Scipy Solutions

1.

```
from scipy import *
print roots([1, 0, 0, -3, 0, 0, 0, 1, 0, 1])
'''outputs [-0.75107001+1.23681826j -0.75107001-1.23681826j  1.36843381+0.j
 -0.88577070+0.j          1.00000000+0.j         -0.30876102+0.74799231j
 -0.30876102-0.74799231j  0.31849948+0.70729763j  0.31849948-0.70729763j]'''
```

2.

```
from pylab import *
from scipy.optimize import fsolve

def func2(x):
    out = [x[0] * x[1] - 2]
    out.append(log(x[0]) * log(x[1]) + log(2))
    return out

x = fsolve(func2,[4, 0.5])
print x
#outputs [ 3.48470406  0.57393683]
```

3.

```
from pylab import *
from scipy.optimize import leastsq
from numpy import random

def residuals(p, y, t):
    err = y - func(t, p)
    return err

def func(t, p):
    return p[0] * t + p[1]

t = array([1, 2, 3, 4, 5])
y = array([2, 3.5, 6.5, 7.8, 11])

p0 = [2, 0]
plsq = leastsq(residuals, p0, args = (y, t))
plot(t, y, 'bo')
plot(t, func(t, plsq[0]), 'ro-')
show()
```

4.

```
from pylab import *
from scipy.optimize import leastsq
from numpy import random

def residuals(p, y, t):
   err = y - func(t, p)
   return err

def func(t, p):
   return p[0] * cos(p[1] * t + p[2])

t = array([0.0,  0.1,  0.2,  0.3,  0.4,  0.5,  0.6, 0.7,  0.8,  0.9,  1.0 ,
          1.1,  1.2,  1.3,  1.4,  1.5,  1.6,  1.7,  1.8,  1.9,  2.0])
y = array([1.78,  1.43 , 1.12, 0.79,  0.30, -0.14, -0.62, -0.99, -1.30, -1.64 ,
          -1.91, -1.97, -2.00, -1.89, -1.74, -1.44, -1.14, -0.78, -0.30,  0.12,
          0.51])

p0 = [1, 2, pi/3]
plsq = leastsq(residuals, p0, args = (y, t))
plot(t, y, 'bo')
plot(t, func(t, plsq[0]), 'ro-')
show()

print "k = ", plsq[0][1] ** 2
#outputs 'k =  5.00094267586'
```

5.

```
from scipy import std, mean, var, stats
data = stats.norm.rvs(loc=3,size=100)
print "Data sample: "
print data
print "Standard deviation:", std(data)
print "Mean:", mean(data)
print "Variance:", var(data)
```

6.

```
from scipy import std, mean, var, stats
data = stats.norm.rvs(loc=0,scale=1,size=10000)
count1 = 0
count2 = 0
count3 = 0
for number in data:
    if abs(number) <= 1:
        count1 += 1
    elif abs(number) <= 2:
        count2 += 1
    elif abs(number) <= 3:
        count3 += 1
print count1/100, "% of the data fall within 1 standard deviation of the mean."
print (count1 + count2)/100, "% of the data fall within 2 standard deviations of the mean."
print (count1 + count2 + count3)/100, "% of the data fall within 3 standard deviations of the mean."
```

7.

```
t_array = [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
x_array = [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55]
outfile = open('C:\\data.txt', 'w')
outfile.write('%10s %10s' % ('time', 'position'))

for i in range(0, len(t_array) - 1):
    outfile.write('\n' + '%10.1f %10d' % (t_array[i], x_array[i]))

 outfile.close()
data_array = load('C:\\data.txt', skiprows=1)
times = data_array[:, 0]
positions = data_array[:, 1]
plot(times, positions)
show()
```