Analysis of Heart Disease Mortality Using Diabetes Data

CO-AUTHORS

SHRUTI BALAJI

PAWAN KARTHIK

GULAM AFROZA MOHAMMAD

SAURAV KUMAR

Presented to: Dylan George

Director for the Center for Forecasting and Outbreak Analytics (CFA)

Analysis of Heart Disease Mortality Using Diabetes Data

THE ISSUE

The data used in this study was collected from the Centre's for Disease Control and Prevention (CDC) website. This comprehensive dataset encompasses various health indicators, including the physical environment of obesity, food access of inactivity, transport of inactivity, economics of obesity, diabetes, obesity, and inactivity, along with their corresponding Social Vulnerability Index (SVI) values. Additionally, the mortality of heart disease data set was obtained from the CDC website. Information regarding states, including their codes, was also collected for regional analysis.
The primary inquiries we aim to address using this extensive dataset are as follows:

- Is there a relationship between diabetes, inactivity and obesity?
- What is relationship between physical environment, economics, and obesity?
- What is the connection between food access, transport and inactivity?
- Will data on obesity and inactivity, as well as the factors influencing them, help in the analysis of diabetes?
- Is there a relationship between heart disease and variables such as diabetes, obesity, and physical inactivity?
- Does the average of diabetes per county rise or fall in relation to the average of heart disease per county?
- Can our predictive model effectively forecast the occurrence of heart disease using data from previous years, demonstrating their reliability and accuracy?
- Does a focus on diabetes prevention help to reduce the risks of heart disease?

THE FINDINGS

After analyzing the data collected from CDC,

- A model using the variables - diabetes, obesity and inactivity was developed which was not completely reliant on predicting diabetes.
- We aimed at finding any possible link for obesity with physical environment, economics and inactivity with food access and transportation. (i.e.) develop a predictive model to find the relationship among these six variables predicting obesity and inactivity. The model showed poor efficiency in predicting the obesity rate and inactivity rate concluding that this data cannot be used for the prediction of obesity and inactivity.
- CDC website has the heart mortality data for all the counties in the USA, which could be used as the additional variable. On analyzing the heart mortality data with the diabetes data, a strong association between high rates of diabetes in particular states and the same for heart disease mortality was seen. This correlation shows that controlling diabetes may be crucial to lowering the risk of dying from heart disease. This association can be visualized in the form of a bar graph.
- We performed a linear regression analysis between the datasets of diabetes and the mortality caused by heart disease during the years 2014, 2016, and 2018 combined. The model showed effect of diabetes on heart diseases.

Overall, this underscores the importance of targeted interventions to reduce diabetes rates, which, in turn, may prevent deaths caused by heart disease.

DISCUSSION

Initially, a multi linear regression model was designed with the variables physical environment (X1) and economics (X2) as independent variables and obesity (y) as dependent variable). Similarly, we also used transport (X1) and food access (X2) to predict Physical-inactivity(y). The result of R-squared values was 0.002 and -0.0007 for obesity and inactivity, respectively, which is unexceptionally low and correlation was inconsistent.

However, the mortality due to heart disease showed consistent results which confirms the theory of diabetes association with mortality caused due to heart disease. These findings suggest that diabetes is indirectly related to death occurring due to heart disease. A bar graph was plotted to analyze the trend and correlation of mortality due to heart disease and diabetes. The states having the lowest mortality rate also had the lowest diabetes rate. To confirm this state-wise analysis, the average of all county's values was taken as the data value for each state.

We discovered that diabetes was a contributor in the deaths of those who suffered from heart disease. We need to further our understanding of heart disease in order to improve our model's potential to fit complex datasets in the future, despite the fact it is a powerful predictor of mortality from heart disease when combined with diabetes data. The p-value indicates that the model makes accurate predictions without any over-fitting of the data. Overall, our model is designed to forecast future data provided we have previous year data and enough dataset to train the model with. The prediction rate and efficiency can be improved on training the model with a better and large dataset.

Appendix A: METHOD

Data collection:

Data was downloaded from the Centers of Disease and Control Prevention (CDC) website as a csv file. The file was then imported into Jupyter Notebook, and we have additionally downloaded the heart disease mortality data from the CDC website. "https://data.cdc.gov/Heart-Disease-Stroke-Prevention/Heart-Disease-Mortality-Data-Among-US-Adults-35-by/s6p7-fvbw"

***Variable creation**: The 2 variables used here are "Heart_Average" and "Diabetes_Average", This represents the average of all the counties present in a state for all the heart and diabetes data. Which acts as a data point for our analysis.

Data Cleaning:

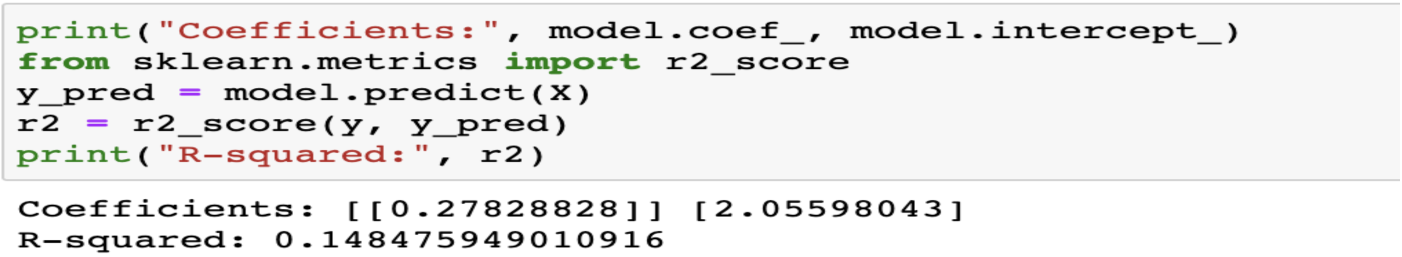
- Null values were removed from the datasets to ensure data accuracy.
- State codes were replaced with state names.
- County-level data was aggregated to the state level by calculating averages. This aggregation simplifies the analysis and reduces data complexity.
- Created a data frame with two variables and analysis was performed

Analytic methods:

- Mathematical statistics: measurement of central tendency (i.e., mean, p-value) coefficient of determination (R-squared) for both the average of heart disease and diabetes.
- Bar graph: For the purpose of visually illustrating relationships and patterns within the data, to compare the heart disease and diabetes datasets bar graphs were created.
- Scatter plot: scatter plots were used to identify the outliers from the dataset of mortality of heart disease and diabetes.
- Linear Regression: We had 2 variables i.e., mortality of heart disease and diabetes. So, naturally, simple linear regression was the best suited model for our analysis. It makes the estimation procedure simple.
- Overall, analytics method will give good picture of understanding the relationship between mortality of heart disease and diabetes data.

