1.

from 10 days log return we can calculate the sigma, which is equal to 0.010485

```
discount = 98.45/100
x = 100
s = 104.36
t = 90
r = -np.log(discount)/90
def bs2(K, S, sigma, r, T, q = 0):
    d1 = (np.log(S/K)+(r-q+sigma**2/2)*T)/(sigma*T**0.5)
    d2 = d1- sigma*T**0.5
    return(norm.cdf(d1)*S-norm.cdf(d2)*K*np.exp(-r*T))
bs2(x , s, sigma,r,t)
```

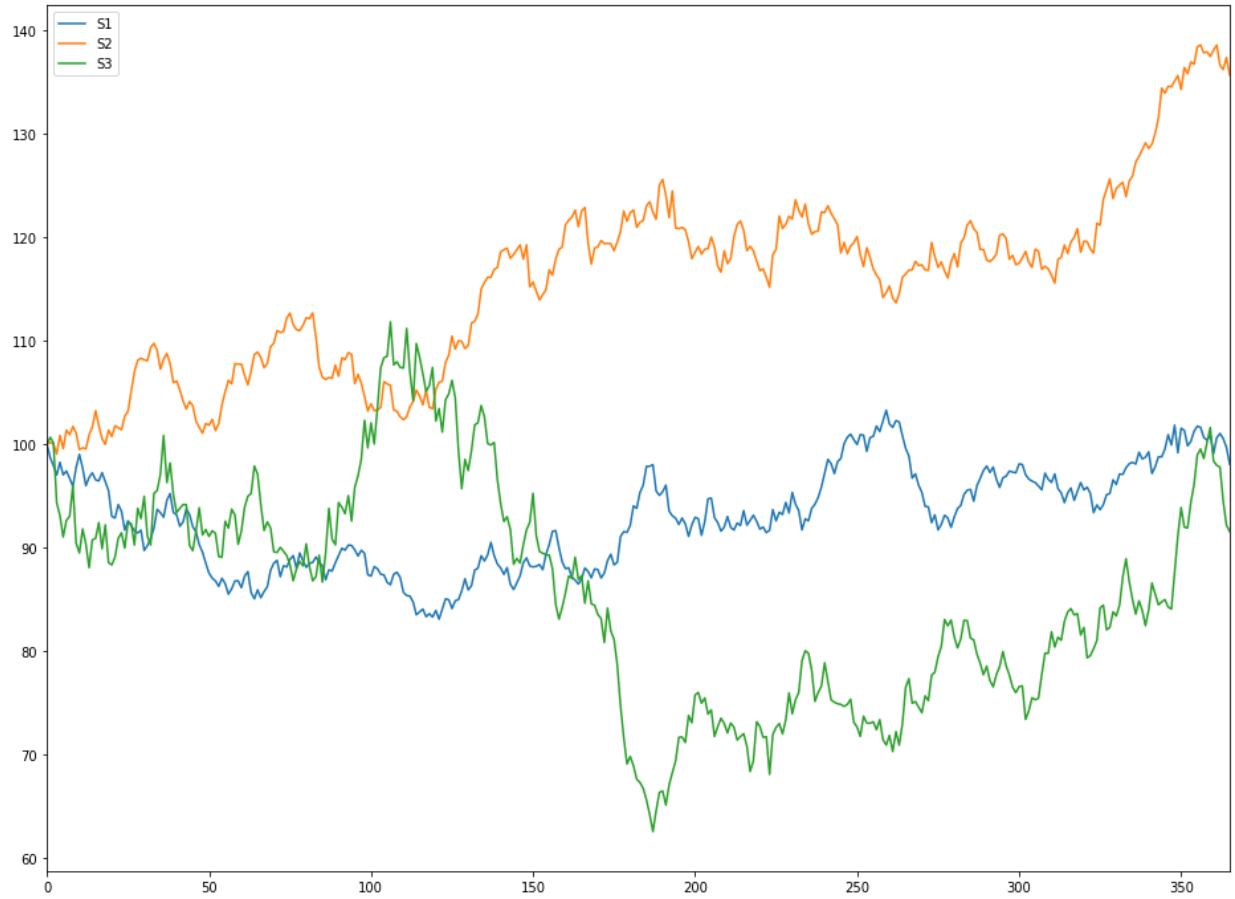
$$C = S \times N(d) - e^{-rt} X \times N(d - \sigma\sqrt{t})$$

Then we input variables into the BS formula, calculate to get the price, which is equal to 7.6484

2.

use MC simulation we get the difference based on their sigma and mean.
and w is the Geometric Brownian motion.
I choose to use the python to do it since i am more familiar with it.

```
path1 = [100]
mean = 0.12
sigma = 0.2
dt = 1/365
random = np.random.standard_normal(365)
for i in range(len(random)):
    path1.append(path1[i] + path1[i]*( mean*dt  + sigma*np.sqrt(dt )*rand
om[i]))
```



Clearly the s1 and s2 are less violate than s3 and s2 is above s1 since it has lager mean.

3.

we could use two month option to derive the implied volatility and use the implied volatility and BS formula to calculate the price for the one month option

```
from scipy.optimize import fsolve
discount = 98.972 /100
x = 105
s = 98.25
t = 60
r = -np.log(discount)/60

def bs2(sigma):
    d1 = (np.log(s/x)+(r+sigma**2/2)*t)/(sigma*t**0.5)
    d2 = d1- sigma*t**0.5
    result = norm.cdf(d1)*s-norm.cdf(d2)*x*np.exp(-r*t) - 2.362
    return result
fsolve(bs2 , 0.001)
```

the sigma equal to 0.01492

And use BS Formula we calculate the first month price equal to 2.6372649