Data Analysis Final Project

Research Objective:

This dataset contains information compiled by the World Health Organization and the United Nations to track factors that affect life expectancy. The data contains 17 rows and 21 columns. The columns include: country, year, adult mortality, life expectancy, alcohol consumption per capita, and country's expenditure on health.

We are going to implement different methods of data, for analyzing the yearly change of rates.

Dataset:

- World Health Organization Global Health Observatory (GHO)
  https://apps.who.int/gho/data/node.home

Formulation:

Contrary to what I had initially thought, the scikit-learn implementation of Linear Regression minimizes a cost function of the form:

$$\min_{w} ||Xw - y||_2^2$$

using the singular value decomposition of X.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

Where,

$\hat{y}$ $-$ *predicted value of* y

$\bar{y}$ $-$ *mean value of* y

We calculate the distance from the line to a given data point by subtracting one from the other. We take the square of the difference because we don't want the predicted values below the actual values to cancel out with those above the actual values. In mathematical terms, the latter can be expressed as follows:

$$\sum_{i}^{n}(y_{pred} - y)^2$$

$$\sum_{i}^{n}(wx - y)^2$$

The cost function used in the scikit-learn library is similar, only we're calculating it simultaneously using matrix operations.

For those of you that have taken a Calculus course, you've probably encountered this kind of notation before.

$$\|\mathbf{x}\| := \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}.$$

x is a vector and we are calculating its magnitude.

In the same sense, when we surround the variable for a matrix (i.e. A) by vertical bars, we are saying that we want to go from a matrix of rows and columns to a scalar. There are multiple ways of deriving a scalar from a matrix. Depending on which one is used, you'll see a different symbol to the right of the variable (the extra 2 in the equation wasn't put there by accident).

$$\min_w ||Xw - y||_2^2$$

The additional 2 implies that we are taking the Euclidean norm of the matrix.4

$$\|A\|_2 = \max_{i=1:n} \sqrt{\lambda_i(A^{\mathrm{T}}A)}$$

where $\lambda_i(A^{\mathrm{T}}A)$ is the i^{th} eigenvalue of $A^{\mathrm{T}}A$.

So that's how we quantify the error. However, that gives rise to a new question. Specifically, how do we actually go about minimizing it? Well, as it turns out, the minimum norm least squares solution (coefficients) can be found by calculating the pseudoinverse of the input matrix X and multiplying that by the output vector y.

$$w = X^+ y$$

$$X^+ = V D^+ U^T$$

$$X = U \Sigma V^T$$

We construct the diagonal matrix D^+ by taking the inverse of the values within the sigma matrix. The + refers to the fact that all the elements must be greater than 0 since we can't divide by 0.

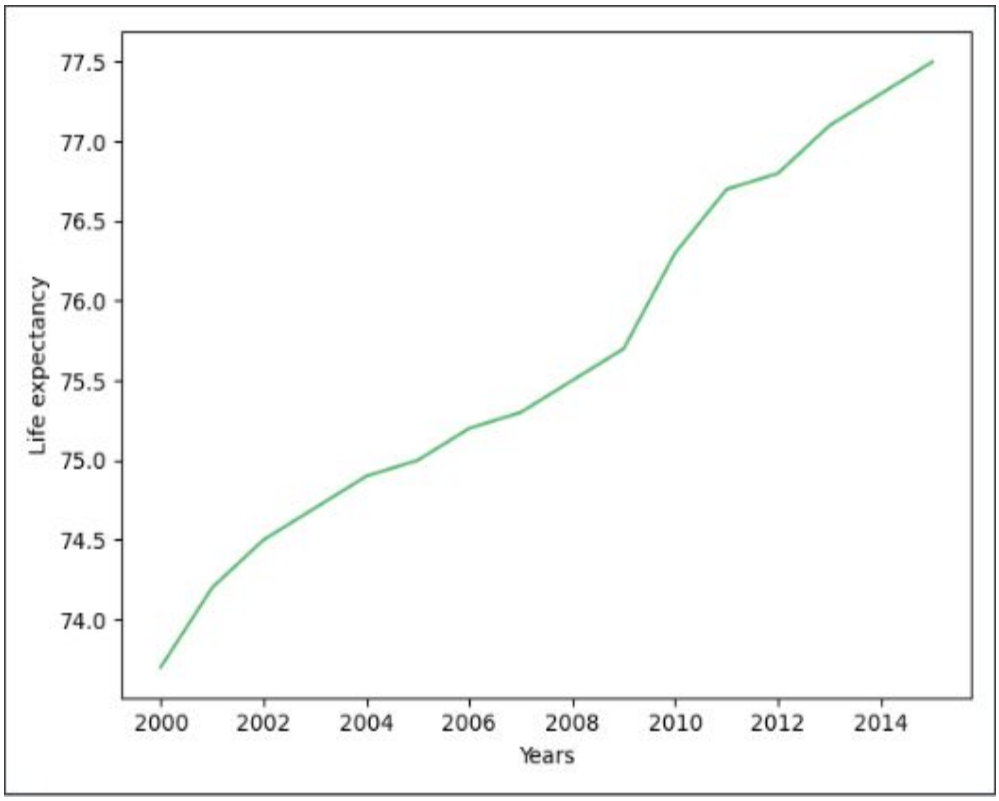
$$\Sigma = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$$