Appendix B: RESULTS

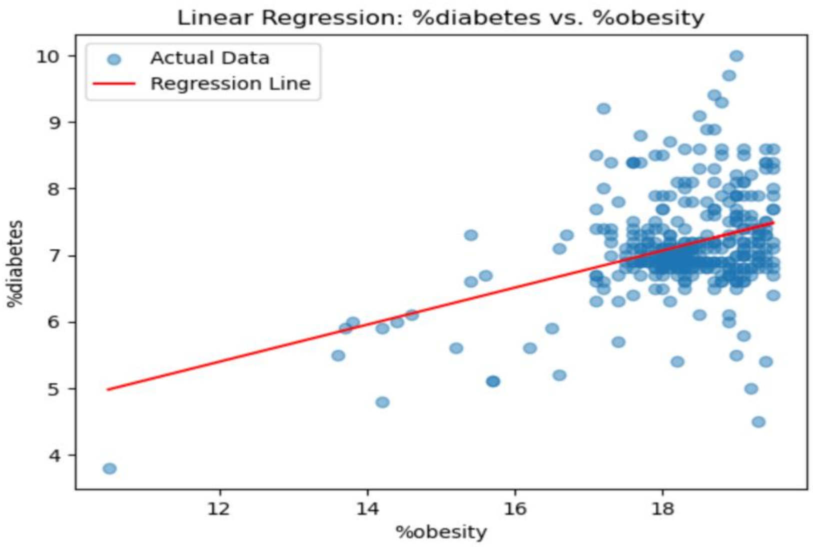
In this analysis, We have started with the correlation between diabetes versus inactivity and obesity. Which results in low R-square value which signifies our independent variable is not dependent to our dependent variable.

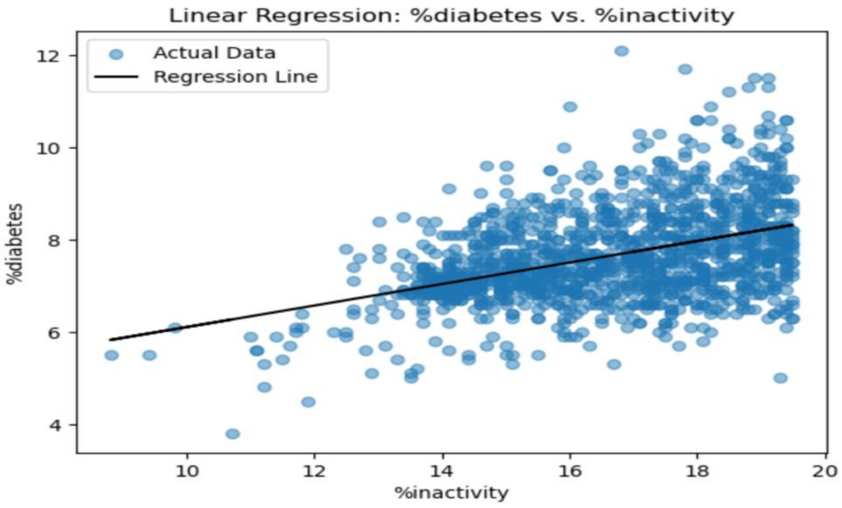


```
print("Coefficients:", model.coef_, model.intercept_)
from sklearn.metrics import r2_score
y_pred = model.predict(X)
r2 = r2_score(y, y_pred)
print("R-squared:", r2)
```

```
Coefficients: [[0.27828828]] [2.05598043]
R-squared: 0.148475949010916
```

With the scatter plot it got clear that the relationship between an independent and dependent variable are not good as the dataset is not evenly distributed on the line of best fit. The accumulation of data set at one place is not helping to get the prediction of further analysis.

