▾ Python Tips for Lab Work

This resource will better help you with general python practise in your experiments. documentation is always useful: https://docs.scipy.org/doc/scipy/ https://numpy.org/doc/ https://matplotlib.org/

```
#Using appropriate imports
#Make sure you use the basic imports such as are listed below.

import numpy as np
import matplotlib.pylab as plt
import astropy as ast #astrophysics libary good for calling constants.
import scipy as sci
import pandas as pd
```

▾ Importing Data

```
# Importing data from a CSV file. e.g. exel sheet.
#using numpy

data = np.genfromtxt('my_CSV.csv',delimiter=',')
#or you can use pandas, a better data organising tool.

Data = pd.read_csv('data.csv')
# or if you have a exel file

DATA = pd.read_excel('data.xlsx')
```

▾ Using Functions to reuse and simply your code.

```
# as an example.
# I need to change from x,y coordinates to polar coordinates.
# I will need to do this multiple times so I will write it in a function.


def change_cords(x,y):
    r = np.sqrt(x**2 +y**2)
    theta = np.arctan(y/x)


    return r, theta


# Now I can call my function when ever I would like.
```

```
r, theta = change_cords(1,2)
print('R :',r, 'Theta :',theta)
```

```
    R : 2.23606797749979 Theta : 1.1071487177940904
```

```
# We can actually use a loop to do this when we want to convert a large number of data.

# lets make a array of x,y
R = np.array([[1,2],[2,3],[4,2],[4,5]])
#print(R)

for i in range(0,3):
    r, theta = change_cords(R[0],R[1])
    print(r, theta)

# to improve this we can have these number be written in to a array. You shoudl do this.
```

```
    [2.23606798 3.60555128] [1.10714872 0.98279372]
    [2.23606798 3.60555128] [1.10714872 0.98279372]
    [2.23606798 3.60555128] [1.10714872 0.98279372]
```

▾ Graphing

If you already have your data in python it is easier to graph and analyse your data here opposed to other software origin.

Below is quite a basic plot by the essential x,y labels and possibly a title shoud be included. The data is random so does not show any relation. When your data does you will want to do some further analysis on it. In origin this analysis is abit of a black box. But here you need to know how to do it and usually implement it yourselves. Sometimes it is useful to use in built functions that numpy or scipy can offer.

```
#Using matplot Lib
# lets just plot the X,Y from R on a graph with lables and title

plt.plot(R[:,0],R[:,1])
plt.xlabel('X (m)')
plt.ylabel("Y (m)")
plt.title('Plot of X vs Y')
plt.show()
```

Plot of X vs Y
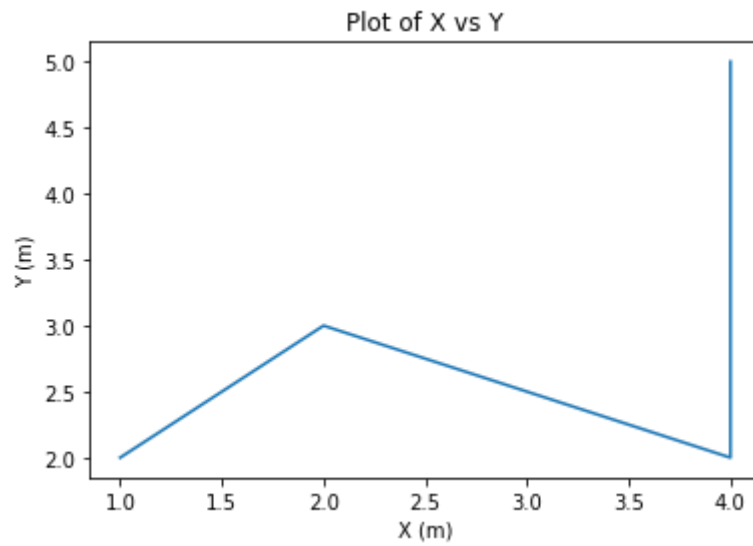
Colab paid products - Cancel contracts here