$$\text{where } \lambda_i > 0$$

$$D^+ = \begin{pmatrix} \frac{1}{\lambda_1} & 0 & 0 \\ 0 & \frac{1}{\lambda_2} & 0 \\ 0 & 0 & \frac{1}{\lambda_3} \end{pmatrix}$$

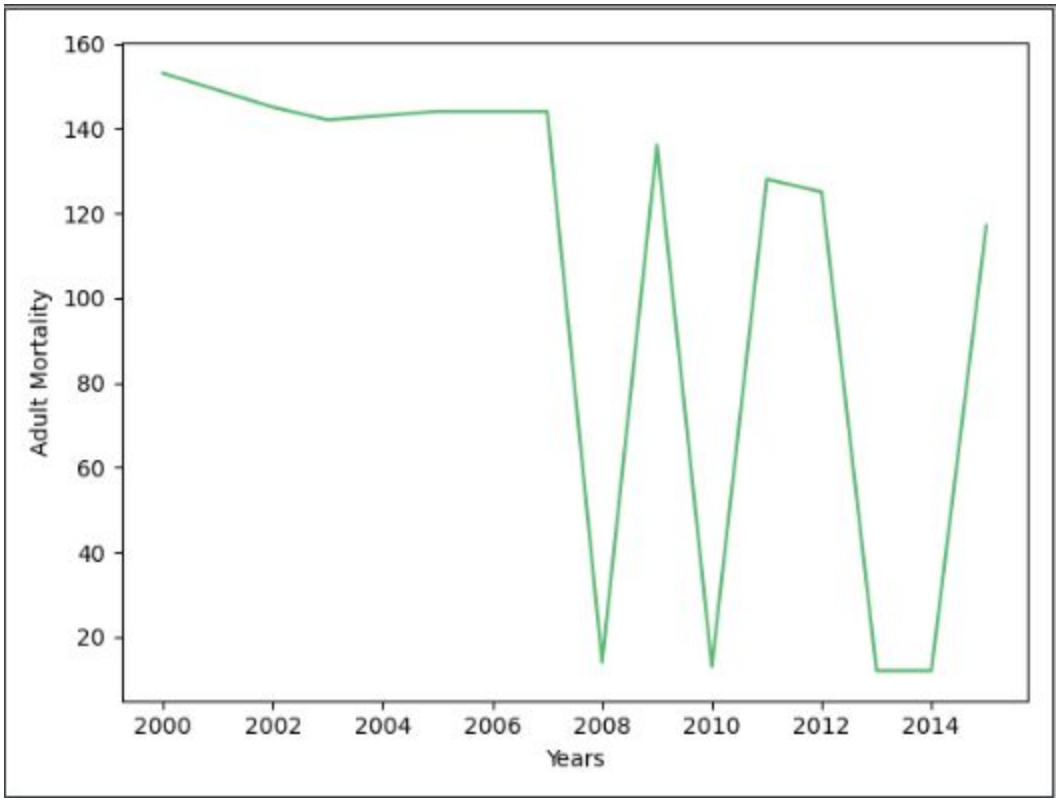
Python Code:

```
A = data['Life expectancy'].values
B = data['Year'].values
plt.plot(B, A, color='#58b970')
plt.xlabel('Years')
plt.ylabel('Life expectancy')
plt.show()
```

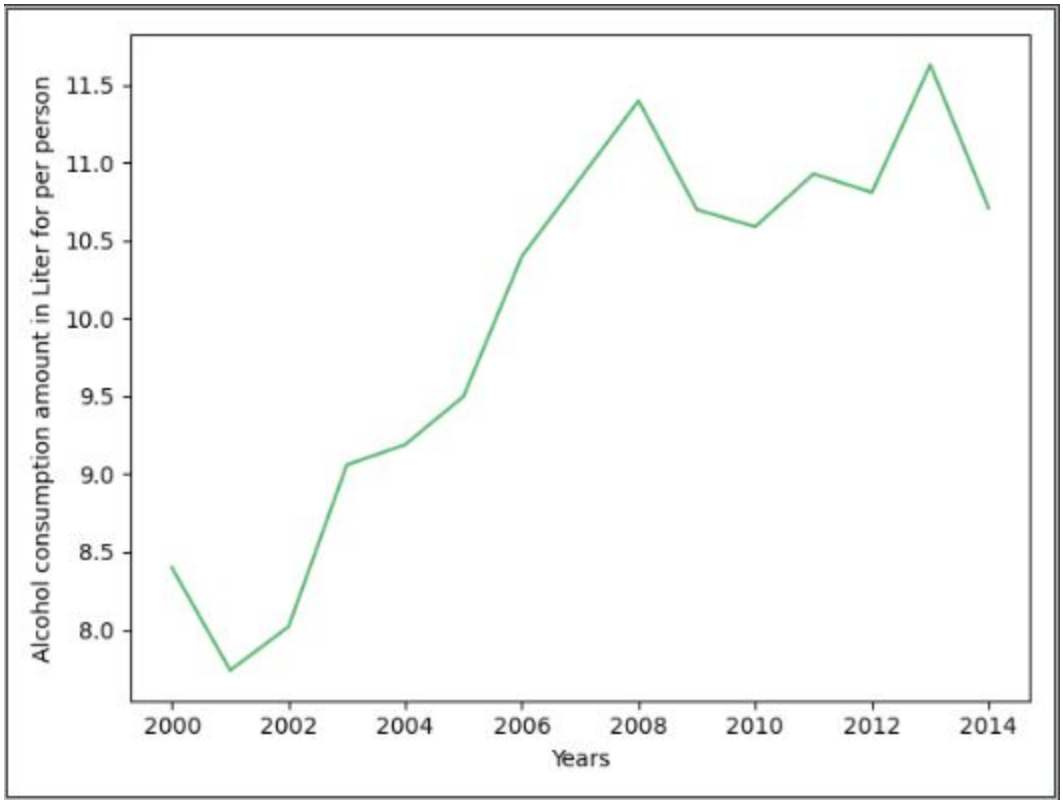


When we are looking at the table, It is obviously seen that from 2000 to 2015, average life duration is expanding in Poland. Between 2009 and 2011 years,there was sharp development.

```
A = data['Adult Mortality'].values
B = data['Year'].values
plt.plot(B, A, color='#58b970')
plt.xlabel('Years')
plt.ylabel('Adult Mortality')
plt.show()
```



This graph shows us Adult Mortality Rates of both sexes (amount of dying between 15 and 60 age for per 1000 population). In 2008, 2010, and 2013-2014 the amount of the death was low.

