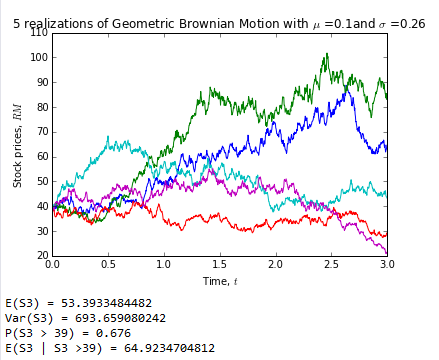
Simulating Geometric Brownian Motion

In task 1 part 1, I will simulate the Geometric Brownian motion based on equation

dS(t) = 0.1 dt + 0.26 dB(t); S(0) = 39.

First and foremost, I have to import the numpy and pylab to perform multidimensional array. Then, I need to declare the value of parameters such as sigma. I also create 1000 simulation runs on each path for 1000 paths within the interval zero to three. Next, I choose the first five realizations of Geometric Brownian Motion and use these five points to plot the graph of stock prices against time.

Furthermore, I use p.array(S[:,-1]) to find the values of stock price at time 3. I also use numpy module to find the mean and variance of the stock prices at time 3. Then I declared mask as a parameter that represent the frequency of stock price at time 3 greater than 39. Example: 1 represents the stock price at time 3 greater than 39, 0 represents the stock price at time 3 do not greater than 39. I use the ratio summation of mask to the total length of mask (total number of stock prices at time 3) to compute the probability of stock price at time 3 greater than 39. I also use the product of stock price at time 3 with mask to obtain the stock prices which greater than 39 at time 3. Then I use the ratio of summation of stock prices which greater than 39 at time 3 to summation of mask (total number of stock prices greater than 39) to compute the conditional expectation of stock price at time 3 given stock price greater than 39 at time 3. Lastly I use print function to present the result for calculation described as above. 

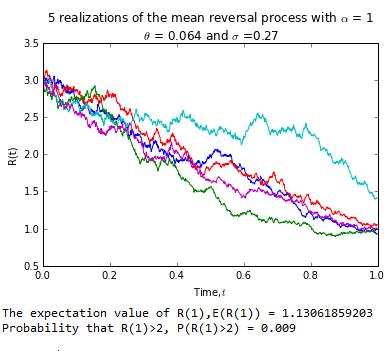
Simulating mean reversal process

In task 1 part 2, I will simulate mean reversal process based on equation

dR(t) = [0.064 - R(t)] dt + 0.27 R(t) dB(t); R(0) = 3.

First and foremost, I have to import the numpy and pylab to perform multidimensional array. Then, I need to declare the value of parameters such as alpha, theta and sigma. I also create 1000 simulation runs on each path for 1000 paths within the interval zero to one. Next, I choose the first five realizations of mean reversal process and use these five points to plot the graph of stock prices against time.

Moreover, I use R[:,-1] to find the values of Riemann sum at time 1. I also use numpy module to find the expectation value of the Riemann sum at time 1. Then I declared mask as a parameter that represent the frequency of Riemann sum at time 1 greater than 2. Example: 1 represents the Riemann sum at time 1 greater than 2, 0 represents the Riemann sum at time 1 does not greater than 2. I use the ratio summation of mask to the total number of path to compute the probability of Riemann sum at time 1 greater than 2. Lastly I use print function to present the result for calculation described as above.



Investigate the FTSE Bursa Malaysia KLCI Index

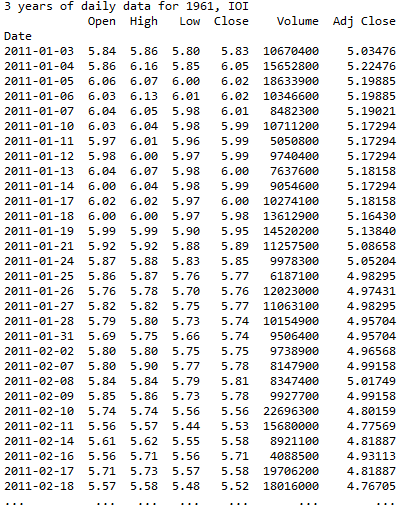
In task 2 part 1, there are 30 components stock as shown in table below.

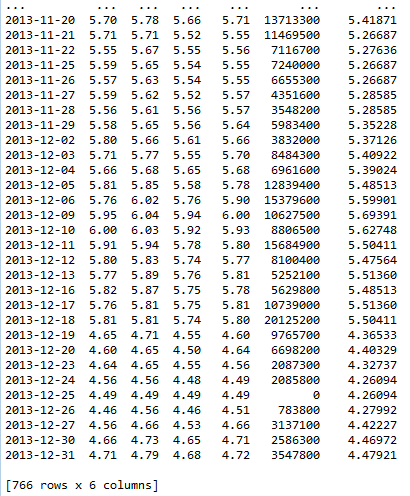
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Stock Name** | **Stock Code** | **Stock Sector** | **Weightage in FTSEKLCI (%)** | **P/E Ratio** | **Net Market Capital (B)** |
| 1 | Public Bank Bhd | 1295 | Banks | 11.6 | 15.24 | **73.84** |
| 2 | Malayan Banking | 1155 | 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 |
|  |  |  |  | 100.01 |  |  |

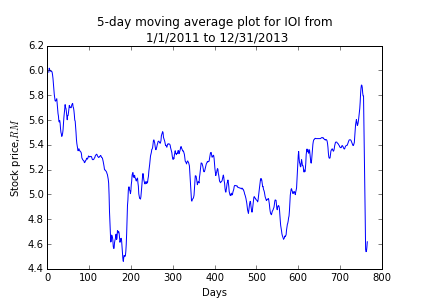
Downloading data

In task 2 part 2, I download daily data for a counter called IOI (code: 1961) from Yahoo!Finance starting from 1 Jan 2011 until 31 Dec 2013.

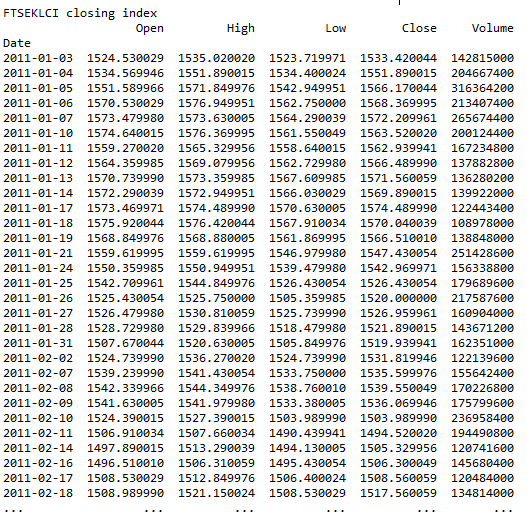
Instead of import numpy and pylab, I also import pandas to handle structured data. Then, I declared the starting and ending date by using dt(year, month, date). I also download the data using DR("1961.KL",'yahoo',start,end). Then I use print function to present the 3 years of daily data for IOI as shown at below.

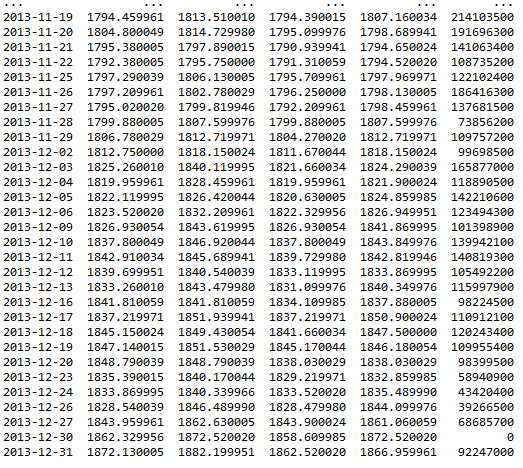


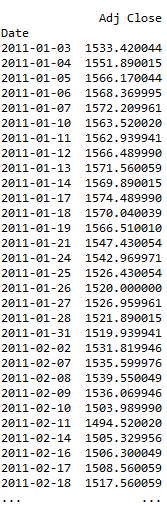
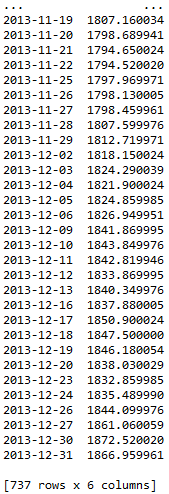


Next, I also declared “ioi” = DR("1961.KL",'yahoo',start,end)['Close'] to use data reader to read the close price. I also use pd.rolling\_mean(ioi,5) which is a built in function in pandas to calculate the 5-day moving average for the downloaded data. Then I use the plot function to plot the graph of stock price against days.

Despite this, I also use DR("^KLSE",'yahoo',start,end) to download daily data for FTSEKLCI and use print function to present the data.





In order to compute the correlation of IOI with FTSEKLCI, I assign “com” as a parameter that consists of IOI and FTSEKLCI data. After that, I use DR(com,'yahoo',start,end)['Close'] to download the close price of both indexes. Then I using corr() which is the built in function to calculate the correlation of the two indexes. Lastly, I use print function to present the result as below.

