Command Line Graphing

These tools have been created to plot data from flat/raw files. If you don't like the API functions, you can use the CLI 's to draw similar graphs.

Line Plots

```
Options:

  -h, --help            show this help message and exit

  -f RAW, --file=RAW    The raw file to be processed

  -s SCALE, --scale=SCALE
                        Can be one of 'logx', 'logy',

                       'logxy' default is linear

  -t TITLE, --title=TITLE
                        The title of your chart

  -x XLABEL, --xlabel=XLABEL
                        The ylabel of your chart

  -y YLABEL, --ylabel=YLABEL
                        The xlabel of your chart

  -m, --multi           select this is you want to graph
                        multiple lines on a
                        single graph
```

Although the help for the command is intuitive, it is worth discussing that it has two plotting modes:

Single Data Series mode (without –m option)

Raw File:

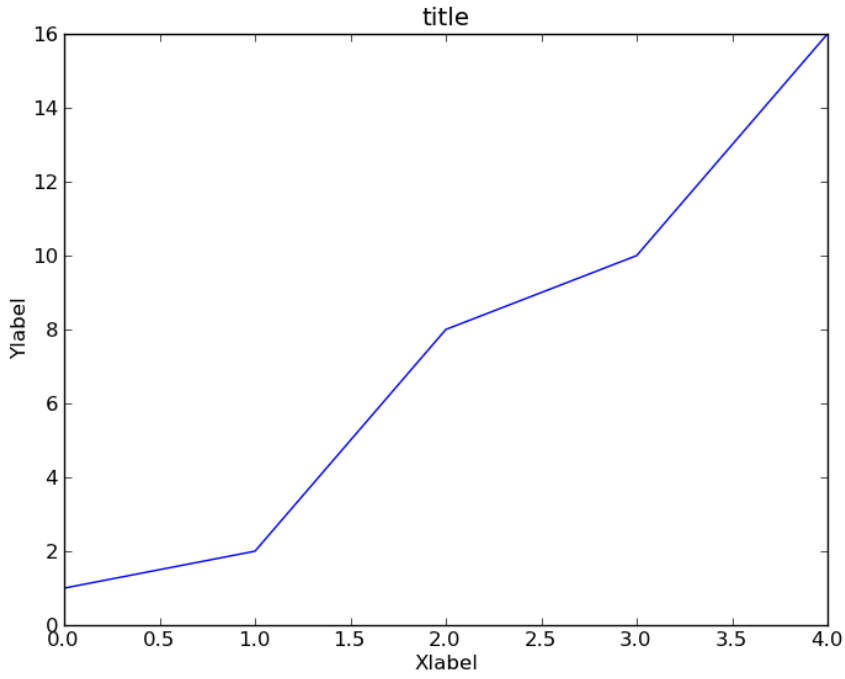
```
1
2
8
10
16
```

Command:

```
line.py -f sample_data/line_data -x "Xlabel" -y "Ylabel"
-t "title"

Processing Data from file sample_data/line_data
Using Linear Scale
Received title as: title
Recieved X-Label as: Xlabel
Recieved Y-Label as: Ylabel
```

The graph gets plotted as:



The multi mode allows you to plot more than line in a single plot. The data must consist of “legend=<name>” as shown in the example below:

Raw file:

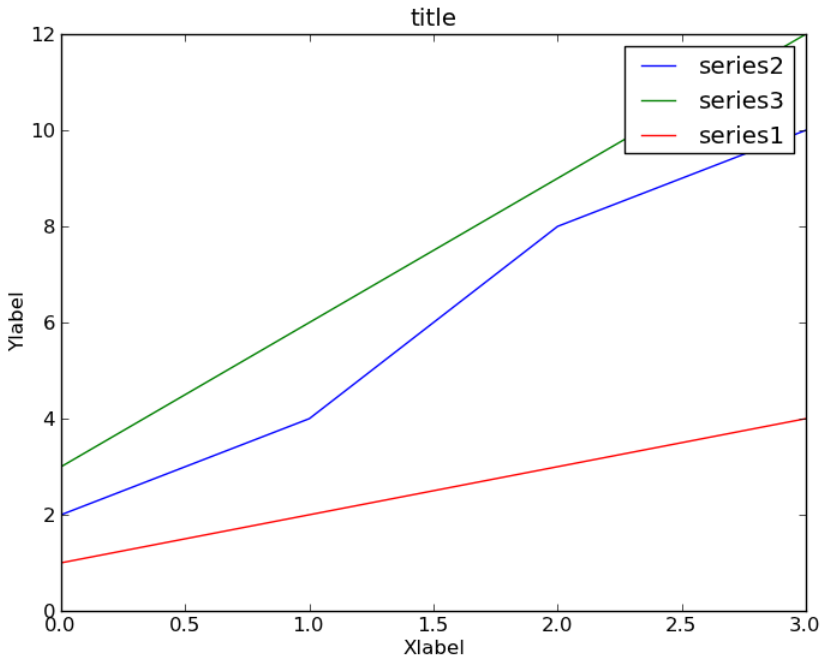
```
legend=series1
1
2
3
4
legend=series2
2
4
```

```
8
10
legend=series3
3
6
9
12
```

