

```
[kev@yogi linear-regression]$ ./line-plot-03 1.5 10 > line1.csv  
[kev@yogi linear-regression]$ ./line-plot-03 2.5 20 > line2.csv  
[kev@yogi linear-regression]$ ./line-plot-03 1.0 30 > line3.csv
```

Edit the top line of each file to have different headings in each in each file. Examples of the tops of each file after editing are below.

```
[kev@yogi linear-regression]$ head -3 line1.csv  
x,line1  
0.000000,10.000000  
0.100000,10.150000  
[kev@yogi linear-regression]$ head -3 line2.csv  
x,line2  
0.000000,20.000000  
0.100000,20.250000  
[kev@yogi linear-regression]$ head -3 line3.csv  
x,line3  
0.000000,30.000000  
0.100000,30.100000
```

Start Jupyter Notebook

```
[kev@yogi linear-regression]$ jupyter notebook
```

Files

Running

Clusters

BeakerX

Select items to perform actions on them.

Upload

New ▾



Notebook:

Clojure

Groovy

Java

Kotlin

Python 3

SQL

Scala

Other:

Create a new notebook with Python 3

File

Edit

View

Insert

Cell

Kernel

Help

Trusted

Python 3 ○



Run Cells

Run Cells and Select Below

Run Cells and Insert Below

Run All

Run All Above

Run All Below

Cell Type

Code

Markdown

Current Outputs

All Output

Raw

Contents will be rendered as HTML and serve as explanatory text

```
In [ ]: 1 # This is a comment
        2
        3 This is a line of code
        4
        5 *My name is John
```

```
In [ ]: 1 # This is a title
        2
        3 This is some text
        4
        5 *My name in italics*
```

Press control-enter and it will get rendered

This is a title

This is some text

My name in italics

Click the plus on the toolbar to add new cells. By default they will be code cells. Press control-enter to run them. The next pages show code for plotting data

```
In [2]: 1 import matplotlib.pyplot as plt
        2 import pandas as pd
```

```
In [3]: 1 data1 = pd.read_csv('line1.csv')
        2 data1.head()
```

Out[3]:

	x	line1
0	0.0	10.00
1	0.1	10.15
2	0.2	10.30
3	0.3	10.45
4	0.4	10.60

```
In [5]: 1 data2= pd.read_csv('line2.csv')
        2 data2.head()
```

Out[5]:

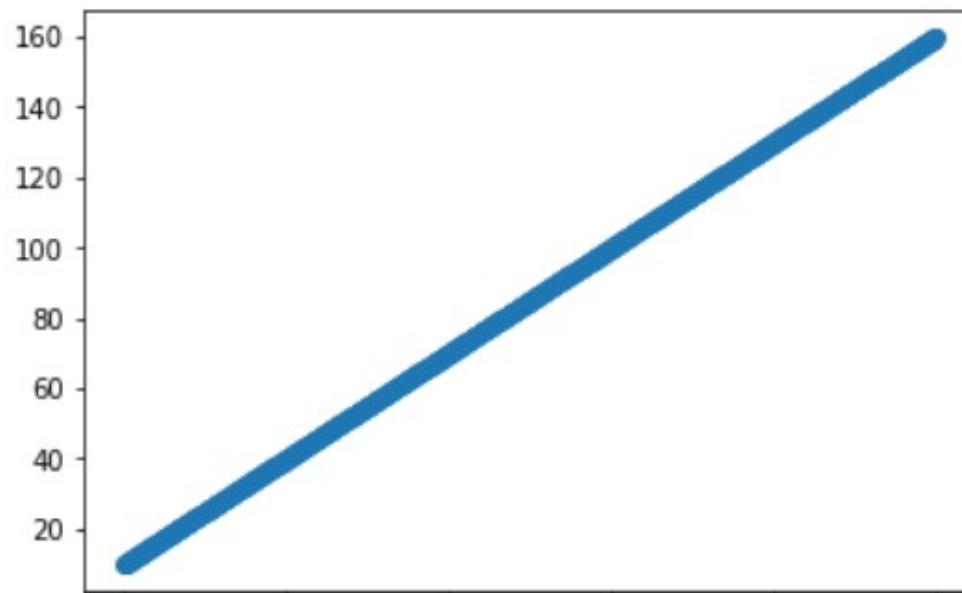
	x	line2
0	0.0	20.00
1	0.1	20.25
2	0.2	20.50

```
In [7]: 1 data3 = pd.read_csv('line3.csv')
        2 data3.head()
```

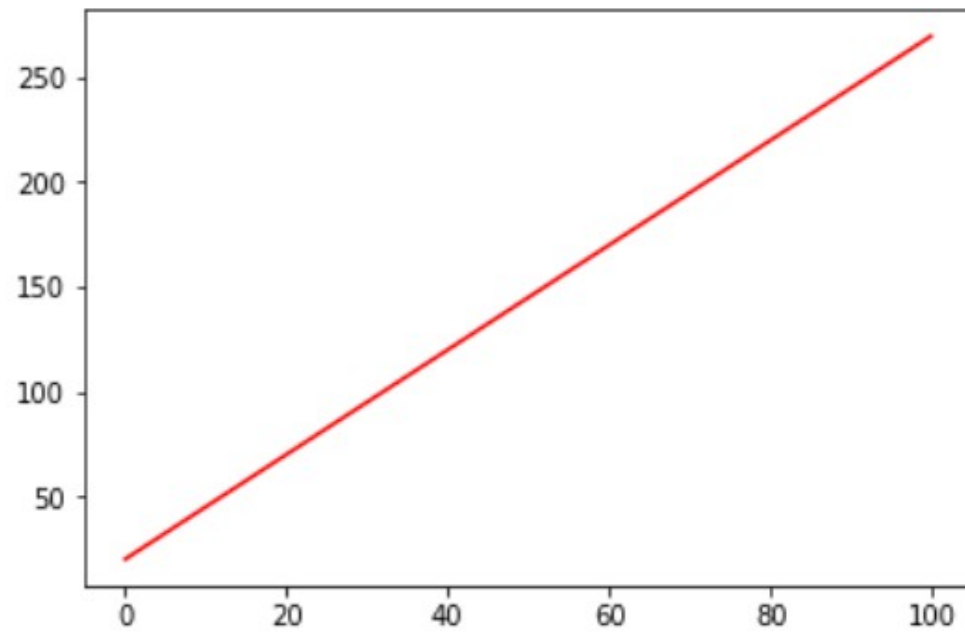
Out[7]:

	x	line3
0	0.0	30.0
1	0.1	30.1
2	0.2	30.2
3	0.3	30.3
4	0.4	30.4

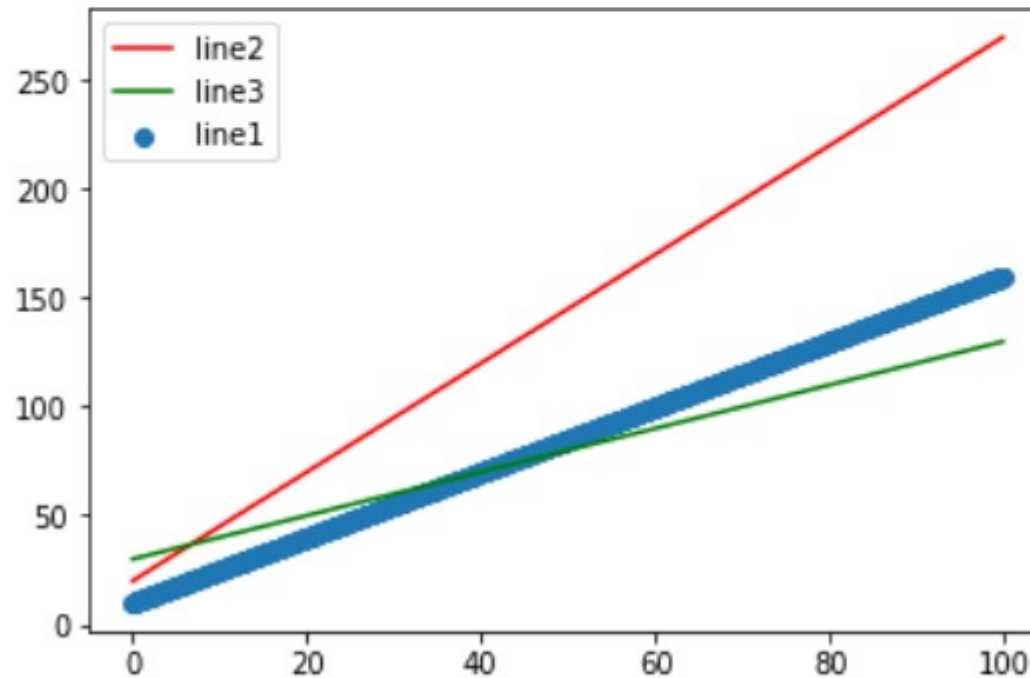
```
In [6]: 1 plt.scatter(data1.x, data1.line1)
        2 plt.show()
```



```
In [9]: 1 plt.plot(data2.x, data2.line2, c='r')  
        2 plt.show()
```



```
In [10]: 1 plt.scatter(data1.x, data1.line1)
          2 plt.plot(data2.x, data2.line2, c='r')
          3 plt.plot(data3.x, data3.line3, c='g')
          4 plt.legend()
          5 plt.show()
```



For your portfolio you can use *scatter* discrete data points and *plot* for the *guess* lines