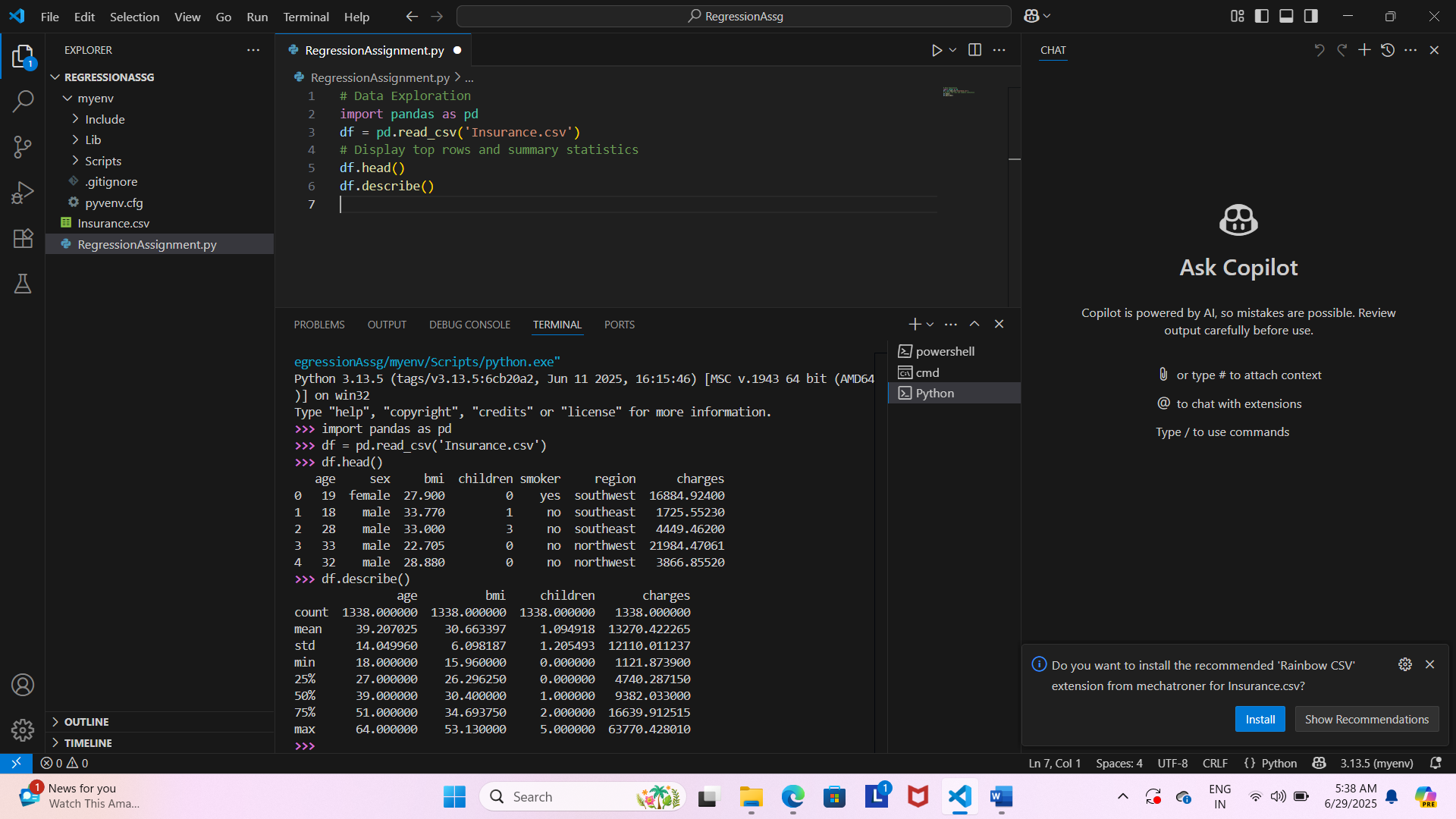
Exercise on Linear Regression Analysis

Exercise 1. Data Cleaning and Exploratory Data Analysis



A screenshot of a computer

AI-generated content may be incorrect.