Command:

```
line.py -m -f sample_data/multi_line_data -x "Xlabel" -y
"Ylabel" -t "title"
Processing Data from file sample_data/multi_line_data
Identifeid series as series1
Identifeid series as series2
Identifeid series as series3
Using Linear Scale
Using Linear Scale
Using Linear Scale
Received title as: title
Recieved X-Label as: Xlabel
Recieved Y-Label as: Ylabel

The graph is plotted as:
```



line_xy is used to plot catesian data. Instead of a single series we will have point data.

```
Options:
  -h, --help            show this help message and exit

  -f RAW, --file=RAW    The raw file to be processed

  -s SCALE, --scale=SCALE
                        Can be one of 'logx', 'logy',
                       'logxy' default is linear

  -t TITLE, --title=TITLE
                        The title of your chart

  -x XLABEL, --xlabel=XLABEL
                        The ylabel of your chart

  -y YLABEL, --ylabel=YLABEL
                        The xlabel of your chart

  -m, --multi           select this is you want to graph
                        multiple lines on a single graph
```

Single Plot Raw Data:

```
535 202319
536 91439
537 322442
538 272258
539 193520
540 139966
```

Multi Plot Raw Data:

```
legend=squares
2 4
3 9
4 16
legend=cubes
1 1
2 8
3 27
legend=roots
```

```
1 1
2 1.41
3 1.732
```

Histogram

hist.py –h

```
  -h, --help            show this help message and exit

  -f RAW, --file=RAW    The raw file to be processed

  -t TITLE, --title=TITLE
                        The title of your chart

  -x XLABEL, --xlabel=XLABEL
                        The ylabel of your chart

  -y YLABEL, --ylabel=YLABEL
                        The xlabel of your chart

  -b BINS, --bins=BINS  The number of bins in the
                        distribution
```

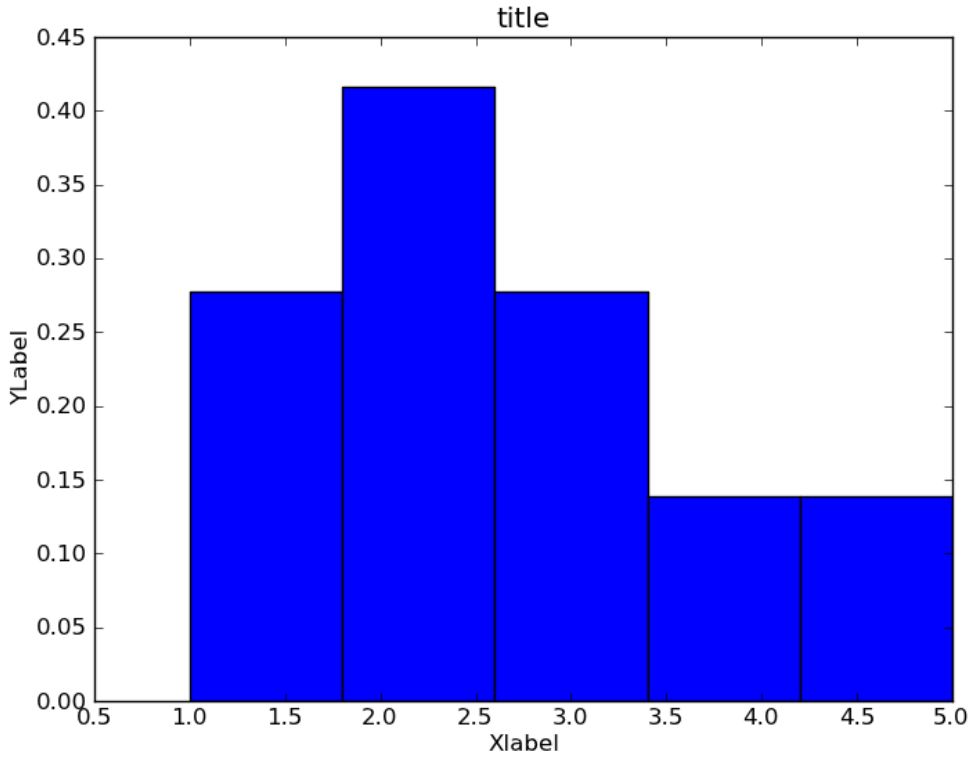
The bins is the number of intervals for the histogram.

Raw Data:

```
1
1
2
2
2
3
3
4
5
```

Command:

```
hist.py -f sample_data/hist -x "Xlabel" -y "YLabel" -t
title -b 5
Processing Data from file sample_data/hist
Received title as: title
Recieved X-Label as: Xlabel
Recieved Y-Label as: YLabel
plotting with: 5
```



Scatter Plot

```
scatter.py -h

  -h, --help            show this help message and exit

  -f RAW, --file=RAW    The raw file to be processed

  -t TITLE, --title=TITLE
                        The title of your chart

  -x XLABEL, --xlabel=XLABEL
                        The ylabel of your chart

  -y YLABEL, --ylabel=YLABEL
                        The xlabel of your chart
```

Raw Data:
1 2
2 4
9 6
4 4
2 1

Command:

```
scatter.py -f sample_data/scatter -x "Xlabel" -y "YLabel"
-t title
Processing Data from file sample_data/scatter
Received title as: title
Recieved X-Label as: Xlabel
Recieved Y-Label as: YLabel
```

The Scatter Plot: