Chapter 6 : Data Analysis with Python

In this chapter, we shall discuss how to do data analysis with Python. In this exercise, you need to call several Python libraries that has been discussed in the earlier chapter (Numpy, Matplotlib and Pandas). For this topic, we choose experiment to prove Hooke's Law. The steps includes:

1. A brief description of what Hooke´s Law is?
2. Collect one set of data to investigate Hooke´s Law.
3. Make a graph of the data and compare it to what you would expect to see.
4. Use your graph to calculate the value of the "spring constant (k").

What is Hooke's Law?

Hooke's Lae says that when an elastic object is stretched, the increase lenght of this object is its extension. In this experiment, we are going to use as the elastic object a sping. In our opinion, the spring is the better object to show this law constant.
The extension of the spring is directly proportional to the force applied to it and we used at the report the following formula, where:

$$F = kx$$

where

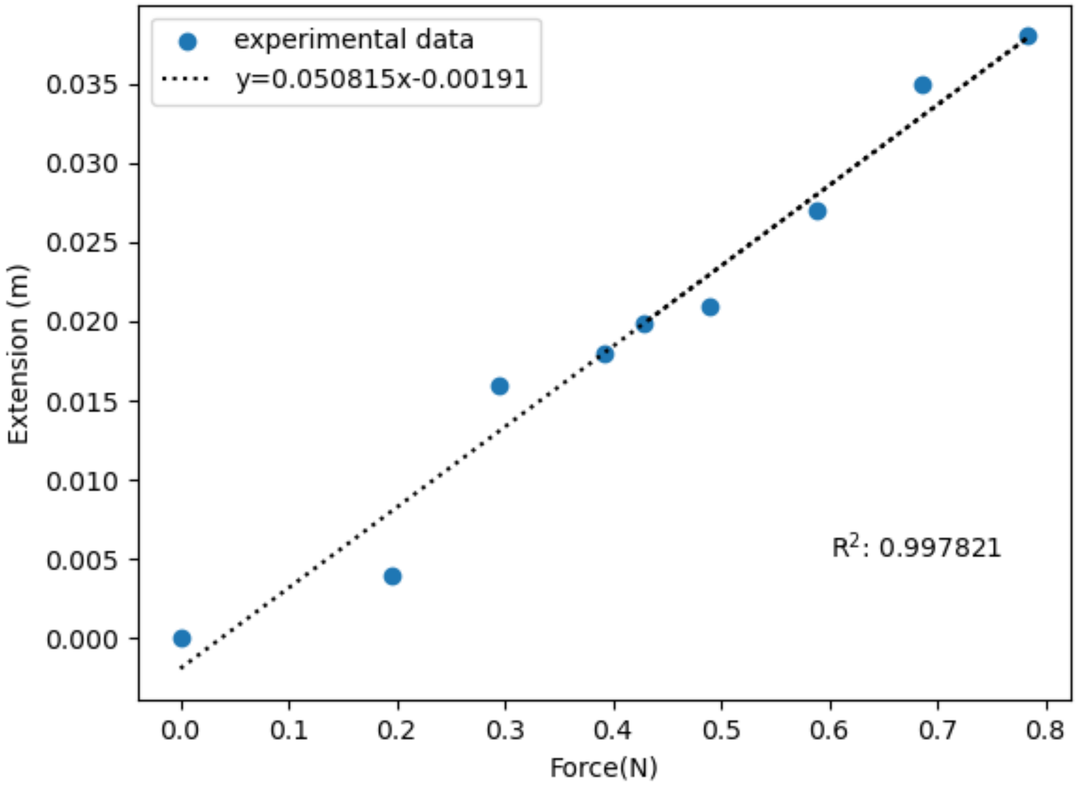
F is the force in newtons, N
k is the 'spring constant' in newtons per metre, N/m
x is the extension in metres, m

This equation works when the elastic limit of the spring or the limit of proportionally is not exceeded. If a spring is stretched too much, for example, it will not return to its original length when the load is removed, and this could be a problem for evaluating when the experiment was finished.

Result:

Trial	Mass (kg)	Force (N)	Position (m)	Extension (m)
1	0.00	0.000	0.310	0.000
2	0.02	0.196	0.314	0.004
3	0.03	0.294	0.326	0.016
4	0.04	0.392	0.328	0.018
5	0.05	0.490	0.331	0.021
6	0.06	0.588	0.337	0.027
7	0.07	0.686	0.345	0.035
8	0.08	0.784	0.348	0.038
Average	0.044	0.429	0.330	0.020

The graph shows the relationship between Force in Newtons and Extension in Meters:



The spring constant in newton per meter (k), is calculated by dividing the force (n)= mass (kg) x (acceleration=9.8 m/s2) by the extension (e) in meters. This is equal to 0.429 : 0.020= 21.45 N/m.

Full program: Hooke.py

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

#read data from pandas
df = pd.read_excel('../Hooke.xlsx')
#print(df)
plt.xlabel('Force(N)')
plt.ylabel('Extension (m)')
plt.scatter(df['Force (N)'],df['Extension (m)'],label='experimental
data')

#Perform linear regression using numpy
#We need to fit y= mx + c
# We have to wewrite the equation as y = Ap, A = [[x 1]], p = [[m],[c]]

x = df['Force (N)']
y = df['Extension (m)']

#rearrange x-value in stack and assign as new matrix A
A = np.vstack([x, np.ones(len(x))]).T
```

```
print(A)

# m is the slope
# c is the intercept
m,c = np.linalg.lstsq(A,y,rcond=None)[0]
print(m,c)

#residuals
residuals = np.abs(y - np.dot(x, m))
#find the r-square of the calculation
r2 = np.average(1 - residuals)
print(f"R-squared: {r2:.6f}")

#plot the least square (linear regression)
plt.plot(x,m*x+c,
'k:',label='y=0.050815x-0.00191')
plt.text(0.6,0.005,f'R$^2$: {r2:.6f}')

#show labels using legend command
plt.legend()

#save figure
plt.savefig('Hooke')

#show plot on the screen
plt.show()
```