COMPUTING SUBJECT:	Machine Learning
TYPE:	WORK ASSIGNMENT
IDENTIFICATION:	Linear Regression
COPYRIGHT:	*Michael Claudius*
DEGREE OF DIFFICULTY:	Easy
TIME CONSUMPTION:	1-2 hours
EXTENT:	< 60 lines
OBJECTIVE:	Basic understanding of linear regression Simple calculations for one independent variable/feaure

COMMANDS:

IDENTIFICATION: Linear Regression/MICL

The Mission

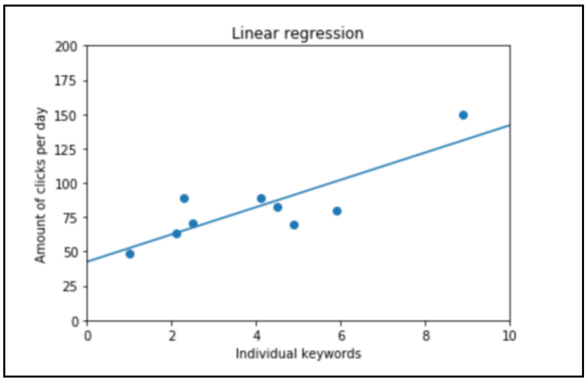
To understand the idea behind linear regression and Root Square Mean Error (RSME).
The context is limited to: Given a variable, y, depending on the independent variable, x,

Precondition

You must have done the Python Basic No. 1 & 2 exercises or have similar knowledge.

The problem

Given a data list with values for y, and another data list with corresponding values for, x, you are to find the values of a and b in the the formula: y = b*x + a, using linear regression; i.e. finding the line which fits the data best. As an example we use the data given in Appendix A and will end up the following plot.



Useful links

https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/regression-analysis/find-a-linear-regression-equation/
https://en.wikipedia.org/wiki/Simple_linear_regression
https://www.dummies.com/education/math/statistics/how-to-calculate-a-regression-line/

Assignment 1: Math behind linear regression

Read the 3 pages (p. 48- 50) in “Machine Learning For Absolute Beginners” Chapter 6 about “Math behind linear regression”. Also given in Appendix A.
Discuss the formula for calculating a and b:

$$a = \frac{(\Sigma y)(\Sigma x^2) - (\Sigma x)(\Sigma xy)}{n(\Sigma x^2) - (\Sigma x)^2}$$

$$b = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{n(\Sigma x^2) - (\Sigma x)^2}$$

Assignment 2: Application program two different lists

We start to calculate the product of elements in two lists and append the result to a third list, xy.

Start Jupyter and a new file, *LinearRegression.*
First, import libraries numpy, pandas and matplotlib.pyplot as plt.
In one cell, declare two lists *x* & *y* of same length and an empty list, *xy,* for the product of elements.

```
list1 = [1, 2, 5]
list2 = [2, 4, 6]
xy = []
for num1, num2 in zip(list1, list2):
    xy.append(num1*num2)
print(xy)
```

Run.

Assignment 3: Function sum of products

Declare a function, def xySum_Prod, to calculate and return the sum of product of elements in in two lists:

```
def xySum_Prod(list1,list2):
......
```

Tip: Similar to assignment 1.
Note: *list2* can be the same as *list1*.

Assignment 4: a, the intersection with the y-axis

Declare a function for calculating and returning a.

```
def find_a(x, y):
    n = len(x)

    xSum = sum(x)
    ySum = sum(y)
    xySum = xySum_Prod(x,y)
    x2Sum = xySum_Prod(x,x)

    a = ((ySum * x2Sum) - .. . . . # fill out the rest yourself.
    return a
```

Call the function for *list1* and *list2*, and print out the result.
Is it correct ?

Assignment 5: b, the slope

In another cell, declare a function for calculating and returning b, the slope:

```
def find_b(x, y):
. . . . . . . . . . .
```

Call the function for *list1* and *list2*, and print out the result.
Is it correct ?

Tip: Very similar to *find_a.*

Assignment 6: Application program

Declare two lists, x and y similar to the data in appendix A

```
#Cost per click of individual keywords
x = [2.3, 2.1, 2.5, 4.5, 5.9, 4.1, 8.9]

#Total amount of clicks per day
y = [89.0, 63.0, 71.0, 70.0, 80.0, 89.0, 150.0]
```

Use your functions to calculate a and b on this data set.

Assignment 7: Application plot of data and line

Use the plot library and plot the diagram and the data points.

```
plt.axis([0, 10, 0, 200])
plt.scatter(x, y)
```

Use plt.title, plt.xlabel and plt.ylabel to apply text according to the plot on page 2.
First, plot a test_line:

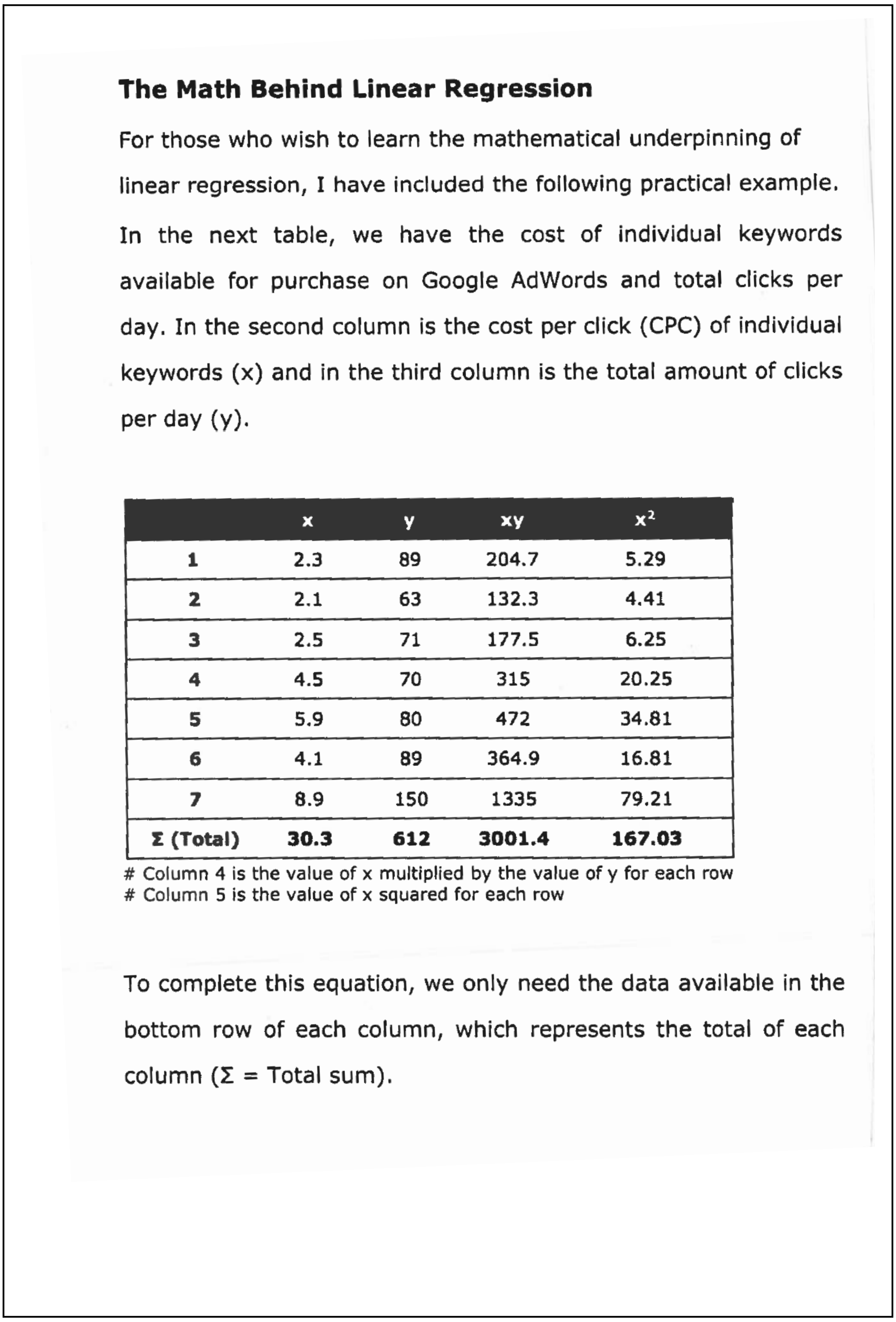
```
test_line = [(10.5*item + 38) for item in [0, 10]]
plt.plot([0, 10], test_line)
```

Finally, plot the regression line b*item + a.

Well done !

Go ahead with the exercise on Root Mean Square Error.

Appendix A



The Math Behind Linear Regression

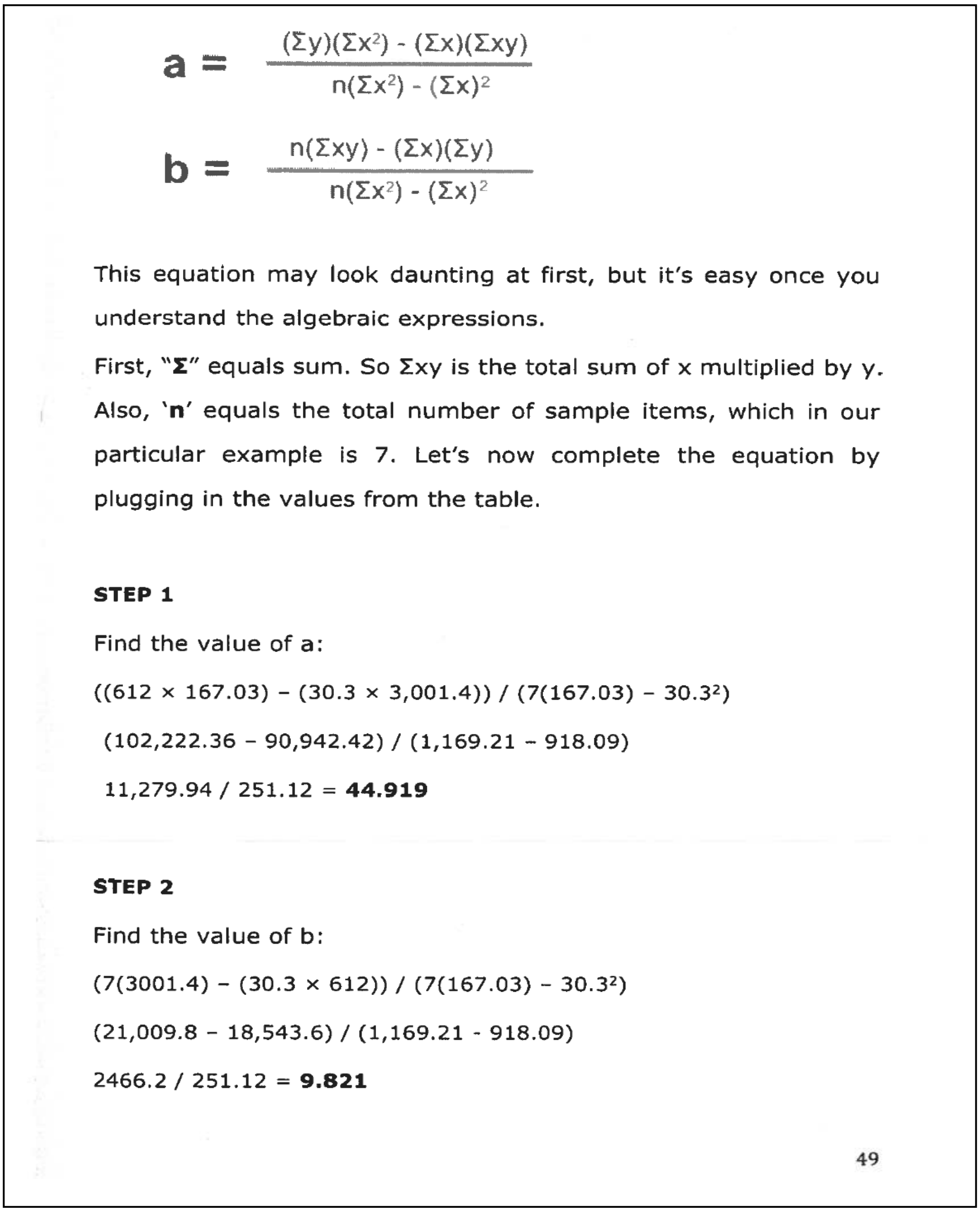
For those who wish to learn the mathematical underpinning of linear regression, I have included the following practical example.

In the next table, we have the cost of individual keywords available for purchase on Google AdWords and total clicks per day. In the second column is the cost per click (CPC) of individual keywords (x) and in the third column is the total amount of clicks per day (y).

	x	y	xy	x^2
1	2.3	89	204.7	5.29
2	2.1	63	132.3	4.41
3	2.5	71	177.5	6.25
4	4.5	70	315	20.25
5	5.9	80	472	34.81
6	4.1	89	364.9	16.81
7	8.9	150	1335	79.21
Σ (Total)	**30.3**	**612**	**3001.4**	**167.03**

Column 4 is the value of x multiplied by the value of y for each row
Column 5 is the value of x squared for each row

To complete this equation, we only need the data available in the bottom row of each column, which represents the total of each column (Σ = Total sum).



$$a = \frac{(\Sigma y)(\Sigma x^2) - (\Sigma x)(\Sigma xy)}{n(\Sigma x^2) - (\Sigma x)^2}$$

$$b = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{n(\Sigma x^2) - (\Sigma x)^2}$$

This equation may look daunting at first, but it's easy once you understand the algebraic expressions.

First, "**Σ**" equals sum. So Σxy is the total sum of x multiplied by y. Also, '**n**' equals the total number of sample items, which in our particular example is 7. Let's now complete the equation by plugging in the values from the table.

STEP 1

Find the value of a:

$((612 \times 167.03) - (30.3 \times 3{,}001.4)) / (7(167.03) - 30.3^2)$

$(102{,}222.36 - 90{,}942.42) / (1{,}169.21 - 918.09)$

11,279.94 / 251.12 = **44.919**

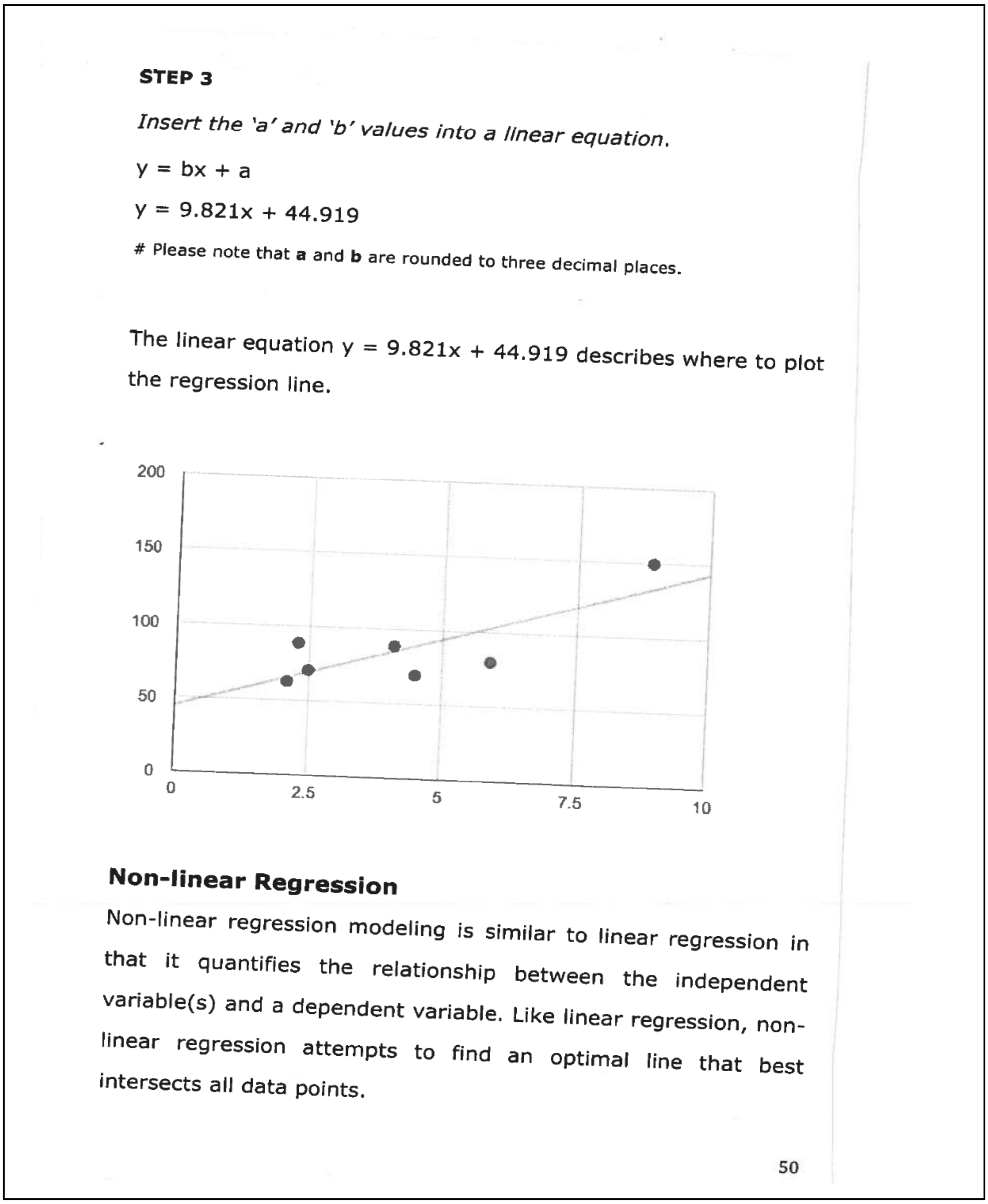
STEP 2

Find the value of b:

$(7(3001.4) - (30.3 \times 612)) / (7(167.03) - 30.3^2)$

$(21{,}009.8 - 18{,}543.6) / (1{,}169.21 - 918.09)$

2466.2 / 251.12 = **9.821**



STEP 3

Insert the 'a' and 'b' values into a linear equation.

y = bx + a

y = 9.821x + 44.919

Please note that **a** and **b** are rounded to three decimal places.

The linear equation y = 9.821x + 44.919 describes where to plot the regression line.

200
150
100
50
0
0
2.5
5
7.5
10

Non-linear Regression

Non-linear regression modeling is similar to linear regression in that it quantifies the relationship between the independent variable(s) and a dependent variable. Like linear regression, non-linear regression attempts to find an optimal line that best intersects all data points.