Basic Matlotlib exercises

The following small steps can be done in jupyter notebook
(if you encounter problems with a strange login screen in your browser start the notebook as follows:

```
jupyter notebook  - -NotenbookApp.token='' - -NotebookApp.password=''
```

1. Plot a sine and a cosine in one plot. Take the range 0 to 2 pi.
   Plot also a legend in the plot. Experiment with the arguments.
   plt.legend(frameon=False, loc=5)
   Use different colors using “ro” for read dots “bx” for blue x.
   Use a label otherwise the legend will not be contain much information.
2. Create a subplot using plt.subplot(211)
   create a x vector containing 100 points between 0 and 5
   plot y = x^3 - 4x
   and plot y = exp(-x^2)
   use a log scale by setting:
   ax = plt.gca()
   ax.set_yscale('log')
   Now add the following:
   f = plt.gcf() #get the figure and set size
   f.setsize_inches(6,8)
   plt.tight_layout()
3. Now let's add some error bars to our plots. We start again.
   Use the following as a start:
   N = 40
   #create a vector x of N points between 0 and 100.0
   sigma = 25.0 * np.ones(N) #look up what this is doing!!
   r = np.random.randn(N)
   y = 10.0 + 3.0*x + sigma*r
   plot the result: plt.errorbar(x,y, yerr=sigma, fmt='0')
4. surface plot:
   create a vector theta between -4 pi and 4 pi, 100 points
   and vector z between -2 and 2 of also 100 points
   r = z**2 + 1
   create 2 vector x and y which use the theta and r to get cartesian points instead of the polar representation.
   add:

```
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
```

```
ax = fig.gca('projection = '3d')
ax.plot(x,yz)
```
we should see a nice contour plot.

5. See also the meshgrid exercise