**Report**

Task 1: Simulating SDE

1. Simulating geometric Brownian motion

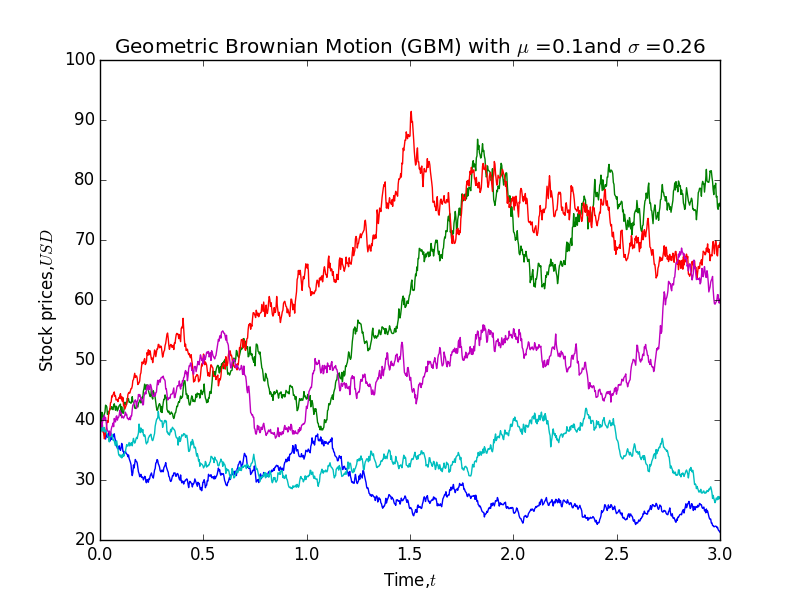
(i)Procedure: Plot 5 realizations of GBM

Before writing the codes, we import pylab and numpy packages which are a collection of useful mathematics functions. After that, parameters are defined and setup. To calculate the theoretical expectation and variance for S(3), formulas of E[S(t)] = S(0)\*exp{(α – δ)\*t} and Var[S(t)] = ( )\* (exp{(α – δ)\*(2t)})\* [(exp{t\*σ\* σ}) -1]. In the question, we are given S(0) =39, t=3.0, (α – δ) = 0.1 and σ = 0.26, by substitute the value into the formula, we get E[S(3)] = 52.6445 and Var[S(3)] = 623.0965.

Now, we are asked to plot the 5 realizations of the geometric Brownian motion (GBM) for 0 < t < 3. We set the partition [0,3] into 1000 equally spaced intervals by using the p.linspace function. Then, we used p.rand to create 1000-by1001 matrix of normally distributed random numbers. Next, we set the first column of the matrix dB = 0. After that, we cumulative sum all the elements in the same row by using function cumsum(axis=1). We created a zero matrix with same size as variable B and defined variable, nu = μ-(/2) and we set the first column of all row as S(0) = 39. Then we simulate 1000 runs of the GBM with the formula, S(0) \* exp{(nu\*t) + sigma\*Bt.

To plot the 5 realizations of the GBM, we pick the first 5 rows among 1000 rows by using S[0:5]. Then we plot the graph by using p.plot function followed by labeled the title, x-axis and y-axis by using the function p.title, p.xlabel and p.ylabel. We showed the graph by using p.show().

(ii) Result: Graph



(iii) Calculation on expectation value of S(3), E[S(3)] and variance of S(3), Var[S(3)] based on the simulation

First, we find the values of S(3) for each path by using p.array(S[:,-1]). Then, we will use np.mean(end\_price) and np.var(end\_price) to find the expectation and variance of S(3). We showed the result by using function print.

(iv) Calculation on P[S(3)>39]

We calculate the number of value that is exceeded 39 and use the function sum to find the total number of values that are exceeded 39. Then, we assign variable, count be the total number of values that are exceeded 39. Finally, we divide the total number of values that are exceeded 39 with number of path, (count/number of path) to get the probability.

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

To calculate this expectation, we first find the total value that is exceeded 39 by using function sum and divided by the number of value that are exceeded 39. This is equivalent to find the mean value for the values that are exceeded 39.

