Exercise1 Matplotlib

April 14, 2018

1 Hochschule Bonn-Rhein-Sieg

2 Learning and Adaptivity, SS18

3 Assignment 01 (15-April-2018)

3.1 Sathiya Ramesh, Pradheep Krishna Muthukrishnan Padmanabhan, Naresh Kumar Gurulingan

4 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.

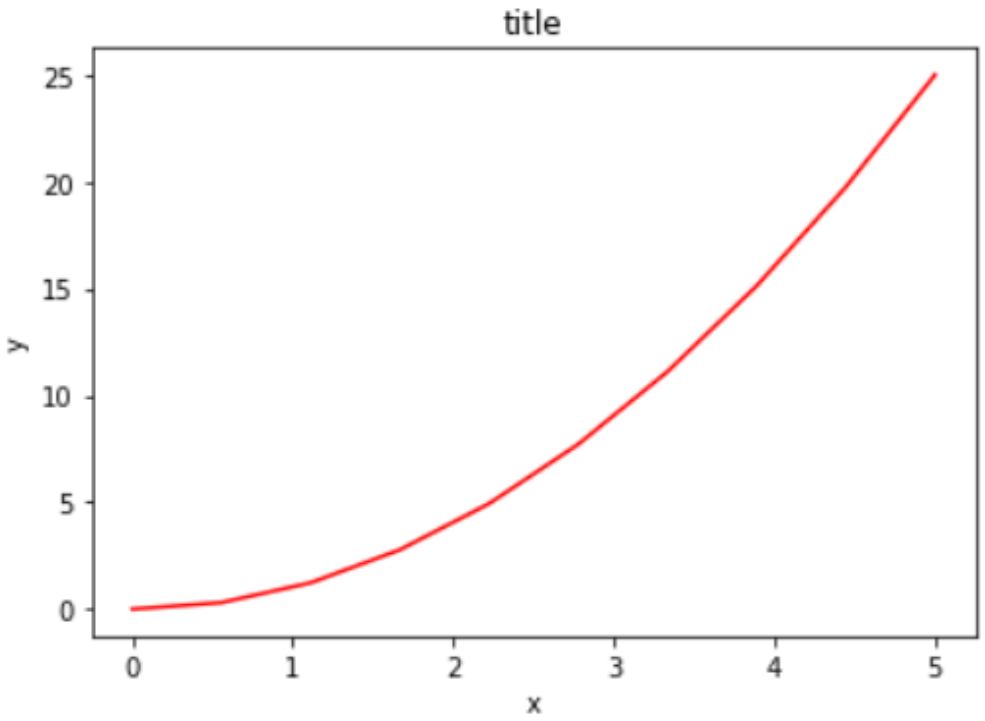
```
In [1]: # needed to display the graphs
        %matplotlib inline
        #from pylab import *
        import numpy as np
        import matplotlib.pyplot as plt
        from IPython.display import Image
        from matplotlib.font_manager import FontProperties
```