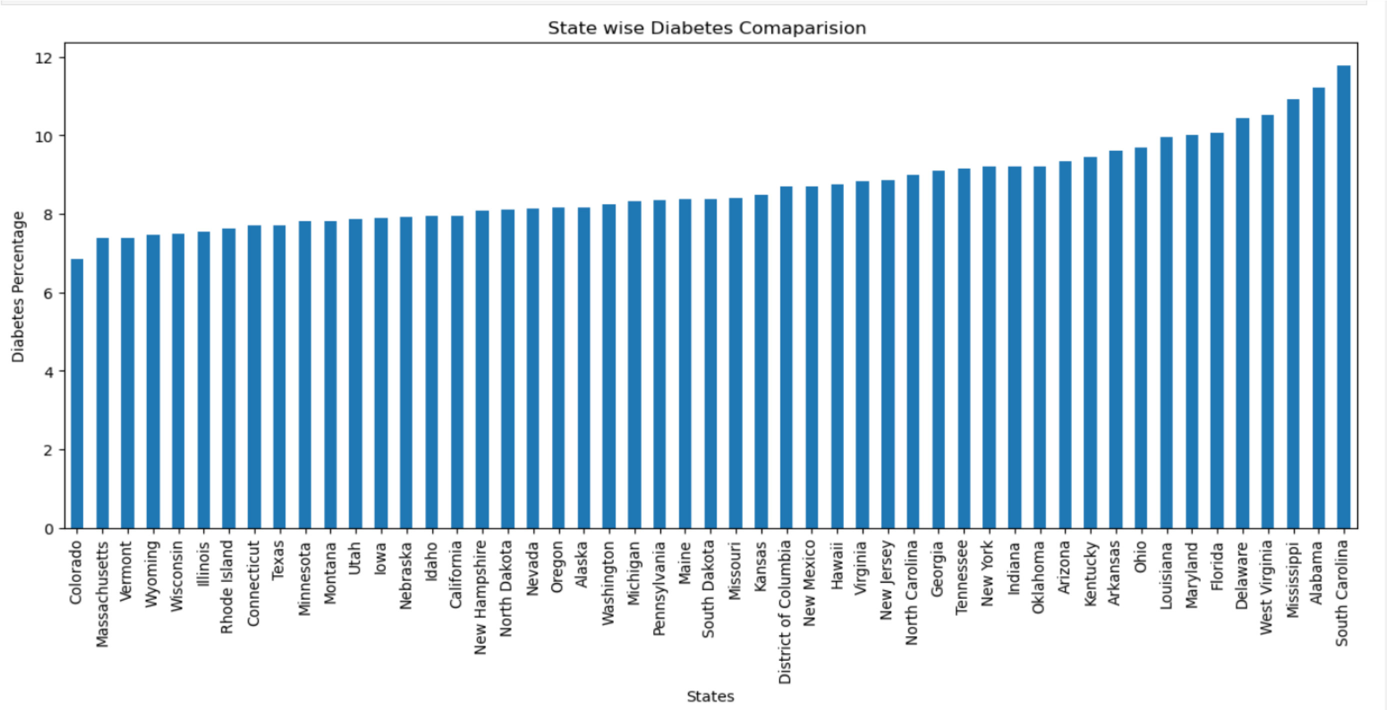


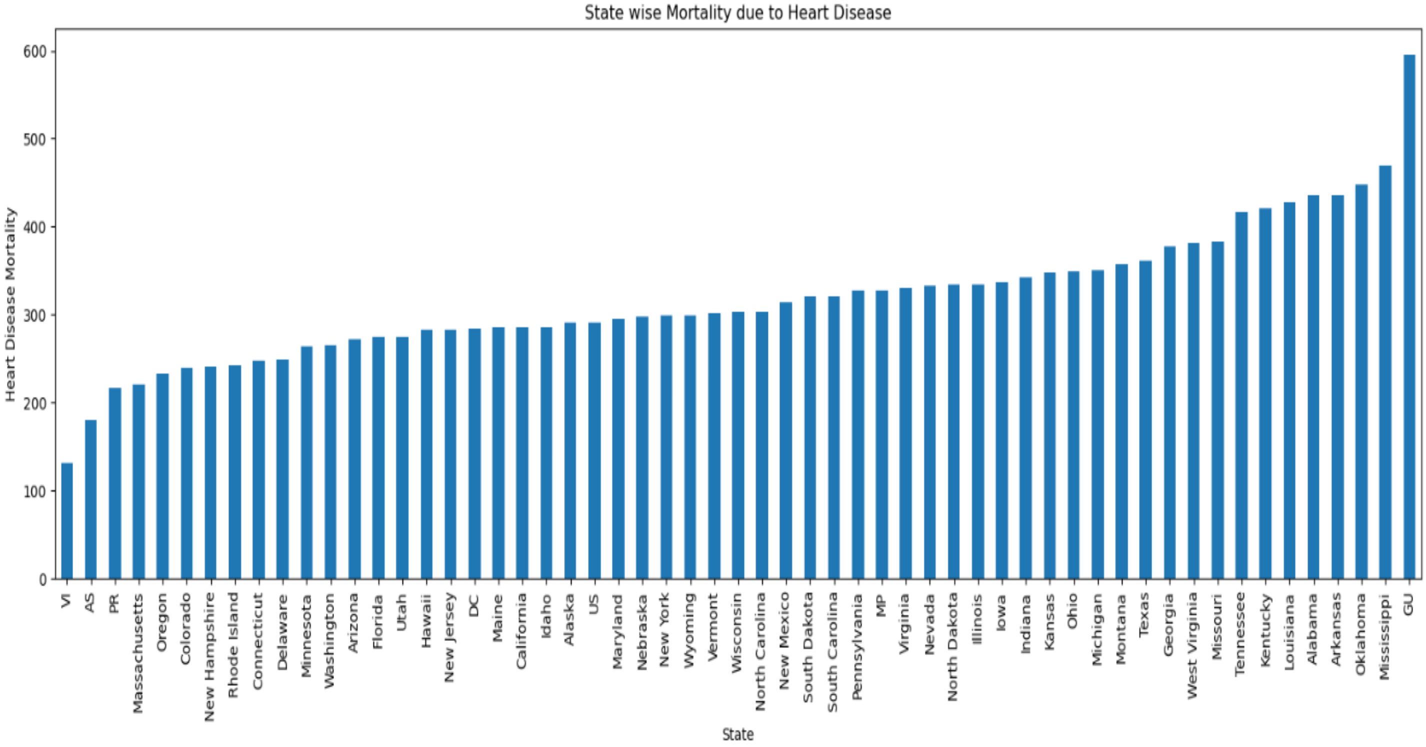


The result section, will start with comparison of the trend in each state of mortality due to heart disease and diabetic below bar graph are giving 1st glance of relationship between diabetic and mortality due to heart disease.

The calculation of mean value for all the counties data of each state is telling that the group of states with higher diabetes value have higher mortality due to heart disease.

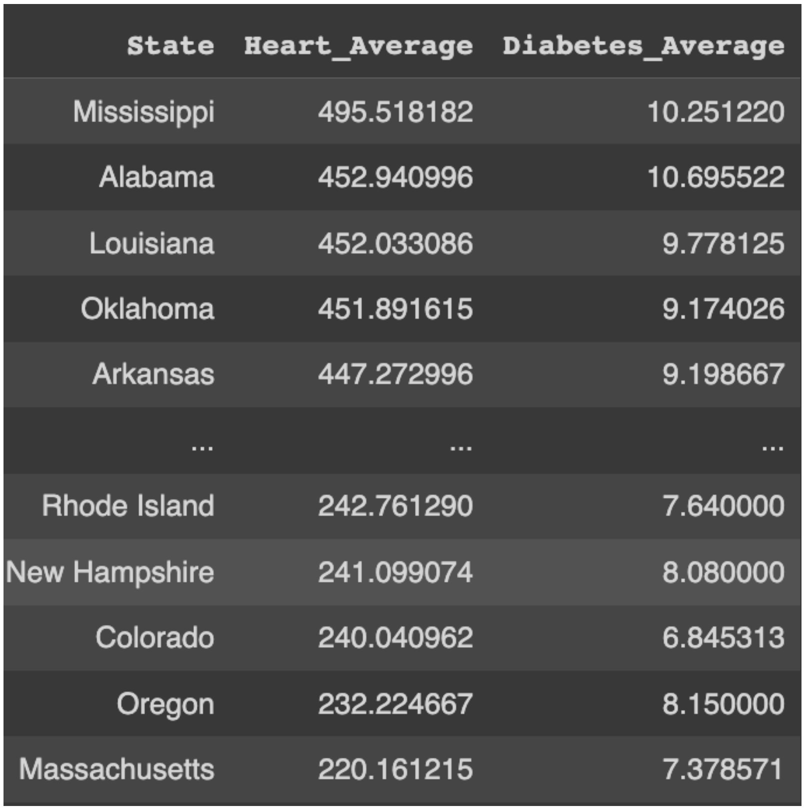
$$\text{mean} = \overline{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$





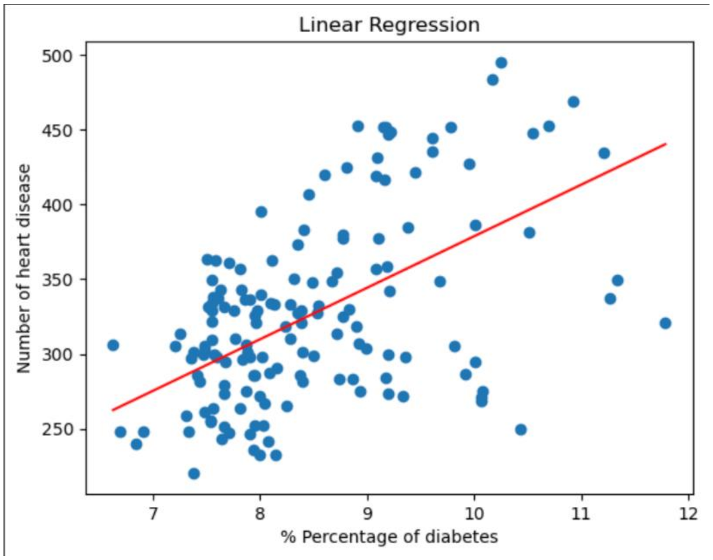
Above bar graph also represented that we have few extra state in our data which has no relation so we have extracted it.

We have created new dataset, one from mortality dataset and other from diabetes.

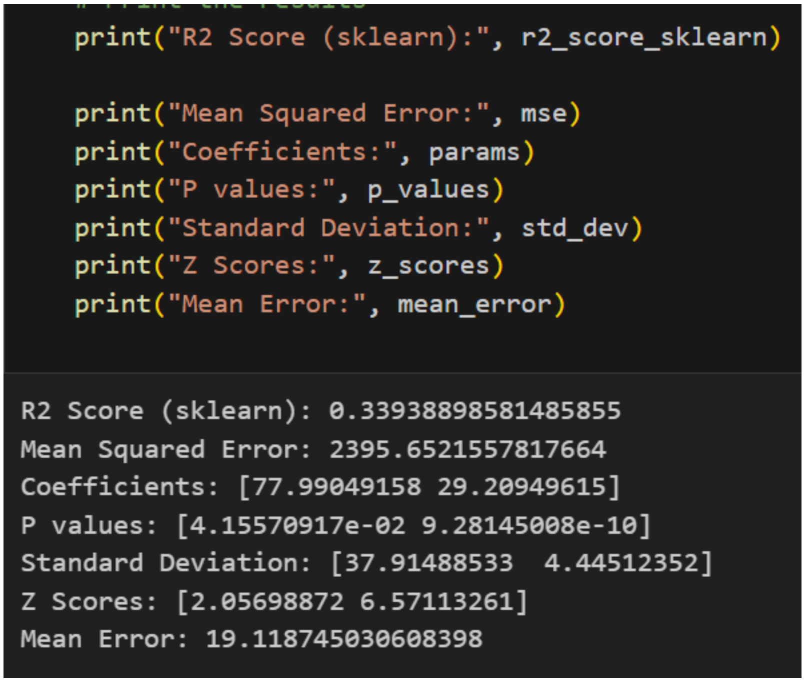


State	Heart_Average	Diabetes_Average
Mississippi	495.518182	10.251220
Alabama	452.940996	10.695522
Louisiana	452.033086	9.778125
Oklahoma	451.891615	9.174026
Arkansas	447.272996	9.198667
...	...	...
Rhode Island	242.761290	7.640000
New Hampshire	241.099074	8.080000
Colorado	240.040962	6.845313
Oregon	232.224667	8.150000
Massachusetts	220.161215	7.378571

Next, results of simple linear regression analysis which has Diabetes_Average as independent variable(X) and Heart_Average as dependent variable(Y) in the equation **Y=mX + C.** This would include the least squares linear regression line plotted. This represent that there is some relation between selected variables because data points are spread across the line and it also tells that the outliners are present in this data with will reduce the chances of good analysis



We used OLS (Ordinary Least Squares) Regression Results to get the details about the relationship between two variables.



```
    # Print the results
    print("R2 Score (sklearn):", r2_score_sklearn)

    print("Mean Squared Error:", mse)
    print("Coefficients:", params)
    print("P values:", p_values)
    print("Standard Deviation:", std_dev)
    print("Z Scores:", z_scores)
    print("Mean Error:", mean_error)
```

```
R2 Score (sklearn): 0.33938898581485855
Mean Squared Error: 2395.6521557817664
Coefficients: [77.99049158 29.20949615]
P values: [4.15570917e-02 9.28145008e-10]
Standard Deviation: [37.91488533  4.44512352]
Z Scores: [2.05698872 6.57113261]
Mean Error: 19.118745030608398
```

The R-squared value of 0.33 indicates that approximately 33% of the variability in the dependent unit that is mortality of heart disease can be explained by the independent unit that is diabetes, which is giving moderate relationship between the two variables.

The coefficient estimation is [77.99 29.20] which represent a strong relationship between the two variables. on average, this indicates one-unit increase of independent variable, will lead the dependent variable to increase by 29 units.

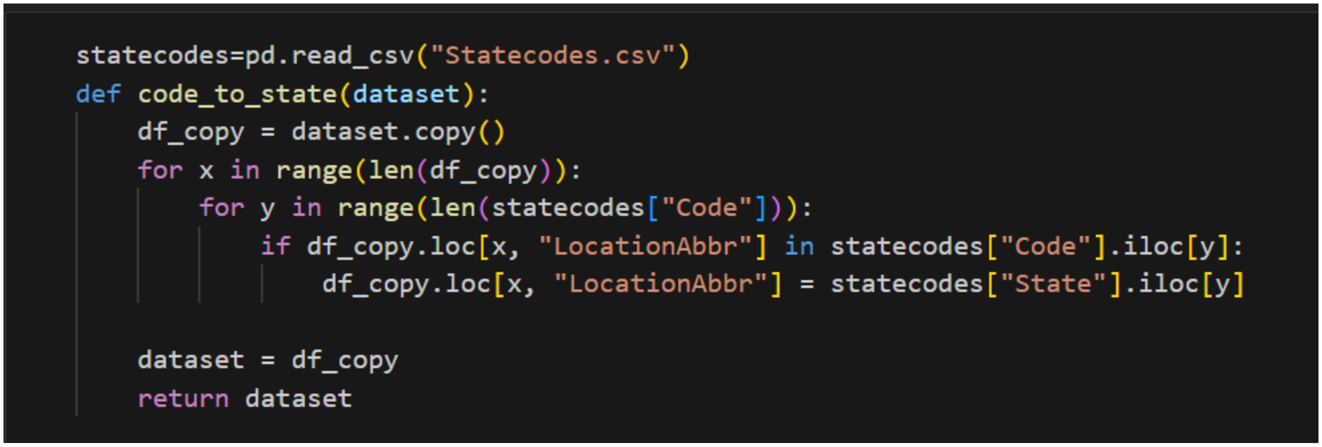
The P-value data is [4.15570917e-02 9.28145008e-10], ideally this data should lie between 0.001 to 0.05 in our case rounding to four decimal place the P-value is 0.0416 which will be consider as statistically significant and also it will tell us that it can be used as sample for the larger population.

