**Introduction to Data Science Programming**

**Week 5**

**What will we do today?**

* What is Linear Regression?
* How to build a data model and displaying the data model on a graph
* Understanding the results
* Multiple Regression

**What is Linear Regression?**

Linear regression is a basic and commonly used type of predictive analysis. The overall idea of regression is to examine two things: (1) does a set of predictor variables do a good job in predicting an outcome (dependent) variable? (2) Which variables in particular are significant predictors of the outcome variable, and in what way do they–indicated by the magnitude and sign of the beta estimates–impact the outcome variable? These regression estimates are used to explain the relationship between one dependent variable and one or more independent variables. The simplest form of the regression equation with one dependent and one independent variable is defined by the formula y = c + b\*x, where y = estimated dependent variable score, c = constant, b = regression coefficient, and x = score on the independent variable.

[What is Linear Regression? - Statistics Solutions](https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/what-is-linear-regression/)

|  |
| --- |
| **Exercise 1 – Consider a scenario where we would want to predict something based on data we already hold. For example someone’s weight based on their height.**  **Write down the data we would use and what it is we would like to predict based on that.** |

Take the following example:

<https://www.kaggle.com/datasets/karthickveerakumar/salary-data-simple-linear-regression>

**How to build a data model and displaying the data model on a graph**

So how can we use Python and Jupyter Notebook to model this information? The first thing we need is the data sets for height and weight. From the graph above, we know that the x axis (height in inches) goes from 40 – 90 and the y axis (weight in lbs) goes from 80 -160.

So let’s start with our Data Set. For this example, we will look at years’ experience and salary expectations:

|  |  |
| --- | --- |
| YearsExperience | Salary |
| 1.1 | 39343 |
| 1.3 | 46205 |
| 1.5 | 37731 |
| 2 | 43525 |
| 2.2 | 39891 |
| 2.9 | 56642 |
| 3 | 60150 |
| 3.2 | 54445 |
| 3.2 | 64445 |
| 3.7 | 57189 |
| 3.9 | 63218 |
| 4 | 55794 |
| 4 | 56957 |
| 4.1 | 57081 |
| 4.5 | 61111 |
| 4.9 | 67938 |
| 5.1 | 66029 |
| 5.3 | 83088 |
| 5.9 | 81363 |
| 6 | 93940 |
| 6.8 | 91738 |
| 7.1 | 98273 |
| 7.9 | 101302 |
| 8.2 | 113812 |
| 8.7 | 109431 |
| 9 | 105582 |
| 9.5 | 116969 |
| 9.6 | 112635 |
| 10.3 | 122391 |
| 10.5 | 121872 |

We are going to start off entering this manually, we will then look at how we can load this from a file.

So within your Notebook add the following:

%matplotlib inline

import numpy as np

import matplotlib.pyplot as plt

x=(1.1,1.3,1.5,2,2.2,2.9,3,3.2,3.2,3.7,3.9,4,4,4.1,4.5,4.9,5.1,5.3,5.9,6,6.8,7.1,7.9,8.2,8.7,9,9.5,9.6,10.3,10.5)

y=(39343,46205,37731,43525,39891,56642,60150,54445,64445,57189,63218,55794,56957,57081,61111,67938,66029,83088,81363,93940,91738,98273,101302,113812,109431,105582,116969,112635,122391,121872)

plt.scatter(x,y)

plt.show

You should, when you run the code, get the following data:

A picture containing text

Description automatically generated

So our code has, based on the data set we had, plotted these values onto a graph.

The next step is to analyse the data and fit a linear regression model to the data. This is done using the polyfit() function. We also use poly1d() to create the function we will use to display the data points :

Chart

Description automatically generated with medium confidence

So now we have a line of linear regression, however this isn’t telling us much, other than the average salary attached to each year of employment. What we now need is a way of telling us, based on a number of years, what our salary is likely to be:

Adding the line print(np.poly1d(p)) will give us the equation. In this case the output is:

9450 x + 2.579e+04

To calculate your salary, derived from the dataset, we would use 9450\*(number of years) + 2.579. This in essence is telling us that the minimum we will earn is £25,790 but for each extra year worked we would receive £9,450.

Therefore if you work for 5 years, your salary would be: (9450 \* 5) + 25790 = £73,040. We can do this in code with:

print(np.poly1d(p(5)))

Which gives the result: 7.304e+04 which is £73,040.

|  |
| --- |
| Exercise 2  What would be the salary expectations for:  3 years\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  10 years\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  16 years\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  25 years\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Initially calculate this by hand and then use your code in the Jupyter Notebook |

|  |
| --- |
| Exercise 3  Using the data set (Health Insurance (Data Science Week 1) write code that will tell me:   1. The predicted cost of health insurance for a 45 year old 2. The predicted cost of health insurance for someone with 4 children |

**Multiple Regression**

Multiple regression is like [linear regression](https://www.w3schools.com/python/python_ml_linear_regression.asp), but with more than one independent value, meaning that we try to predict a value based on **two or more** variables.

