

Summer Internship Project Report

On

“DATA SCIENCE”

Submitted in the partial fulfillment of the requirement for the award of

Master of Computer Application.

Noida International University, Greater Noida
ACADEMIC SESSION
(2021 – 2023)



Under the guidance of:

Saumya Tiwari

Senior HR Manager

Company: TEACHNOOK

Location: INDIA

Date: 26/08/2023

Submitted By:

Koppula Suresh Babu

MCA 2nd Semester

Roll no:

122114142007

TO WHOMSOEVER IT MAY CONCERN

This is to certify that KOPPULA SURESH BABU a student of MCA from Noida International University(Batch 2022-2024).Roll no: 122114142007 has undergone his/her project research on DATA SCIENCE (Linear regression) using python libraries the topic under my guidance and supervision from 01/06/2023 to 31/07/2023.

Signature of Internship Guide

Internship Guide Name

Designation

Company

This is to certify that KOPPULA SURESH BABU a student of MCA from Noida International University (Batch 2022-2024). Roll no: 122114142007 has successfully completed his summer internship training in “DATA SCIENCE” using Python libraries (Linear regression) at TEACHNOOK from 01/06/2023 to 31/07/2023 in our Organization.

During the above training period, we found him punctual, honest, hardworking and sincere towards his work.

CERTIFICATE

This is to certify that MR. “Koppula Suresh Babu” Roll no: 122114142007
Is a student at Noida International University has done his/her Research paper
Titled “DATA SCIENCE” at TEACHNOOK from 01/06/2023 to 31/07/2023.

The work tangible in the report is original and is of the standard expected of a
MCA student and has not been submitted in part or full to this or any other
Institution,

For the award of MASTER'S Degree.

He has completed all requirements for Internship and the report work is fit for
evaluation.

Signature of Head of Institution:

Saumya Tiwari

Senior HR Manager

TEACHNOOK.

DECLARATION

I, Koppula Suresh Babu hereby declare that the Research paper work title

“DATA SCIENCE” using Python libraries (Linear regression) at TEACHNOOK from 01/06/2023 to 31/07/2023 completed and submitted in partial fulfilment of the award of the degree of “Masters of computer science” by Noida International University and work was carried out with the help of under Guidance of “SAUMYA TIWARI” during the time period in internship from 01/06/2023 to 31/07/2022.

I further declared this internship has not formed the basis for the award of any other Degree/Diploma of any other University/Institution.

Date: 31/07/2023

Koppula Suresh Babu

122114142007

MCA (2022-2024)

ACKNOWLEDGEMENT

The internship opportunity I had with “**TEACHNOOK**” was a great chance for learning and professional development. Therefore, I consider myself very lucky individual as I was provided with an opportunity to be a part of it. I am also grateful for having a chance to meet so many wonderful people and professionals who led me through this Internship period.

I am using this opportunity to express my deepest gratitude and special thanks to “**SAMUYA TIWARI**” HR Manager in TEACHNOOK.

592,3 rd. Block,

Koramangala,

Bengaluru,

Karnataka 560068.

It is my radiant sentiment to place on record my best regards, deepest sense of gratitude to prof. DEEPTI GAUTAM. NOIDA INTERNATIONAL UNIVERSITY (Powdered by SUNSTONE) for his careful and precious guidelines though out of campus and on campus which were extremely valuable for my study both theoretically and practically.

I sincerely want to thank prof. DEEPTI GAUTAM for being our programme coordinator helping me at every stage of internship programme and studies.

I perceive this opportunity as a big role for my career development. I will strive to use gained skills and knowledge in the best possible way, and still, I continue to work on the improvement, in order to attain desire career objectives. Hope to continue cooperation with all of you in the future.

Sincerely,

KOPPULA SURESH BABU.

Introduction

In this article, we will analyse a business problem with linear regression in a step-by-step manner and try to interpret the statistical terms at each step to understand its inner workings. Although the linear regression algorithm is simple, for proper analysis, one should interpret the statistical results.

First, we will take a look at simple linear regression and after extending the problem to multiple linear regression.

What is Linear Regression?

Regression is the statistical approach to find the relationship between variables. Hence, the **Linear Regression** assumes a linear relationship between variables. Depending on the number of input variables, the regression problem is classified into

- 1) Simple linear regression
- 2) Multiple linear regression

Business problem

In this article, we are using the **Advertisement data set**. Let's consider there is a company and it has to improve the sales of the product. The company spends money on different advertising media such as TV, radio, and newspaper to increase the sales of its products. The company records the money spent on each advertising media (in

thousands of dollars) and the number of units of product sold (in thousands of units).

Now we have to help the company to find out the most effective way to spend money on advertising media to improve sales for the next year with a less advertising budget.

Simple Linear Regression

Multiple Linear Regression:

In multiple linear regression, we will analyse the relationship between sales and three advertising media collectively.

$$\text{Sales} = \beta_0 + \beta_1 * TV + \beta_2 * \text{Radio} + \beta_3 * \text{Newspaper} + \text{epsilon}$$

Now let's follow the steps similar to the simple linear regression,

1] Estimating the Coefficients:

	coef	std err	t	P> t	[0.025	0.975]
Intercept	2.9389	0.312	9.422	0.000	2.324	3.554
TV	0.0458	0.001	32.809	0.000	0.043	0.049
radio	0.1885	0.009	21.893	0.000	0.172	0.206
newspaper	-0.0010	0.006	-0.177	0.860	-0.013	0.011

Table 4: Multiple linear regression of sales on TV, radio, newspaper

The above table shows the multiple regression coefficient estimates when TV, radio, and newspaper advertising budgets are used to predict product sales using the Advertising data.

$$\text{Sales} = 2.94 + 0.045 * TV + 0.189 * \text{Radio} + (- 0.001) * \text{Newspaper}$$

We can analyse that the coefficient estimate for the newspaper is close to zero and the p-value is no longer significant ($p\text{-value} \gg 0.005$) with a value around 0.86. This shows that money spent on newspaper advertising media has no relation to the sale of the product.

Linear Regression (Python Implementation)

Simple Linear Regression

Simple linear regression is an approach for predicting a response using a single feature. It is one of the most basic machine learning models that a machine learning enthusiast gets to know about. In linear regression, we assume that the two variables dependent and independent variables are linearly related. Hence, we try to find a linear function that predicts the response value(y) as accurately as possible as a function of the feature or independent variable(x). Let us consider a dataset where we have a value of response y for every feature x:

x	0	1	2	3	4	5	6	7	8	9
y	1	3	2	5	7	8	8	9	10	12

For generality, we define:

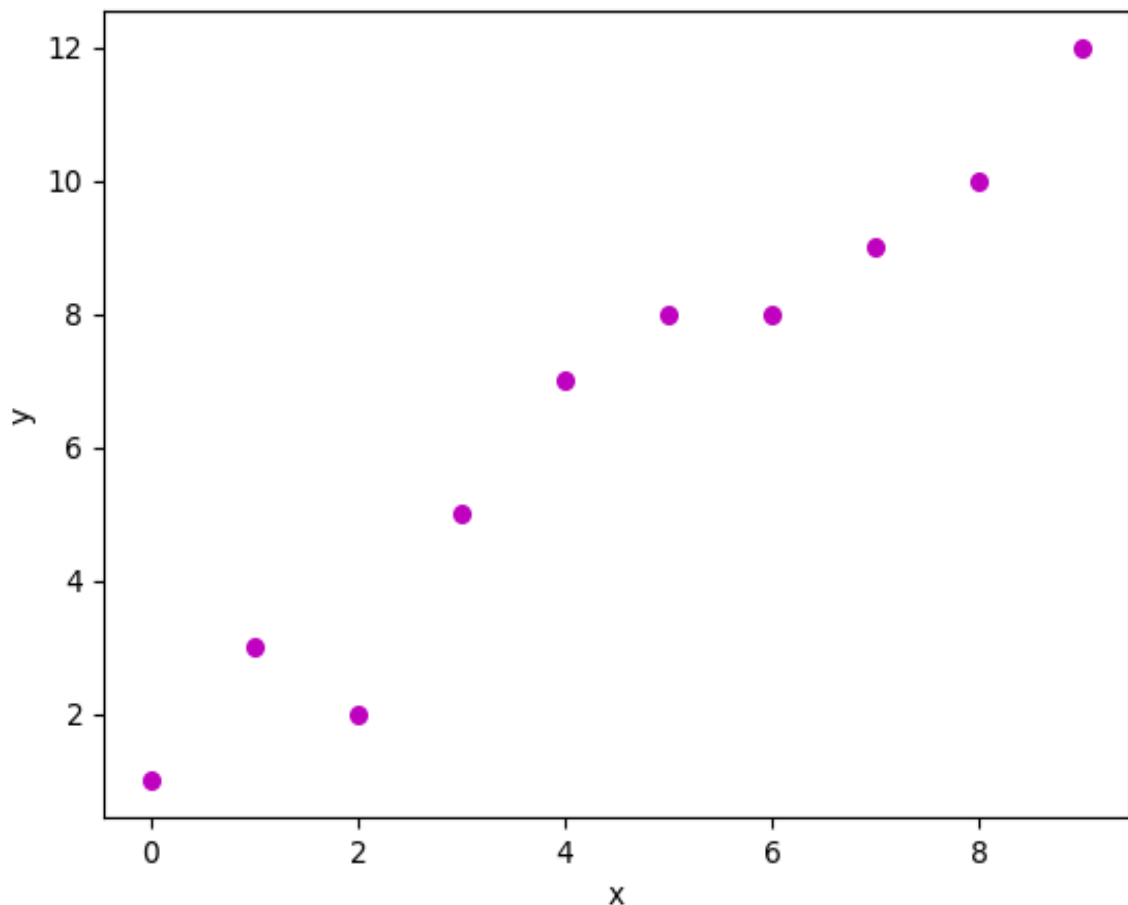
x as feature vector,

$x = [x_1, x_2, \dots, x_n]$,

y as response vector,

$y = [y_1, y_2, \dots, y_n]$

for n observations (in the above example, $n=10$). A scatter plot of the above dataset looks like this: -