The Z Scores value is [2.05698872 6.57113261] which is telling about the deviation of our data from the mean and the minimum value in our model is 2 which suggests a moderate deviation from the mean and higher value is 6 which indicates an extremely large deviation, probably because of outliners present in our dataset.

Appendix C: Data and Code

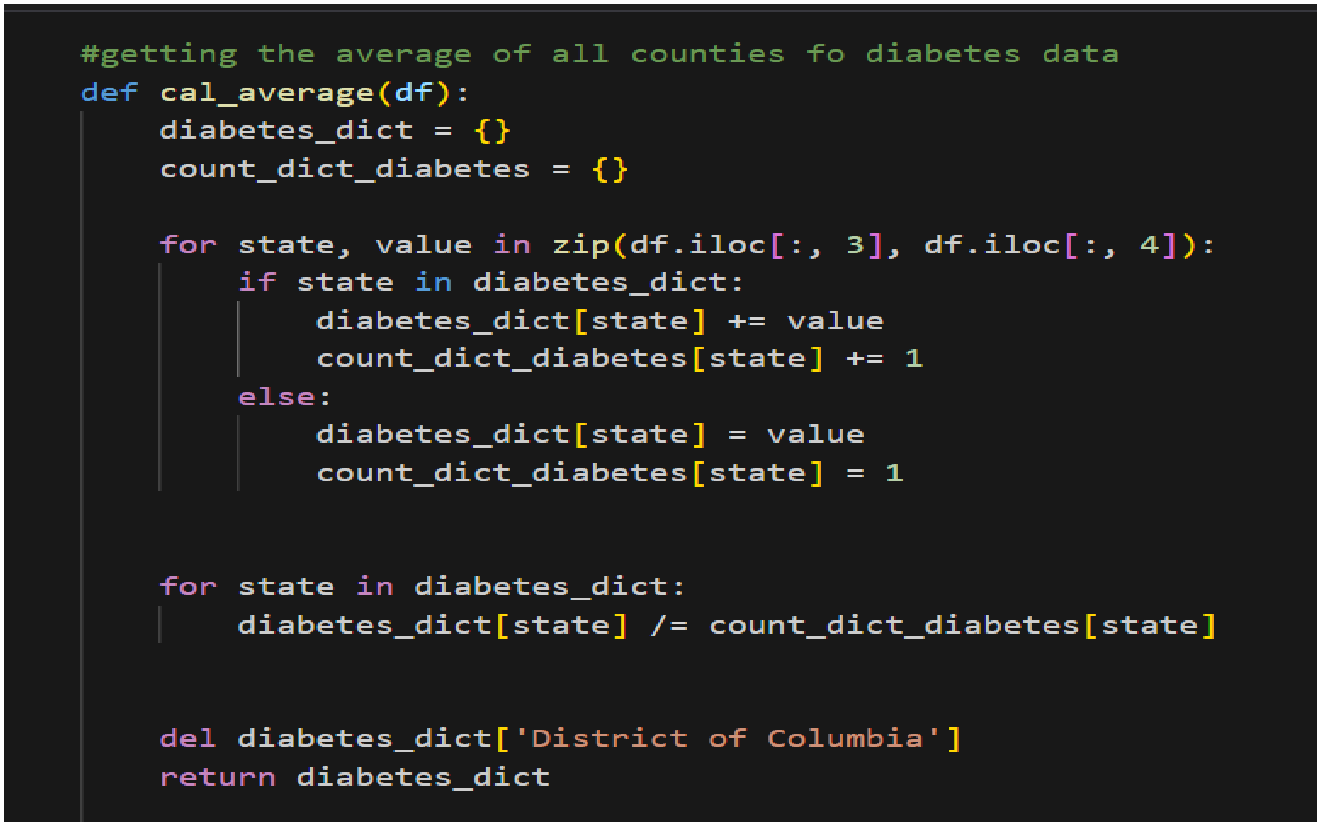
In this appendix anyone can replicate our analysis with the help of python code. Use the git hub repository https://github.com/Milkyy-way/-Analysis-of-Heart-Disease-Mortality-Using-Diabetes-Data.git

1) Data cleaning for converting state code into state



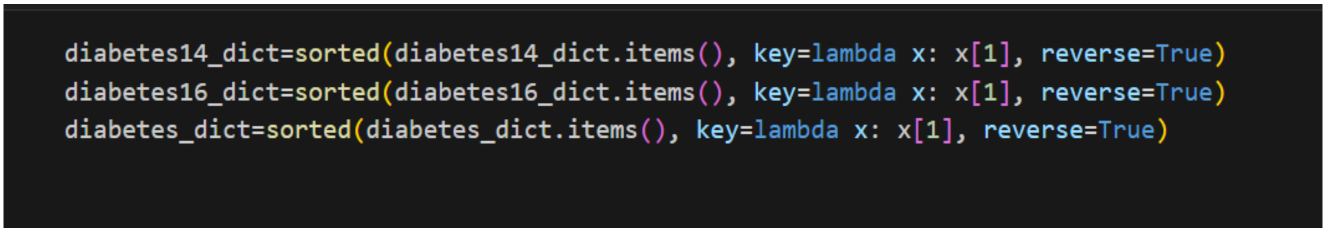
```
statecodes=pd.read_csv("Statecodes.csv")
def code_to_state(dataset):
    df_copy = dataset.copy()
    for x in range(len(df_copy)):
        for y in range(len(statecodes["Code"])):
            if df_copy.loc[x, "LocationAbbr"] in statecodes["Code"].iloc[y]:
                df_copy.loc[x, "LocationAbbr"] = statecodes["State"].iloc[y]

    dataset = df_copy
    return dataset
```

