Exercise1 Matplotlib

April 15, 2018

1 Matplotlib

Documentation: http://matplotlib.org/

Matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

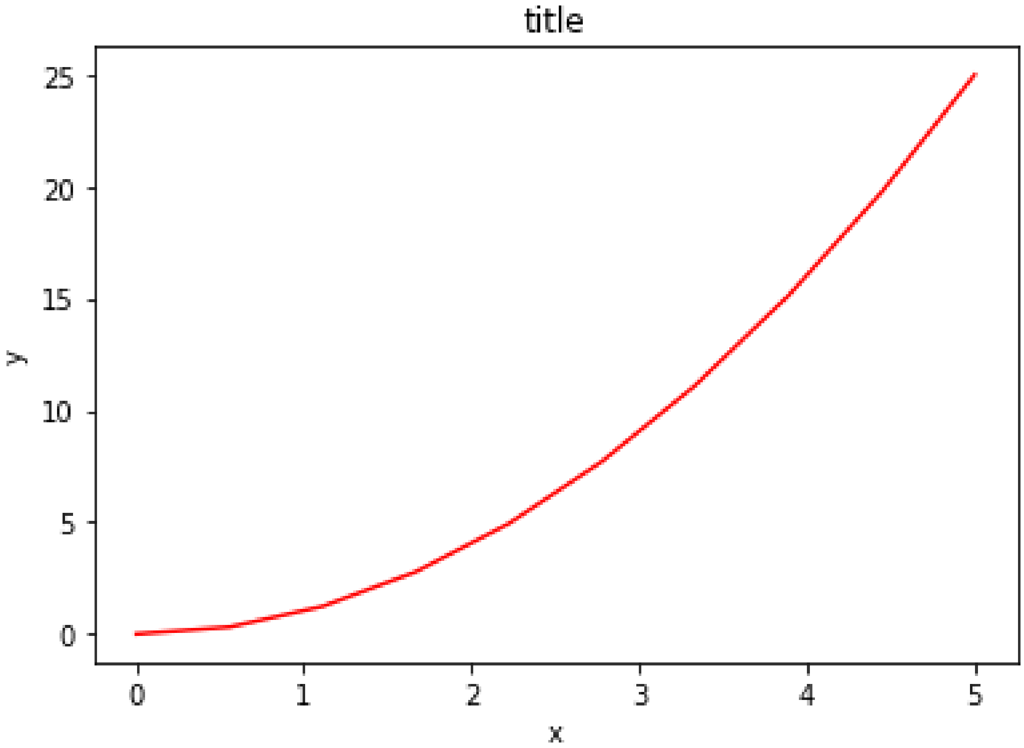
You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc.

```
In [1]: # needed to display the graphs
        %matplotlib inline
        from pylab import *

In [2]: import numpy as np
        import matplotlib.pyplot as plt

        np.random.seed(1)
```

