**Task 1 -- Simulating SDE**

#### Simulating geometric Brownian motion

What is the expectation value of S(3)?

What is the variance of S(3)?

**From the GBM given , we can use the theoretical expectation** \mathbb{E}(S_t)= S_0e^{\mu  t}, **and theoretical variance** \operatorname{Var}(S_t)= S_0^2e^{2\mu t} \left( e^{\sigma^2 t}-1\right),**to compute the value.**

**For detail in scripting, please refer to gbm.py under #theoretical expectation and variance**

**The answers :**

**Theoretical Expectation = 52.6444934955**

**Theoretical Variance = 623.09647233**

* Plot only 5 realizations of the GBM with proper labels.

In order to draw 5 independent random path with 1000 runs

#Create Brownian paths

t=p.linspace(0,period,n+1);

dB=p.randn(n\_path,n+1)/p.sqrt(n/period);

dB[:,0]=0;

B=dB.cumsum(axis=1);

#Calculate stock prices

nu=mu-sigma\*sigma/2.0

S=p.zeros\_like(B);

S[:,0]=S0

S[:,1:]=S0\*p.exp(nu\*t[1:]+sigma\*B[:,1:])

#Plotting only 5 realizations of the GBM

S\_plot=S[0:5]

label = 'Time , $t$' ; p.xlabel(label)

label = 'Stock prices, $S$' ; p.ylabel(label)

para1 = '\n with $\mu$ = ' + str(mu)

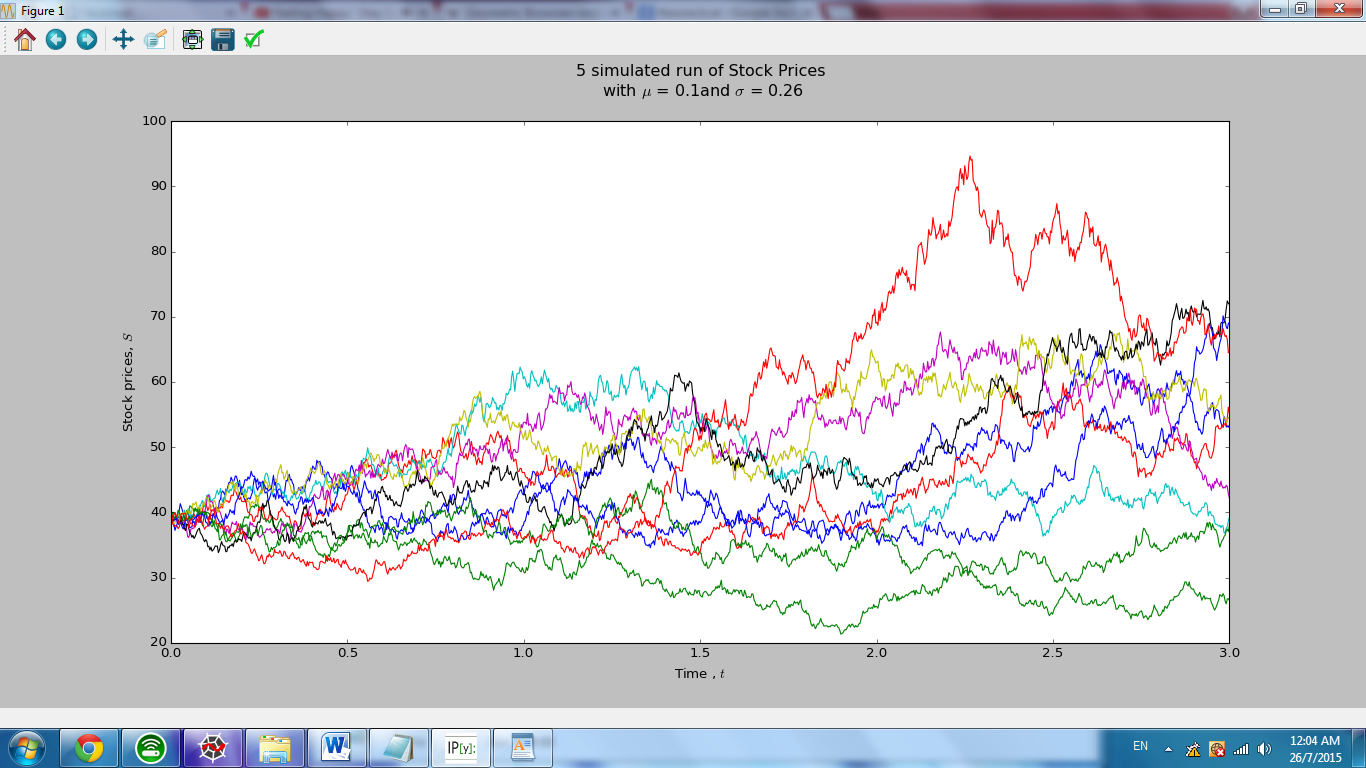
para2 = 'and $\sigma$ = ' + str(sigma) + '\n'