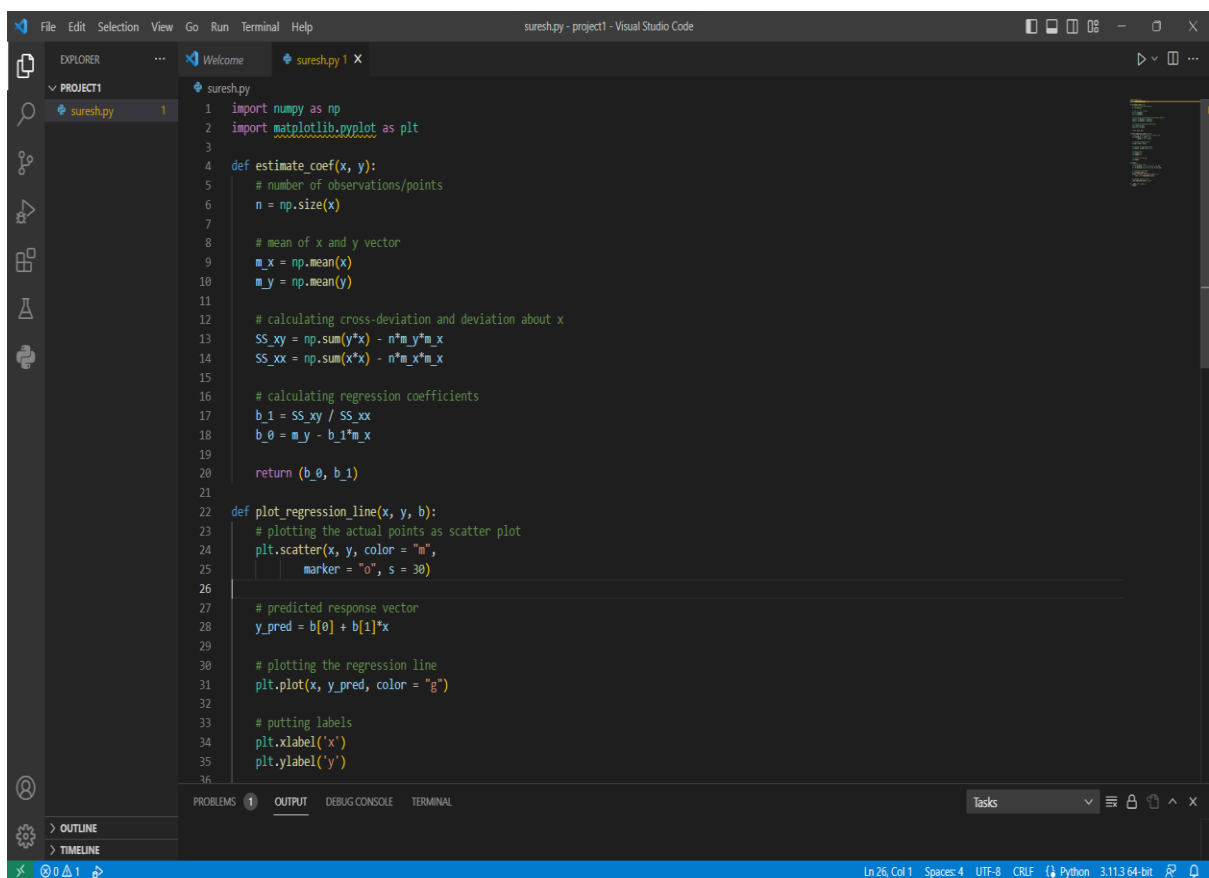
Scatter plot for the randomly generated data.

Now, the task is to find a line that fits best in the above scatter plot so that we can predict the response for any new feature values. (a value of x not present in a dataset) This line is called as regression line. The equation of the regression line is represented as:

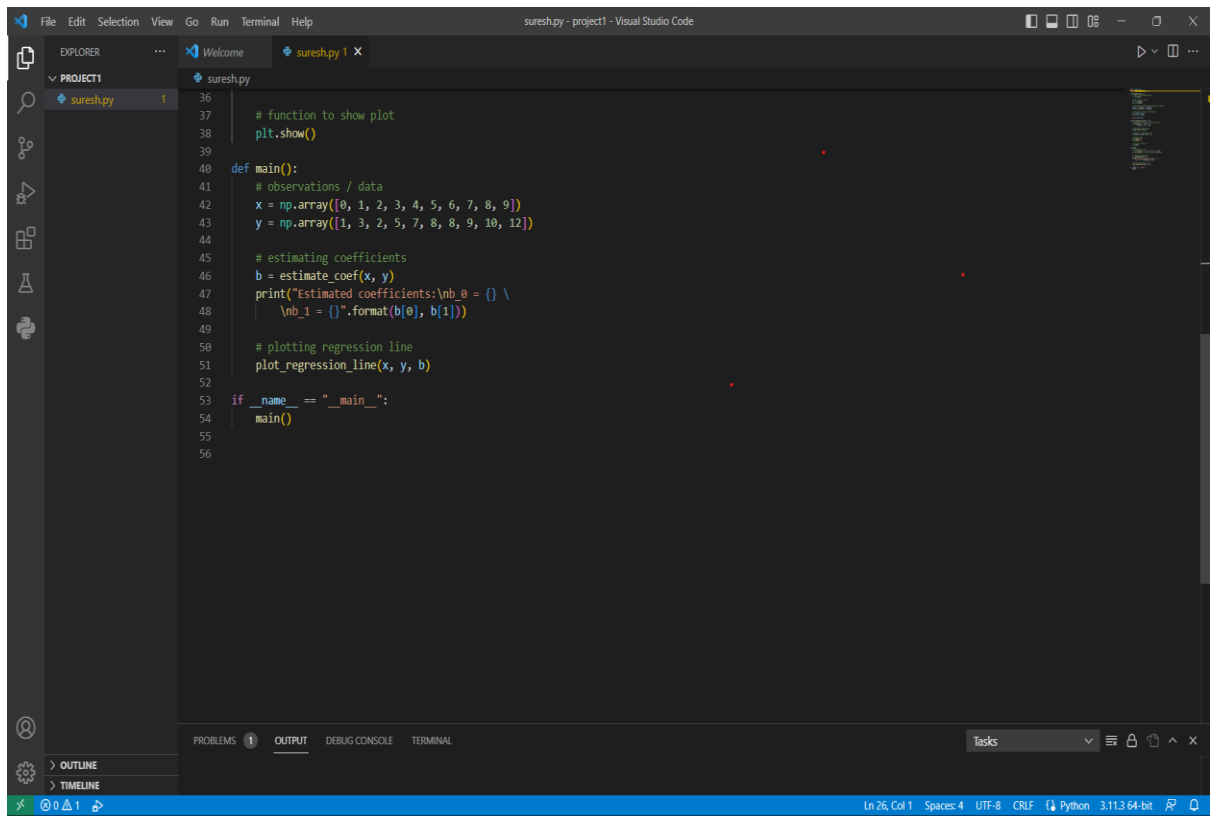
Python implementation of linear Regression