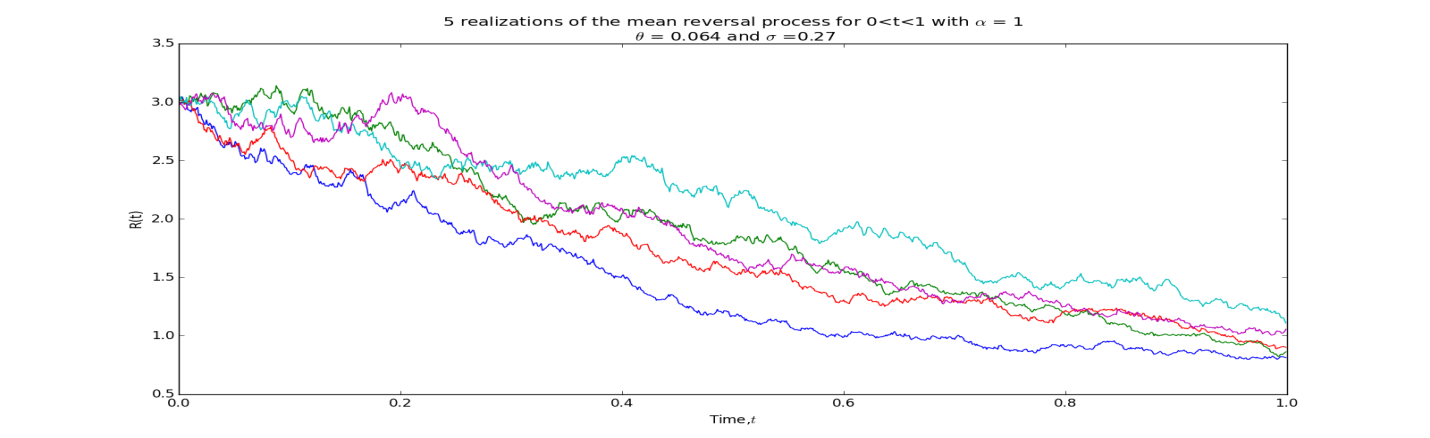
2. Simulating mean reversal process

(i) Procedure: Plot 5 realizations of the mean reversal process

We import pylab and numpy packages before we write the codes. Then, we defined all the parameters. After that, we create the Brownian motion paths. We set the partition [0,1] into 1000 equally spaced intervals by using function p.linspace. Next, we used function p.rand to create 1000-by1001 matrix of normally distributed random numbers followed by set the first column of variable, dB = 0. We used function cumsum(axis=1) to find the cumulative sum of all elements in the same row and set it as the variable, B.

After that, we create a zero matrix, r which is same size with variable, B. We set the first column of the matrix r as 3. Then we run the simulation by using for loop with the equation given. After the simulation, we pick the 5 realizations and plot the graph by using function p.plot followed by labeled the title, x-axis, and y-axis by using functions p.title, p.xlabel, and p.ylabel. We show the graph by using function p.show.

(ii) Result: Graph of 5 realizations of the mean reversal process



(ii) Calculation on expectation value of R(1) based on the simulation

First, we find the values of R(1) by using r[:,-1] and assign it to variable, r\_t1. Then, we find the mean of the R(1) values. To do this, we used function np.mean(r\_t1). Finally, we show the result by using function print.

(iii) Calculate P[R(1)>2]

We calculate the total number of values of R(1) that are exceeded 2 and divide it with number of paths to get the probability of R(1) greater than 2. Finally we used function print to show the result.

Task 2: Downloading and manipulating stock data

1. FTSE Bursa Malaysia KLCI Index

(i)There are 30 components in FTSE Bursa Malaysia KLCI Index.

(ii) The following table summarized information including Stock Name, Stock Code, Stock Sector, Weightage in FTSEKLCI, PE Ratio, and Net Market Capital for all the component stocks.

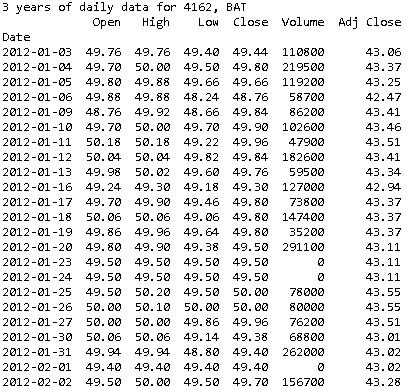
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Stock Name | Stock Code | Stock Sector | Weightage in FTSEKLCI | PE Ratio | Net Market Capital (Billion) |
| 1 | Public Bank Bhd | 1295 | Banks | 11.6 | 15.24 | **73.84** |
| 2 | Malayan Banking | 1155 | Banks | 9.32 | 12.37 | **87.66** |
| 3 | Tenaga Nasional | 5347 | Alternative Electricity | 9.28 | 9.17 | 69.42 |
| 4 | CIMB Group Holdings | 1023 | Banks | 5.76 | 17.52 | 46.694 |
| 5 | Axiata Group Bhd | 6888 | Mobile Telecommunications | 5.62 | 24.29 | 55.44 |
| 6 | Sime Darby Bhd | 4197 | Diversified Industrials | 5.51 | 22.71 | 52.4 |
| 7 | Digi.com | 6947 | Mobile Telecommunications | 4.16 | 20.74 | 41.29 |
| 8 | Genting | 3182 | Hotels | 3.68 | 19.8 | 30.88 |
| 9 | PETRONAS Chemicals Group Bhd | 5183 | Commodity Chemicals | 3.55 | 22.2 | 51.52 |
| 10 | Maxis Bhd | 6012 | Mobile Telecommunications | 3.45 | 30.2 | 49.48 |
| 11 | Petronas Gas | 6033 | Exploration & Production | 3.4 | 22.84 | 42.82 |
| 12 | IHH Healthcare | 5225 | Health Care Providers | 3.28 | 63.46 | 48.76 |
| 13 | IOI | 1961 | Farming & Fishing | 2.99 | 73.65 | 27.45 |
| 14 | Telekom Malaysia | 4863 | Fixed Line Telecommunications | 2.96 | 32.71 | 25.14 |
| 15 | Genting Malaysia Bhd | 4715 | Hotels | 2.5 | 20.45 | 25.53 |
| 16 | MISC | 3816 | Marine Transportation | 2.45 | 16.08 | 35.44 |
| 17 | AMMB Holdings | 1015 | Banks | 2.38 | 8.88 | 17.09 |
| 18 | Kuala Lumpur Kepong | 2445 | Farming & Fishing | 2.28 | 29.18 | 24.02 |
| 19 | SapuraKencana Petroleum | 5218 | Oil Equipment & Services | 1.98 | 12.03 | 14.26 |
| 20 | PBB Group | 4065 | Food Products | 1.8 | 17.73 | 17.83 |
| 21 | British American Tobacco (Malaysia) | 4162 | Tobacco | 1.7 | 20.11 | 18.50 |
| 22 | Hong Leong Bank | 5819 | Banks | 1.67 | 11.18 | 25.68 |
| 23 | YTL Corp | 4677 | Multiutilities | 1.63 | 14.85 | 17.38 |
| 24 | UMW Holdings | 4588 | Automobiles | 1.37 | 20.57 | 11.96 |
| 25 | Astro Malaysia Holdings | 6399 | Broadcasting & Entertainment | 1.22 | 28.59 | 15.97 |
| 26 | Petronas Dagangan Bhd | 5681 | Intrgrated Oil & Gas | 1.21 | 36.99 | 20.43 |
| 27 | RHB Capital | 1066 | Banks | 1.06 | 9.37 | 19.52 |
| 28 | Westports Holdings | 5246 | Transportation Services | 0.93 | 26.84 | 14.05 |
| 29 | Hong Leong Financial | 1082 | Banks | 0.64 | 10.01 | 16.32 |
| 30 | KLCC Prop & Reits - Stapled Sec | 5235SS | Real Estate Holding & Development | 0.63 | 13.63 | 12.71 |

2. Downloading data

(i) Download 3 years of daily data for British American Tobacco (BAT) from 1/1/2012 -12/31/2014

We import DataReader from pandas.io.data and import datetime from datetime in order to read the data from Yahoo!Finance. Besides, we import pandas, pylab and numpy in order to complete tasks.

First, we set the starting date and ending date for data we want to read. Then, we used DataReader to read the data in the date set by us. We use function print to show the data. The following table is the daily data for BAT.



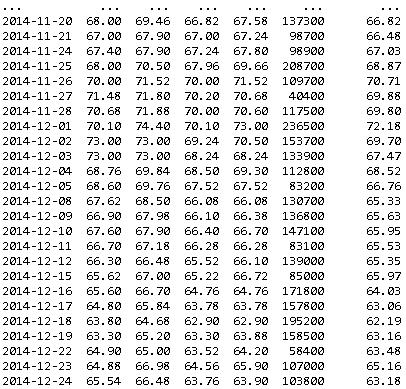
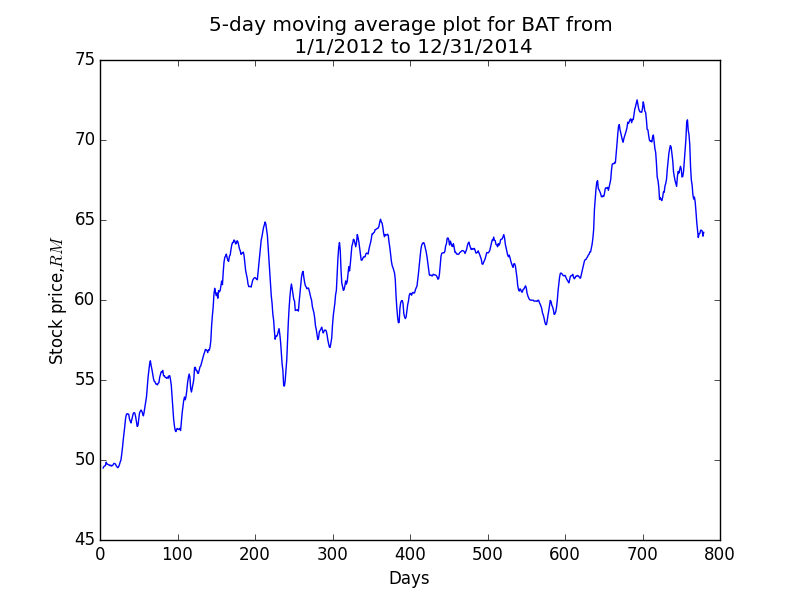


Figure 1: 3 years of daily data for British American Tobacco (BAT)

(ii) Plot the 5-day moving average graph for British American Tobacco (BAT)

We used the built in function pandas.rolling\_mean to find the moving average. First, we read the closing price for BAT by using data reader. Then, we use the function pd.rolling\_mean to calculate the moving average. After that, we use function p.plot to plot the moving average function followed by labeled the title, x-axis, and y-axis by using functions p.title, p.xlabel, and p.ylabel. We will show the graph by using p.show().

The following graph is the 5-day moving average for BAT:



(iii) Download FTSEKLCI daily data from 1/1/2012 to 31/12/2014

We used DataReader to read the 3 years daily data from 1/1/2012 to 31/12/2014 from Yahoo!Finance by typing klci = DR(“^KLSE”,’yahoo’,start,end), where ^KLSE is the code in Yahoo!Finance, start and end are the starting date and ending date set respectively. Then, we used function print to show the result.

(iv) Calculate correlation between British American Tobacco and FTSEKLCI

We first read and combine the data for the closing price for both BAT and FTSEKLCI by using DataReader. Then, we used the function corr() to find the correlation between BAT and FTSEKLCI. We show the result by using function print.

Finally, we transform the combined data into array and show it by using function print. This will allow us to see the closing stock price corresponds to the closing index.