Interactive Plotting

FOSSEE

1 Starting up...

```
  $ ipython -pylab
```

Exiting

```
In [2]: (Ctrl-D)^D
Do you really want to exit ([y]/n)? y
```

2 Plotting

2.1 linspace

`In []: x = linspace(start, stop, num)`
`linspace` returns array of length `num`, for which `x[0] = start` and `x[num-1] = stop`
Please note indices of array starts from zero(0)

2.2 plot

`In []: plot(X, Y)`
For given arrays of equal length(above case X and Y), `plot` plots the corresponding *x* and *y* pairs taken from X and Y.

2.3 Colors of plots

`In []: plot(y, sin(y), 'g')`
Plots graph with green color. Other options available are:

```
  'r' ---> Red
  'b' ---> Blue
  'r' ---> Red
  'c' ---> Cyan
  'm' ---> Magenta
  'y' ---> Yellow
  'k' ---> Black
  'w' ---> White
```

One can set the width of the plotline using optional argument `linewidth`. For example:

```
In []: plot(x, cos(x), 'r', linewidth=2)
```

Plots the line with linewidth = 2

2.4 label and title

```
In []: xlabel('Length') #sets *x* axis label to Length
In []: ylabel('Time') #sets *y* axis label to Time.
In []: title('Sinusoids') #sets title of plot
```

Additionally

Pylab accepts TeX equation expressions in any text expression. To get something like:

$\sigma_i = 15$

on title of figure use:

```
In []: title('$\sigma_i=15$')
```

Same way one can have TeX expression on xlabel, ylabel etc.

2.5 legends

```
In []: legend('sin(x)',loc=center)
```

Places a legend on the current plot at location *loc*.

Apart from `center`, some other `loc` which can be specified are:

```
'best'
'right'
'upper right'
'upper left'
'lower left'
'lower right'
'center left'
'center right'
'lower center'
'upper center'
```

One can also mention explicit co-ordinates for placement of legend.

```
In []: legend(['sin(2y)'], loc=(.8,.1))
```

`loc = 0, 1` (top left position of graph)
`loc = 0.5, 0.5` (center of graph).

2.6 Annotate

```
In []: annotate('local max', xy=(1.5, 1))
```

Annotates current plot with text, 'local max', at position specified to `xy`.

2.7 Saving figures

```
In []: savefig('sinusoids.png')
```

Saves the current figure with file name 'sinusoids.png' in current working directory. One can save figure in any of these formats: png, pdf, ps, eps and svg.

2.8 Miscellaneous

```
In []: clf() #Clears the current plot area
In []: close() #Closes the figure
```

3 Saving and running scripts

- `%hist`
  It returns the logs of all commands(including mistakes) used in IPython interpreter.
- `%hist -n`
  It disables the line number representation of logs.
- `%save four_plot.py 16 18-27`
  For creating a script named four_plot which includes line 16 and line 18 to 27 of logs.
- `%run -i four_plot.py`
  Running the python script inside IPython interpreter.

4 Example

```
In []: x = linspace(0, 2*pi, 50)
In []: plot(x, sin(x), ’g’)
In []: plot(x, cos(x), ’r’, linewidth=2)
In []: xlabel(’x’)
In []: title(’Sinusoidal Waves’)
In []: legend([’sin(x)’, ’cos(x)’])
In []: annotate(’origin’, xy=(0, 0))
In []: xmin, xman = xlim()  # returns current X axis limits.
In []: ymin, ymax = ylim()
In []: xlim(0, 2 * pi) # sets the X axis limits to passed values
In []: ylim(ymin - 0.2, ymax + 0.2)

In []: savefig(’sin.png’)   # Save figure
In []: close()
```

5 References

- For documentation on IPython refer:
  http://ipython.scipy.org/moin/Documentation
- Plotting(matplotlib) related documentation are available at:
  http://matplotlib.sourceforge.net/contents.html
- Explore examples and plots based on matplotlib at
  http://matplotlib.sourceforge.net/examples/index.html