MF703 Problem Set 1 Solutions Outline

September 26, 2020

Problem 1: Historical Analysis of Sector ETFs

(a)

For downloading equity price data I recommend using Pandas DataReader. The syntax looks something like this:

```python
from pandas_datareader import data
tickers = ['stock1', 'stock2']
source = 'yahoo'
start_date = '01-01-2018'
end_date = '01-01-2020'
prices = data.DataReader(tickers,source,start_date,end_date)
```

(b)

There are multiple ways of calculating annualized return, below we outline 2:

Method 1

- Find the mean daily return for each ticker.
- Multiply each mean daily return by 252 to annualize

Method 2

- Total return over entire period is last price divided by first price $\rightarrow r_{total} = \left(\frac{S_{last}}{S_{first}}\right)$
- Find daily return by solving the following equation for $r_{daily} \rightarrow r_{total} = (1 + r_{daily})^n$, where n is the total number of days in your sample
- Then annualize the daily return by $\rightarrow r_{annual} = (1 + r_{daily})^{252}$

Note: You could also use log returns for your analysis. If you do that, remember that log returns are additive in nature.

For the standard deviation, simply compute the standard deviation of the time series of returns for each ETF and annualize by multiplying by $\sqrt{252}$

(c)

You should have found that the covariances are higher with monthly returns purely because monthly returns are usually higher than daily returns. However, note that comparing covariances is not very insightful. We can normalize (to compare apples to apples) by dividing by the variances of the two stocks which will give us the correlation. Comparing correlations is much more meaningful and insightful than comparing covariances.

(d)

The correlations should be bounded by -1 and 1 at any given time. Most sector ETFs will have high positive correlation with the market during most time periods. Variations can be explained by stages of a bull (or bear) market as well as changing weights of the sector ETFs in the market index.

(e)

The easiest way to calculate the rolling betas in this case would be using the rolling method on a Pandas Series. Since

$$\beta = \frac{Cov\left(r_{ETF}, r_{SPY}\right)}{Var\left(r_{SPY}\right)}$$

you can get the rolling 90 day covariance as well as the rolling 90 day variance of the market and just divide the two series. This should give you a time series of betas.
Alternatively, you can write a loop where you are taking 90 data points at a time from each time series, run a regression each time and extract the regression coefficient.

(f)

Here you can run the regression for each ETF, but you need to look at the p value (or t-value) of the coefficient α. If testing at 5% significance, we need t-value > 1.96 (since we have a lot of observations this is almost normal).
Alternatively, you can plot the ACF or PACF (they are equivalent at lag 1) function of each time series and check if the value is significant from the plot. The documentation on how to plot the ACF and PACF function can be found here and here respectively.

Problem 2: Exotic Option Pricing via Simulation

We simulate by discretizing the equation. We can do this by first defining a partition of the time interval $[0, T]$, $\mathbf{t} = \{t_0, t_1, ..., t_n\}$ such that $\Delta t = t_i - t_{i-1}$, $\forall i \leq n$, and $t_0 = 0, t_n = T$ where T is the time to maturity. Then we discretize the SDE by the following steps:

- $dS_t \approx S_{t_{i+1}} - S_{t_i}$
- $dt \approx t_{i+1} - t_i = \Delta t$
- $dW_t \approx W_{t_{i+1}} - W_{t_i} = \Delta W \sim \mathcal{N}(0, \Delta t)$

Finally, plugging the above into $dS_t = rS_t dt + \sigma S_t dW_t$. we get the following:

$$S_{t_{i+1}} = S_{t_i} + rS_{t_i}\Delta t + \sigma S_{t_i}\Delta W$$

Note: The analytical solution to this equation is actually known, so you could also use that to simulate instead. In that case, your discretized equation would look like this:

$$S_{t_{i+1}} = S_{t_i} exp\left(\left(r - \frac{\sigma^2}{2}\right)\Delta t + \sigma\Delta W\right)$$

(a)

The mean of the terminal value should be close to 100, and the standard deviation should be close to $25\% \cdot 100 = 25$

(b)

Most payoffs should be close to 0. The mean payoff should be close to 9.9477 and the standard deviation around 12.6.

(c)

Since the question specifies $r = 0$, the average discounted payoff is equal to the mean payoff from part (b).

(d)

The Black-Scholes formula applied to the same put option, gives a price of \$9.9477. Simulations have a randomness component to them so you will not get the exact same price every time. The more paths you generate in each simulation, your price will become more stable and converge to the theoretical price.

(e)

The price should be roughly \$17.67

(f)

The premium is just the difference between the two prices. It can never be negative since:

$$\begin{aligned} Premium &= Max(K - Min_{t \le T}(S_t), 0) - Max(K - S_T, 0) \\ &\ge Max(K - S_T, 0) - Max(K - S_T, 0) = 0 \end{aligned}$$

(g)

The higher the σ, the higher the premium should be. The bigger σ is, the more likely it is that the paths will take on more extreme values, thus making the lookback payoffs much higher.