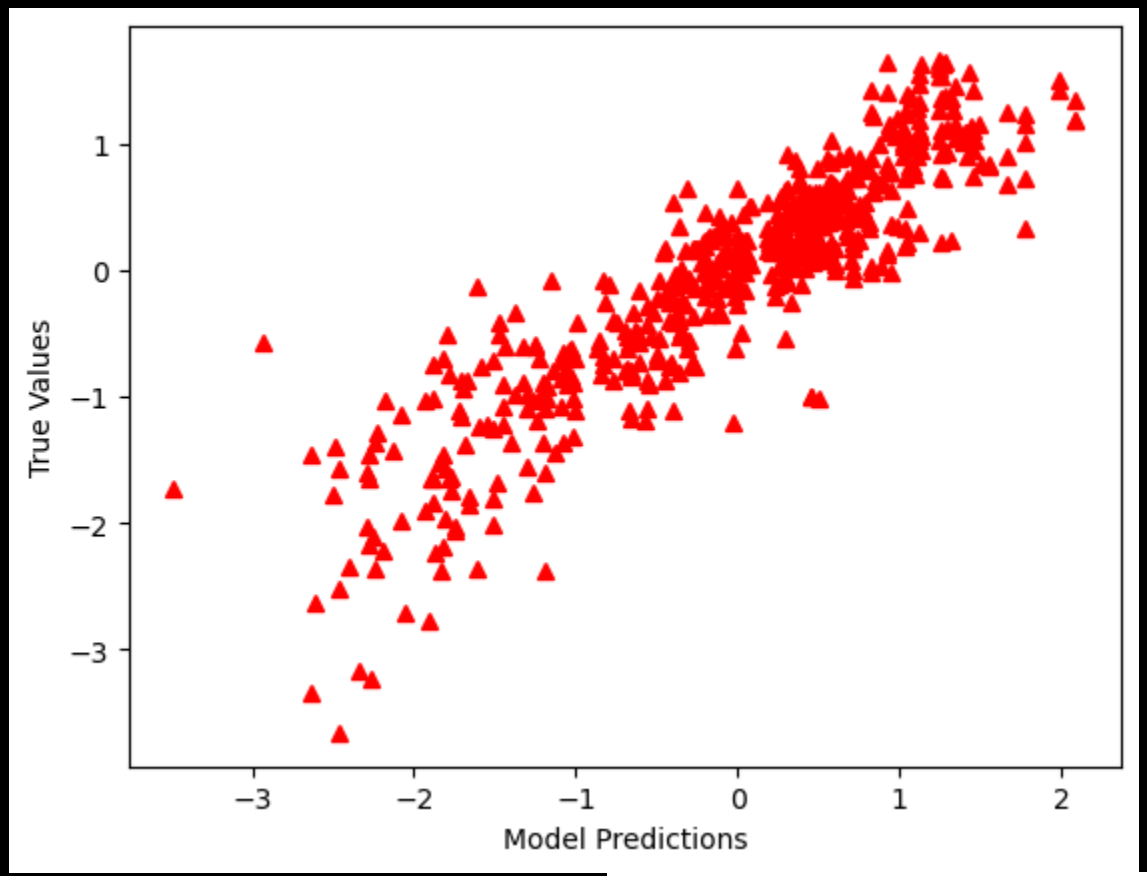
```
xgb_model = xgb.XGBRegressor(objective='reg:squarederror')
xgb_model.fit(X_train, y_train)
```

EVALUATING MODEL


```
y_predict_lr = regresssion_model_sklearn.predict(X_test)
y_predict_xgb = xgb_model.predict(X_test)
```

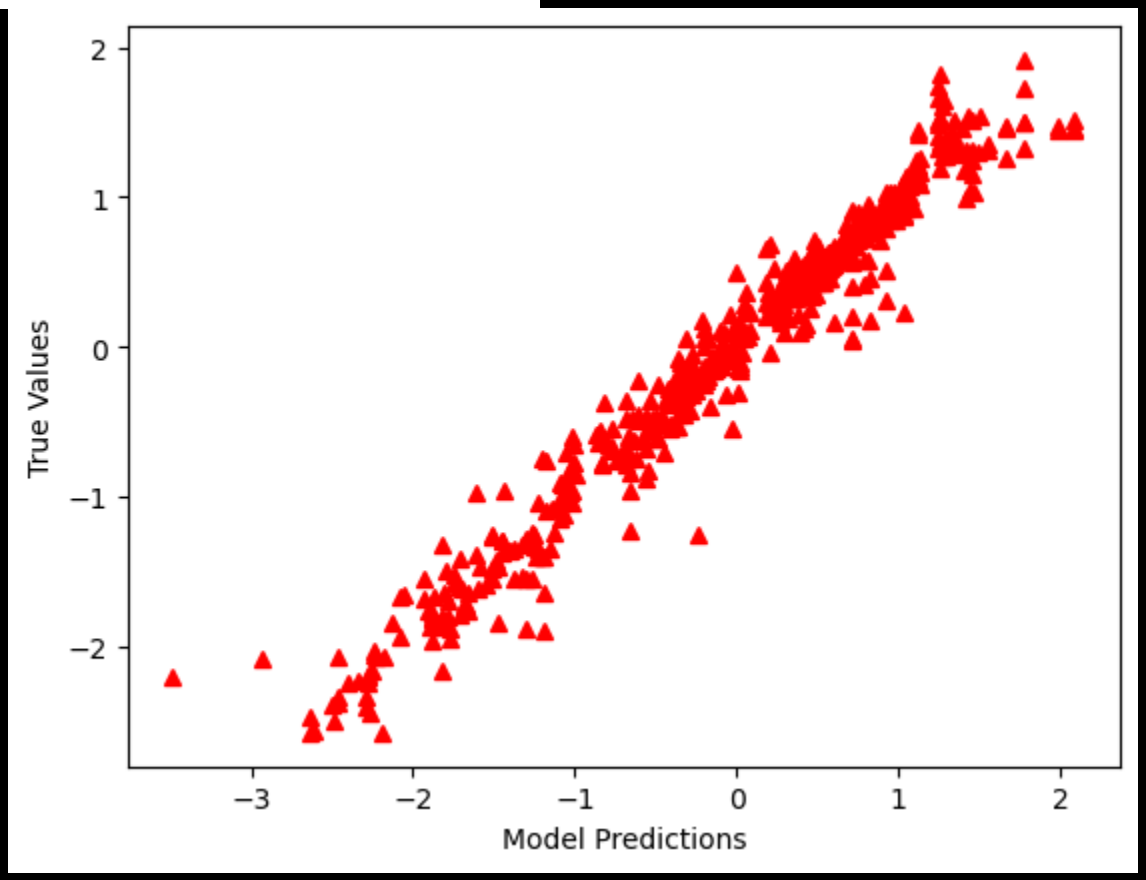
```
plt.plot(y_test, y_predict_lr, "^", color = 'r')
plt.xlabel('Model Predictions')
plt.ylabel('True Values')

plt.plot(y_test, y_predict_xgb, "^", color = 'r')
plt.xlabel('Model Predictions')
plt.ylabel('True Values')
```

	LR	XG BOOST
RMSE	4.213	4.213
MSE	17.749678	0.04095177
MAE	3.0903466	0.13452429
R2	0.8115424751819664	0.9611257301193765
ADJ R2	0.8045502348618627	0.9596833985513675


True Values
Model Predictions
1
0
−1
−2
−3
−3
−2
−1
0
1
2

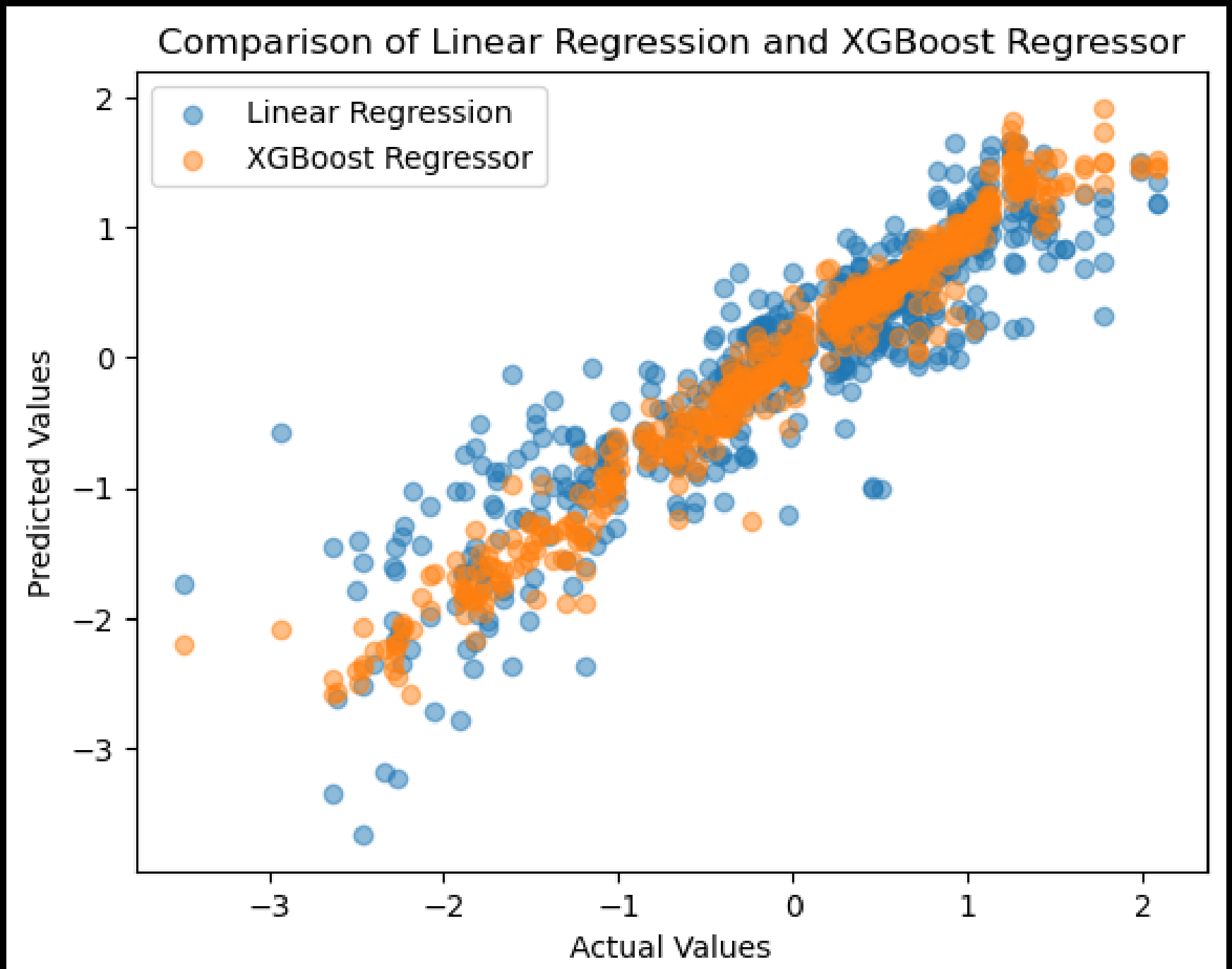
LR

XG BOOST


True Values
Model Predictions
2
1
0
−1
−2
−3
−2
−1
0
1
2

COMPARING MODELS

```
plt.scatter(y_test, y_predict_lr, label='Linear Regression',
alpha=0.5)
plt.scatter(y_test, y_predict_xgb, label='XGBoost Regressor',
alpha=0.5)
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.legend()
plt.title('Comparison of Linear Regression and XGBoost
Regressor')
plt.show()
```


Comparison of Linear Regression and XGBoost Regressor
Linear Regression
XGBoost Regressor
Predicted Values
Actual Values
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2

THE END

Github Link for this project →

Yuaan Hussain Raheem