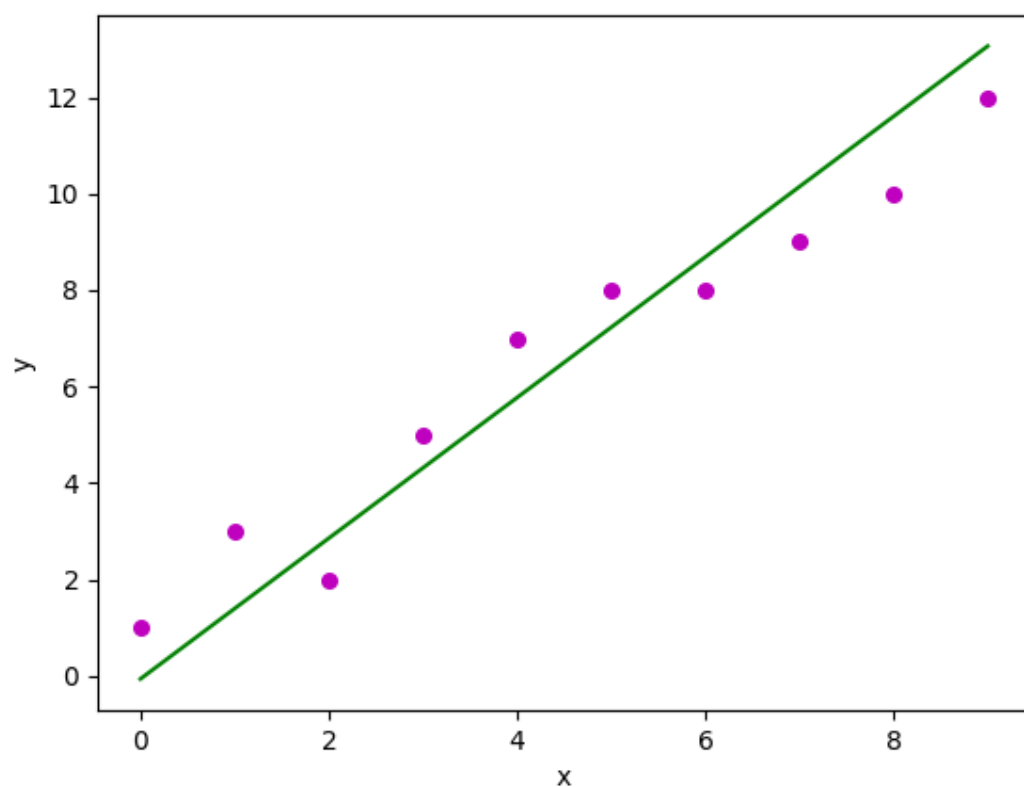
We can use the PYTHON language to learn coefficient of linear regression models. For plotting the input data and best-fitted line we will use the matplotlib library.

It is one of the most used Python libraries for plotting graphs.



```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def estimate_coef(x, y):
5     # number of observations/points
6     n = np.size(x)
7
8     # mean of x and y vector
9     m_x = np.mean(x)
10    m_y = np.mean(y)
11
12    # calculating cross-deviation and deviation about x
13    SS_xy = np.sum(y*x) - n*m_y*m_x
14    SS_xx = np.sum(x*x) - n*m_x*m_x
15
16    # calculating regression coefficients
17    b_1 = SS_xy / SS_xx
18    b_0 = m_y - b_1*m_x
19
20    return (b_0, b_1)
21
22 def plot_regression_line(x, y, b):
23     # plotting the actual points as scatter plot
24     plt.scatter(x, y, color = "m",
25                marker = "o", s = 30)
26
27     # predicted response vector
28     y_pred = b[0] + b[1]*x
29
30     # plotting the regression line
31     plt.plot(x, y_pred, color = "g")
32
33     # putting labels
34     plt.xlabel('x')
35     plt.ylabel('y')
```





