Problem Set 2 Solutions

Problem 1(a)

$$Q(t) = \int_0^t P_0 e^{rt} dt = \frac{P_0}{r}(e^{rt} - 1)$$

$$t = \frac{\ln(1 + \frac{rQ}{P_0})}{r}$$

$$\frac{\partial t}{\partial r} = \frac{1}{1 + \frac{rQ}{P_0}} \frac{Q}{P_0} \frac{1}{r} - \left(\ln(1 + \frac{rQ}{P_o}) \right) \frac{1}{r^2}$$

$$\frac{\partial t}{\partial Q} = \frac{Q}{rP_0 + r^2 Q} - \left(\ln(1 + \frac{rQ}{P_o}) \right) \frac{1}{r^2}$$

$$\frac{\partial t}{\partial Q} = \frac{1}{r} \left(\frac{1}{1 + \frac{rQ}{P_0}} \frac{r}{P_0} \right)$$

$$\frac{\partial t}{\partial Q} = \frac{1}{P_0 + rQ}$$

PoE

$$\sigma^2 = \left[\left(\frac{\partial t}{\partial r} \sigma_r \right)^2 + \left(\frac{\partial t}{\partial Q} \sigma_Q \right)^2 \right]^{1/2}$$

With: *r=0.027*, *Q=1000*, *P=5*,

$$\frac{\partial t}{\partial r} = -1388.95$$

$$\frac{\partial t}{\partial Q} = 0.03125$$

With $\sigma_r = 0.002$, $\sigma_Q = (0.1)(1000) = 100$, the PoE becomes:

$$\sigma^2 = \left[\left((-1388.95)(0.002) \right)^2 + \left((0.03125)(100) \right)^2 \right]^{1/2}$$

$$\sigma^2 = \left[2.778^2 + 3.125^2 \right]^{1/2}$$

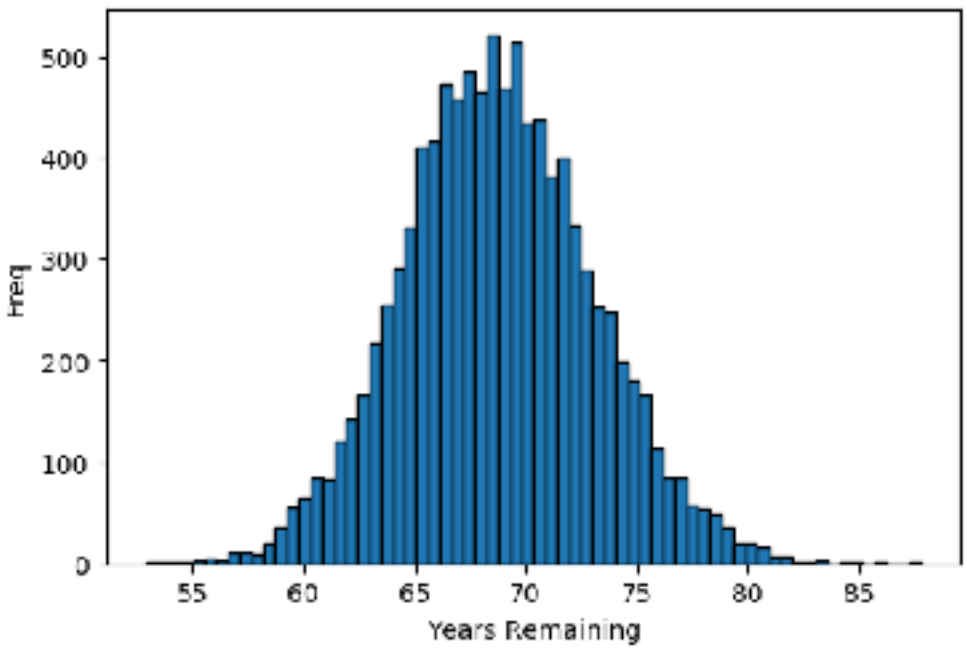
$$\sigma = 4.181 \text{ years}$$

Problem 1(b)

```
def coalt(Q,r,P0):
   return (np.log((Q*r/P0)+1))/r
def MC(Q,r,P0,EQ,Er,Num):
    t=[]
    for sample in range(Num):
        myQ=ss.norm.rvs(Q,EQ)
        myr=ss.norm.rvs(r,Er)
        t.append(coalt(myQ,myr,P0))
    mean=np.mean(t)
    sigma=np.std(t, ddof=1) #sample st.dev
    print "stats:", mean, sigma
    return t

Timeleft=MC(1000,0.027,5, 100, 2e-3,10000)
plt.figure()
plt.hist(Timeleft, bins='auto', edgecolor='black')
plt.xlabel('Years Remaining')
plt.ylabel('Freq')
```

Output




```
>>stats: 68.7424339151 4.26931437762
```

The lower and upper limits at the 95% CL (with z=1.96) is = 68.7 ± (1.96)4.2 = {60.5, 76.9} years .

