**World Quant University**

**Professor: Ritabrata Bhattacharyya**

**Python II**

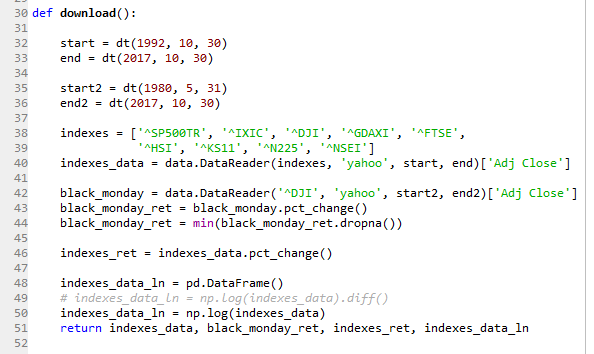
Nikolas Lippmann Pareschi - [nikolaslippmann@gmail.com](mailto:nikolaslippmann@gmail.com)

**Final Project:  The Misbehavior of Markets**

Introduction: Spyder PEP8 checker is truly a time saver, as I discovered in Mini Project 4 reading the Piazza forum. So, I have used it again for the Final Project, despite the fact that once in a while I have to reinitiate the kernel in Spyder. I have tried to avoid writing modular code in the Final Project, following the orientation received in Mini Projects 1 and 2.

1. Write a python program(s) to download end-of-day data last 25 years the major global stock market indices from Google Finance, Yahoo Finance, Quandl, CityFALCON, or another similar source.

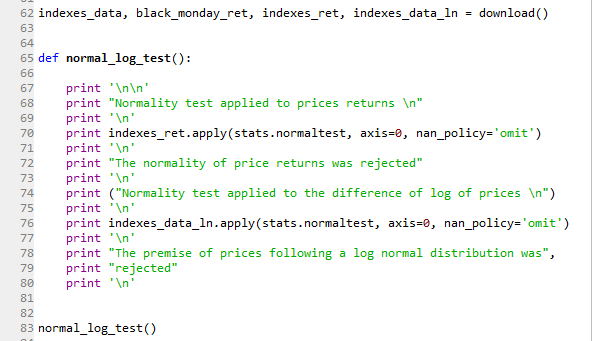
I have done this using this function. I used 2 start dates, the second was to calculate the daily drop in the black Monday 1987 event:



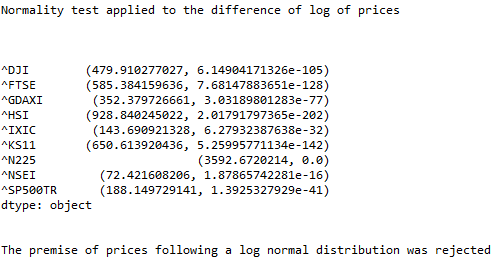
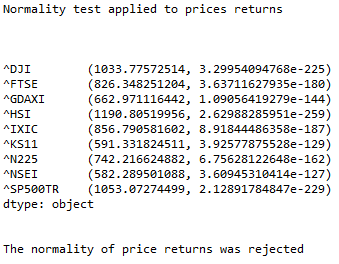
1. It is a common assumption in quantitative finance that stock returns follow a normal distribution whereas prices follow a lognormal distribution For all these indices check how closely price movements followed a log-normal distribution.
2. Verify whether returns from these broad market indices followed a normal distribution?

Steps 2 and 3 were done using the apply function. The logs of prices were taken directly in the function download() explained in the step 1. I avoided the Shapiro-Wilk test because the data was bigger than 5000 data points.