2) Taking average



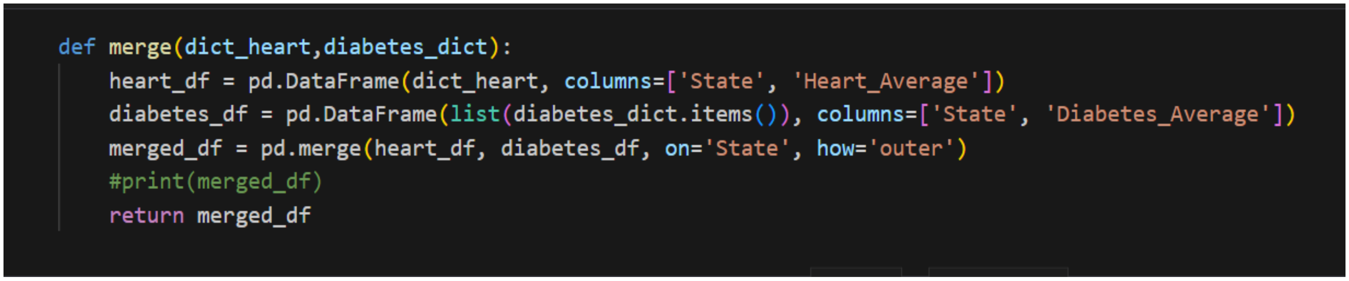
```
#getting the average of all counties fo diabetes data
def cal_average(df):
    diabetes_dict = {}
    count_dict_diabetes = {}

    for state, value in zip(df.iloc[:, 3], df.iloc[:, 4]):
        if state in diabetes_dict:
            diabetes_dict[state] += value
            count_dict_diabetes[state] += 1
        else:
            diabetes_dict[state] = value
            count_dict_diabetes[state] = 1


    for state in diabetes_dict:
        diabetes_dict[state] /= count_dict_diabetes[state]


    del diabetes_dict['District of Columbia']
    return diabetes_dict
```

3) Sorting



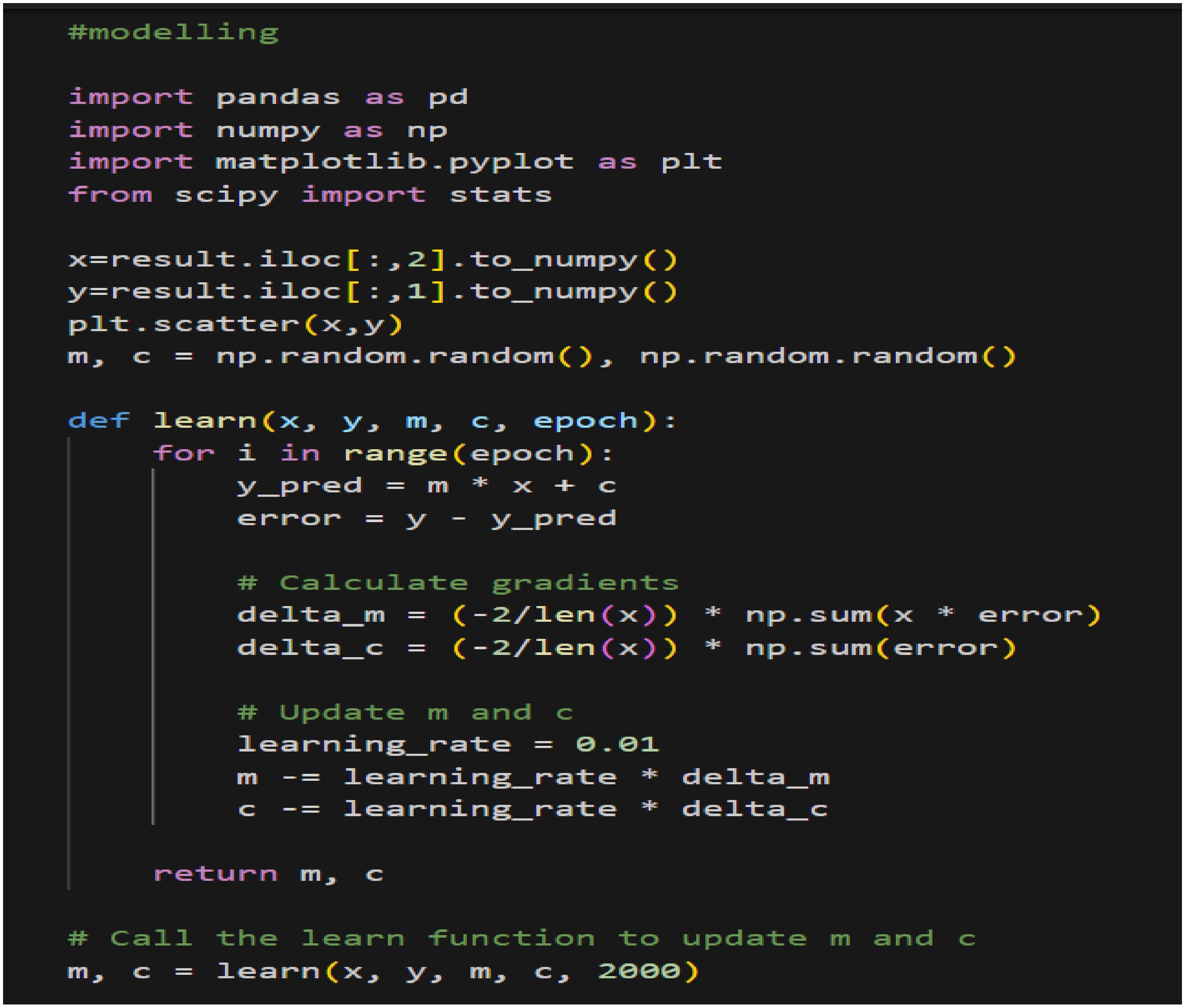
```
diabetes14_dict=sorted(diabetes14_dict.items(), key=lambda x: x[1], reverse=True)
diabetes16_dict=sorted(diabetes16_dict.items(), key=lambda x: x[1], reverse=True)
diabetes_dict=sorted(diabetes_dict.items(), key=lambda x: x[1], reverse=True)
```