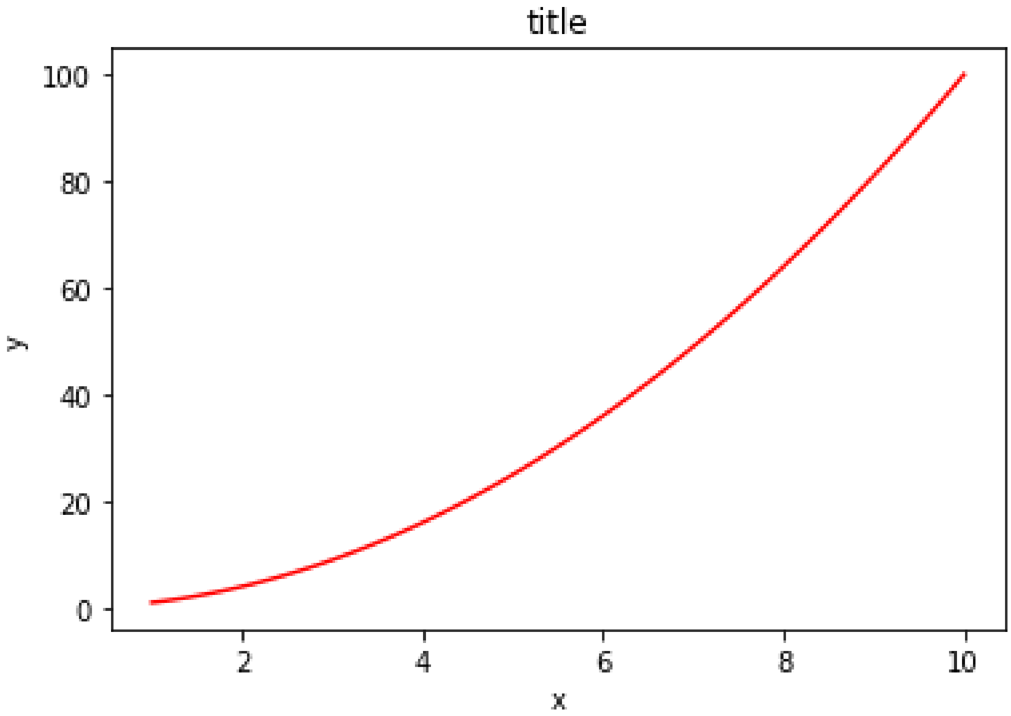
1.1 Task 1

- Create a plot $y = x^2$ for $x \in [1 : 10]$
- Add Title and Axes (Replicate the plot below)

```
In [5]:
```



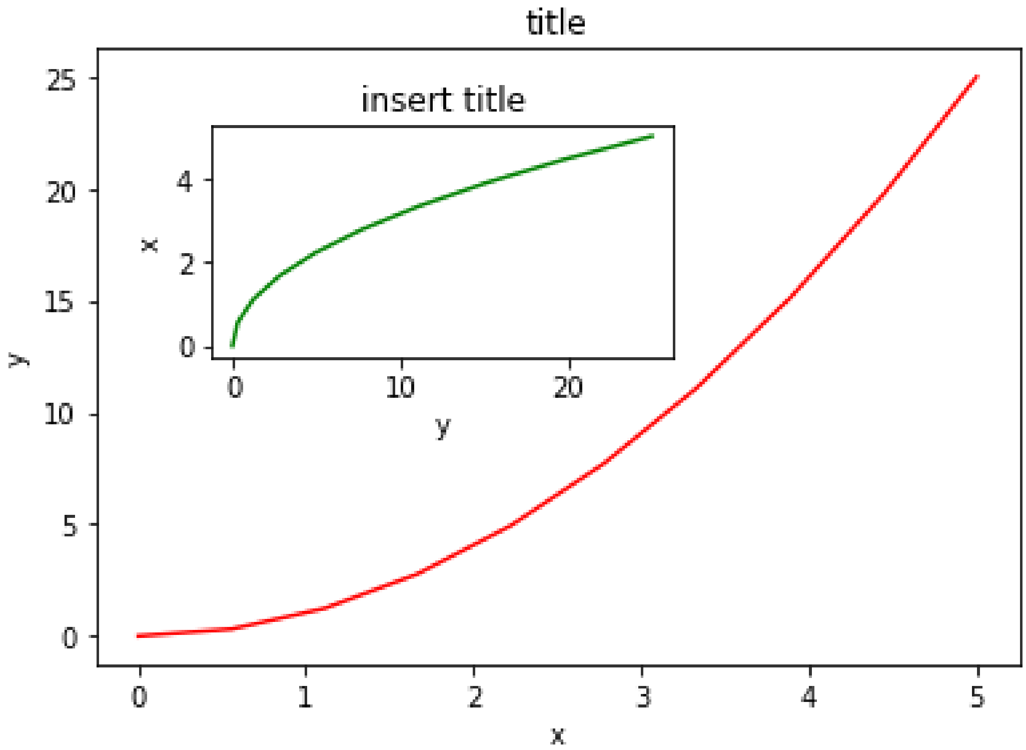
```
In [3]: x = np.linspace(1,10,100)
        y = x ** 2

        plt.plot(x,y,'r')

        plt.xlabel('x')
        plt.ylabel('y')

        plt.title('title')

        plt.show()
```



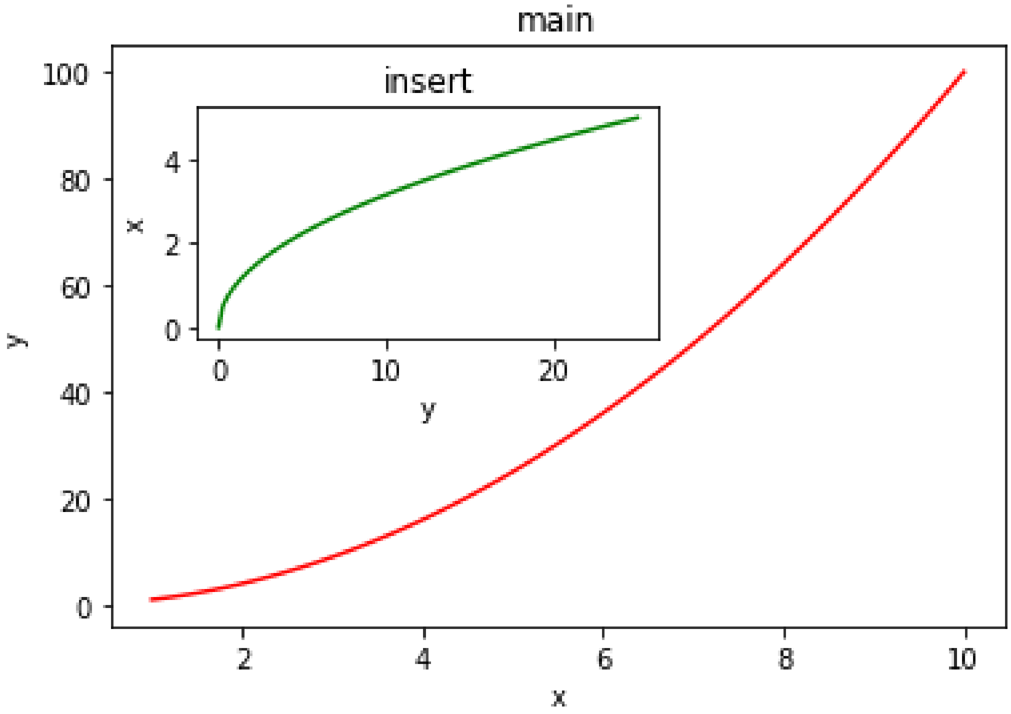
1.2 Task 2

Create two plots: 'main' and 'insert' and place them such that - The 'insert' plot are included into the 'main' plot - The 'insert' is next to the 'main' plot (Replicate the plots below)

```
In [6]:
```



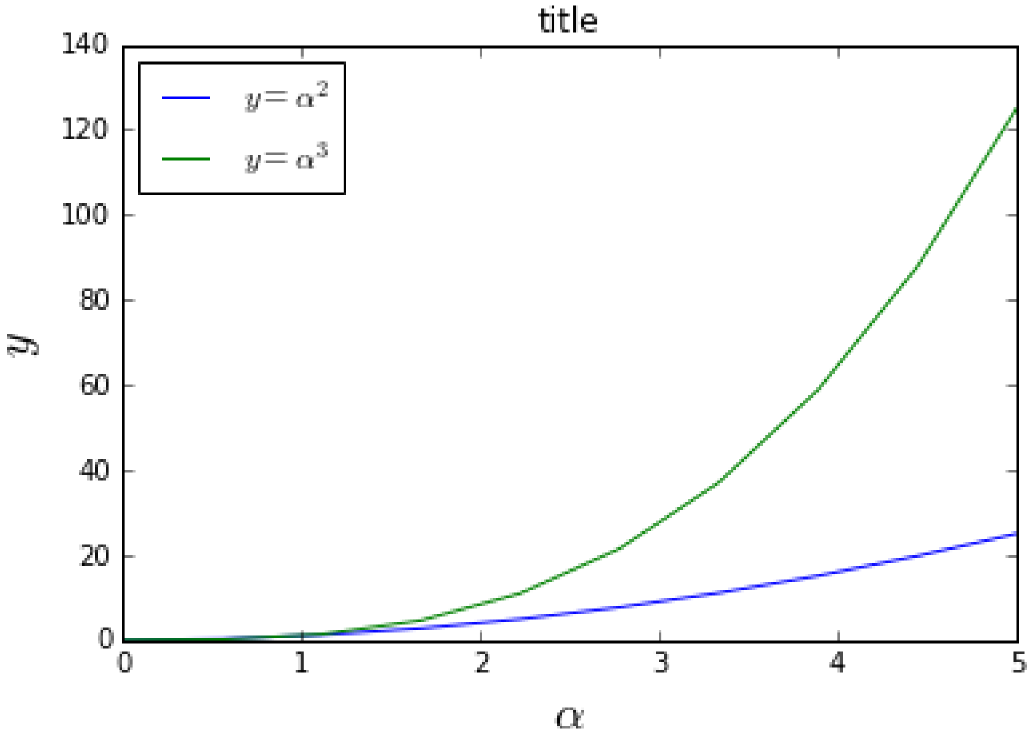
```
In [4]: # Plot the main plot

        x = np.linspace(1,10,100)
        y = x ** 2

        plt.plot(x,y,'r')

        plt.xlabel('x')
        plt.ylabel('y')

        plt.title('main')

        # Plot subplot
        a = plt.axes([0.2, 0.5, 0.4, 0.3])

        insert_y = np.linspace(0, 25, 100)
        insert_x = np.sqrt(insert_y)

        plt.plot(insert_y, insert_x, 'g')

        plt.xlabel('y')
        plt.ylabel('x')
        plt.title('insert')
plt.show()
```



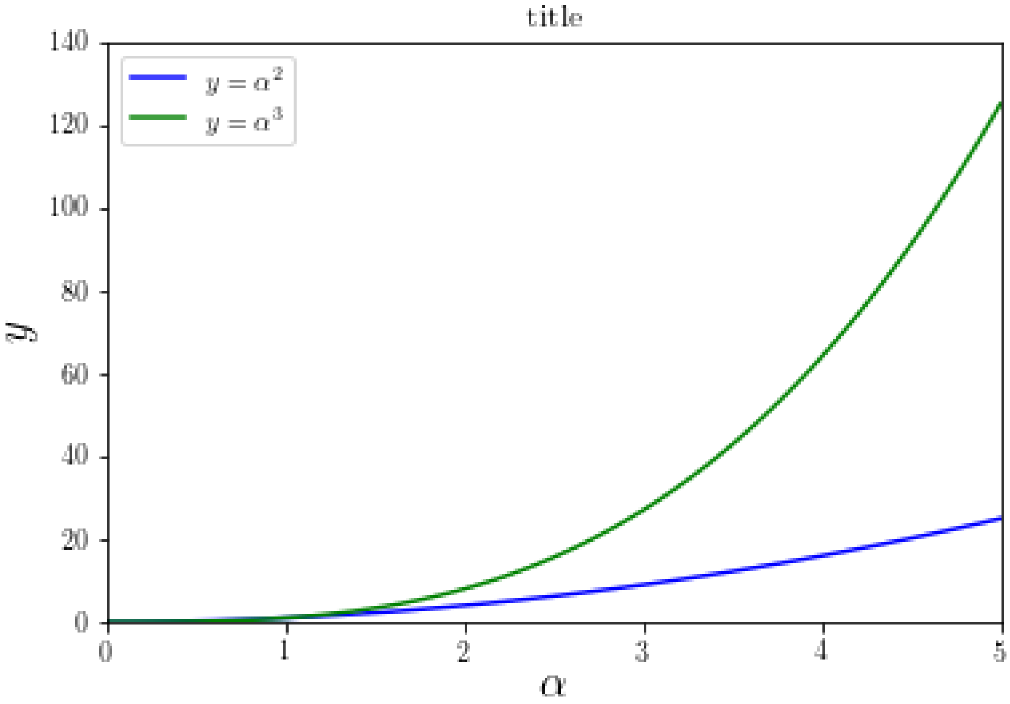
1.3 Task 3

Create a plot with a legend and latex symbols

```
In [8]:
```



```
In [5]: plt.rc('text', usetex=True)

        alpha = np.linspace(0, 5, 100)
        alpha_2 = alpha ** 2
        alpha_3 = alpha ** 3

        plt.rc('text', usetex=True)
        plt.rc('font', family='serif')

        plt.plot(alpha, alpha_2, 'b', label=r'$y = \alpha ^ 2$')
        plt.plot(alpha, alpha_3, 'g', label=r'$y = \alpha ^ 3$')

        plt.xlabel(r'\LARGE{$\alpha$}')
        plt.ylabel(r'\LARGE{$y$}')
        plt.title('title')

        plt.xlim(0, 5)
        plt.ylim(0, 140)


        plt.legend()
        plt.show()
```



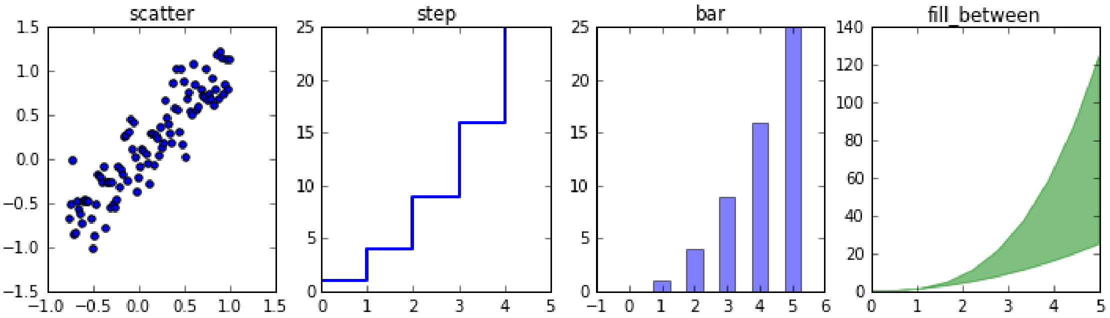
1.4 Task 4

Other plot styles. Given:

```
In [6]: xx = np.linspace(-0.75, 1., 100)
        n = array([0,1,2,3,4,5])
```

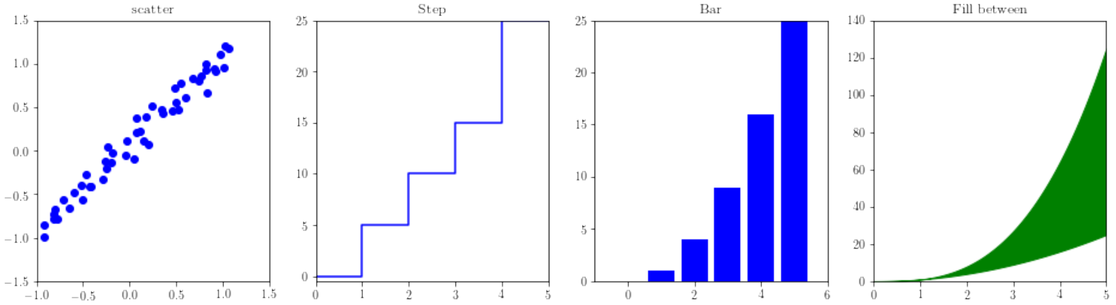
Generate: scatter, step, bar, fill_between

```
In [11]:
```



```
In [7]: plt.figure(figsize=(16,4))

        # Scatter plot
        plt.subplot(141)

        x = np.linspace(-1, 1, 50)
        y = x
        plt.scatter(x + np.random.rand(len(x)) / 5, y + np.random.rand(len(y)) / 3, marker='o',

        plt.xlim(-1.0, 1.5)
        plt.ylim(-1.5, 1.5)

        plt.title('scatter')


        # Step plot
        plt.subplot(142)

        x = np.array([0, 1, 2, 3, 4, 5])
        y = np.array([0, 0, 5, 10, 15, 25])

        plt.step(x, y, where='pre', color='b')


        plt.xlim(0, 5)
        plt.ylim(-0.5, 25)
        plt.title('Step')


        # Bar chart
        plt.subplot(143)

        x = np.array([0, 1, 2, 3, 4, 5])
        y = x ** 2

        plt.bar(x, y, color='b')

        plt.xlim(-1, 6)
        plt.ylim(0, 25)
        plt.title('Bar')

        # Bar chart
        ax = plt.subplot(144)

        alpha = np.linspace(0, 5, 100)
        alpha_2 = alpha ** 2
        alpha_3 = alpha ** 3
```

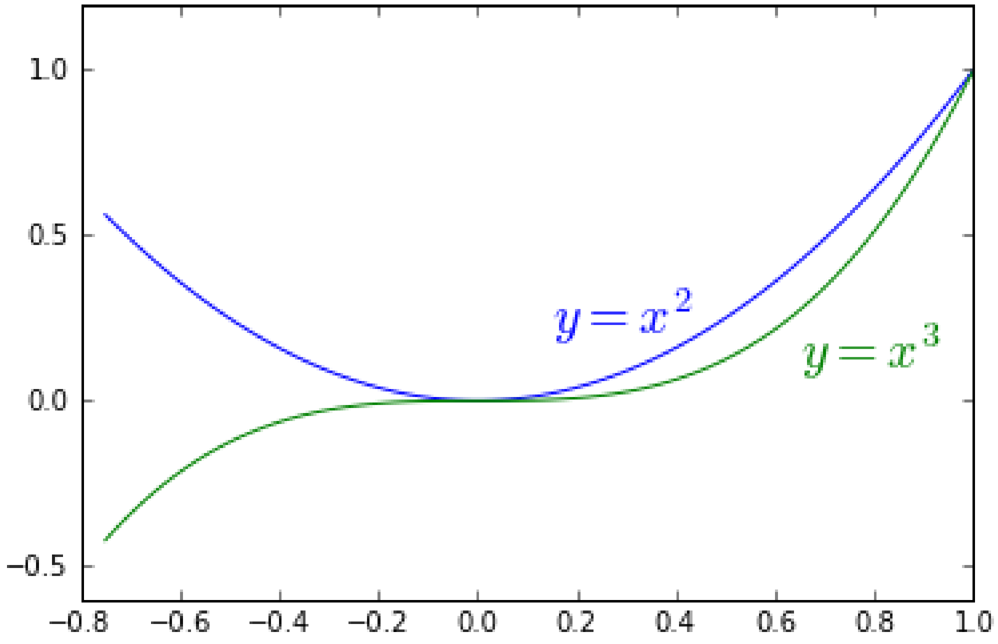
```
ax.fill_between(alpha, alpha_2, alpha_3, color='g')

plt.xlim(0, 5)
plt.ylim(0, 140)
plt.title('Fill between')

plt.show()
```



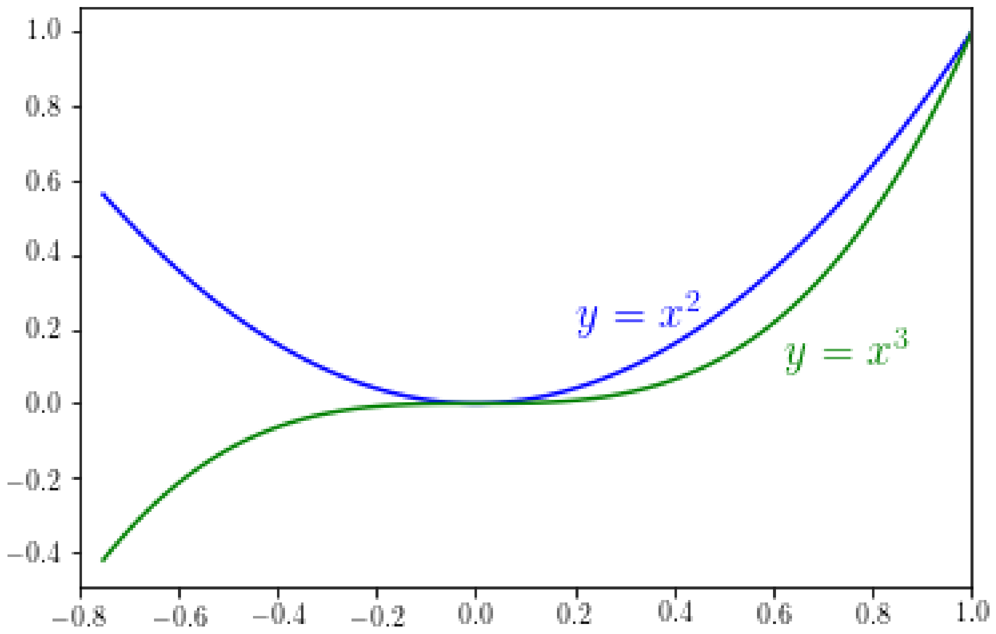
1.5 Task 5

Create a plot with annotations of the curves.

In [12]:



```
In [8]: x = np.linspace(-0.75, 1.0, 100)
        y1 = x ** 2
        y2 = x ** 3

        plt.plot(x, y1, 'b')
        plt.plot(x, y2, 'g')

        plt.xlim(-0.8, 1.0)

        plt.text(0.2, 0.2, r'\LARGE{$y=x^2$}', color='b')
        plt.text(0.62, 0.1, r'\LARGE{$y=x^3$}', color='g')

        plt.show()
```



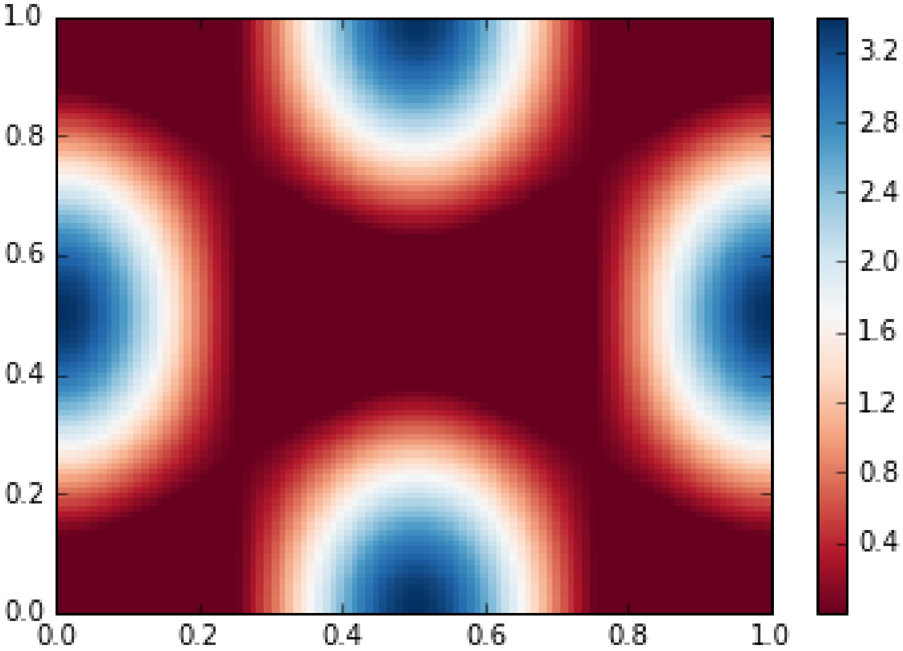
1.6 Task 6

Create a color map using pcolor and colorbar functions for the following X,Y and Z

```
In [9]: alpha = 0.7
        phi_ext = 2 * pi * 0.5

        def flux_qubit_potential(phi_m, phi_p):
            return ( + alpha - 2 * cos(phi_p)*cos(phi_m) -
                alpha * cos(phi_ext - 2*phi_p))

        phi_m = linspace(0, 2*pi, 100)
```