Scatterplot of the points along with the regression line

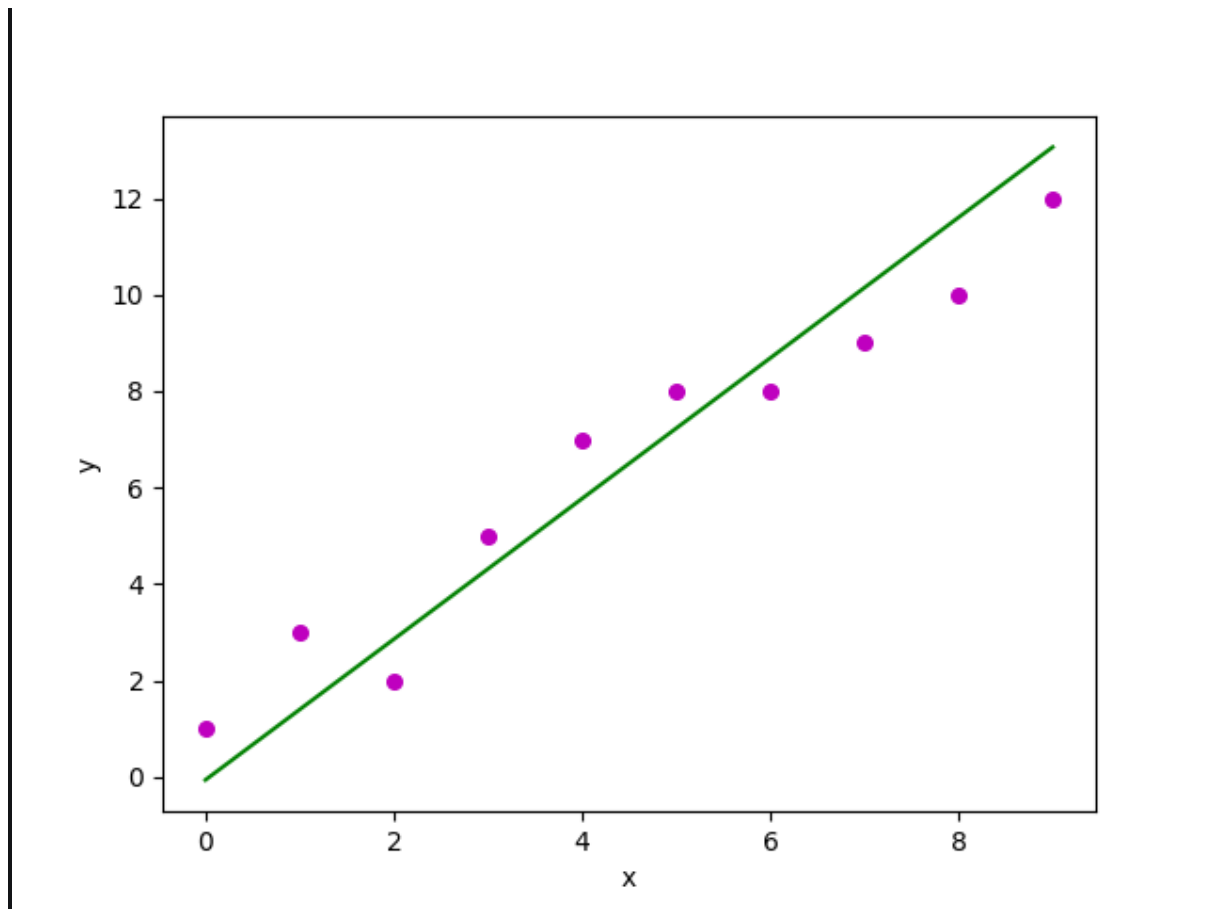
Output:

Estimated coefficients:

$b_0 = -0.0586206896552$

$b_1 = 1.45747126437$

And the graph obtained looks like this:



Scatterplot of the points along with the regression line:

Multiple linear regression

Multiple linear regression attempts to model the relationship between two or more features and a response by fitting a linear equation to the observed data.

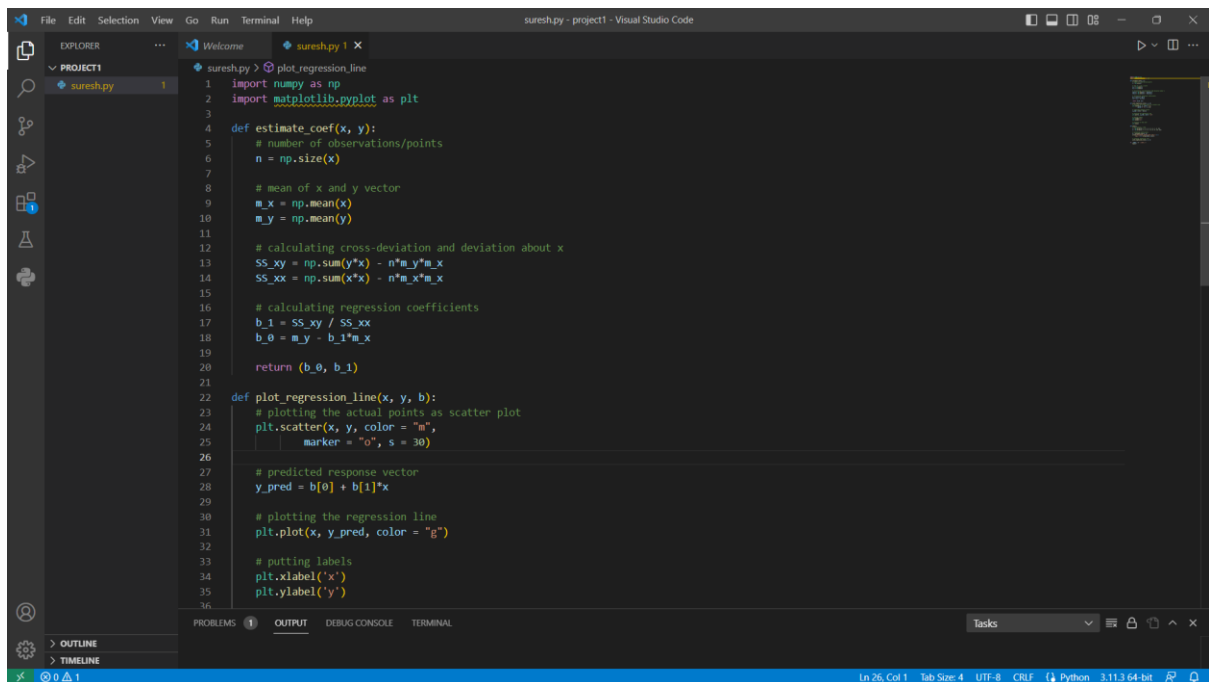
Clearly, it is nothing but an extension of simple linear regression. Consider a dataset with p features (or independent variables) and one response (or dependent