4) Merging the two data frame

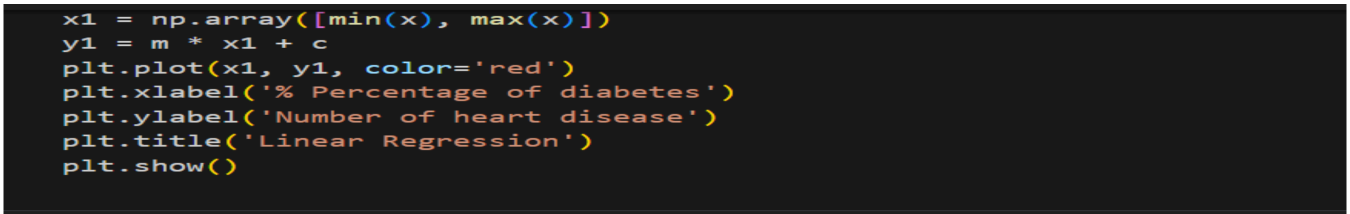


```
def merge(dict_heart,diabetes_dict):
    heart_df = pd.DataFrame(dict_heart, columns=['State', 'Heart_Average'])
    diabetes_df = pd.DataFrame(list(diabetes_dict.items()), columns=['State', 'Diabetes_Average'])
    merged_df = pd.merge(heart_df, diabetes_df, on='State', how='outer')
    #print(merged_df)
    return merged_df
```

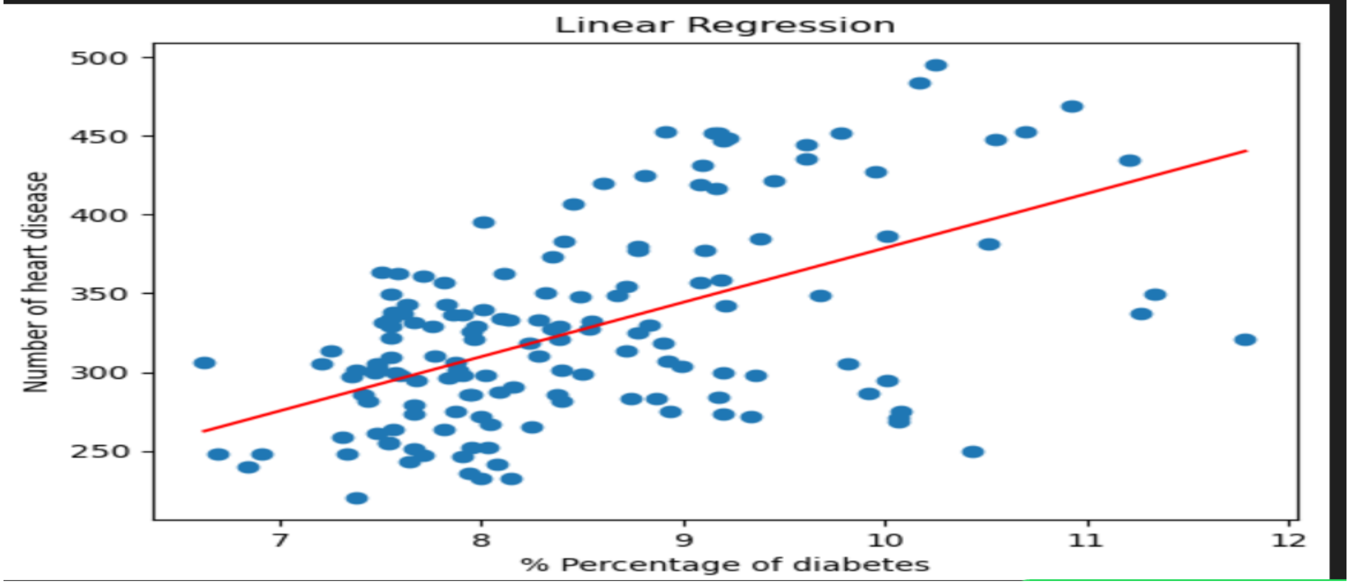
5) Modeling



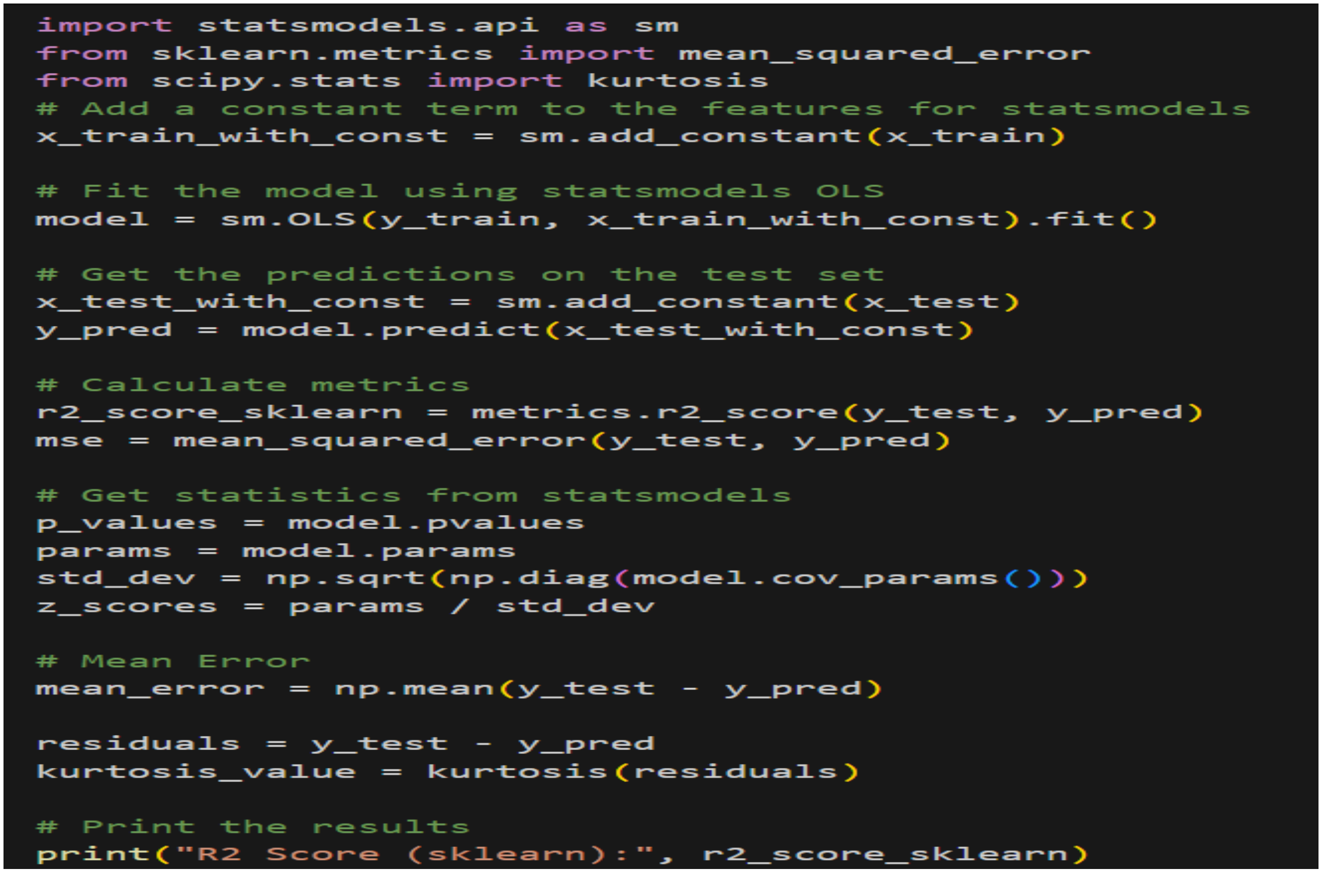
```
#modelling

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats

x=result.iloc[:,2].to_numpy()
y=result.iloc[:,1].to_numpy()
plt.scatter(x,y)
m, c = np.random.random(), np.random.random()

def learn(x, y, m, c, epoch):
    for i in range(epoch):
        y_pred = m * x + c
        error = y - y_pred

        # Calculate gradients
        delta_m = (-2/len(x)) * np.sum(x * error)
        delta_c = (-2/len(x)) * np.sum(error)

        # Update m and c
        learning_rate = 0.01
        m -= learning_rate * delta_m
        c -= learning_rate * delta_c

    return m, c

# Call the learn function to update m and c
m, c = learn(x, y, m, c, 2000)
```