4.1 Task 1

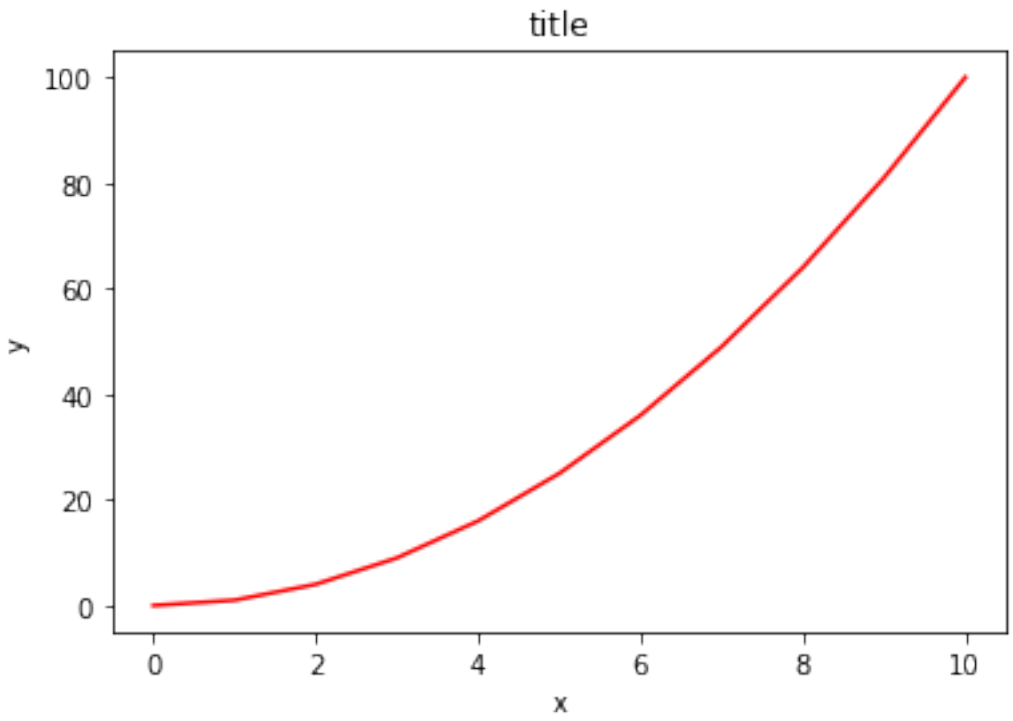
- Create a plot $y = x^2$ for $x \in [1 : 10]$
- Add Title and Axes (Replicate the plot below)

```
In [2]: Image('./images/ex1_task1.png', width= 430, height= 430)
```

```
Out[2]:
```




```
In [3]: def plot_task_1(xlim= None, ylim= None, title= 'title', swap_xy= False, color= 'r'):
            x = np.arange(0, 11, 1)
            y = x**2
            xy = [x, y, 'x', 'y', xlim, ylim]
            if swap_xy:
                xy = [y, x, 'y', 'x', ylim, xlim]
            plt.plot(xy[0], xy[1], color= color)
            plt.xlabel(xy[2])
            plt.ylabel(xy[3])
            plt.xlim(xy[4])
            plt.ylim(xy[5])
            plt.title(title)

        plot_task_1()
        plt.show()
```

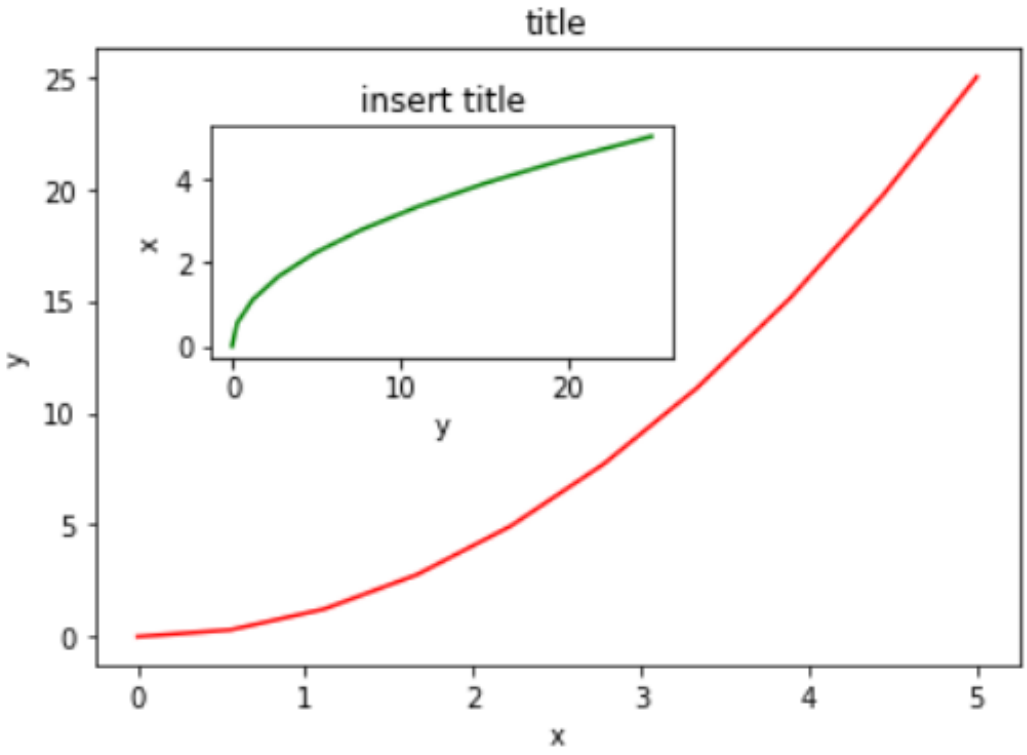


4.2 Task 2

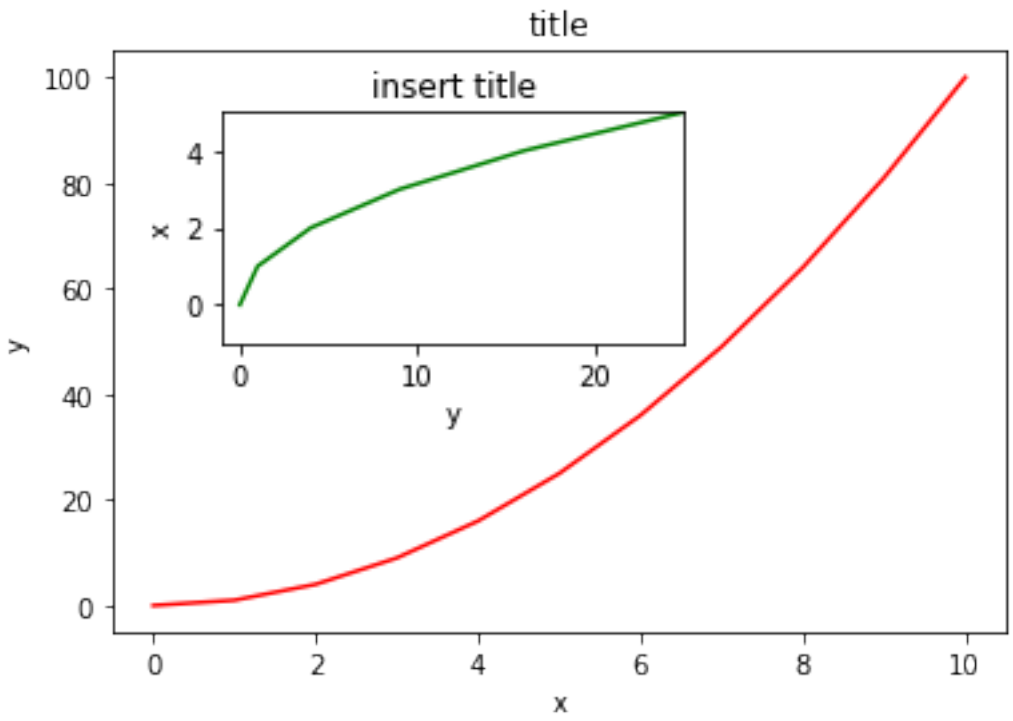
Create two plots: 'main' and 'insert' and place them such that - The 'insert' plot are included into the 'main' plot - The 'insert' is next to the 'main' plot (Replicate the plots below)

```
In [4]: Image('./images/ex1_task2.png', width= 430, height= 430)
```

```
Out[4]:
```




```
In [5]: plot_task_1()
        plt.axes([.22, .5, .4, .3])
        plot_task_1(xlim= (-1, 5), ylim= (-1, 25), title= 'insert title',
                    swap_xy= True, color= 'g')
        plt.show()
```

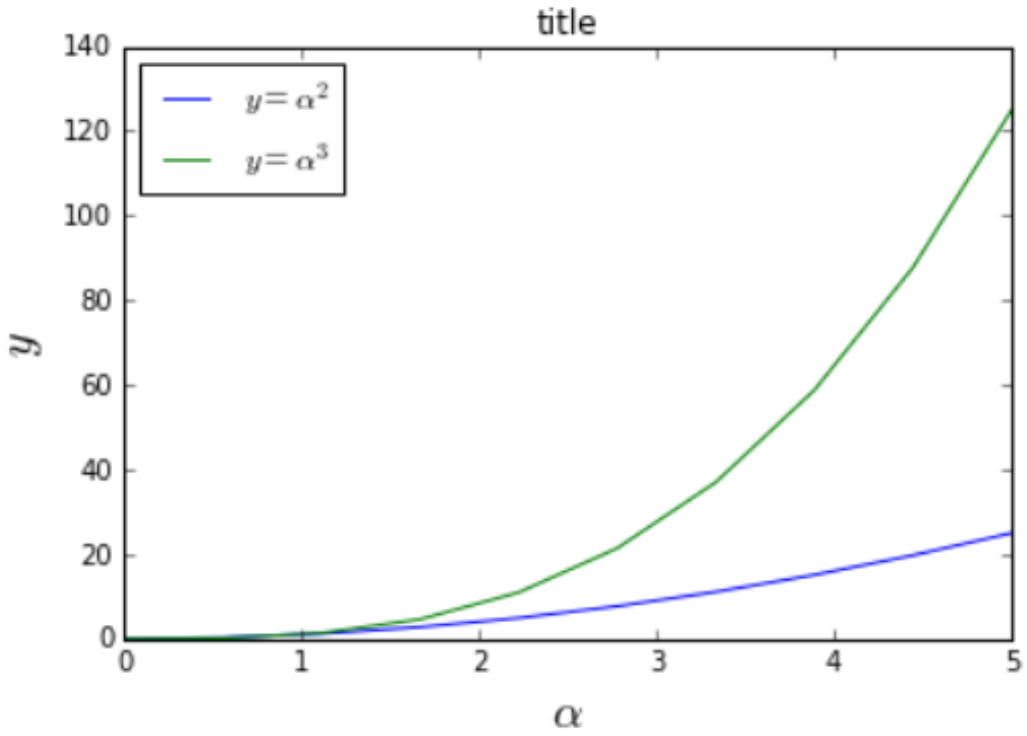


4.3 Task 3

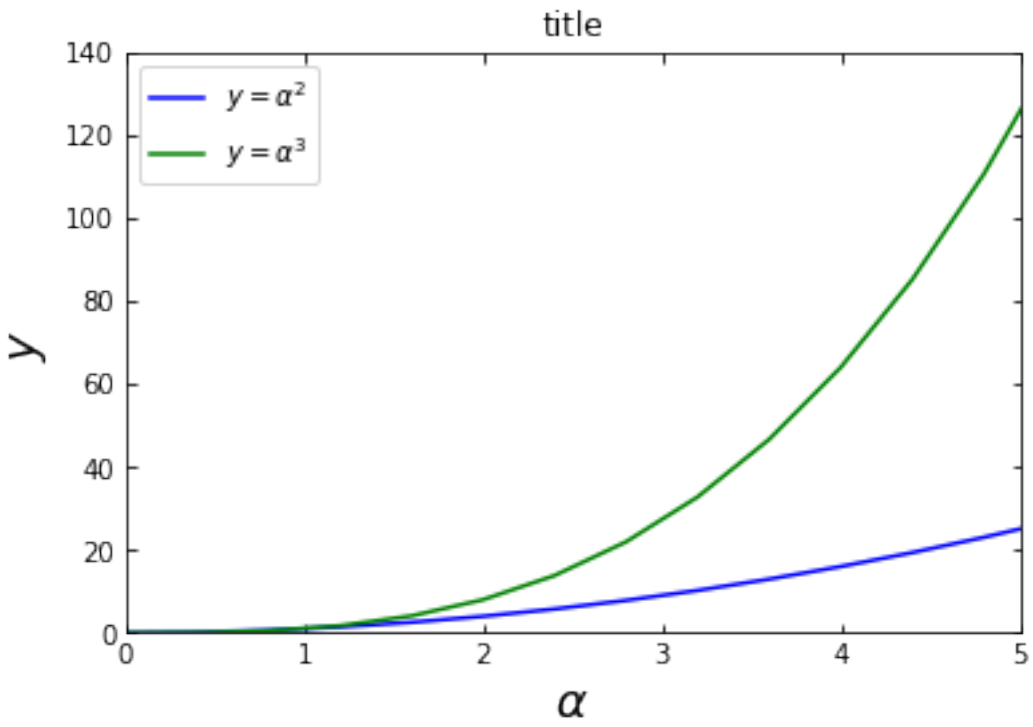
Create a plot with a legend and latex symbols

```
In [6]: Image('./images/ex1_task3.png', width= 430, height= 430)
```

```
Out[6]:
```




```
In [7]: alpha = np.arange(0, 10, 0.4)
        y1 = alpha**2
        y2 = alpha**3
        plt.plot(alpha, y1, color= 'b', label= r'$y=\alpha^2$')
        plt.plot(alpha, y2, color= 'g', label= r'$y=\alpha^3$')
        plt.tick_params(right= 'on', direction= 'in', top= 'on')
        plt.xlim(0,5)
        plt.ylim(0, 140)
        plt.xlabel(r'$\alpha$', fontsize= 18)
        plt.ylabel(r'$y$', fontsize= 18)
        plt.title('title')
        plt.legend()
        plt.show()
```



4.4 Task 4

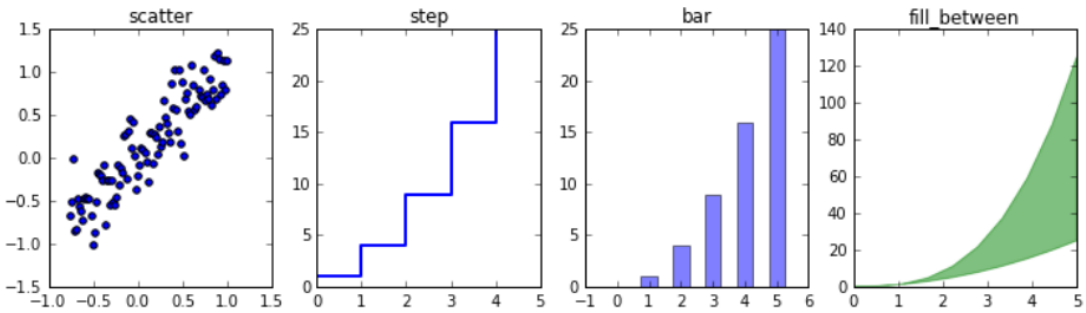
Other plot styles. Given:

```
In [8]: xx = np.linspace(-0.75, 1., 100)
        n = np.array([0,1,2,3,4,5])
```

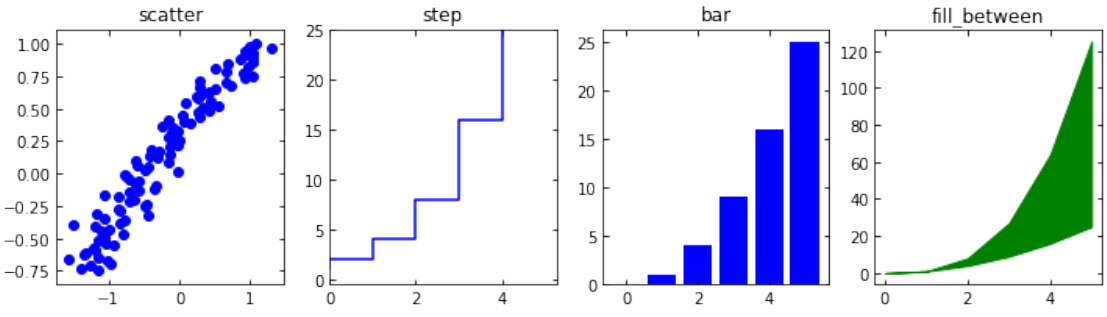
Generate: scatter, step, bar, fill_between

```
In [9]: Image('./images/ex1_task4.png', width= 730, height= 530)
```

Out[9]:



In [10]:
```
np.random.seed(0)
x = np.linspace(-1.5, 1, 100) + np.random.normal(0., 0.2, size= 100)

figure = plt.figure()
figure.set_figwidth(12)
figure.set_figheight(3)
figure.add_subplot(1, 4, 1)
plt.tick_params(right= 'on', direction= 'in', top= 'on')
plt.scatter(x, xx, color= 'b')
plt.title('scatter')
x = np.arange(0, 6)
step_y = [2**i for i in range(0,6)]
figure.add_subplot(1, 4, 2)
plt.tick_params(right= 'on', direction= 'in', top= 'on')
plt.step(x, step_y, color= 'b')
plt.xlim(xmin=0)
plt.ylim(ymax=25)
plt.title('step')
figure.add_subplot(1, 4, 3)
plt.tick_params(right= 'on', direction= 'in', top= 'on')
bar_y = [i**2 for i in x]
plt.bar(x, bar_y, color= 'b')
plt.title('bar')
fill_curve_1 = bar_y
fill_curve_2 = [i**3 for i in x]
figure.add_subplot(1, 4, 4)
plt.tick_params(right= 'on', direction= 'in', top= 'on')
plt.fill_between(x, fill_curve_1, fill_curve_2, color= 'g')
plt.title('fill_between')
plt.show()
```

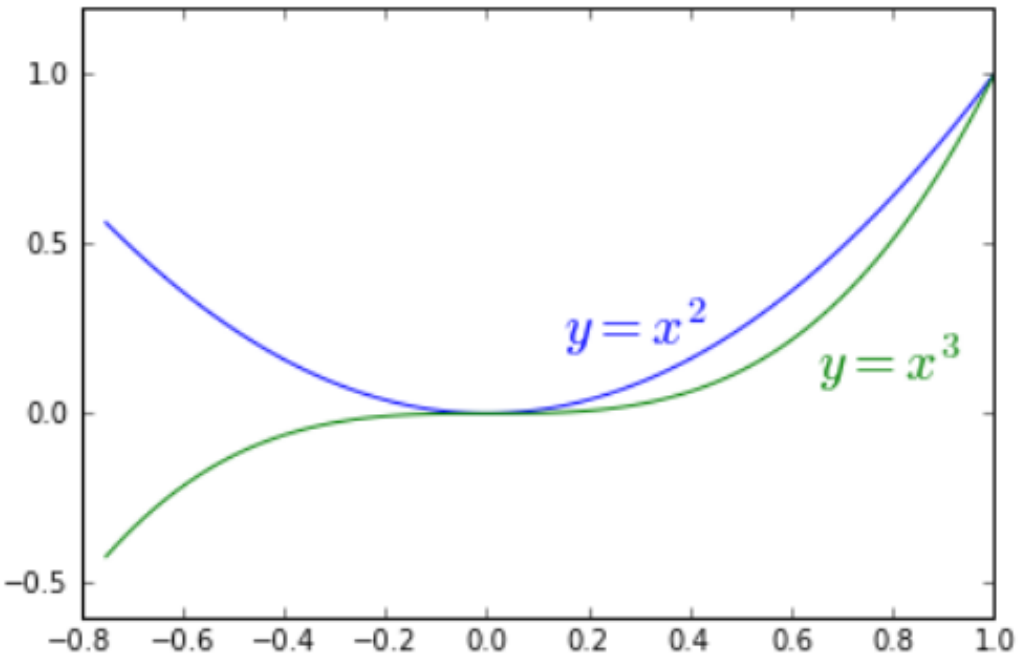


4.5 Task 5

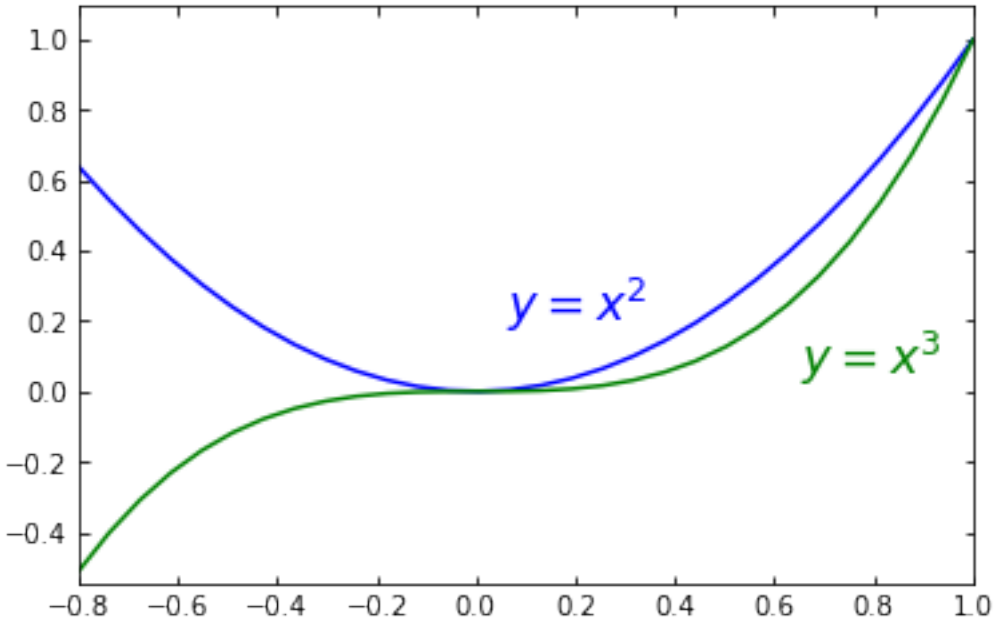
Create a plot with annotations of the curves.

```
In [11]: Image('./images/ex1_task5.png', width= 430, height= 430)
```

Out[11]:



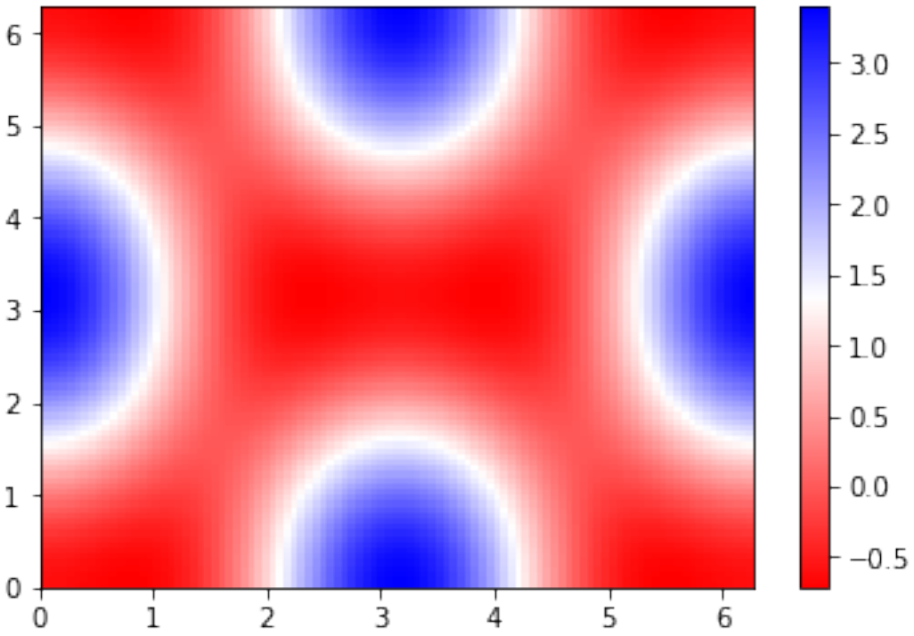

```
In [12]: x = np.linspace(-0.8, 1, 30)

         fig = plt.figure()
         ax = fig.add_subplot(111)
         ax.plot(x, x**2, color= 'b')
         ax.plot(x, x**3, color= 'g')
         ax.tick_params(right= 'on', direction= 'in', top= 'on')
         plt.xlim(-0.8, 1)
         plt.ylim(-0.55, 1.1)

         ax.text(0.06, 0.2, r'$y=x^2$', style='italic', fontsize= 18, color= 'b')
         ax.text(0.65, 0.05, r'$y=x^3$', fontsize= 18, color= 'g', style= 'italic')
         plt.show()
```



4.6 Task 6

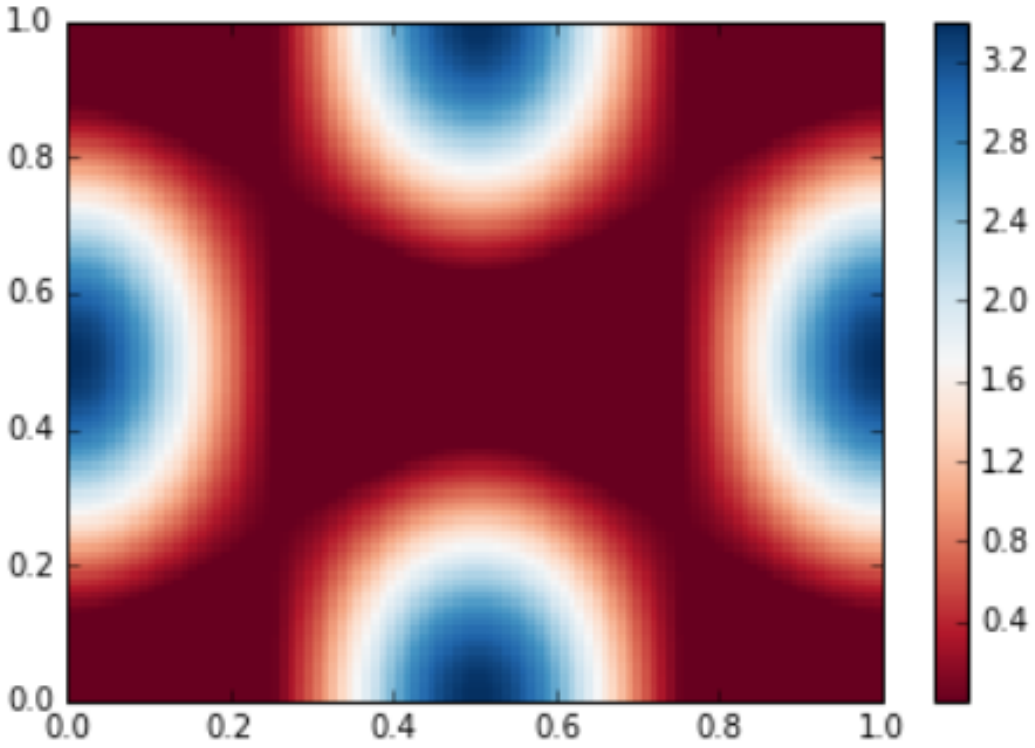
Create a color map using pcolor and colorbar functions for the following X,Y and Z

```
In [13]: alpha = 0.7
         phi_ext = 2 * np.pi * 0.5

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

         phi_m = np.linspace(0, 2*np.pi, 100)
         phi_p = np.linspace(0, 2*np.pi, 100)
         X,Y = np.meshgrid(phi_p, phi_m)
         Z = flux_qubit_potential(X, Y).T
         color_plot = plt.pcolor(X, Y, Z, cmap=plt.cm.bwr_r)
         plt.colorbar(color_plot)
         plt.show()
```



In [14]: Image('./images/ex1_task6.png', width= 430, height= 430)

Out[14]:



4.7 Task 7

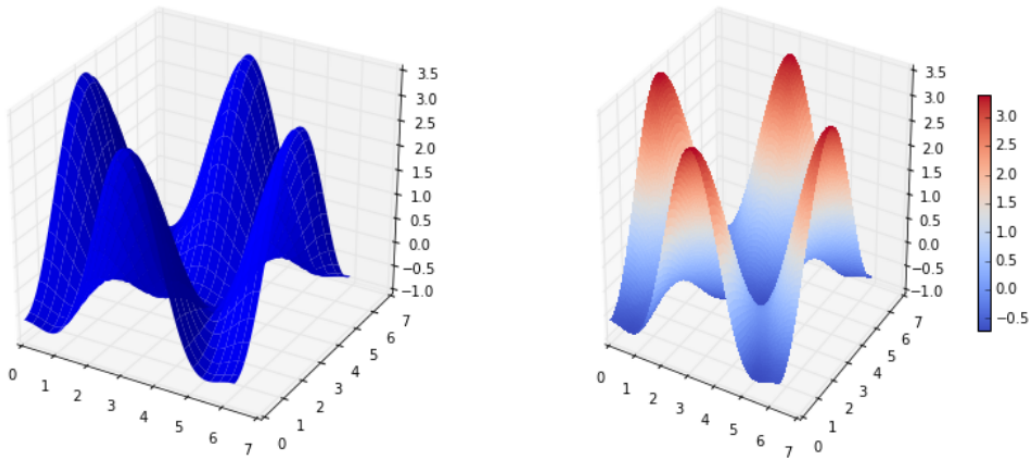
For the same data (i.e. X,Y and Z) create plot_surface, plot_wireframe, contour plot with projections, using

```
In [15]: from mpl_toolkits.mplot3d.axes3d import Axes3D
         from matplotlib.ticker import LinearLocator, FormatStrFormatter
```

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

```
In [16]: Image('./images/ex1_task7.png', width= 730, height= 430)
```

Out[16]:




```
In [17]: fig = plt.figure()
         fig.set_figwidth(13)
         fig.set_figheight(6)
         ax = fig.add_subplot(121, projection='3d')
         fig.set()
         # Plot the surface
         ax.plot_surface(X, Y, Z, color='b')
         ax.set_zlim(-1, 3.5)
         ax.set_xlim(0, 7)
         ax.set_ylim(0, 7)

         ax = fig.add_subplot(122, projection='3d')
         surf = ax.plot_surface(X, Y, Z, cmap=plt.cm.coolwarm,
                                linewidth=0, antialiased=False)
```

```
# Customize the z axis.
ax.set_zlim(-1, 3.5)
ax.set_xlim(0, 7)
ax.set_ylim(0, 7)
ax.zaxis.set_major_locator(LinearLocator(10))
ax.zaxis.set_major_formatter(FormatStrFormatter('%.01f'))

# Adding a color bar which maps values to colors.
fig.colorbar(surf, shrink=0.7, aspect=20)
plt.show()
```