variable).

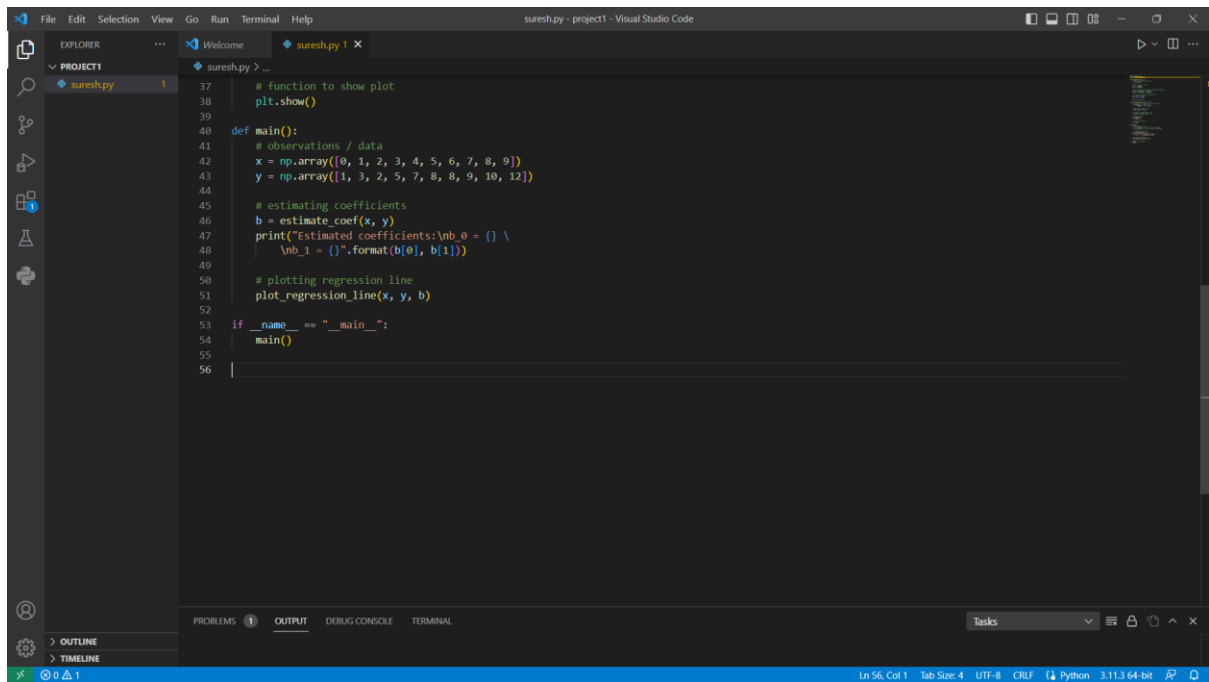
Also, the dataset contains n rows/observations.

We define:

X (feature matrix) = a matrix of size $n \times p$ where x denotes the values of the j feature for i observation.



```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def estimate_coef(x, y):
5     # number of observations/points
6     n = np.size(x)
7
8     # mean of x and y vector
9     m_x = np.mean(x)
10    m_y = np.mean(y)
11
12    # calculating cross-deviation and deviation about x
13    SS_xy = np.sum(y*x) - n*m_y*m_x
14    SS_xx = np.sum(x*x) - n*m_x*m_x
15
16    # calculating regression coefficients
17    b_1 = SS_xy / SS_xx
18    b_0 = m_y - b_1*m_x
19
20    return (b_0, b_1)
21
22 def plot_regression_line(x, y, b):
23     # plotting the actual points as scatter plot
24     plt.scatter(x, y, color = "m",
25                marker = "o", s = 30)
26
27     # predicted response vector
28     y_pred = b[0] + b[1]*x
29
30     # plotting the regression line
31     plt.plot(x, y_pred, color = "g")
32
33     # putting labels
34     plt.xlabel('x')
35     plt.ylabel('y')
```



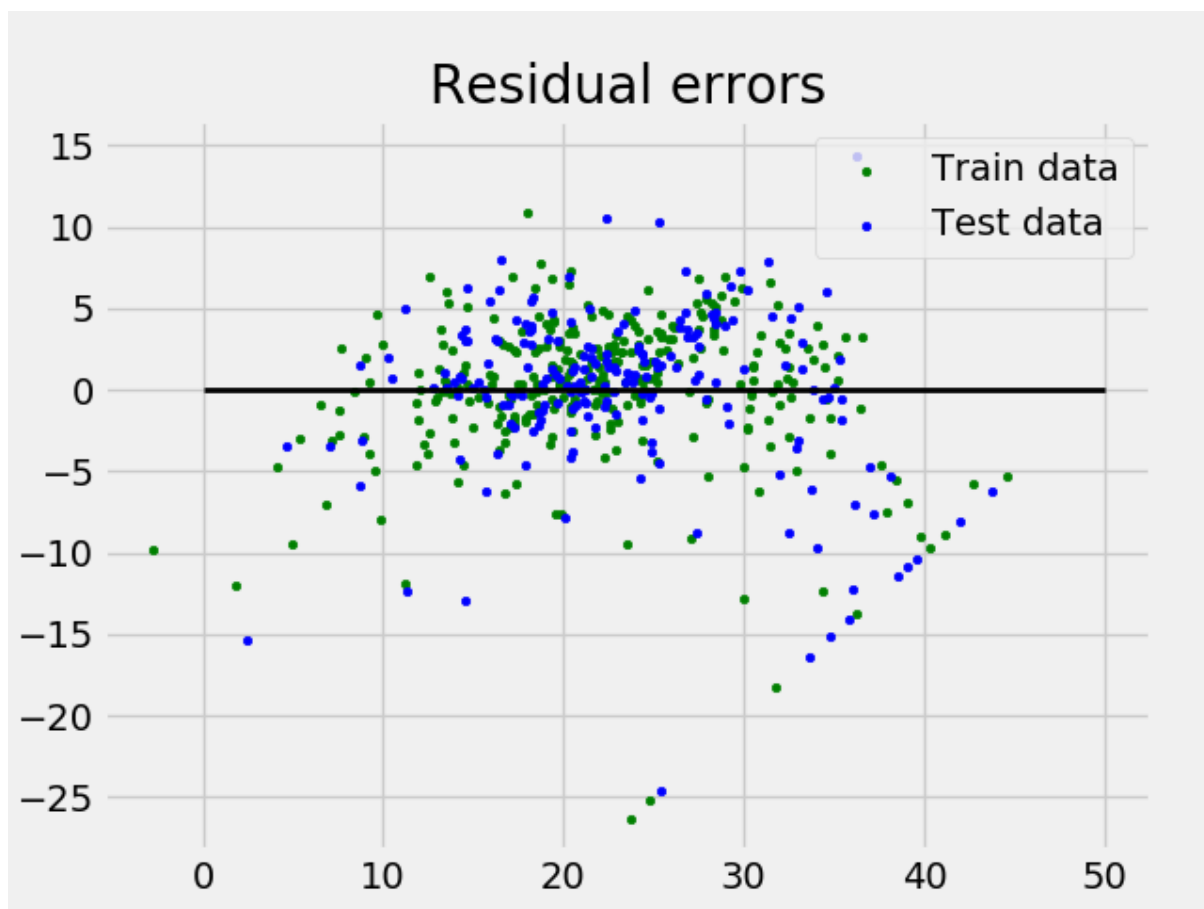
```
37 # function to show plot
38 plt.show()
39
40 def main():
41     # observations / data
42     x = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
43     y = np.array([1, 3, 2, 5, 7, 8, 8, 9, 10, 12])
44
45     # estimating coefficients
46     b = estimate_coef(x, y)
47     print("Estimated coefficients:\nb_0 = {} \
48         \nb_1 = {}".format(b[0], b[1]))
49
50     # plotting regression line
51     plot_regression_line(x, y, b)
52
53 if __name__ == "__main__":
54     main()
55
56
```

Output:

Coefficients:

```
[-8.80740828e_02  6.72507352e_02
 5.10280463e-02   2.18879172e+00
-1.72283734e+01  3.62985243e+00
 2.139933641e-03 -1.36531300e+00
 2.88788067e-01  -1.22618657e-02
-8.36014969e+00  9.53058061e-03
-5.05036163e-01]
```

Variance source :0.720898784611



Residual Error Plot for the Multiple Linear Regression

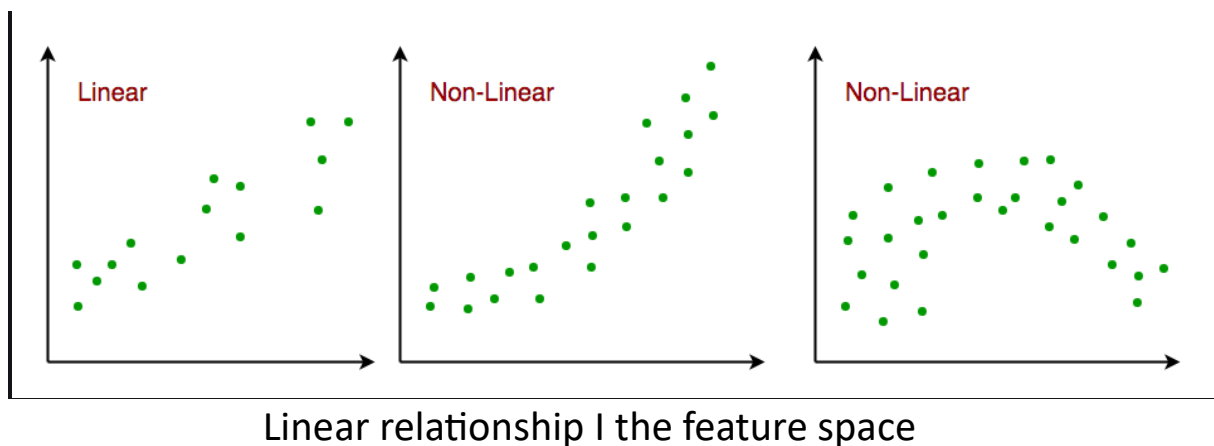
Assumptions We Make in a Linear Regression Model:

Given below are the basic assumptions that a linear regression model makes regarding a dataset on which it is applied:

Linear relationship:

The relationship between response and feature variables should be linear. The linearity assumption can be tested using scatter plots. As shown below, 1st figure represents linearly related variables whereas

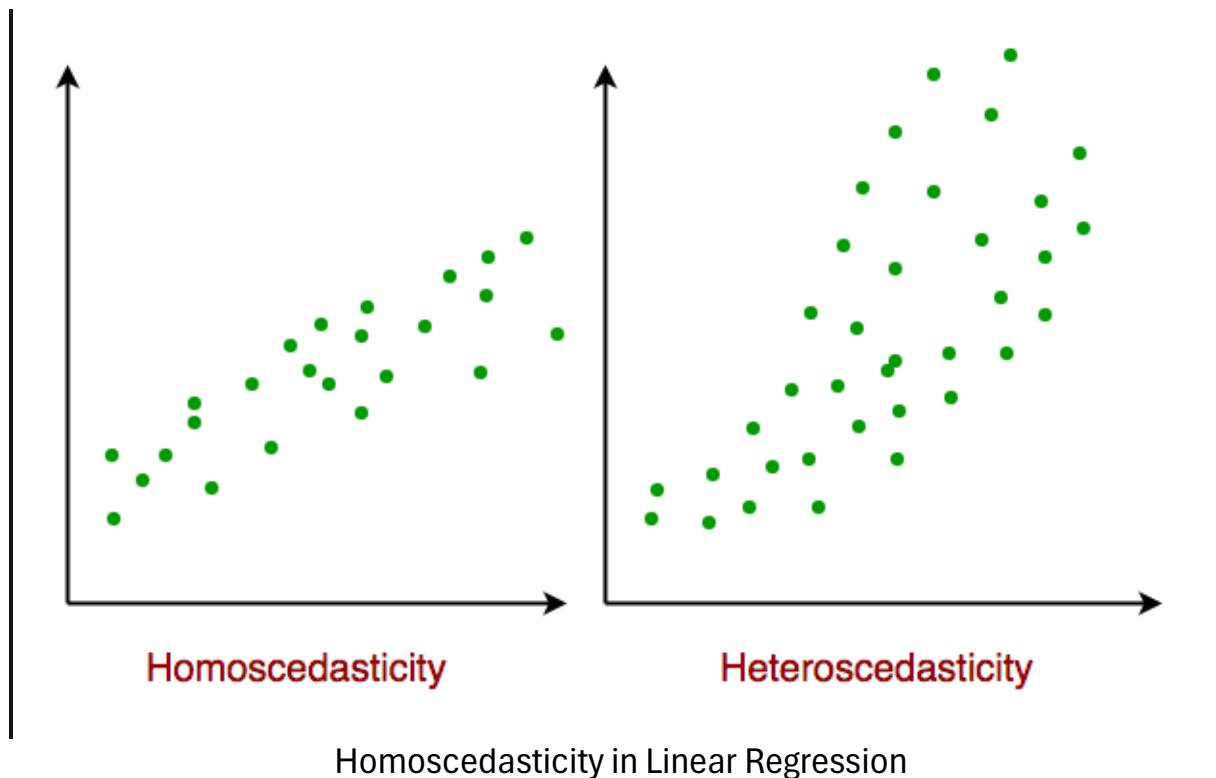
variables in the 2nd and 3rd figures are most likely non-linear. So, 1st figure will give better predictions using linear regression.



Little or no multi-collinearity: It is assumed that there is little or no multicollinearity in the data. Multicollinearity occurs when the features (or independent variables) are not independent of each other. Little or no autocorrelation: Another assumption is that there is little or no autocorrelation in the data. Autocorrelation occurs when the residual errors are not independent of each other. You can refer here for more insight into this topic.

Homoscedasticity:

Homoscedasticity describes a situation in which the error term (that is, the “noise” or random disturbance in the relationship between the independent variables and the dependent variable) is the same across all values of the independent variables. As shown below, figure 1 has homoscedasticity while Figure 2 has heteroscedasticity.



Homoscedasticity in Linear Regression

Applications of Linear Regression:

Trend lines: A trend line represents the variation in quantitative data with the passage of time (like GDP, oil prices, etc.). These trends usually follow a linear relationship. Hence, linear regression can be applied to predict future values.

However, this method suffers from a lack of scientific validity in cases where other potential changes can affect the data.

Economics:

Linear regression is the predominant empirical tool in economics. For example, it is used to predict consumer spending, fixed investment spending, inventory investment, purchases of a country's exports, spending on imports, the demand to hold liquid assets, labour demand, and labour supply.

Finance:

The capital price asset model uses linear regression to analyse and quantify the systematic risks of an investment.

Biology: Linear regression is used to model causal relationships between parameters in biological systems.

Resources

Lastly, I would like to mention a few great resources which you can use to learn more about linear regression.

[Linear Regression \(Stats models Documentation\)](#)

Conclusion

In this article, we went over what Linear Regression is, how it works and how can we analyse the results at each step of model building with python implementation.

