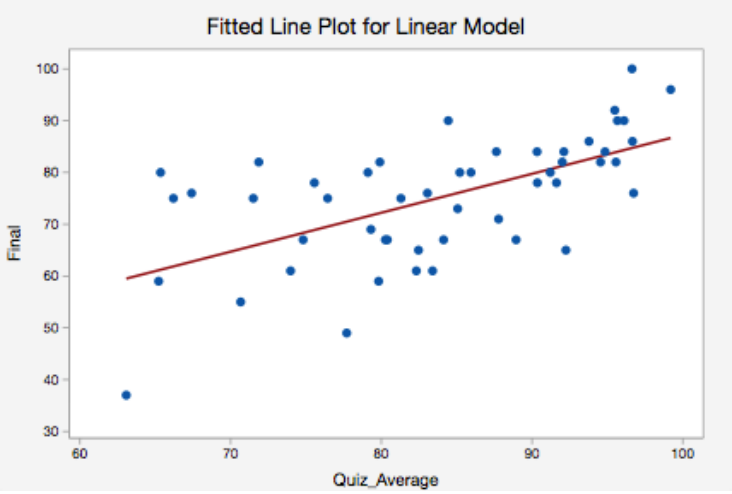
LINEAR REGRESSION FROM SCRATCH WITH PYTHON

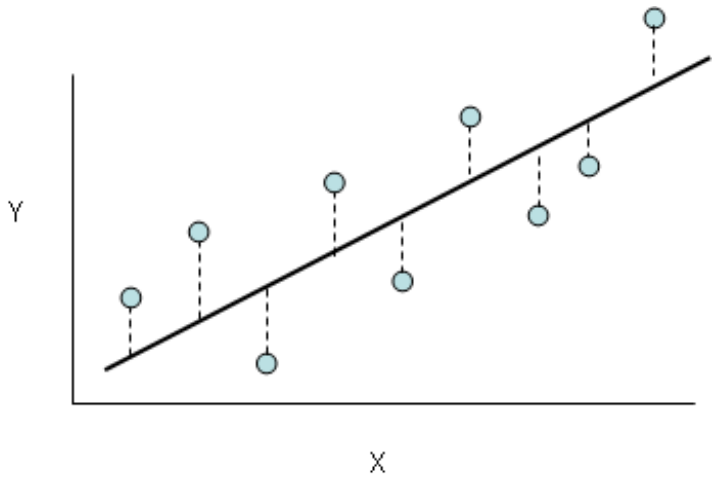
Ventje Jeremias Lewi Engel, M.T.

Ordinary Least Mean Square

In the class we discussed estimating the relationship between **X** and **Y** to a line. For example, we get sample inputs and outputs and we plot these scatter point on a 2d graph, we something similar to the graph below :



The line seen in the graph is the actual relationship we are going to accomplish, And we want to minimize the error of our model. This line is the best fit that passes through most of the scatter points and also reduces error which is the distance from the point to the line itself as illustrated below.



And the total error of the linear model is the sum of the error of each point. I.e. ,

$$\sum_{i=1}^{n} r_i^2$$

r*i* = Distance between the line and **i***th* point.

n =Total number of points.

We are squaring each of the distance's because some points would be above the line and some below. We can minimize the error of our linear model by minimizing **r** thus we have

$$\beta_i = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \bar{\beta}_1 \bar{x}$$

where $\bar{x}$ is the mean of the input variable **X** and $\bar{y}$ being the mean of the output variable **Y.**

Now let's implement this method in python (the fun part).

To follow on, you need python and following libraries:

- numpy
- pandas
- matplotlib

We are going to be using a dataset containing head size and brain weight of different people. This dataset is available in this link.