Take a look at the data set below, it contains some information about cars.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Car | Model | Volume | Weight | CO2 |
| Toyota | Aygo | 1000 | 790 | 99 |
| Mitsubishi | Space Star | 1200 | 1160 | 95 |
| Skoda | Citigo | 1000 | 929 | 95 |
| Fiat | 500 | 900 | 865 | 90 |
| Mini | Cooper | 1500 | 1140 | 105 |
| VW | Up! | 1000 | 929 | 105 |
| Skoda | Fabia | 1400 | 1109 | 90 |
| Mercedes | A-Class | 1500 | 1365 | 92 |
| Ford | Fiesta | 1500 | 1112 | 98 |
| Audi | A1 | 1600 | 1150 | 99 |
| Hyundai | I20 | 1100 | 980 | 99 |
| Suzuki | Swift | 1300 | 990 | 101 |
| Ford | Fiesta | 1000 | 1112 | 99 |
| Honda | Civic | 1600 | 1252 | 94 |
| Hundai | I30 | 1600 | 1326 | 97 |
| Opel | Astra | 1600 | 1330 | 97 |
| BMW | 1 | 1600 | 1365 | 99 |
| Mazda | 3 | 2200 | 1280 | 104 |
| Skoda | Rapid | 1600 | 1119 | 104 |
| Ford | Focus | 2000 | 1328 | 105 |
| Ford | Mondeo | 1600 | 1584 | 94 |
| Opel | Insignia | 2000 | 1428 | 99 |
| Mercedes | C-Class | 2100 | 1365 | 99 |
| Skoda | Octavia | 1600 | 1415 | 99 |
| Volvo | S60 | 2000 | 1415 | 99 |
| Mercedes | CLA | 1500 | 1465 | 102 |
| Audi | A4 | 2000 | 1490 | 104 |
| Audi | A6 | 2000 | 1725 | 114 |
| Volvo | V70 | 1600 | 1523 | 109 |
| BMW | 5 | 2000 | 1705 | 114 |
| Mercedes | E-Class | 2100 | 1605 | 115 |
| Volvo | XC70 | 2000 | 1746 | 117 |
| Ford | B-Max | 1600 | 1235 | 104 |
| BMW | 2 | 1600 | 1390 | 108 |
| Opel | Zafira | 1600 | 1405 | 109 |
| Mercedes | SLK | 2500 | 1395 | 120 |

We can predict the CO2 emission of a car based on the size of the engine, but with multiple regression we can throw in more variables, like the weight of the car, to make the prediction more accurate.

**How Does it Work?**

In Python we have modules that will do the work for us. Start by importing the Pandas module.

import pandas

Learn about the Pandas module in our [Pandas Tutorial](https://www.w3schools.com/python/pandas_tutorial.asp).

The Pandas module allows us to read csv files and return a DataFrame object.

The file is meant for testing purposes only, you can download it here: [cars.csv](https://www.w3schools.com/python/cars.csv)

df = pandas.read\_csv("cars.csv")

Then make a list of the independent values and call this variable X.

Put the dependent values in a variable called y.

X = df[['Weight', 'Volume']]  
y = df['CO2']

We will use some methods from the sklearn module, so we will have to import that module as well:

from sklearn import linear\_model

From the sklearn module we will use the LinearRegression() method to create a linear regression object.

This object has a method called fit() that takes the independent and dependent values as parameters and fills the regression object with data that describes the relationship:

regr = linear\_model.LinearRegression()  
regr.fit(X, y)

Now we have a regression object that are ready to predict CO2 values based on a car's weight and volume:

#predict the CO2 emission of a car where the weight is 2300kg, and the volume is 1300cm3:  
predictedCO2 = regr.predict([[2300, 1300]])

### **Example**

See the whole example in action:

import pandas  
from sklearn import linear\_model  
  
df = pandas.read\_csv("cars.csv")  
  
X = df[['Weight', 'Volume']]  
y = df['CO2']  
  
regr = linear\_model.LinearRegression()  
regr.fit(X, y)  
  
#predict the CO2 emission of a car where the weight is 2300kg, and the volume is 1300cm3:  
predictedCO2 = regr.predict([[2300, 1300]])  
  
print(predictedCO2)

### **Result:**

[107.2087328]

|  |
| --- |
| **Exercise 4**  Practice using the datasets and questions from the following link in Kaggle:  [Datasets for regression analysis | Kaggle](https://www.kaggle.com/code/rtatman/datasets-for-regression-analysis/notebook) |