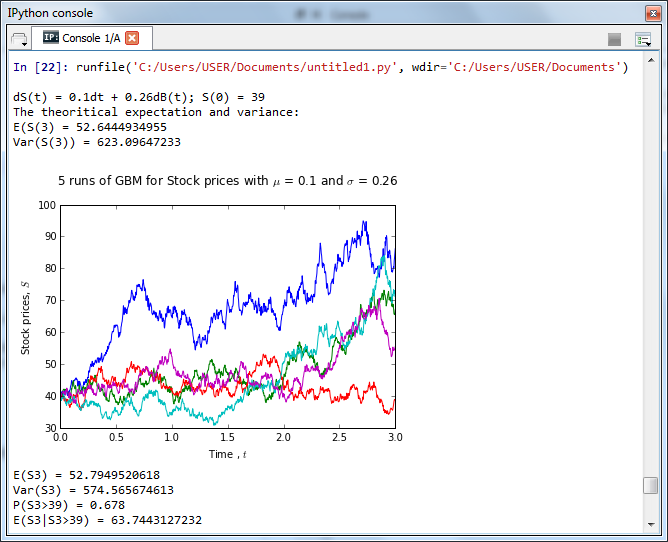
**Task 1 – Part 1: Simulating Geometric Brownian Motion**

Running the code stored in file named “gbm”, we obtained the following output:



Initially, we have stimulated 1000 runs of Geometric Brownian Motion for 0<t<3. Now, we going to make use of these to obtain E[S(3)], Var[S(3)], P[S(3)>39] and E[S(3)|S(3)>39].

An array is created to store the value of 1000 runs of Geometric Brownian Motion at time 3.

1. **To find E[S(3)]**
   * A numpy function, *np.mean*, is used to find the average of the value stored in this array.

**2. To find Var[S(3)]**

* + A numpy function, *np.var*, is used to find the variance of the value stored in this array.

**3. To find P[S(3)>39]**

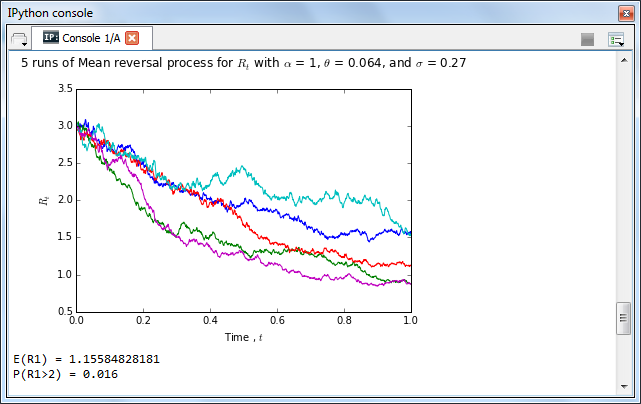
* + A variable, *count*, is created. Whenever the value in the array is greater than 39, *True* will be returned. The function, *sum(count)*, calculate how many *True* are returned.
  + *len(ST\_3)* calculate number of value stored in the array mentioned above.
  + A variable, P\_S3\_39, is created. This is to calculate the probability of S(3) greater than 39, by looking at the ratio of *sum(count)* to *len(ST\_3)*.

**4. To find E[S(3)|S(3)>39]**

* A variable, *S3\_39*, is created. Function *sum(S3\_39)* calculates the sum of value in the array that is greater than 39. Divide the sum by the total number of values that is greater than 39.
* To find E[S(3)|S(3)>39], variable *E\_S3\_39* is created. This can be found by looking at the ratio of *sum(S3\_39)* to the total number of values that is greater than 39, *sum(count)*.

**Task 1 – Part 2: Simulating Mean Reversal Process**

Running the code stored in file named “mr”, we obtained the following output:



At the beginning, we stimulated 1000 runs of Geometric Brownian Motion for 0<t<1. We then modify these to become a mean reversal process by using the Euler Method. The mean reversal process is generated by:

*for col in range(n):*

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

The value of 1000 runs of Mean Reversal Process at time 1 are stored in variable *R1*.

1. **To calculate E[R(1)]**
   * A numpy function, *np.mean*, is used to find the average of the value stored in this array.
2. **To calculate P(R(1)>2)**
   * A variable, *mask*, is created. Whenever the value in *R1* is greater than 2, *True* will be returned. Variable *mask* stored total numberof *True* that have been returned.
   * Another variable, *P\_R1*, is created. This is to calculate the probability of R(1) greater than 2, by looking at the ratio of *R1* to the total number of path, *n\_path*.

**Task 2 – Part 1: FTSE Bursa Malaysia KLCI Index**

FTSE Bursa Malaysia KLCI is made up of 30 largest listed companies by market value.

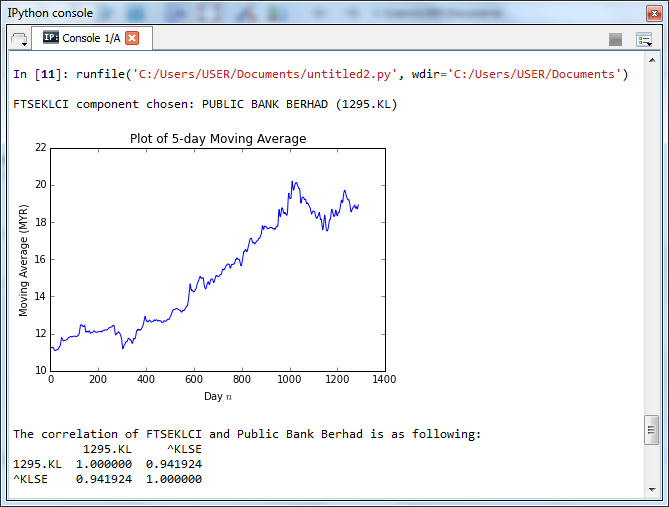
The details about the 30 components are extracted on 18th July 2015 from FTSE Group and Kenanga KenTrade, and the summary is as following:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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 | 14,356.10 |
| 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: Downloading data**

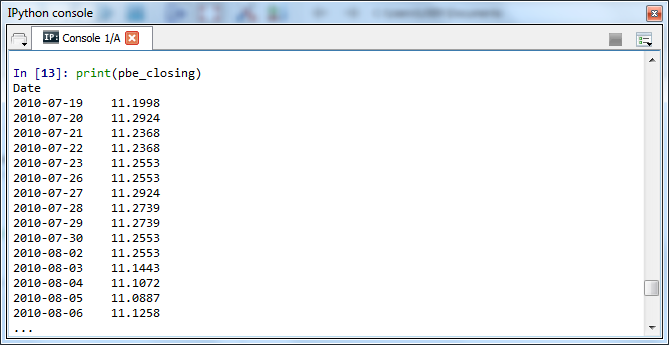
Public Bank Berhad is chosen in this task. The daily closing price of Public Bank Berhad from 17th July 2010 to 16th July 2015 is downloaded from Yahoo! Finance.

Running the code stored in file named “download\_data”, we obtained the following output:



The 5-day moving average is calculated by adding the closing price of Public Bank Berhad for 5 days, and then dividing this total by 5.

The closing price of Public Bank Berhad from 17th July 2010 onwards is as following:



The following shows how to calculate moving average using the closing price of Public Bank :

The moving average on 5th day (The average of the first subset of data, from 19/7/2010-23/7/2010):

The moving average on 6th day (The average of the second subset of data, from 20/7/2010-26/7/2010):

Then, continue calculating each 5-year average, until reaching the end of the data.

From the above example, we can know that the general formula to calculate the moving average is as following:

We are able to verify the 5 day moving average obtained from hand calculation by comparing it with the 5 day moving average obtained from the Python code:

