Final Project: The Misbehavior of Markets

Project Instructions:

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.

* + 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.
  + Verify whether returns from these broad market indices followed a normal distribution?
  + 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.
  + 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.
  + 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.
  + 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.

\*Use the following stock market indices for this project:

Python Libraries Used:

The following libraries are used and can be installed using pip, e.g. “pip install pandas”

pandas 0.22.0 - see https://pandas.pydata.org/pandas-docs/stable/ for more information.

Urllib3 1.22 – see https://urllib3.readthedocs.io/en/latest/ for more information.

fix\_yahoo\_finance – see https://pypi.org/project/fix-yahoo-finance/#description

scipy.stats

numpy

Code Classes and Methods:

I created a Behaviour object(class) that implements the following methods. The \_\_init\_\_() method is the class constructor where all the necessary variable were declared and initialized either as an empty string or empty list and or empty pandas DataFrame.

**1, get\_market\_indices(self, data)**

**2, daily\_return(self, indexes, data)**

**3, do\_log\_normal(self, data, dcolor)**

**4, do\_normal\_dist(self, data)**

**main(self)**

Solution Steps

**Step 1:**

I wrote a method in my solution that connects to yahoo finance and downloads end-of-day data for the specified indices. The indices and their varous tickers had been prepared and saved in a csv file that is read into the application. So this method retrieved all the needed data all at one into a pandas dataframe with eight columns and the date columns serves as table index.

**Step 2:**

In another method using the lognorm method of the statsmodel module of scipy I fit a log-normal curve for into the histogram of the price movement of the end-of-day of data for all the indices that was downloaded in step 1

**Step 3:**

To verify if the returns follow a normal distribution, I wrote a method to calculate daily returns on the vartious indices that had been downloaded, I also presented the returns in an eight column dataframe from where I passed the individual columns to the normal distribution method that performed a normal curve fit on the histogram plot of the returns as calculated.

**Step 4:**

Here I used the students T-test for each of the two parameters (price movements and stock returns) as statistical measures to show the degree of deviation from the ideal distributions.

**Step 5:**

**Step 6:**

**Step 7:**