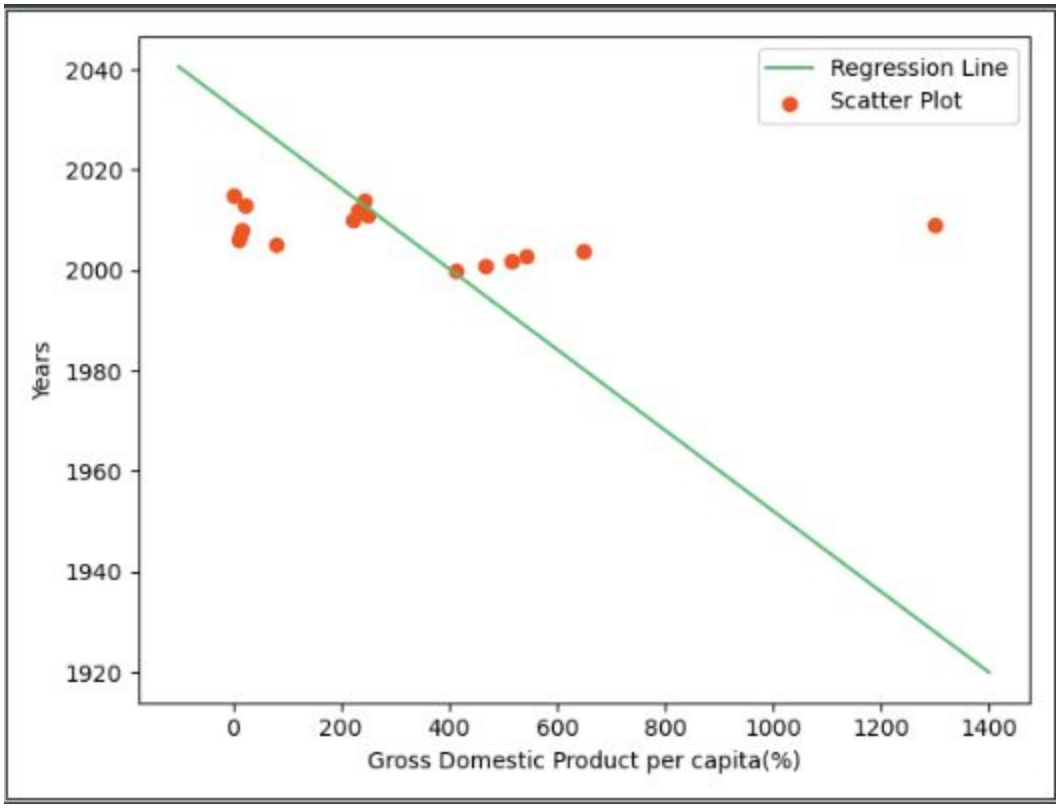


Alcohol, recorded per capita (15+) consumption (in litres of pure alcohol), The lowest consumption of alcohol per person is shown in 2001 years.

```
X = data['percentage expenditure'].values
Y = data['Year'].values

# Mean X and Y
mean_x = np.mean(X)
mean_y = np.mean(Y)

# Total number of values
n = len(X)

# Using the formula to calculate 'm' and 'c'
numer = 0
denom = 0
for i in range(n):
    numer += (X[i] - mean_x) * (Y[i] - mean_y)
denom += (X[i] - mean_x) ** 2
m = numer / denom
c = mean_y - (m * mean_x)

# Printing coefficients
print("Coefficients")
print(m, c)
```

```python
# Calculating R2 Score
ss_tot = 0
ss_res = 0
for i in range(n):
    y_pred = c + m * X[i]
    ss_tot += (Y[i] - mean_y) ** 2
    ss_res += (Y[i] - y_pred) ** 2
r2 = 1 - (ss_res/ss_tot)
print("R2 Score")
print(r2)
```

```
  COUNTRY  Year  ...  Alcohol  percentage expenditure
0  Poland  2000  ...     8.40               412.432397
1  Poland  2001  ...     7.74               466.738311
2  Poland  2002  ...     8.02               516.055439
3  Poland  2003  ...     9.06               542.023500
4  Poland  2004  ...     9.19               648.074345

[5 rows x 7 columns]
Coefficients
-0.08039696280579925 2032.4779107838099
RMSE
25.439247020064123
R2 Score
-29.45436653872195

Process finished with exit code 0
```



Expenditure on health as a percentage of Gross Domestic Product for per person(%).