p.title('5 simulated run of Stock Prices' + para1 + para2)

p.plot(t,S\_plot.transpose());

p.show();

**The result :**



* Calculate the expectation value of S(3) based on the simulation.
* Calculate the variance of S(3).

In order to calculate the expectation and the variance of stock S at time 3,i.e. S(3) , we generate an array of stock S(3) and extract the Stock price at the last column by using S[:,-1]. Then we apply np.mean and np.var of (S3) to get the expectation and variance respectively.

#Calculate the expectation value of S(3) based on the simulation

S3=np.array(S[:,-1]) #generate array and extract the stock price at last column

E\_S3=np.mean(S3)

print('E(S3) = ' + str(E\_S3))

#Calculate the variance of S(3)

Var\_S3=np.var(S3)

print('Var(S3) = ' + str(Var\_S3))

**The answers :**

**E(S3) = 52.5518293598**

**Var(S3) = 684.231893023**

* Calculate P[S(3)> 39].

We use a Boolean command called “mask” to take the position of the values that are more than 39 from the array of S(3) , then apply the probably formula to get P[S(3)> 39]

#Calculate P[S(3)> 39]

mask = S3 > 39

non\_zero=mask\*S3

num=sum(non\_zero)

den=sum(S3)+0.0

P\_S3\_more\_than\_39 = num/denprint('P(S3 > 39) = ' + str(P\_S3\_more\_than\_39))

**The answers :**

**P(S3 > 39) = 0.811653**

* Calculate E[S(3) | S(3) > 39].

We use a Boolean command called “mask” to take the position of the values that are more than 39 from the array of S(3) , then by multiplying the selected S3 array to “mask” ( Boolean) to get the values of S >39. Then apply the conditional formula to get E[S(3)|S(3)> 39] ,by setting the sum(mask) as denominator to sum(S3\_more\_than\_39)

#Calculate E[S(3) | S(3) > 39]

mask2 = S3 > 39 #number of values more than 39

S3\_more\_than\_39 = S3 \* mask #extracting values more than 39

num = sum(S3\_more\_than\_39)

den = sum(mask2)+0.0

E\_S3\_more\_than\_39 = num / den

print('E(S3 | S3 > 39) = ' + str(E\_S3\_more\_than\_39))

**The answers :**

**E(S3 | S3 > 39) = 64.966700798**

#### \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### Simulating mean reversal process

Plot only 5 realizations of the mean reversal process with proper labels.

