Statistical Analysis in Python with Real-World Datasets

Introduction

Statistical analysis is a fundamental aspect of data science and machine learning. By understanding and applying statistical methods, you can derive meaningful insights from datasets, make predictions, and inform decision-making. In this session, we will explore the basics of statistical analysis using Python, leveraging libraries like pandas, numpy, and scipy. We will use various datasets, including the built-in Diabetes dataset from the sklearn library, and explore how to apply statistical methods to datasets from sources like Kaggle and the UCI Machine Learning Repository.

Basic Concepts in Statistics

Population and Sample

- Population: The complete set of all possible observations or measurements.
- Sample: A subset of the population selected for the actual study.

Need for Sampling in Statistics

- **Cost Efficiency**
- **Time Efficiency**
- **Feasibility**
- **Manageability**

Benefits of Sampling

- **Accuracy**
- **Less Data Overhead**
- **Focused Research**

Descriptive Statistics

Measures of Central Tendency

- **Mean**: The average value.
- **Median**: The middle value.
- **Mode**: The most frequent value.

Measures of Variability

- **Standard Deviation**: Measures the amount of variation.
- **Variance**: The average of the squared differences from the mean.
- **Range**: The difference between the maximum and minimum values.

Additional Descriptive Measures

- **Percentiles and Quartiles**
- **Interquartile Range (IQR)**
- **Skewness**
- **Kurtosis**

Inferential Statistics

- **Hypothesis Testing**
- **Confidence Intervals**
- **Regression Analysis**
- **ANOVA (Analysis of Variance)**
- **Chi-Square Test**

Getting Started with Python for Statistical Analysis

Before we dive into coding, make sure you have Python installed along with the necessary libraries. You can install the required libraries using the following commands:

In [5]:
```
!pip install scikit-learn
```

```
Requirement already satisfied: scikit-learn in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (1.5.1)
Requirement already satisfied: numpy>=1.19.5 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from scikit-learn) (2.1.0)
Requirement already satisfied: scipy>=1.6.0 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from scikit-learn) (1.14.1)
Requirement already satisfied: joblib>=1.2.0 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from scikit-learn) (3.5.0)
[notice] A new release of pip is available: 24.0 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [46]:
```
pip install statsmodels
```

```
Requirement already satisfied: statsmodels in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (0.14.2)
Requirement already satisfied: numpy>=1.22.3 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from statsmodels) (2.1.0)
Requirement already satisfied: scipy!=1.9.2,>=1.8 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from statsmodels) (1.14.1)
Requirement already satisfied: pandas!=2.1.0,>=1.4 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from statsmodels) (2.2.2)
Requirement already satisfied: patsy>=0.5.6 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from statsmodels) (0.5.6)
Requirement already satisfied: packaging>=21.3 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from statsmodels) (24.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from pandas!=2.1.0,>=1.4->statsmodels) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from pandas!=2.1.0,>=1.4->statsmodels) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from pandas!=2.1.0,>=1.4->statsmodels) (2024.1)
Requirement already satisfied: six in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from patsy>=0.5.6->statsmodels) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
[notice] A new release of pip is available: 24.0 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [8]:
```
!pip install seaborn
```

```
Collecting seaborn
  Downloading seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from seaborn) (2.1.0)
Requirement already satisfied: pandas>=1.2 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from seaborn) (2.2.2)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from seaborn) (3.9.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.3.0)
Requirement already satisfied: cycler>=0.10 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (4.53.1)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.7)
Requirement already satisfied: packaging>=20.0 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (24.1)
Requirement already satisfied: pillow>=8 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.1.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from pandas>=1.2->seaborn) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from pandas>=1.2->seaborn) (2024.1)
Requirement already satisfied: six>=1.5 in c:\users\amrut\appdata\local\programs\python\python312\lib\site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.16.0)
Downloading seaborn-0.13.2-py3-none-any.whl (294 kB)
   ---------------------------------------- 0.0/294.9 kB ? eta -:--:--
   ---------------------------------------- 0.0/294.9 kB ? eta -:--:--
   ---- ----------------------------------- 30.7/294.9 kB 660.6 kB/s eta 0:00:01
   --------- ------------------------------ 71.7/294.9 kB 653.6 kB/s eta 0:00:01
   ------------------- -------------------- 143.4/294.9 kB 1.1 MB/s eta 0:00:01
   -------------------------- ------------- 194.6/294.9 kB 1.1 MB/s eta 0:00:01
   -------------------------------------- - 286.7/294.9 kB 1.3 MB/s eta 0:00:01
   ---------------------------------------- 294.9/294.9 kB 1.2 MB/s eta 0:00:00
Installing collected packages: seaborn
Successfully installed seaborn-0.13.2
[notice] A new release of pip is available: 24.0 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Applying Statistical Methods to Datasets

Descriptive Statistics in Python

Descriptive statistics help us summarize and describe the main features of a dataset.

Inferential Statistics in Python

Inferential statistics involve drawing conclusions about a population based on a sample.

Applying Statistical Methods

- ## Load the dataset into Python using pandas.

In [11]:
```
# Sample Code for Statistical Analysis in Python

