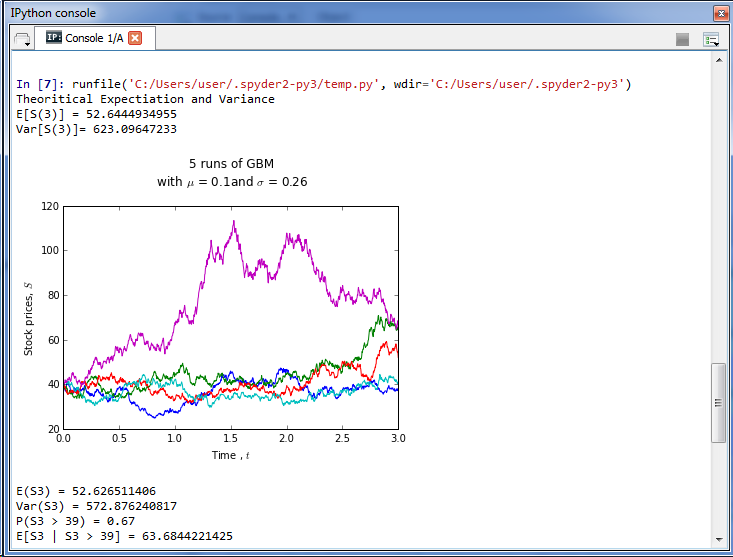
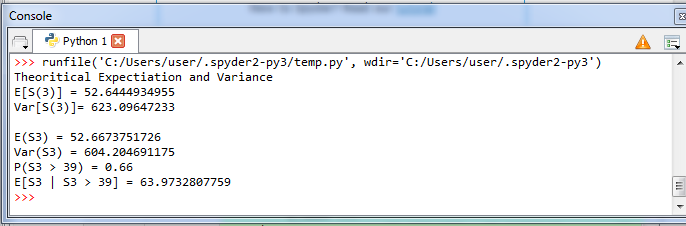
Task 1 (Part 1)

Output of coding:



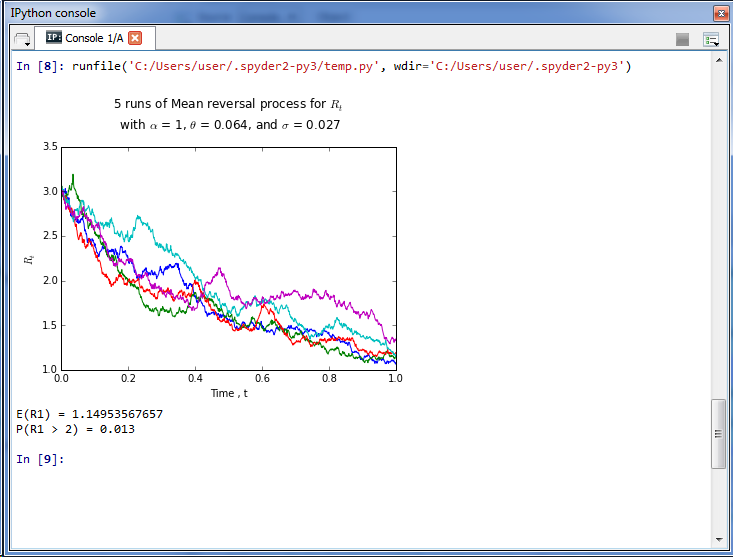


We have simulated 1000 runs of Geometric Brownian Motion for 0<t<3

Create an array(S\_T3) to store the value of last column of every row from the Geometric Brownian Motion we simulated.

1. To find E(S3)
   1. Use the numpy mean function, np.mean, to find the average of the values stored in S\_T3.
2. To find Var(S3)
   1. Use the numpy variance function, np.var, to find the average of the values stored in this S\_T3.
3. To obtain the value of P[S(3)> 39]
   1. Create an array (count) to store the statement of S\_T3>39, for any number that larger than 39 in S\_T3, TRUE will be returned and store in the count. Else, False will be returned and store in count.
   2. Sum up the array count, while TRUE return 1 and FALSE return 0.
   3. P[S(3)> 39] is the Sum of count divided by the total length of array S\_T3.
4. To find E[S3|S3>39]
   1. Multiply count with S\_T3, we will only get the number that is larger than 39 in S\_T3, and store in array S3\_MT39
   2. Sum up the array S3\_MT39
   3. E[S3|S3>39] is S3\_MT39 divide by the total number the value that greater than 39.

Task 1 (part 2)

Output of coding:

We first create 1000 Brownian paths. Then we create a variable name R, which the all the first column of this array start with 3, and all remaining columns are 0. After that, to complete R array, we need to use Euler Method. Mean reversal process is generated by using

for col in range(n\_partition):

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

1. To obtain E[R(1)]:
   1. Use the numpy mean function, np.mean, to find the average of the last column values stored in R.
2. To obtain P[R(1)> 2]
   1. We first create a variable name mask to store the values return by statement

R[:,-1] > 2

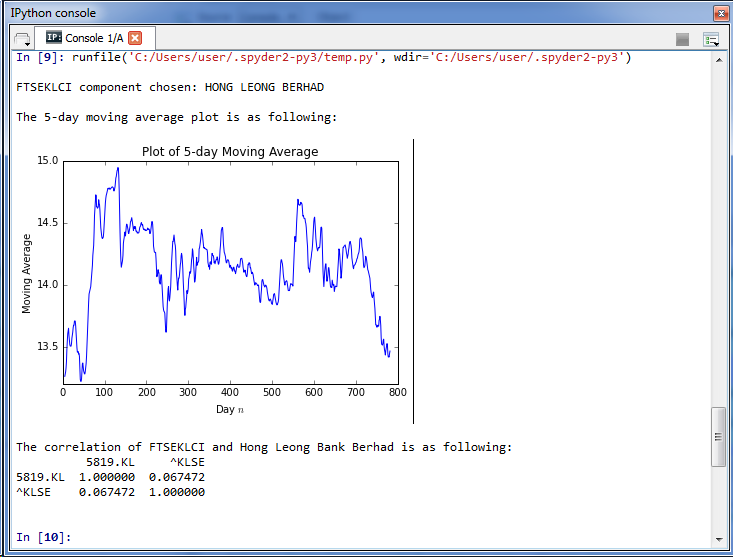
* 1. Sum up the value in mask while TRUE in mask returns 1 and FALSE in mask return 0.
  2. The Sum of mask divide by the length of mask, we will get P[R(1)> 2]

Task 2 (Part 1)

At the close of trading on 18th July 2015, the 30 constituent companies of the FTSE Bursa Malaysia KLCI with the summary of their details are as following:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stock Code** | **Stock Name** | **Stock Sector** | **Weightage** | **PE Ratio** | **Net Market Capital**  **(MYR in millions)** |
| 1295 | Public Bank Bhd | Financials | 11.60 | 16.31 | 73682.979 |
| 1155 | Malayan Banking Bhd | Financials | 9.32 | 13.07 | 87750.513 |
| 5347 | Tenaga Nasional Bhd | Trading Services | 9.28 | 10.79 | 69754.896 |
| 1023 | CIMB Group Holdings | Financials | 5.76 | 14.97 | 46524.058 |
| 6888 | Axiata Group Bhd | Trading Services | 5.62 | 23.60 | 55544.741 |
| 4197 | Sime Darby Bhd | Trading Services | 5.51 | 15.91 | 53354.208 |
| 6947 | DiGi.Com Bhd | Infrastructure Proj | 4.16 | 20.71 | 42062.750 |
| 3182 | Genting Bhd | Trading Service | 3.68 | 20.77 | 31068.523 |
| 5183 | PETRONAS Chemicals Group Bhd | Industrial Products | 3.55 | 20.77 | 51200.000 |
| 6012 | Maxis Bhd | Trading Services | 3.45 | 28.46 | 48883.526 |
| 6033 | Petronas Gas Bhd | Industrial Products | 3.40 | 22.91 | 42226.141 |
| 5225 | IHH Healthcare Bhd | Trading Services | 3.28 | 63.97 | 48251.265 |
| 1961 | IOI Corp Bhd | Plantation | 2.99 | 8.25 | 27837.135 |
| 4863 | Telekom Malaysia Bhd | Trading Services | 2.96 | 30.22 | 25140.584 |
| 4715 | Genting Malaysia Bhd | Trading Services | 2.50 | 20.18 | 23985.949 |
| 3816 | MISC Bhd | Trading Services | 2.45 | 16.28 | 35888.904 |
| 1015 | AMMB Holdings Bhd | Financial | 2.38 | 9.94 | 17723.408 |
| 2445 | Kuala Lumpur Kepong Berhad | Plantation | 2.28 | 24.76 | 24552.615 |
| 5218 | SapuraKencana Petroleum Bhd | Trading Services | 1.98 | 9.88 | 14153.159 |
| 4065 | PPB Group Bhd | Consumer Products | 1.80 | 19.45 | 17829.920 |
| 4162 | British American Tobacco (Malaysia) Bhd | Consumer Products | 1.70 | 21.08 | 19016.298 |
| 5819 | Hong Leong Bank Bhd | Financial | 1.67 | 12.02 | 25265.977 |
| 4677 | YTL Corp Berhad | Trading Services | 1.63 | 11.11 | 17270.384 |
| 4588 | UMW Holdings Bhd | Consumer Product | 1.37 | 18.21 | 11869.867 |
| 6399 | Astro Malaysia Holdings Berhad | Trading Services | 1.22 | 30.45 | 15813.168 |
| 5681 | Petronas Dagangan Bhd | Trading Services | 1.21 | 40.88 | 20504.891 |
| 1066 | RHB Capital Berhad | Financial | 1.06 | 9.67 | 19698.345 |
| 5246 | Westports Holdings Berhad | Trading Services | 0.93 | 32.98 | 14356.10 |
| 1082 | Hong Leong Financial Group Berhad | Financial | 0.64 | 11.20 | 16654.790 |
| 5235SS | KLCC Real Estate Investment Trust | Trusts | 0.63 | 15.31 | 12637.331 |

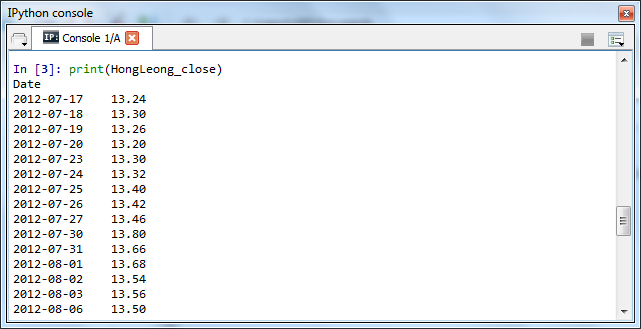
Task 2 (Part 2)

Output of coding:

Moving average is calculated several times for several subset of a data. For example, if you want a 3-year moving average for a data set from 2011, 2012, 2013, 2014 and 2015, you would find averages for the subsets 2011/2012, 2012/2013, 2013/2014 and 2014/2015.

To understand in depth on how to calculate the moving average, we use the data we obtain from Python code stored in file named “download\_data” and work out on calculating its 5-days moving average.

So, the following is the closing price of Hong Leong Financial Group Berhad from 17th July 2012 onwards:



The mean closing price for the first five days (17/7/2012-23/7/2012) is calculated by finding the mean from the first five days. This gives you the moving average for the fifth day.

The mean closing price for the second subset of five days (18/7/2012-24/7/2012) is

Continue calculating each 5-days moving average, until reaching the end of the set.

In general, the 5-days moving average is calculated using this formula:

*where MA is moving average and CP is closing price*

To verify whether the above hand calculation of 5 day moving average is working correctly, we can compare the values obtained above with the values obtained from running the Python code stored in file named “download\_data”.