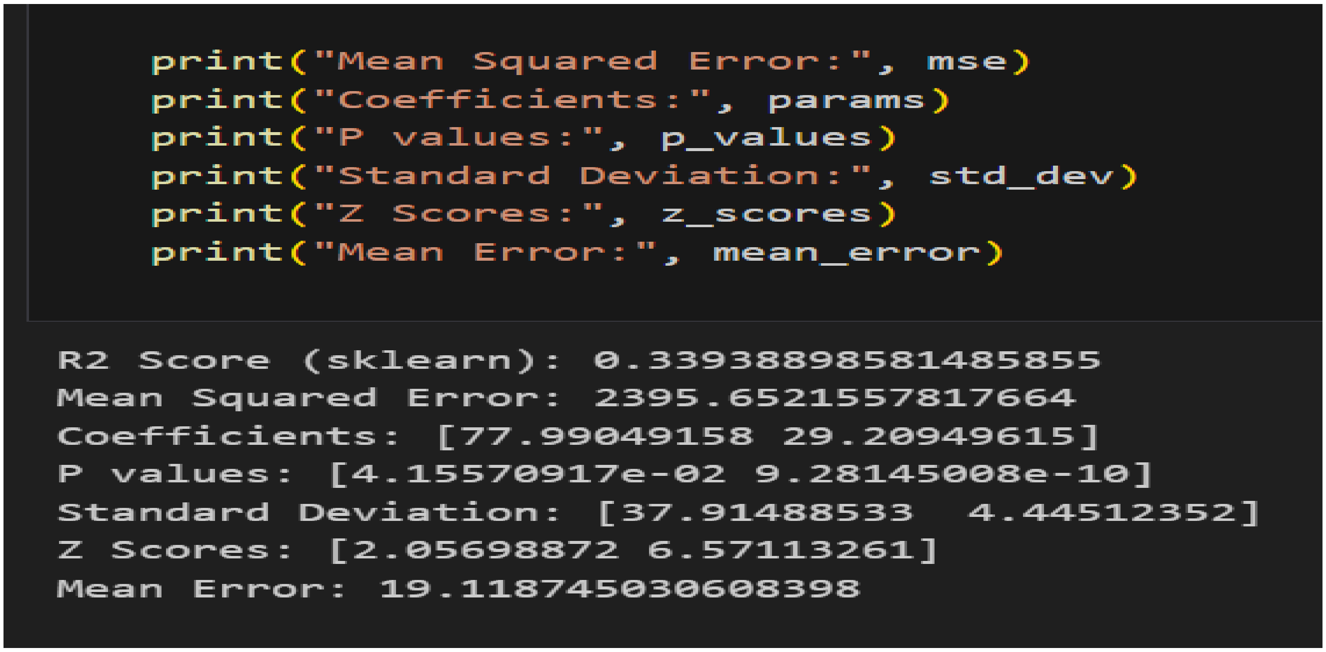
```
    x1 = np.array([min(x), max(x)])
    y1 = m * x1 + c
    plt.plot(x1, y1, color='red')
    plt.xlabel('% Percentage of diabetes')
    plt.ylabel('Number of heart disease')
    plt.title('Linear Regression')
    plt.show()
```



6) Calculating Statistic using OLS



```
import statsmodels.api as sm
from sklearn.metrics import mean_squared_error
from scipy.stats import kurtosis
# Add a constant term to the features for statsmodels
x_train_with_const = sm.add_constant(x_train)

# Fit the model using statsmodels OLS
model = sm.OLS(y_train, x_train_with_const).fit()

# Get the predictions on the test set
x_test_with_const = sm.add_constant(x_test)
y_pred = model.predict(x_test_with_const)

# Calculate metrics
r2_score_sklearn = metrics.r2_score(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)

# Get statistics from statsmodels
p_values = model.pvalues
params = model.params
std_dev = np.sqrt(np.diag(model.cov_params()))
z_scores = params / std_dev

# Mean Error
mean_error = np.mean(y_test - y_pred)

residuals = y_test - y_pred
kurtosis_value = kurtosis(residuals)

# Print the results
print("R2 Score (sklearn):", r2_score_sklearn)
```



```
    print("Mean Squared Error:", mse)
    print("Coefficients:", params)
    print("P values:", p_values)
    print("Standard Deviation:", std_dev)
    print("Z Scores:", z_scores)
    print("Mean Error:", mean_error)
```

```
R2 Score (sklearn): 0.33938898581485855
Mean Squared Error: 2395.6521557817664
Coefficients: [77.99049158 29.20949615]
P values: [4.15570917e-02 9.28145008e-10]
Standard Deviation: [37.91488533  4.44512352]
Z Scores: [2.05698872 6.57113261]
Mean Error: 19.118745030608398
```

Contribution:

	Shruti	Pawan	Afroza	Saurav
Analysis	25%	25%	25%	25%
Coding	30%	20%	20%	30%
Report Writing	20%	30%	30%	20%