## Loading the Diabetes Dataset
import pandas as pd
from sklearn.datasets import load_diabetes

# Load the dataset
diabetes = load_diabetes()
df = pd.DataFrame(data=diabetes.data, columns=diabetes.feature_names)
df['target'] = diabetes.target

# Display the first few rows
print(df.head())
```

```
        age       sex       bmi        bp        s1        s2        s3  \
0  0.038076  0.050680  0.061696  0.021872 -0.044223 -0.034821 -0.043401
1 -0.001882 -0.044642 -0.051474 -0.026328 -0.008449 -0.019163  0.074412
2  0.085299  0.050680  0.044451 -0.005670 -0.045599 -0.034194 -0.032356
3 -0.089063 -0.044642 -0.011595 -0.036656  0.012191  0.024991 -0.036038
4  0.005383 -0.044642 -0.036385  0.021872  0.003935  0.015596  0.008142

         s4        s5        s6  target
0 -0.002592  0.019907 -0.017646   151.0
1 -0.039493 -0.068332 -0.092204    75.0
2 -0.002592  0.002861 -0.025930   141.0
3  0.034309  0.022688 -0.009362   206.0
4 -0.002592 -0.031988 -0.046641   135.0
```

Exploring Regression Analysis on a New Dataset

- Perform a linear regression analysis to determine the relationship between two or more variables.
- Interpret the coefficients, p-values, and R-squared value from the regression model summary.
- Create visualizations to illustrate the relationships between variables and the regression line.

Calculate the mean, median, mode, standard deviation, and variance for all the relevant features.

In [12]:
```
##Performing Descriptive Statistics
# Calculate basic descriptive statistics
print("Mean:\n", df.mean())
print("\nMedian:\n", df.median())
print("\nMode:\n", df.mode().iloc[0])
print("\nStandard Deviation:\n", df.std())
print("\nVariance:\n", df.var())

# Additional descriptive statistics
print("\nRange:\n", df.max() - df.min())
print("\nSkewness:\n", df.skew())
print("\nKurtosis:\n", df.kurt())
```

```
Mean:
 age      -1.444295e-18
sex       2.543215e-18
bmi      -2.255925e-16
bp       -4.854086e-17
s1       -1.428596e-17
s2        3.898811e-17
s3       -6.028360e-18
s4       -1.788100e-17
s5        9.243486e-17
s6        1.351770e-17
target    1.521335e+02
dtype: float64

Median:
 age         0.005383
sex        -0.044642
bmi        -0.007284
bp         -0.005670
s1         -0.004321
s2         -0.003819
s3         -0.006584
s4         -0.002592
s5         -0.001947
s6         -0.001078
target    140.500000
dtype: float64

Mode:
 age        0.016281
sex       -0.044642
bmi       -0.030996
bp        -0.040099
s1        -0.037344
s2        -0.001001
s3        -0.013948
s4        -0.039493
s5        -0.018114
s6         0.003064
target    72.000000
Name: 0, dtype: float64

Standard Deviation:
 age        0.047619
sex        0.047619
bmi        0.047619
bp         0.047619
s1         0.047619
s2         0.047619
s3         0.047619
s4         0.047619
s5         0.047619
s6         0.047619
target    77.093005
dtype: float64

Variance:
 age          0.002268
sex          0.002268
bmi          0.002268
bp           0.002268
s1           0.002268
s2           0.002268
s3           0.002268
s4           0.002268
s5           0.002268
s6           0.002268
target    5943.331348
dtype: float64

Range:
 age         0.217952
sex         0.095322
bmi         0.260831
bp          0.244442
s1          0.280694
s2          0.314401
s3          0.283486
s4          0.261629
s5          0.259694
s6          0.273379
target    321.000000
dtype: float64

Skewness:
 age      -0.231382
sex       0.127385
bmi       0.598148
bp        0.290658
s1        0.378108
s2        0.436592
s3        0.799255
s4        0.735374
s5        0.291754
s6        0.207917
target    0.440563
dtype: float64

Kurtosis:
 age      -0.671224
sex      -1.992811
bmi       0.095094
bp       -0.532797
s1        0.232948
s2        0.601381
s3        0.981507
s4        0.444402
s5       -0.134367
s6        0.236917
target   -0.883057
dtype: float64
```

calculate T-statistic and p_value

In [13]:
```
##  Performing Inferential Statistics
from scipy import stats

# Example data: BMI values
bmi_values = df['bmi']

# Hypothetical population mean for BMI
population_mean = 0.05

# Perform one-sample t-test
t_stat, p_value = stats.ttest_1samp(bmi_values, population_mean)

print(f"T-Statistic: {t_stat}")
print(f"P-Value: {p_value}")
```

```
T-Statistic: -22.074985843710174
P-Value: 2.7634312235044638e-73
```

Calculate a 95% confidence interval for the mean of a selected feature.

In [14]:
```
## 6.4 Confidence Intervals
import numpy as np
from scipy import stats

# Sample mean and standard error for BMI
sample_mean = np.mean(bmi_values)
standard_error = stats.sem(bmi_values)

# Compute 95% confidence interval for BMI
confidence_interval = stats.norm.interval(0.95, loc=sample_mean, scale=standard_error)

print(f"95% Confidence Interval for BMI: {confidence_interval}")
```

```
95% Confidence Interval for BMI: (np.float64(-0.004439332370169141), np.float64(0.0044393323701686915))
```

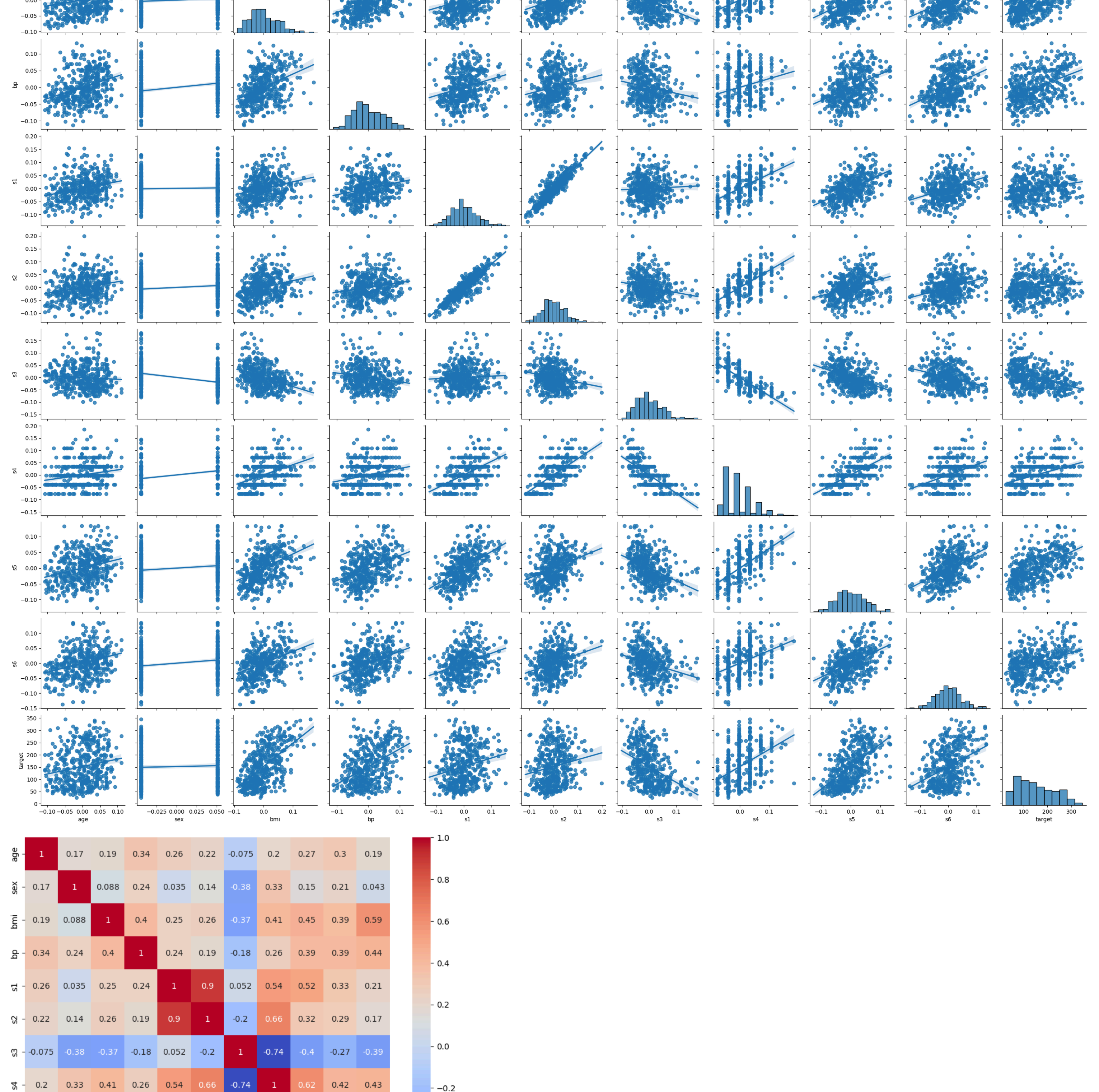
calculate regression

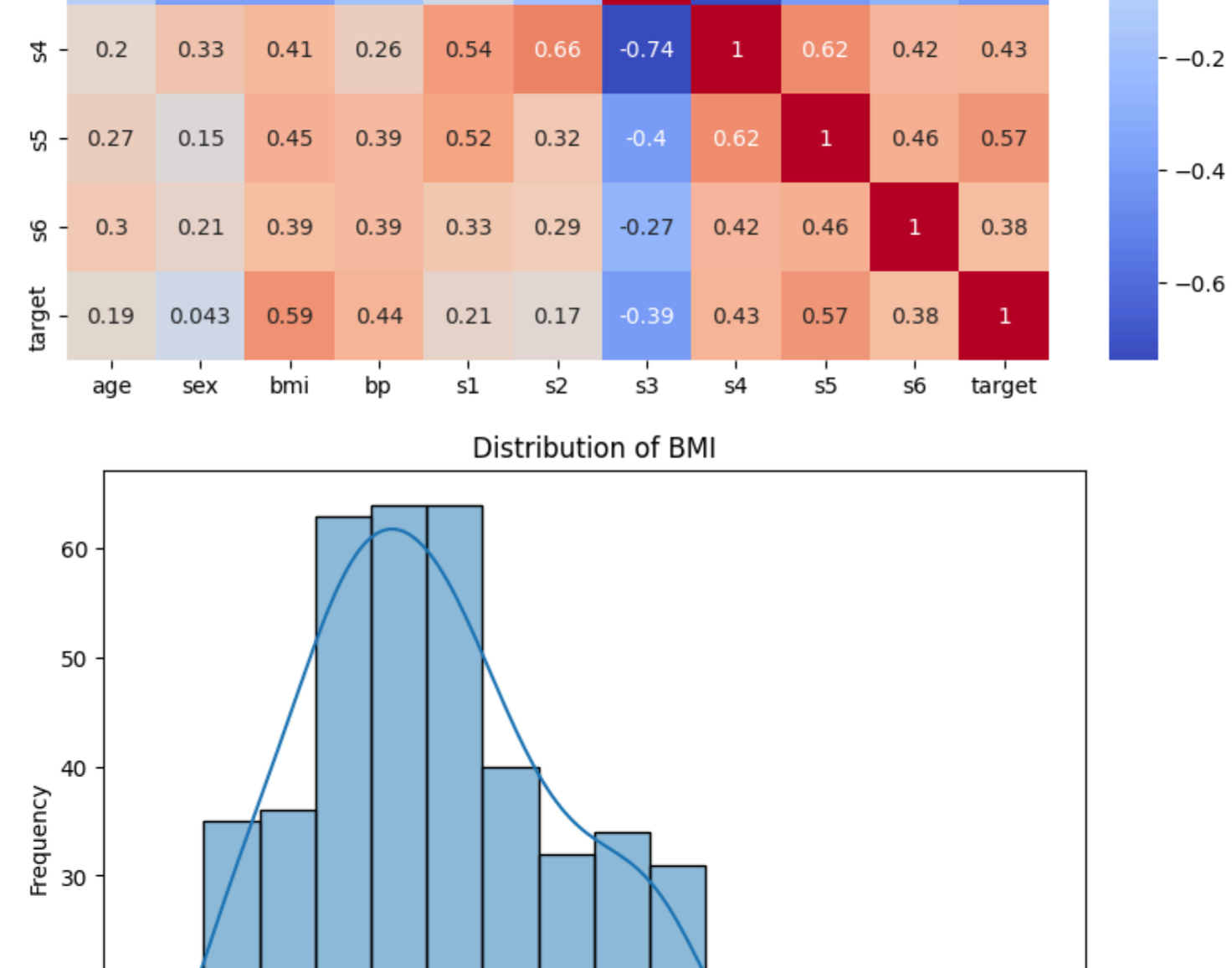
In [15]:
```
## 6.5 Regression Analysis
import statsmodels.api as sm

# Define independent variable (add constant for intercept)
X = sm.add_constant(df['bmi'])

# Define dependent variable
y = df['target']

# Fit linear regression model
model = sm.OLS(y, X).fit()

# Print model summary
print(model.summary())
```

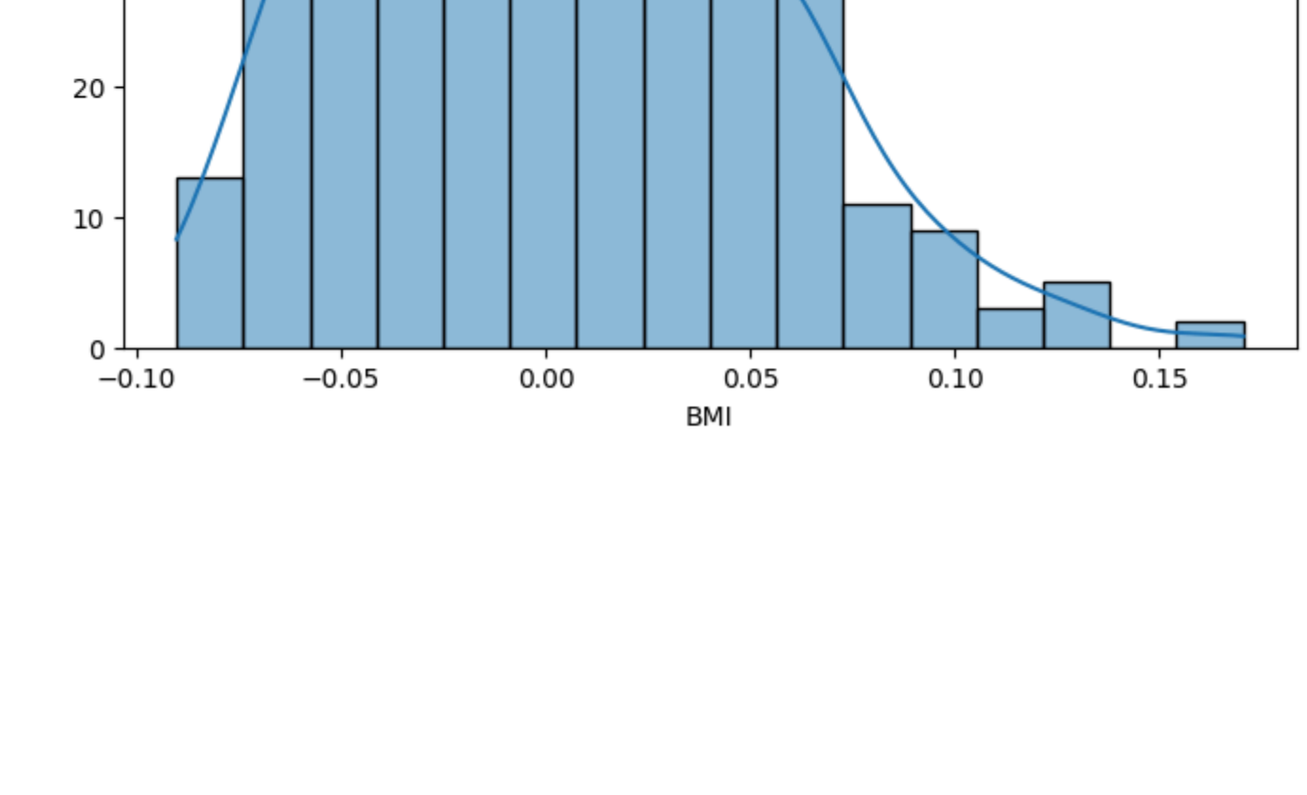
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 target   R-squared:                       0.344
Model:                            OLS   Adj. R-squared:                  0.342
Method:                 Least Squares   F-statistic:                     230.7
Date:                Thu, 05 Sep 2024   Prob (F-statistic):           3.47e-42
Time:                        19:47:37   Log-Likelihood:                -2454.0
No. Observations:                 442   AIC:                             4912.
Df Residuals:                     440   BIC:                             4920.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        152.1335      2.974     51.162      0.000     146.289     157.978
bmi          949.4353     62.515     15.187      0.000     826.570    1072.301
==============================================================================
Omnibus:                       11.674   Durbin-Watson:                   1.848
Prob(Omnibus):                  0.003   Jarque-Bera (JB):                7.310
Skew:                           0.156   Prob(JB):                       0.0259
Kurtosis:                       2.453   Cond. No.                         21.0
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

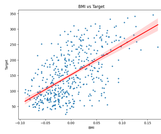
Create visualizations to illustrate the relationships between variables

In [35]:
```
## 6.6 Data Visualization

import seaborn as sns
import matplotlib.pyplot as plt

# Pairplot with regression lines
sns.pairplot(df, kind='reg')
plt.show()

plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.show()

plt.figure(figsize=(8, 6))
sns.histplot(df['bmi'], kde=True)
plt.title('Distribution of BMI')
plt.xlabel('BMI')
plt.ylabel('Frequency')
plt.show()

plt.figure(figsize=(8, 6))
sns.regplot(x='bmi', y='target', data=df, scatter_kws={'s':10}, line_kws={'color':'red'})
plt.title('BMI vs Target')
plt.xlabel('BMI')
plt.ylabel('Target')
plt.show()
```








BMI vs Target
350
300
250
200
150
100
50
Target
−0.10
−0.05
0.00
0.05
0.10
0.15
bmi