In order to plot the 5 generated stock price , we use the following scripts:

import pylab as p

import numpy as np

#Setup parameters

alpha = 1; theta = 0.064

sigma = 0.27; R0 = 3

time = 1

n\_path = 1000; n\_partition= 1000

#Create Brownian paths

dt = time / n\_partition

t = p.linspace(0,time,n\_partition+1)[:-1];

dB = p.randn(n\_path,n\_partition+1) \* p.sqrt(dt); dB[:,0] = 0;

B = dB.cumsum(axis=1);

# Generating Variable R

R = p.zeros\_like(B)

R[:,0] = R0

for col in range(n\_partition):

R[:,col+1] = R[:,col] + (theta-R[:,col])\*dt + sigma\*R[:,col]\*dB[:,col+1]

#Plotting only 5 realizations of R

R\_plot = R[0:5:,:-1]

p.plot(t,R\_plot.transpose())

label = 'Time , t' ; p.xlabel(label)

label = '$R\_t$' ; p.ylabel(label)

para1 = '\n with $\\alpha$ = 1'

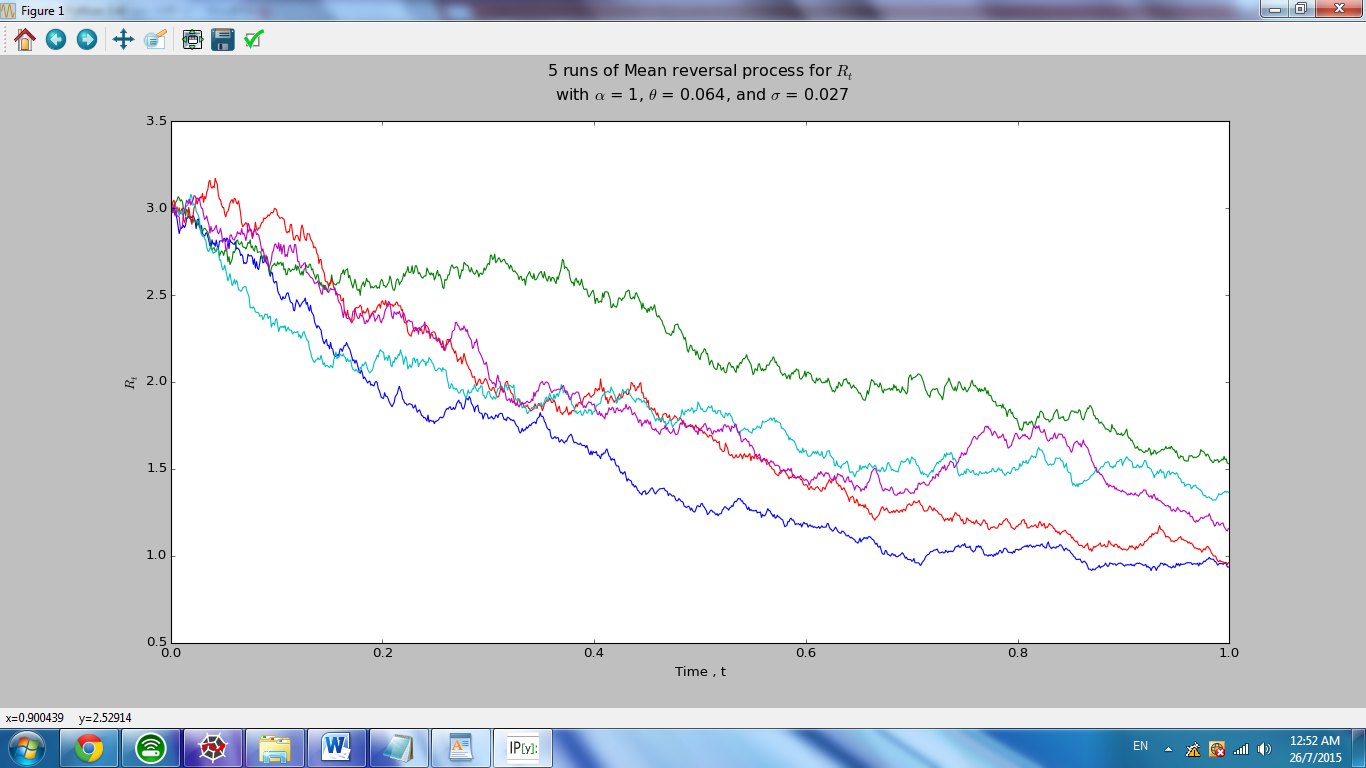
para2 = ', $\\theta$ = 0.064'

para3 = ', and $\sigma$ = 0.027 \n'

p.title('5 runs of Mean reversal process for ' + label + para1 + para2 + para3)

p.show();

**The result :**



Calculate the expectation value of R(1) based on the simulation.

Firstly we generates arrays of R

# Generating Variable R

R = p.zeros\_like(B)

Then we choose over the first column of the array R

R[:,0] = R0

As we know the dRt = Rt+dt -Rt

Then we can interpret as dRt = R[:,col+1] - R[:,col]

for col in range(n\_partition):

R[:,col+1] = R[:,col] + (theta-R[:,col])\*dt + sigma\*R[:,col]\*dB[:,col+1]

By combining dRt = R[:,col+1] - R[:,col] into dR(t) = [0.064 - R(t)] dt + 0.27 R(t) dB(t) ,

We are able to calculate the mean of entire array of R at time 1 using following script

#Calculate the expectation value of R(1) based on the simulation

E\_R1= np.mean(p.array(R[:,-1]))

print('E(R1) = ' + str(E\_R1))

**The answer: E(R1) = 1.15536182056**

Calculate P[R(1)> 2].

We are able to calculate the probability covering entire array of R at time 1 using the following script by setting a command “mask” (Boolean) to extract all the values that fulfil R(1)> 2 then Using Probability formula P\_R1 to get the answer.

#calculation for P[R(1)> 2]

mask = R[:,-1] > 2

P\_R1 = sum(mask)/n\_path

print('P(R1 > 2) = ' + str(P\_R1))

**The answers :**

**P(R1 > 2) = 0.012**

***Task 2 -- Downloading and manipulating stock data***

How many components stocks are there?

**The answer: 30**

Create a table list the following information for all the component stocks: Stock Name, Stock Code, Stock Sector, Weightage in FTSEKLCI, PE Ratio, Net Market Capital.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Stock Name | Code | Sector | Weightage | PE Ratio | Market Capital ( Mil ) |
| 1 | Public Bank Bhd | 1295 | Banks | 11.6 | 15.47 | 72287 |
| 2 | Malayan Banking | 1155 | Banks | 9.32 | 12.87 | 87712 |
| 3 | Tenaga Nasional | 5347 | Alternative Electricity | 9.28 | 9.42 | 7.545 |
| 4 | CIMB Group Holdings | 1023 | Banks | 5.76 | 17.27 | 45251 |
| 5 | Axiata Group Bhd | 6888 | Mobile Telecomunication | 5.62 | 24.58 | 55155 |
| 6 | Sime Darby Bhd | 4197 | Diversified Industrials | 5.51 | 21.02 | 52919 |
| 7 | Digi.com | 6947 | Mobile Telecommunication | 4.16 | 21.5 | 43540 |
| 8 | Genting | 3182 | Hotel | 3.68 | 18.51 | 29967 |
| 9 | Petronas Chemical Group Bhd | 5183 | Commodity Chemicals | 3.55 | 21.78 | 50560 |
| 10 | Maxis Bhd | 6012 | Mobile Telecommunication | 3.45 | 29.09 | 47831 |
| 11 | Petronas Gas | 6033 | Exploration &Production | 3.4 | 22.43 | 42068 |
| 12 | IHH Healthcare | 5225 | Health Care Providers | 3.28 | 60.65 | 46503 |
| 13 | IOI | 1961 | Farming & Fishing | 2.99 | 61.91 | 25741 |
| 14 | Telekom Malaysia | 4863 | Fixed Line Telecommunication | 2.96 | 32.43 | 24325 |
| 15 | Genting Malaysia Bhd | 4715 | Hotel | 2.5 | 18.72 | 30302 |
| 16 | MISC | 3816 | Marine Transportation | 2.45 | 16.22 | 35755 |
| 17 | AMMB Holdings | 1015 | Banks | 2.38 | 9.47 | 18176 |
| 18 | Kuala Lumpur Kepong | 2445 | Farming & Fishing | 2.28 | 27.75 | 22790 |
| 19 | SapuraKencana Petroleum | 5218 | Oil Equipment & Services | 1.98 | 12.35 | 14621 |
| 20 | PPB Groups | 4065 | Food Products | 1.8 | 17.88 | 17972 |
| 21 | British American Tobacco | 4162 | Tobacco | 1.7 | 19.32 | 17703 |
| 22 | Hong Leong Bank | 5819 | Banks | 1.67 | 11.18 | 24104 |
| 23 | YTL Cotp | 4677 | Multiutilities | 1.63 | 14.55 | 16149 |
| 24 | UMW Holdings | 4588 | Automobiles | 1.37 | 20.17 | 11846 |
| 25 | Astro Malaysia Holdings | 6399 | Broadcasting & Entertainment | 1.22 | 28.64 | 16021 |
| 26 | Petronas Dagangan Bhd | 5681 | Integrated Oil & Gas | 1.21 | 37.02 | 20445 |
| 27 | RHB Capital | 1066 | Banks | 1.06 | 9.24 | 19077 |
| 28 | Westports Holdings | 5246 | Transportation Services | 0.93 | 27.69 | 14493 |
| 29 | Hong Leong Financial | 1082 | Banks | 0.64 | 9.79 | 15960 |
| 30 | KLCC Prop&Reits Stapled Sec | 5235 | Real Estate Holding & Development | 0.63 | 13.59 | 12673 |

# Plot a 5-day moving average plot for the downloaded data. Explain how you calculate the 5-day moving average.

# 

from pandas.io.data import DataReader as DR

from datetime import datetime as dt

import datetime

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

import pylab as pl

%Download data of tenaga nasional

start = dt(2012,1,1)

end = dt(2015,1,1)

tenaga = DR("5347.KL",'yahoo',start,end)

klse = DR("^KLSE",'yahoo',start,end)

%Draw tenaga nasional stock chart

fig=plt.figure()

fig.patch.set\_facecolor('white')

ax1=fig.add\_subplot()

tenaga['Close'].plot(ax=ax1,color='b',lw=1,label='TNB Stock Price')

ax2=fig.add\_subplot()

%Find moving average and draw the moving average graph using the panda command “rolling mean”

pd.rolling\_mean(tenaga['Close'],5).plot(ax=ax2,color='r',lw=1,label='5DaysMA')

plt.ylabel('Stock Price,RM')

plt.legend(loc='upper left')

**Compute the correlation of your counter X with FTSEKLCI**

%Download data of klse

start = dt(2012,1,1)

end = dt(2015,1,1)

klse = DR("^KLSE",'yahoo',start,end)

%To find the correlation between tenaga nasional and klse

from pandas.io.data import DataReader as DR

from datetime import datetime as dt

start1 = dt(2012,1,1)

end1 = dt(2015,1,1)

start2 = dt(2011,10,27)

end2 = dt(2015,1,1)

tenaga = DR("5347.KL",'yahoo',start1,end1)

klse = DR("^KLSE",'yahoo',start2,end2)

import datetime

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

import pylab as pl

x=tenaga['Close']

y=klse['Close']

z=np.corrcoef(x,y)

print('Correlation bwtween KLSE and TNB ' + str(z))

**The answer: 0.87540651**

**\*\*Side note : due to some of the trading date is lacking in TNB, so I have to adjust the duration of KLSE to an earlier starting line,but both TNB and KLSE are both intact with 3 years data (ends at 1-1-2015)**