LINEAR REGRESSION WITH NumPy and Python

CO's:

1. Implementing the gradient descent algorithm from scratch.
2. Perform univariate linear regression with NumPy and Python.
3. Create data visualizations and plots using matplotlib.

The objective of Linear Regression is to minimize the cost function J(Ɵ). The cost is the error our model made in estimating a value.

Linear Regression:

- It is used to estimate the value of a variable based of another variable.
- The variable we approximate is called the dependent variable.
- The variable we are using in this process of estimation of the dependent variable is called the independent variable.
- This form of analysis estimates the coefficients of the linear equation.
- There can be one or many independent variables that best predict the value of the dependent variable.
- Linear regression fits a straight line or surface that minimizes the discrepancies between the predicted and actual output values.
- Most of the Linear Regression calculators use the "Least Squares" method to find the best fit line.
- Then the value of x (dependant variable) from y (independent variable).

To apply the Linear Regression, make sure that data can be analysed using the procedure below and must pass through the required assumptions such as:

- The variables must be continuous. Example: sales, weight, test scores etc.
- Use scatterplot to see if there is a linear relationship between the variables.
- The observations should be independent of each other such that the variance of the dependent variable doesn't change with the value of the independent variable.
- The data must not have any significant outliers. Significant outliers are the data points that differs substantially from the other observations.
- Check for "HOMOSCEDASTICITY", a statistical concept in which the variance along the best-fit line remains similar all through the line.

For gradient descent info and explanation: Gradient descent in Linear Regression