We start by importing the dataset and our dependencies

```
#import libraries
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
import pandas as pddataset = pd.read_csv('dataset.csv')
print(dataset.shape)
dataset.head()(237, 4)
```

	Gender	Age Range	Head Size(cm^3)	Brain Weight(grams)
0	1	1	4512	1530
1	1	1	3738	1297
2	1	1	4261	1335
3	1	1	3777	1282
4	1	1	4177	1590

Let's find the relationship between the Head Size and Brain weights.

```
X = dataset['Head Size(cm^3)'].values
Y = dataset['Brain Weight(grams)'].values

x_mean = np.mean(X)
y_mean = np.mean(Y)#total number of values
n = len(X)

# using the formula to calculate the b1 and b0
numerator = 0
denominator = 0
for i in range(n):
   numerator += (X[i] - x_mean) * (Y[i] - y_mean)
   denominator += (X[i] - x_mean) ** 2

b1 = numerator / denominator
b0 = y_mean - (b1 * x_mean)

#printing the coefficient
print(b1, b0)
```

output : 0.26342933948939945 325.57342104944223

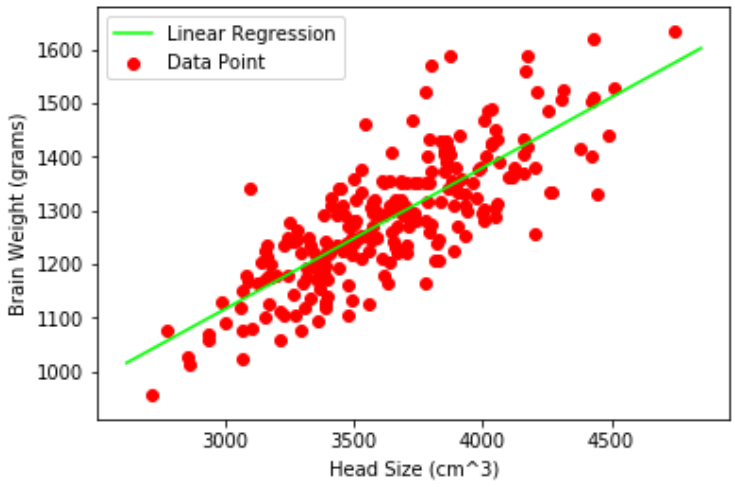
Now we have our coefficient and bias/ intercept. In mathematical terms :

*Brain weights = 325.57342104944223 + 0.26342933948939945 * Head size*

Now we have a linear model.

Lets plot it graphically.

```
#plotting values
x_max = np.max(X) + 100
x_min = np.min(X) - 100
#calculating line values of x and y
x = np.linspace(x_min, x_max, 1000)
y = b0 + b1 * x
#plotting line
plt.plot(x, y, color='#00ff00', label='Linear Regression')
#plot the data point
plt.scatter(X, Y, color='#ff0000', label='Data Point')
plt.xlabel('Head Size (cm^3)')
plt.ylabel('Brain Weight (grams)')plt.legend()
plt.show()
```



We need to able to measure how good our model is (accuracy). There are many methods to achieve this but we would implement **Root mean squared error** and **coefficient of Determination** (**R² Score**).

Root Mean Squared Error is the square root of the sum of all errors divided by the number of values, or Mathematically,

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

Here ***yj^*** is the ***ith*** predicted output values. Now we will find RMSE.

```
rmse = 0
for i in range(n):
  y_pred=  b0 + b1* X[i]
  rmse += (Y[i] - y_pred) ** 2

rmse = np.sqrt(rmse/n)

print(rmse)

#output : 72.1206213783709
```

Let's find our R² score to be able to measure the accuracy of our linear model, mathematically :

$$R^2 = \frac{SSR}{SST} = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2}$$

SST or TSS is the total sum of squares and **SSR or RSS** is the total sum of squares of residuals.

R^2 Score usually ranges from 0 to 1. It will also become negative if the model is completely wrong.

Now we will find the **R^2** Score.

```
sumofsquares = 0
sumofresiduals = 0

for i in range(n) :
  y_pred = b0 + b1 * X[i]
  sumofsquares += (Y[i] - y_mean) ** 2
  sumofresiduals += (Y[i] - y_pred) **2

score  = 1 - (sumofresiduals/sumofsquares)

print(score)

#output : 0.6393117199570003
```

0.63 is certainly not bad, but we can improve the score by :

- Getting more datasets
- Improving the features
- Add more variables, etc

Conclusion

Linear Regression is the base of all machine learning algorithms and the easiest to pick up, we have implemented the **Ordinary Least Mean Square** method to predict Brain weights from Head Size and also measured the accuracy with **Root mean squared error** and **coefficient of Determination** (**R^2 Score**).

Stay Healthy and Cheers,

Ventje Jeremias Lewi Engel, M.T.

Do something today that your future self will thank you for