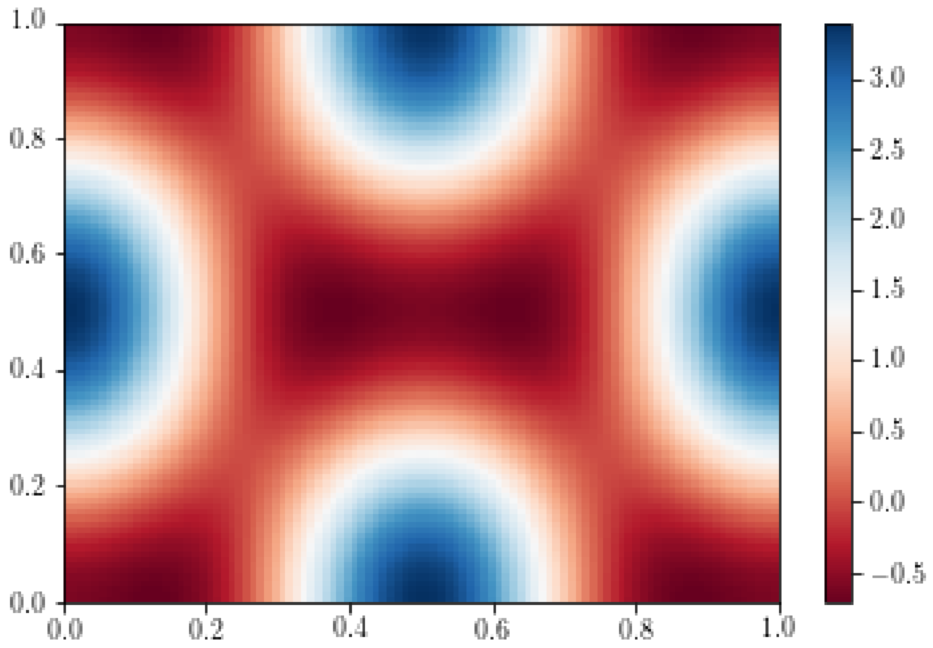
```
phi_p = linspace(0, 2*pi, 100)
X,Y = meshgrid(phi_p, phi_m)
Z = flux_qubit_potential(X, Y).T
```

In [14]:



In [10]:

```
# Ref: http://www.southampton.ac.uk/~fangohr/training/python14/notebooks/Matplotlib.htm
fig, ax = plt.subplots()

p = ax.pcolor(X/(2*np.pi), Y/(2*np.pi), Z, cmap=plt.cm.RdBu)
cb = fig.colorbar(p, ax=ax)
```



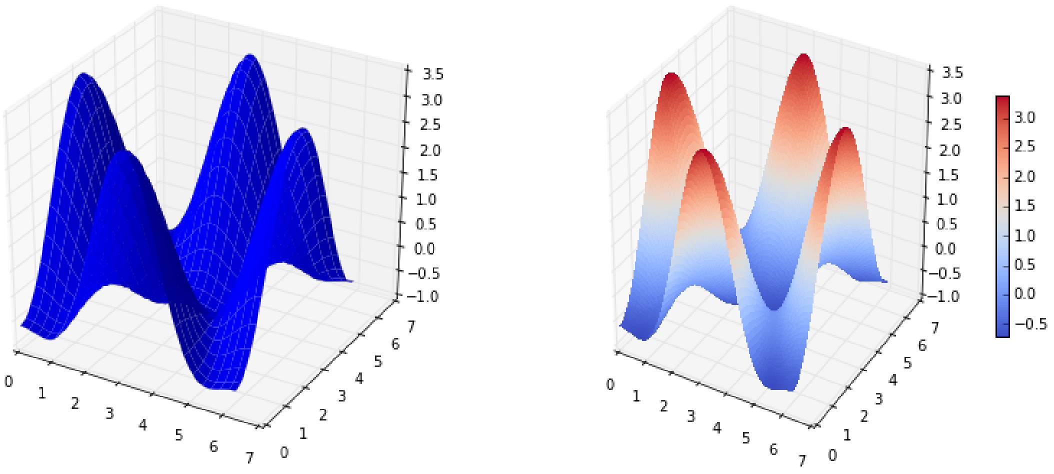
1.7 Task 7

For the same data (i.e. X,Y and Z) create plot_surface, plot_wireframe, contour plot with projections, using

