Simple Linear Regression:

Simple linear regression is a statistical method used to model the relationship between two continuous variables. It assumes that there exists a linear relationship between the predictor variable (independent variable) and the target variable (dependent variable).

The equation for simple linear regression can be represented as:

$Y = \beta_0 + \beta_1 X + \varepsilon$

Where:

- Y is the dependent variable.
- X is the independent variable.
- β_0 is the intercept.
- β_1 is the slope.
- ε is the error term.

The goal of simple linear regression is to estimate the coefficients $\beta 0$ and $\beta 1$ that best fit the observed data.

Multiple Linear Regression:

Multiple linear regression extends the concept of simple linear regression by considering more than one independent variable. It models the relationship between multiple independent variables and one dependent variable. The equation for multiple linear regression is:

$Y = \beta_0 + \beta_1 X1 + \beta_2 X2 + ... + \beta n Xn + \varepsilon$

Where:

- Y is the dependent variable.
- X is the independent variable.
- β_0 is the intercept.
- β_1 is the slope.
- ε is the error term.

The goal of multiple linear regression is to estimate the coefficients 0, β1, β2, ..., βn that best fit the observed data.

Practical Example with Python Code:

Let's consider a practical example of simple and multiple linear regression using Python. We'll use the popular scikit-learn library for building the regression models.

First, let's import the necessary libraries and generate some synthetic data for our example.

```
[2]  import numpy as np
     import pandas as pd
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LinearRegression
     from sklearn.metrics import mean_squared_error

     # Generate synthetic data
     np.random.seed(0)
     X_simple = 2 * np.random.rand(100, 1)
     y_simple = 3 + 4 * X_simple + np.random.randn(100, 1)

     X_multiple = 2 * np.random.rand(100, 2)
     y_multiple = 3 + 4 * X_multiple[:,0] + 5 * X_multiple[:,1] + np.random.randn(100)

     # Split the data into training and testing sets
     X_simple_train, X_simple_test, y_simple_train, y_simple_test = train_test_split(X_simple, y_simple, test_size=0.2, random_state=0)
     X_multiple_train, X_multiple_test, y_multiple_train, y_multiple_test = train_test_split(X_multiple, y_multiple, test_size=0.2, random_state=0)
```

Now, let's fit the simple linear regression model and evaluate its performance.

```
# Simple Linear Regression
simple_reg = LinearRegression()
simple_reg.fit(X_simple_train, y_simple_train)

# Predictions
y_simple_pred = simple_reg.predict(X_simple_test)

# Model evaluation
mse_simple = mean_squared_error(y_simple_test, y_simple_pred)
print("Simple Linear Regression MSE:", mse_simple)
```

```
Simple Linear Regression MSE: 1.0434333815695171
```

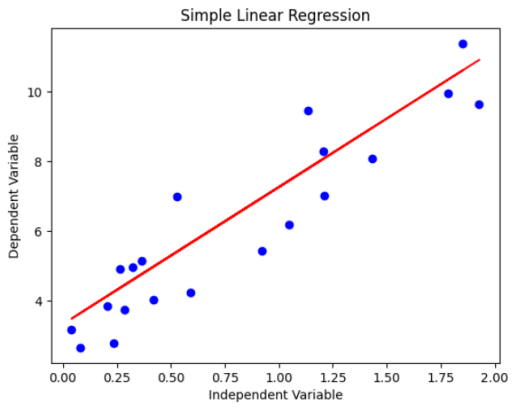
Next, let's fit the multiple linear regression model and evaluate its performance.

```
[4] # Multiple Linear Regression
    multiple_reg = LinearRegression()
    multiple_reg.fit(X_multiple_train, y_multiple_train)

    # Predictions
    y_multiple_pred = multiple_reg.predict(X_multiple_test)

    # Model evaluation
    mse_multiple = mean_squared_error(y_multiple_test, y_multiple_pred)
    print("Multiple Linear Regression MSE:", mse_multiple)
```

```
Multiple Linear Regression MSE: 1.351540019899241
```

Let's enhance our example by including visualization using matplotlib.

```
[5] import matplotlib.pyplot as plt

    # Simple Linear Regression Visualization
    plt.scatter(X_simple_test, y_simple_test, color='blue')
    plt.plot(X_simple_test, y_simple_pred, color='red')
    plt.title('Simple Linear Regression')
    plt.xlabel('Independent Variable')
    plt.ylabel('Dependent Variable')
    plt.show()
```



The following code will plot the scatter plot of the testing data along with the regression line predicted by the simple linear regression model.

[6]
```python
# Multiple Linear Regression Visualization
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(X_multiple_test[:,0], X_multiple_test[:,1], y_multiple_test, c='blue', marker='o')
ax.plot_trisurf(X_multiple_test[:,0], X_multiple_test[:,1], y_multiple_pred, color='red', alpha=0.5)
ax.set_title('Multiple Linear Regression')
ax.set_xlabel('Independent Variable 1')
ax.set_ylabel('Independent Variable 2')
ax.set_zlabel('Dependent Variable')
plt.show()
```