Problem 2

```
import numpy as np
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt

Tc=[440.6, 440.3, 439.7, 438.2, 437.3, 434.4, 431.7, 429.7]
phi=[4.5, 3.4, 3.2, 2.7, 2.1, 1.0, 0.8, 0.5]

kb= 8.6173303E-5 #Boltzmann constant in eV

def genTemp(temps, err): # generate normally distributed crystallisation
temperature data for each heating rate
    randT=[]
    for t in temps:
        randT.append(np.random.normal(t, err))
    return randT


def genPhi(rates, err): #generate normally distributed heating rate data
for each heating rate
    randR=[]
    for p in phi:
        randR.append(np.random.normal(p, 0.1))
    return randR

def line(x,m,c):  #define a linear funcion to fit the data
    return m*x+c

def GenE_a(Tx, rate, Terr, raterr): #generate the activation energy
    newT=genTemp(Tx,Terr) #calculate the random temperatures
    newrates=genPhi(rate, raterr) #calculate the random heating rates
    recipT=[]
    for i in newT:
        recipT.append(1.0/i) #prepare the reciprocal temperature data (x-
axis)

    lograteT2=[]
    for i in range(len(rate)):
        #print newrates[i], newT[i]
        lograteT2.append( np.log(newrates[i]/(newT[i]**2))) #calculate
the y-axis data, which is the ln(phi/T^2)

    params=curve_fit(line, recipT, lograteT2)[0] #fit a line to
theeciprocal temperature vs  ln(phi/T^2) data
    return -params[0]*kb #return the gradient of the line multiplied by
boltzmanns constant, this is the activation energy

def MC(N):
    Ea=[]
    for i in range(N):
        Ea.append(GenE_a(Tc, phi, 0.2,0.2))
    return Ea
```

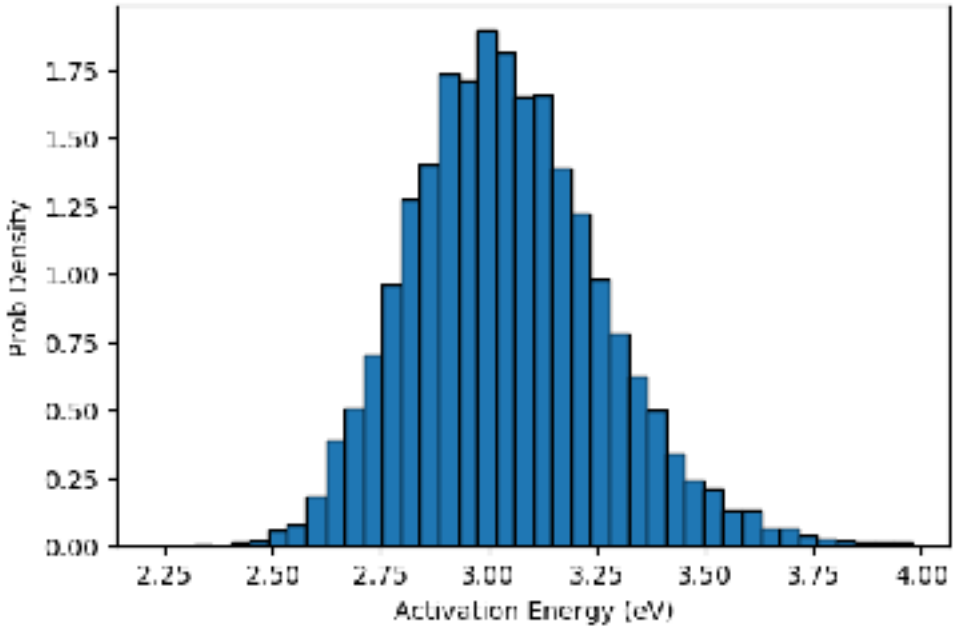
```
data=MC(10000)

mean=np.mean(data)
sigma=np.std(data, ddof=1) #sample st.dev
print "stats:", mean, sigma

plt.figure()
plt.hist(data, bins=40, align='left',  range=(2.25,4),
normed=1,edgecolor='black')
plt.xlabel('Activation Energy (eV)')
plt.ylabel('Prob Density')

>>stats: 3.06533834076 0.226083967706
```

Problem 2(b)



Problem 2(c)

The standard deviation is $\sigma = 0.226$

Problem 2(d)

The standard deviation is $\sigma_{SE} = \dfrac{0.226}{\sqrt{10000}} = 0.00226$ eV

Problem 2(e)

The 95% (z=1.96) confidence limits within which we know the mean = 3.065 ± (1.96)0.00226 = {3.06057, 3.06942} eV.

Note, these limits are purely a measure of how well the simulation predicts the true mean. They are not the limits that should be quoted as measurement error. For that we would use the standard deviation because we only used the data from 1 experiment (N=1 and therefore $\sigma_{SE} = \dfrac{\sigma}{\sqrt{1}} = \sigma$). The measurement limits are, therefore, 3.065 ± (1.96)0.226 = {2.62, 3.52} eV.