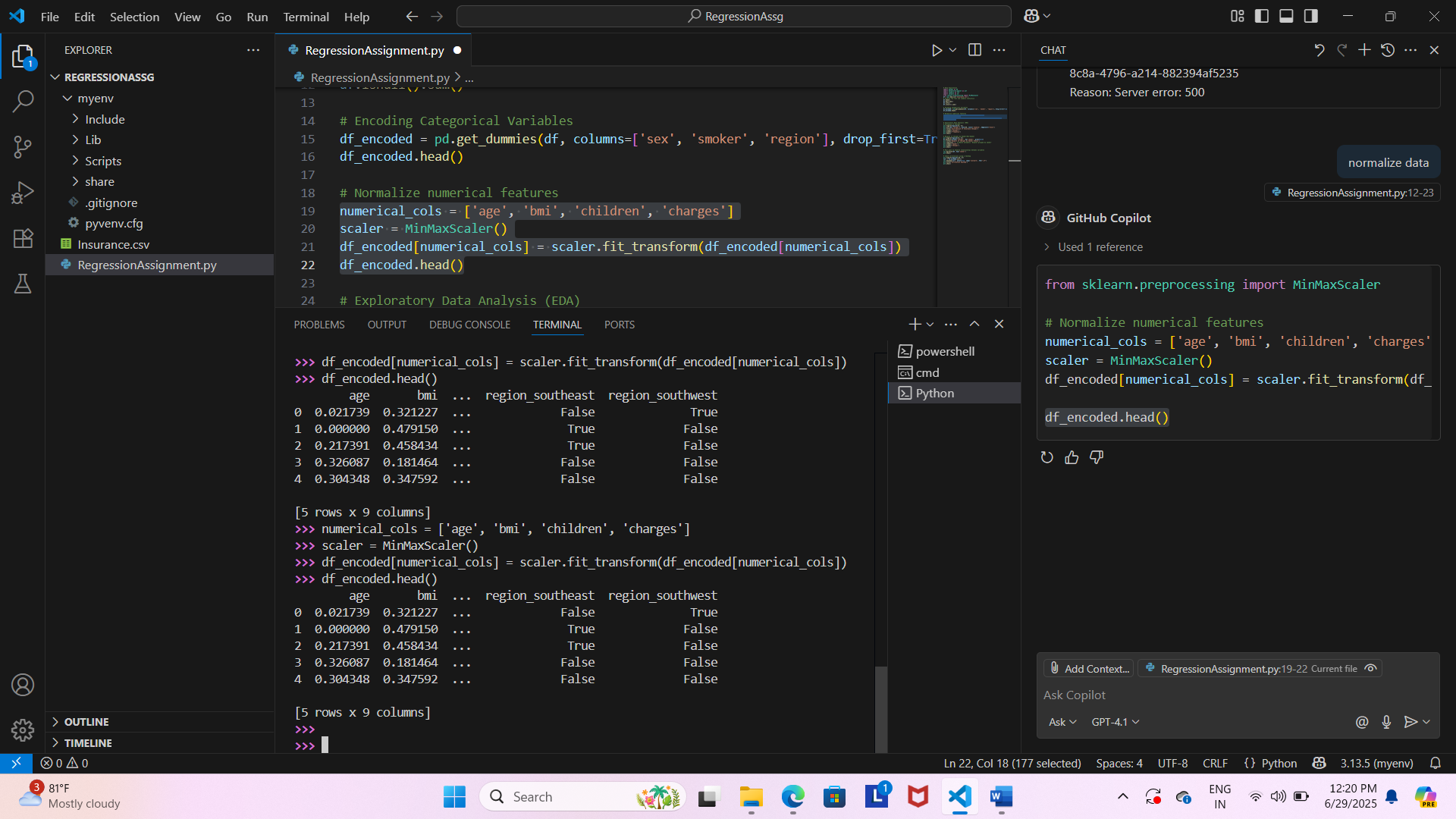
Explain categorical variables:

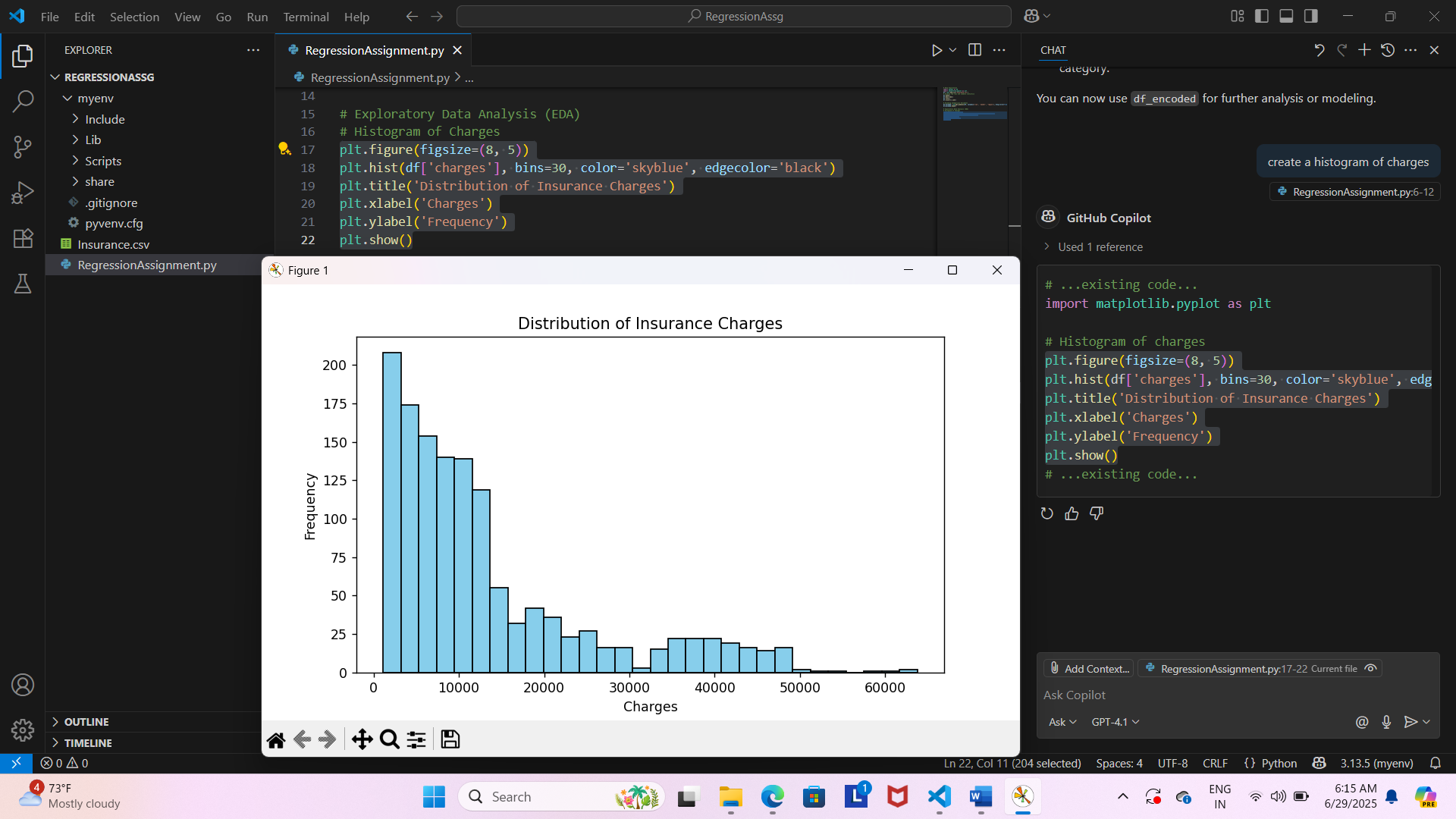
In the dataset **categorical variables** are columns that contain discrete values representing categories rather than numbers. These can be identified by running ‘df.info()’. Identify categorical variables by looking for columns with the data type object.

In our insurance datasets, the categorical variables are:

* **sex** (e.g., 'male', 'female')
* **smoker** (e.g., 'yes', 'no')
* **region** (e.g., 'northeast', 'southeast', 'southwest', 'northwest')

These columns contain text values representing categories and not numbers.  
Categorical data represents discrete values or categories, such as gender, country, or product type. Machine learning algorithms, however, require numerical input, making it essential to convert categorical data into a numerical format. This process is known as encoding. Categorical variables often need to be encoded (e.g., using one-hot encoding or label encoding) before being used in machine learning models.





A computer screen shot of a computer screen

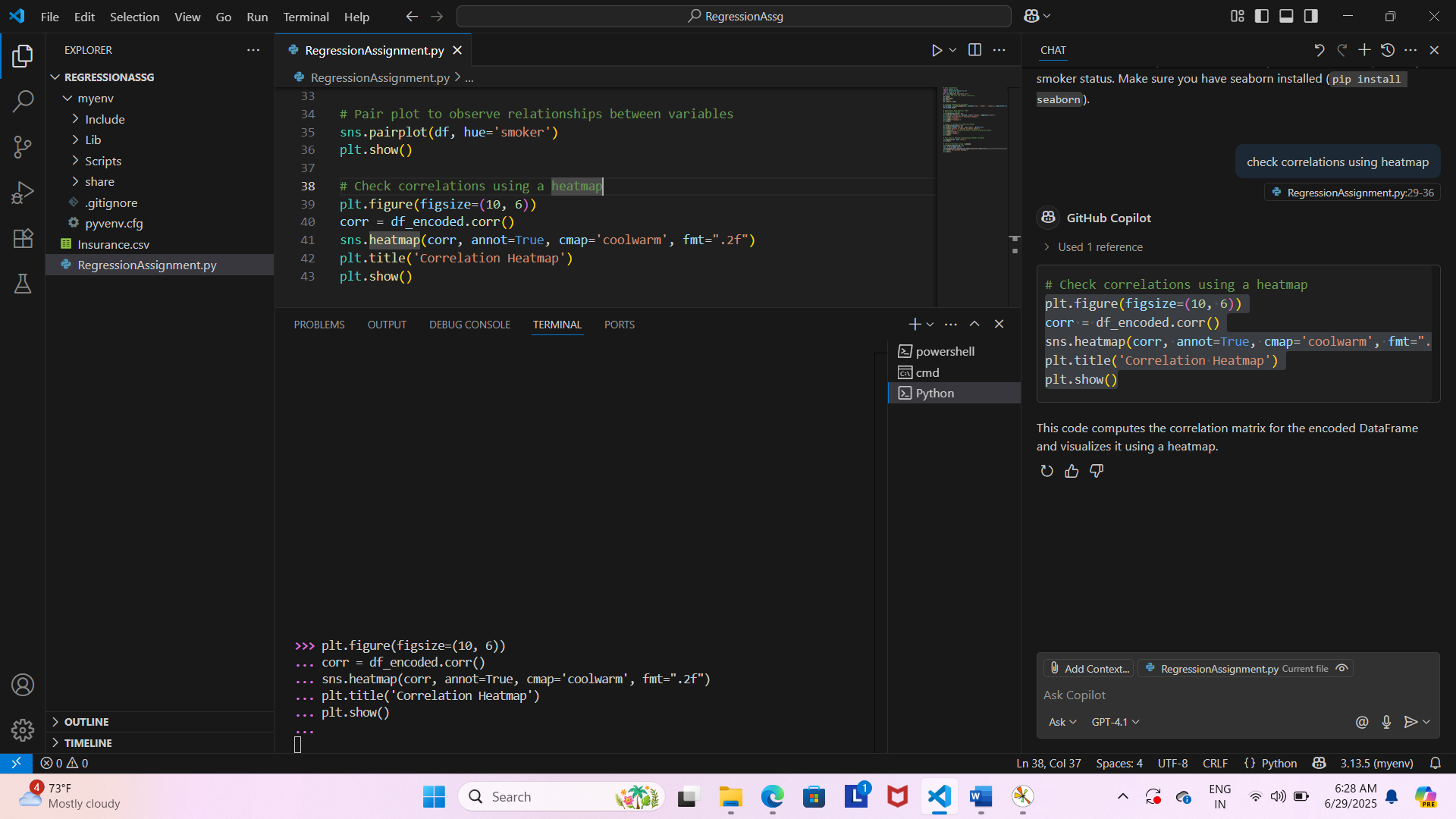
AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.



A screenshot of a computer

AI-generated content may be incorrect.

**Brief report describing key observations from data exploration:**

Based on exploratory data analysis the insights suggests that smoking status, age and BMI are important predictors of insurance charges.

Age: Age is positively correlated with insurance charges, older individuals pay higher insurance charges. Hence, insurance charge would increase with increase in age.

BMI: Higher BMI would result in higher insurance charges.

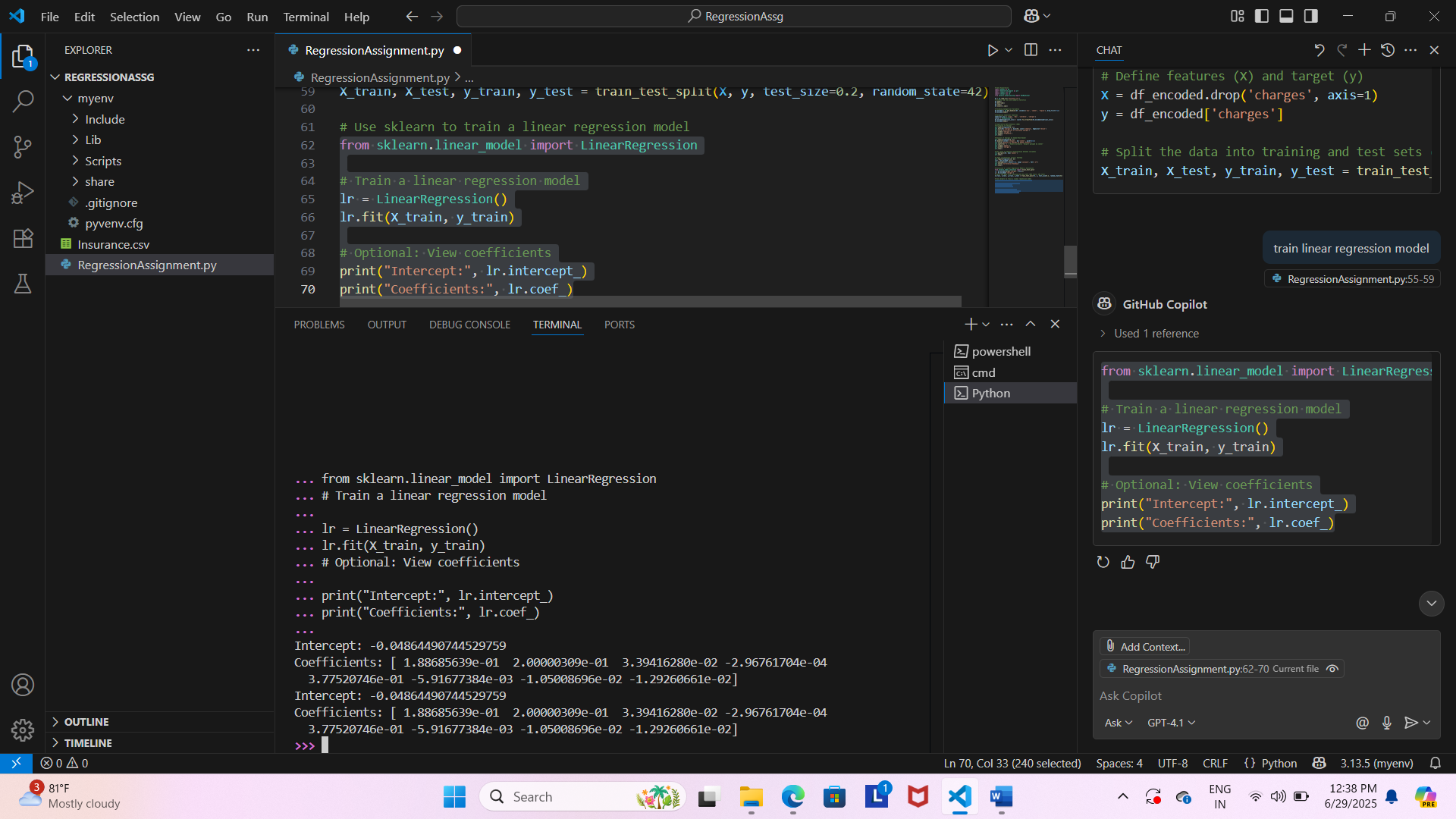
Smoking Status: This is the most effecting factor and very strong predictor. Smokers have much high insurance changes compared to non-smokers.

Other factors like Children, Sex or Region does not show a clear trend to analyze there impact on insurance charges.

Exercise 2: Linear Regression model & Evaluation

A screenshot of a computer

AI-generated content may be incorrect.



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

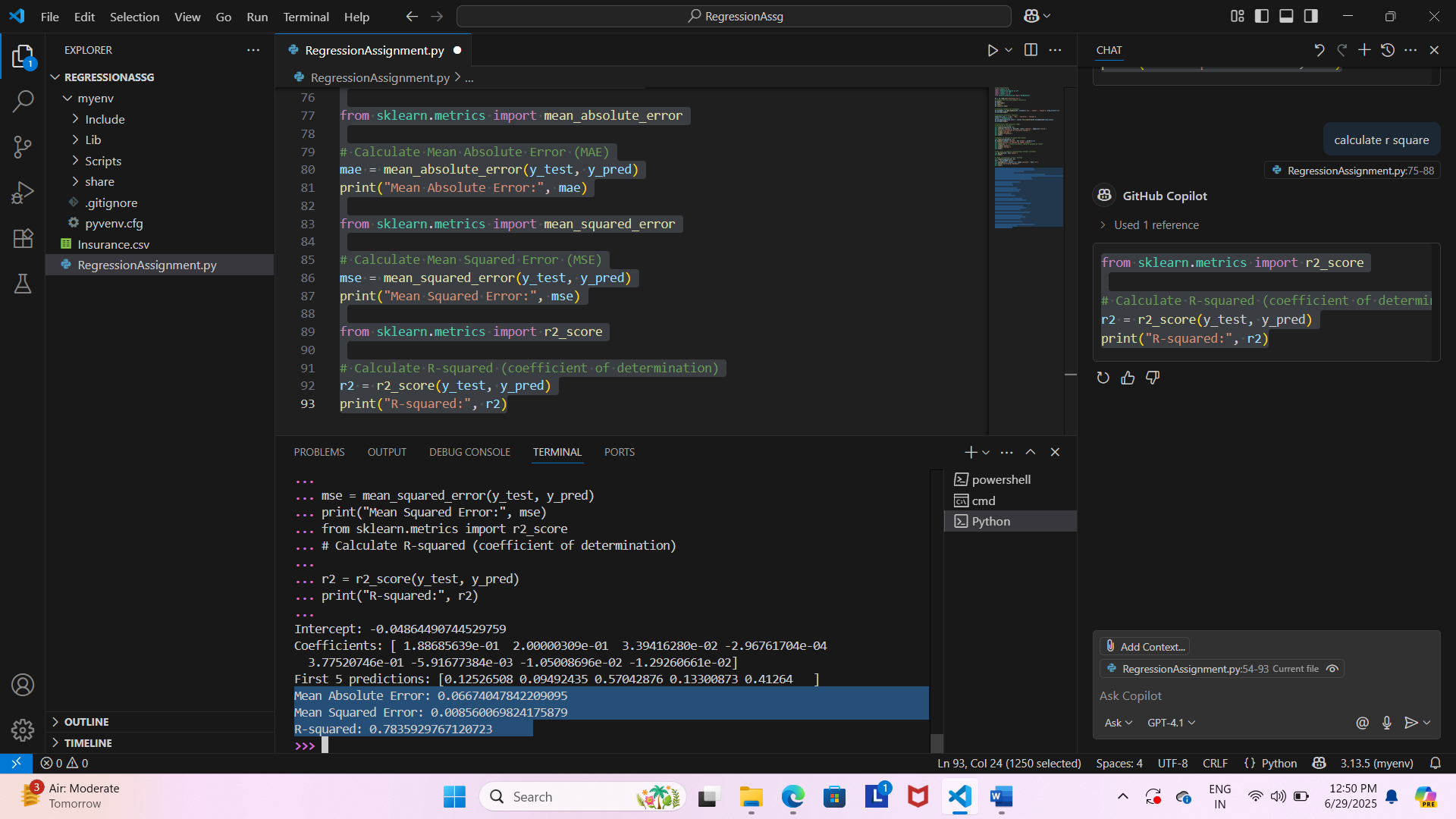
AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.



A screen shot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Interpreting the insights from linear regression model:**

The R-Squared value of 0.7835929767120723 suggests that the model explains approximately 78% of the variance in insurance changes, which is good.

Lower values of ‘Mean Squared Error’ and ‘Mean Absolute Error’

**Python Code for Linear Regression**

# Exercise 1: Data Cleaning & Exploratory Data Analysis

# Data Exploration

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import sklearn as sk

from sklearn.preprocessing import MinMaxScaler

df = pd.read\_csv('Insurance.csv')

# Display top rows and summary statistics

df.head()

df.describe()

df.info()

df.isnull().sum()

# Encoding Categorical Variables

df\_encoded = pd.get\_dummies(df, columns=['sex', 'smoker', 'region'], drop\_first=True)

df\_encoded.head()

# Normalize numerical features

numerical\_cols = ['age', 'bmi', 'children', 'charges']

scaler = MinMaxScaler()

df\_encoded[numerical\_cols] = scaler.fit\_transform(df\_encoded[numerical\_cols])

df\_encoded.head()

# Exploratory Data Analysis (EDA)

# Histogram of Charges

plt.figure(figsize=(8, 5))

plt.hist(df['charges'], bins=30, color='skyblue', edgecolor='black')

plt.title('Distribution of Insurance Charges')

plt.xlabel('Charges')

plt.ylabel('Frequency')

plt.show()

# Boxplot of Charges by Smoker/Non-Smoker

plt.figure(figsize=(8, 5))

df.boxplot(column='charges', by='smoker', grid=False)

plt.title('Boxplot of Charges by Smoker Status')

plt.suptitle('')  # Remove automatic 'Boxplot grouped by smoker'

plt.xlabel('Smoker')

plt.ylabel('Charges')

plt.show()

# Pair plot to observe relationships between variables

sns.pairplot(df, hue='smoker')

plt.show()

# Check correlations using a heatmap

plt.figure(figsize=(10, 6))

corr = df\_encoded.corr()

sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f")

plt.title('Correlation Heatmap')

plt.show()

# Excercise 2: Linear Regression Model & Evaluation

from sklearn.model\_selection import train\_test\_split

X = df\_encoded.drop('charges', axis=1)

y = df\_encoded['charges']

# Split the data into training and test sets (80% train, 20% test)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Use sklearn to train a linear regression model

from sklearn.linear\_model import LinearRegression

# Train a linear regression model

lr = LinearRegression()

lr.fit(X\_train, y\_train)

# View coefficients

print("Intercept:", lr.intercept\_)

print("Coefficients:", lr.coef\_)

# Predict on test set

y\_pred = lr.predict(X\_test)

# View first 5 predictions

print("First 5 predictions:", y\_pred[:5])

from sklearn.metrics import mean\_absolute\_error

# Calculate Mean Absolute Error (MAE)

mae = mean\_absolute\_error(y\_test, y\_pred)

print("Mean Absolute Error:", mae)

from sklearn.metrics import mean\_squared\_error

# Calculate Mean Squared Error (MSE)

mse = mean\_squared\_error(y\_test, y\_pred)

print("Mean Squared Error:", mse)

from sklearn.metrics import r2\_score

# Calculate R-squared (coefficient of determination)

r2 = r2\_score(y\_test, y\_pred)

print("R-squared:", r2)

# Visualize Actual vs Predicted Charges

plt.figure(figsize=(8, 5))

plt.scatter(y\_test, y\_pred, color='teal', alpha=0.6)

plt.plot([y\_test.min(), y\_test.max()], [y\_test.min(), y\_test.max()], color='red', linestyle='--')

plt.xlabel('Actual Charges')

plt.ylabel('Predicted Charges')

plt.title('Actual vs Predicted Insurance Charges')

plt.show()

# Residual plot

residuals = y\_test - y\_pred

plt.figure(figsize=(8, 5))

plt.scatter(y\_pred, residuals, color='purple', alpha=0.6)

plt.axhline(0, color='red', linestyle='--')

plt.xlabel('Predicted Charges')

plt.ylabel('Residuals')

plt.title('Residual Plot')

plt.show()