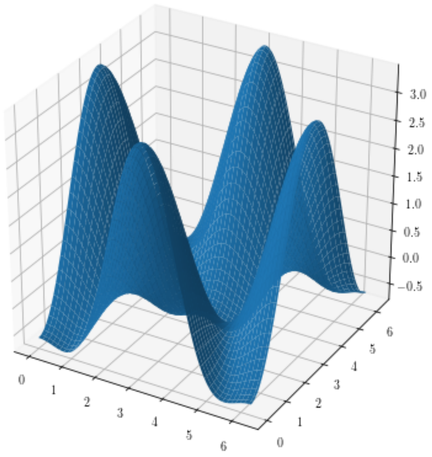
```
In [11]: from mpl_toolkits.mplot3d.axes3d import Axes3D
```

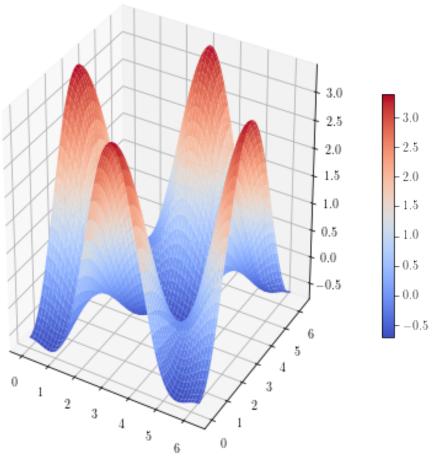
Replicate the plots introduced below (you can use your own data for this)

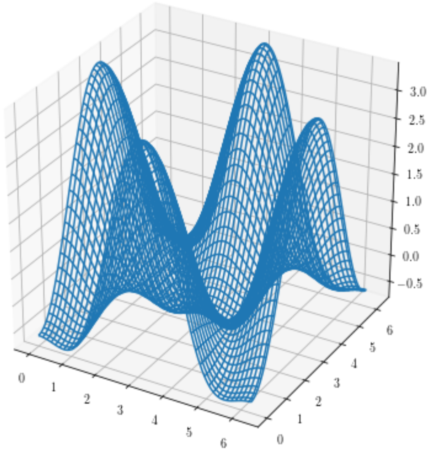
```
In [16]:
```

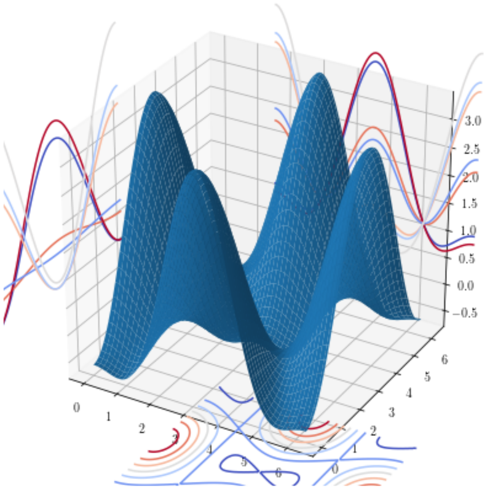


```
In [12]: # Ref: http://www.southampton.ac.uk/~fangohr/training/python14/notebooks/Matplotlib.htm

         fig = plt.figure(figsize=(16,16))

         ax = fig.add_subplot(221, projection='3d')
         p = ax.plot_surface(X, Y, Z)

         ax = fig.add_subplot(222, projection='3d')
         p = ax.plot_surface(X, Y, Z, cmap=plt.cm.coolwarm)

         cb = fig.colorbar(p, shrink=0.5)

         ax = fig.add_subplot(223, projection='3d')
         p = ax.plot_wireframe(X, Y, Z)

         ax = fig.add_subplot(224, projection='3d')
         ax.plot_surface(X, Y, Z)
         cset = ax.contour(X, Y, Z, zdir='z', offset=-pi, cmap=plt.cm.coolwarm)
         cset = ax.contour(X, Y, Z, zdir='x', offset=-pi, cmap=plt.cm.coolwarm)
         cset = ax.contour(X, Y, Z, zdir='y', offset=3*pi, cmap=plt.cm.coolwarm)
```


3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
0
1
2
3
4
5
6
0
1
2
3
4
5
6


3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
0
1
2
3
4
5
6
0
1
2
3
4
5
6
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5


3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
0
1
2
3
4
5
6
0
1
2
3
4
5
6


3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
0
1
2
3
4
5
6
0
1
2
3
4
5
6