Code:



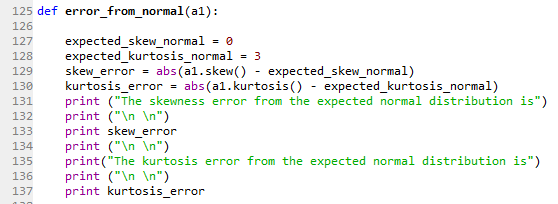
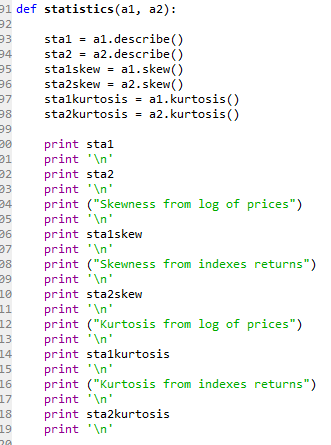
Results:



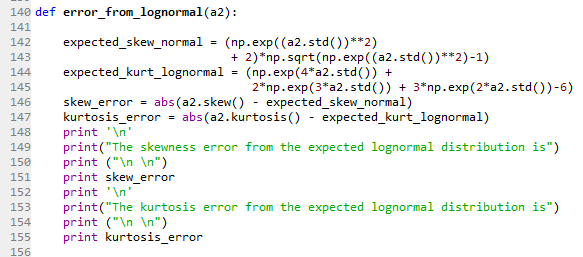
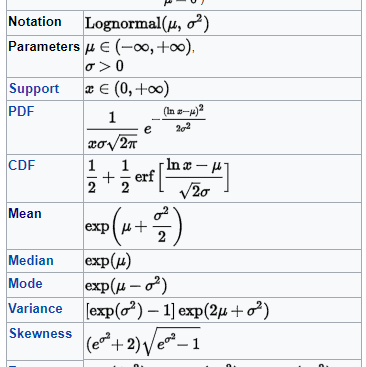
The normality of price returns was clearly rejected with immense t-scores and all p-values near zero. The lognormality of prices (normality of the log of prices) was also rejected with immense t-scores and near zero p-values.

1. For each of the above two parameters (price movements and stock returns) come up with specific statistical measures that clearly identify the degree of deviation from the ideal distributions. Graphically represent the degree of correspondence.

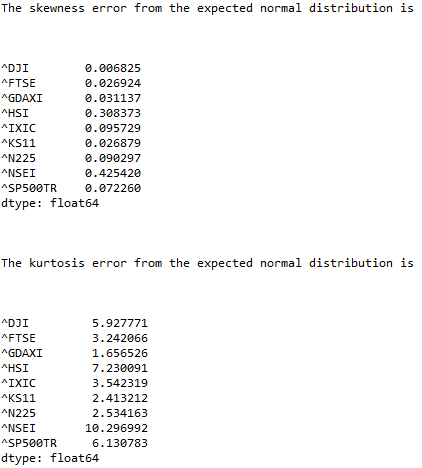
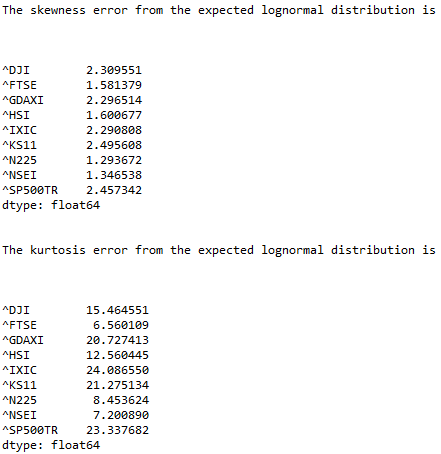
For this step I calculate the skewness and kurtosis of the return distributions and the prices and printed. Then I calculated the error, i.e., the expected skewness/kurtosis for a normal distribution minus the real skewness/ kurtosis and the expected skewness/kurtosis for the log normal distribution minus the real skewness/kurtosis:



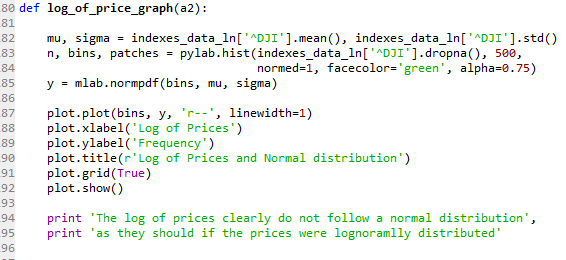
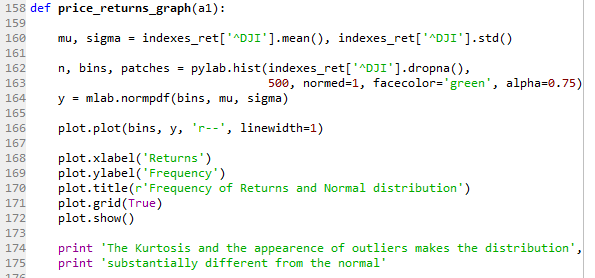
For the log normal I used Wikipedia to hardcode the equation to find the expected skewness and kurtosis:



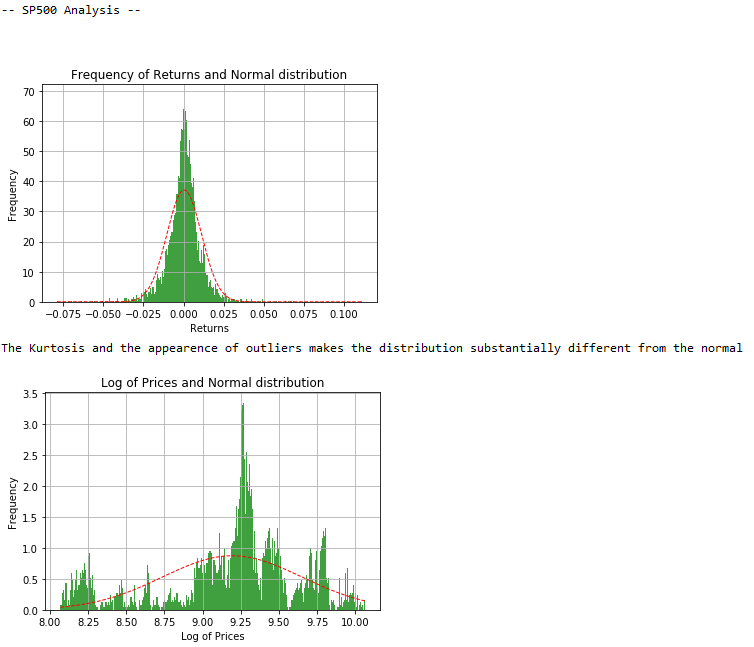
Results:



As we can observe, we have huge numbers in kurtosis, in both the distributions. We graphically represented the errors too:

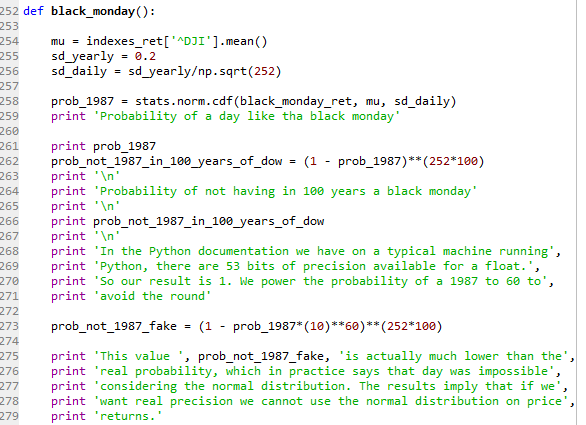


I will only copy 1 picture, in the program we have graphs for all the indexes. All indexes behave similarly with high kurtosis, well above the expected considering the distributions:



1. One of the most notable hypothesis about stock market behavior is the “Efficient market hypothesis” which also internally assume that market price follows a random-walk process. Assuming that Stock Index prices follow a geometric Brownian motion and hence index returns were normally distributed with about 20% historical volatility, write a program sub-module to calculate the probability of an event like the 1987 stock market crash happening? Explain in simple terms what the results imply.

I have found data in yahoo for the Dow Jones from dates before 1990, so I used Dow to calculate the daily mean of prices since 1980, to have the 1987 data. I took the given yearly standard deviation and calculated the daily standard deviation. Then I calculated how likely was a 1987 day with the stats.norm.cdf function. One should observe nevertheless that the Dow Jones have something like 1 century of data, so I calculated the probability of NOT HAVING a black Monday day in a year and in 100 years. The probability was so low that Python rounded to 1. I had to make a boundary powering the probability of 1987. Yet, even with the fake boundary the event of not having a day like 1987 had the probability near 1. In other terms, it was an impossible event under the normality of returns and lognormality of prices.



Here we had the round problem due to Python having 53 bits of precision for a Python float:





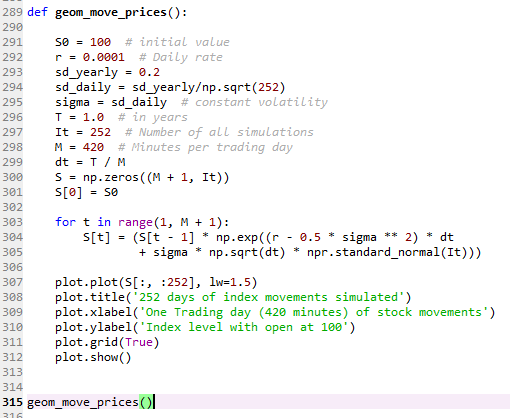
With the boundary:

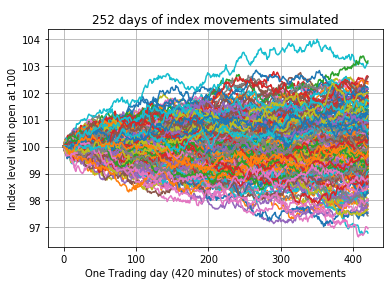




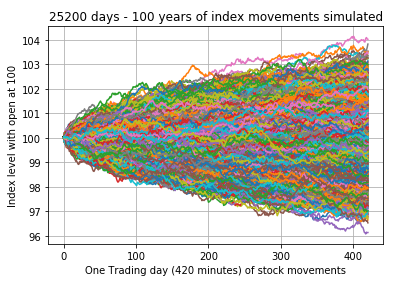
After this we tried the approach of making Geometric Brownian simulation to see if we could accomplish some dark days:

Simulation of 252 days:





We were still very far from a black Monday -0.2261 return. I have simulated so 25200 days. One century. I got this:

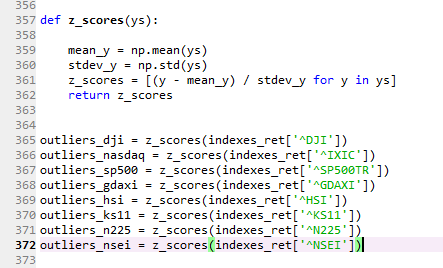


Still very far. I have commented the code for the 25200 days because it took 5 minutes to process.

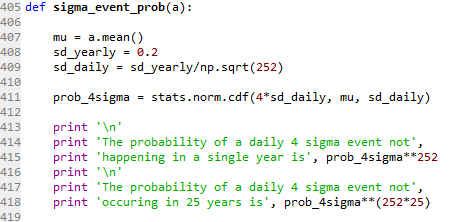
All these calculations and simulations imply that we must carefully use the assumption of normality of returns and the assumptions of lognormality of prices. We would only have gray Mondays, never black Mondays like that.

1. What does "fat tail" mean? Plot the distribution of price movements for the downloaded indices (in separate subplot panes of a graph) and identify fat tail locations if any.

I used the function below to calculate Z-scores in each of the indexes:



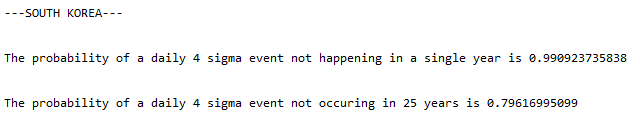
I then calculated the 4 sigma events (returns) in each of the indexes:



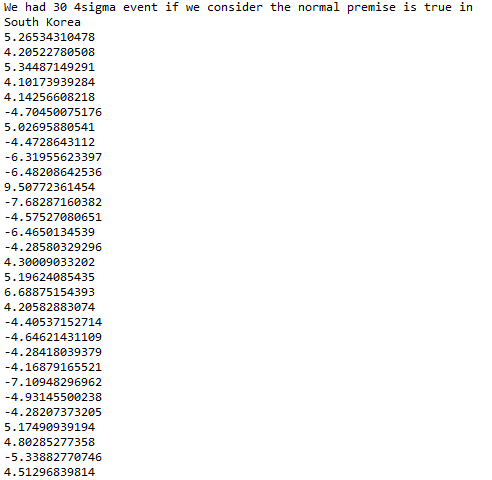
This is in the main function:



Results are like this:

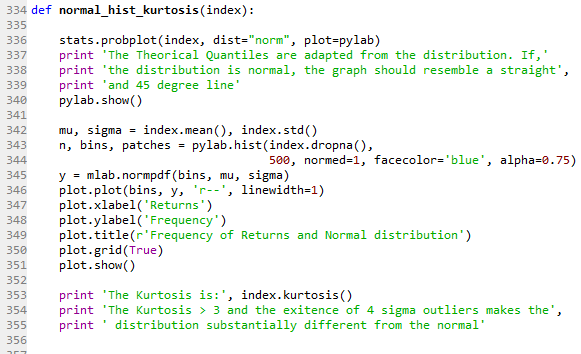


But in real life we had a bunch of 4 sigma days returns, implying in fat tails:

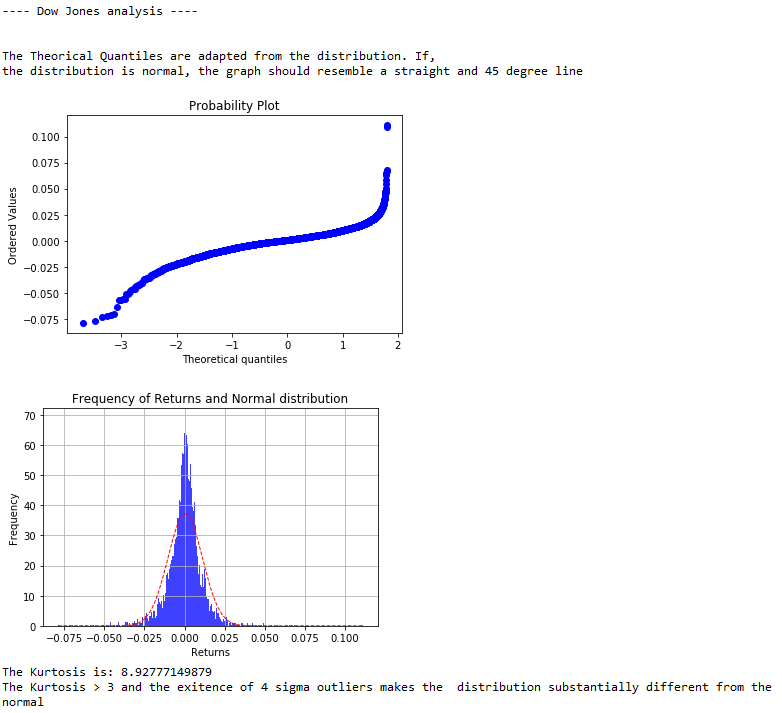


We then proceed to graphically display the distribution of returns and the expected normal. We also print the kurtosis. When the kurtosis is above 3 we usually have fat tails distribution. We also plot a QQ plot to demonstrate the existence of outliers:

Code:

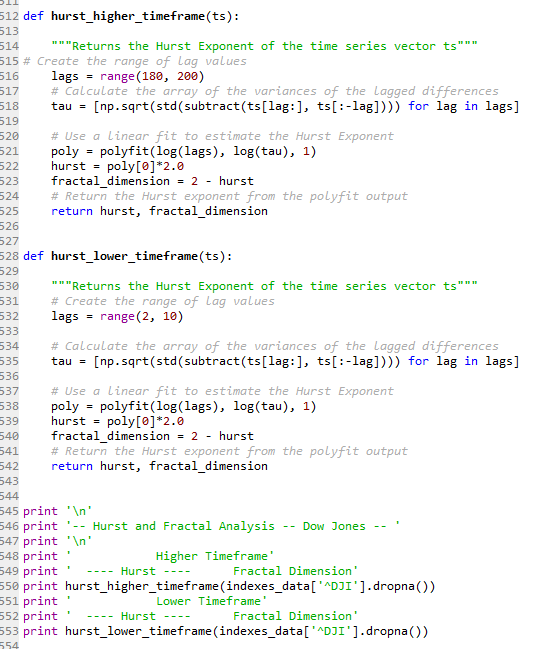
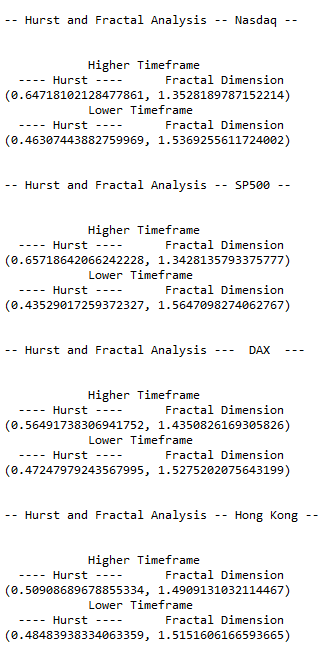


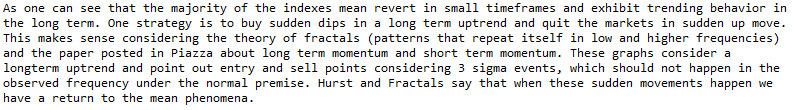
Example of results:



1. It is often claimed that fractals and multi-fractals generate a more realistic picture of market risks than log-normal distribution. Considering last 10 year daily price movements of NASDAQ, write a program to check whether fractal geometrics could have better predicted stock market movements than log-normal distribution assumption. Explain your findings with suitable graphs.

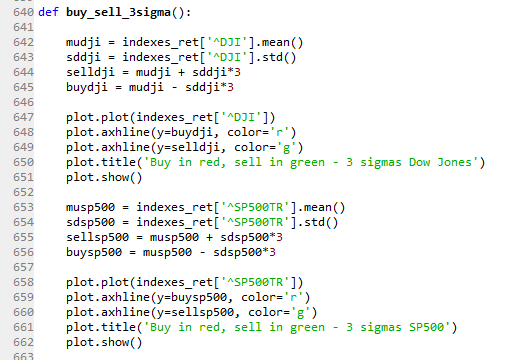
We first calculate the Fractal dimension using Hurst Coefficient. After some simulations we were able to conclude that higher time frames usually present a Hurst Exponent higher than 0.5 (Fractal Dimension higher than 1.5) and the lower timeframes usually present a Hurst Exponent lower than 0.5 (Fractal Dimension lower than 1.5). Code and some results:



PS: I changed the print to the main method in the picture above so the code could be more modular. I have maintained the picture because it is more understandable.

We then elaborate a simple strategy of buying and selling in the events that tend to mean revert, considering an uptrend (equities tend to rise in the long term). We would not even think in this strategy (because of few occurences) in a world of normality:



Nikkei example:

