

Introduction to Linear Regression

- **Linear regression** is a method used to model the relationship between a dependent variable and one or more independent variables.
- It assumes that there is a linear relationship between the variables and tries to find the line of best fit that minimizes the sum of squared errors.

Simple Linear Regression

- Simple linear regression involves modeling the relationship between a dependent variable y and a single independent variable x .
- The line of best fit can be represented by the equation $y = b_0 + b_1 * x$, where b_0 is the y-intercept and b_1 is the slope of the line.

Multiple Linear Regression

- Multiple linear regression involves modeling the relationship between a dependent variable y and multiple independent variables x_1, x_2, \dots, x_n .
- The line of best fit can be represented by the equation $y = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n$.

Implementing Linear Regression in Python

- We can use the `LinearRegression` class from the `scikit-learn` library to perform linear regression in Python.
- Here is an example of how to fit a simple linear regression model:

```
from sklearn.linear_model import LinearRegression

# define the data
X = [[0], [1], [2]]
y = [0, 1, 2]

# create and fit the model
model = LinearRegression()
model.fit(X, y)

# make predictions
predictions = model.predict([[3], [4]])
```

Visualizing Results

- We can use visualization libraries such as matplotlib or seaborn to create scatter plots and visualize the line of best fit.
- Here is an example of how to create a scatter plot with a fitted line:

```
import matplotlib.pyplot as plt
```

```
# define the data
```

```
X = [0, 1, 2]
```

```
y = [0, 1, 2]
```

```
# plot the data
```

```
plt.scatter(X, y)
```

```
# plot the fitted line
```

```
plt.plot(X, y)
```

```
# show the plot
```

```
plt.show()
```

Is there anything **else** you would like to know?