# Introduction/Abstract:

Geometric Brownian Motion is widely used to model market behavior of stock prices. This has been mostly performed on traditional financial assets such as stock options. With the adoption and explosive popularity of cryptocurrencies, it brings a new asset class to attempt modeling via geometric Brownian motion. INCOMPLETE.

# Geometric Brownian Motion Derivation:

# Methodology:

Standard Geometric Brownian motion assumes that the noise is normally distributed. This assumption should be checked on the available data to determine if standard geometric Brownian motion would be an acceptable method to model/forecast market prices. Stock prices are log-normally distributed but returns (noise) follow a gaussian behavior. The returns histogram will be plotted against a normal distribution line, and a Q-Q plot will also be plotted to determine if the behavior can be described by a normal distribution.

The multiple day (step) predictions can be applied over a set period to determine how well it forecasts future values of market price. In the research paper **[INSERT RESEARCH CITATION]**, it was determined that a training set of 60 days produced the best predictions based on the mean squared value (MSE).

Instead of re-creating the procedure set in that research paper, root mean squared error (RSME) will replace the MSE because RSME is in the same units as the dependent variable, and thus provides better interpretability. Additionally, because this paper will be looking across three different types of financial assets (stock options, ETFs, cryptocurrencies), the RSME will be normalized (nRSME) so that the values can be compared across all assets since RSME will only make sense within the domain that it is calculated.

100,000 30-day geometric Brownian motion simulations will be generated from training sets of 30 to 100 days each, and the resulting simulations are compared to the actual (test) stock prices to obtain the RMSE, nRMSE, and MAPE.

A two-tailed t-test with a critical value of 5% will be conducted to determine if there is actually any significant differences between the forecasting capabilities of the varying training set sizes.

Another interesting methodology that the same research paper conducts is examining the experimental probability of predicting the correct direction of the price change. According to the same research paper, 100-day sets produced the most accurate direction prediction. To perform this experiment, 100,000 one-day simulations will be generated for training sizes from 30 to 100 each. The resulting experimental prices will be subtracted by the true *s(t-1)* price, and the direction will be checked against the true price change direction of *s(t)-s(t-1)*.

Finally, sample Geometric Brownian motion realizations will be generated and plotted against the actual data points.

Determine 95% confidence interval?

# Results:

## AMD Stock:

### Analysis of Returns:

The distribution of returns on AMD stock is normal with a couple outliers on the right end tail, however the behavior appears to be normal. Similarly, the Q-Q plot shows that the behavior can mostly be described by a normal distribution, but the points that trail off at the end indicate a bit of right skewedness.

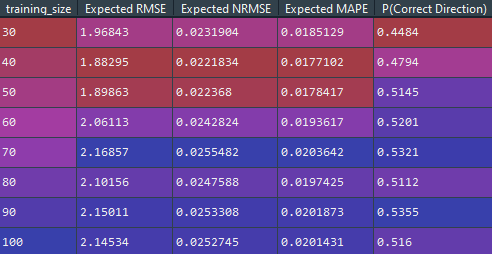
Chart, line chart

Description automatically generated

### Forecasting Predictions:

The expected RMSE and expected MAPE for the 40-day set is the lowest amongst the training sets. This differs from the findings in **[INSERT RESEARCH CITATION**], which concluded a 60-day training set to be the most accurate.

In comparison to the same paper’s results of the 100-day set being the most accurate training size when it comes to predicting the correct direction of the price change, the testing set on AMD stock prices results in a training size of 90 being the most accurate.

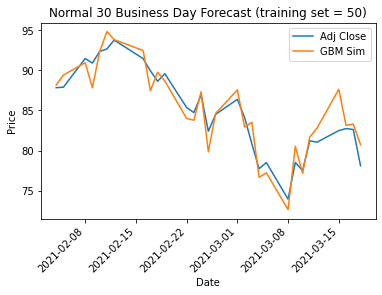
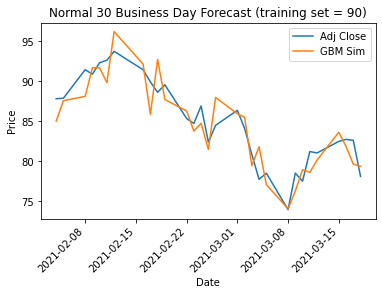


The expected RMSE of each training size are all similar; to determine if there is a statistically significant difference between each training size’s expected RMSE, a two-tailed t-test will be performed against each permutation:



The table above displays the calculate p-values of each tested permutation of training sizes. With a critical value of 5%, this test reveals that for the most accurate training set of 40 days, there’s no statistically significant difference between 40 and 50 days, but there is a difference between the rest. Unfortunately, this means that there has to be compromises between RSME versus P(Correct Direction). Given that the 50-day training set results in a higher P(Correct Direction), that should be the better choice between the two.

### Sample 30 Business Day Forecasts:

## S&P500 ETF (SPY):

### Analysis of Returns:

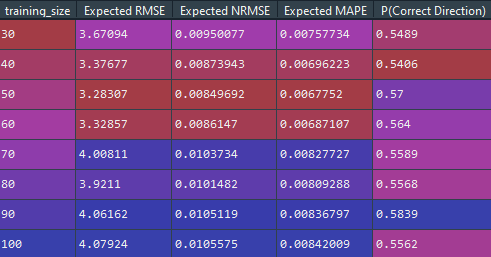
The distribution of returns on SPY ETF is a bit skewed to the left. However, it does still hold a general normal shape. The Q-Q plot reinforces this assessment since only a few of the points trail off in the beginning, indicating left skewedness.

Chart, line chart, scatter chart

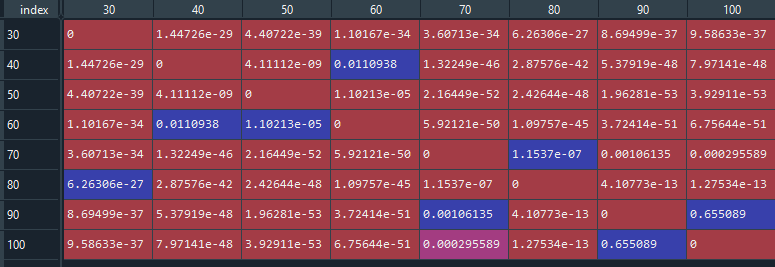
Description automatically generated

### Forecasting Predictions:

For the S&P500 ETF, the training size that resulted in the lowest RMSE and MAPE is the 50-day training set. When attempting to determine the best training size to measure direction accuracy, the size that produced the most accurate direction predictions is the 90-day set. These results are similar to the AMD results.

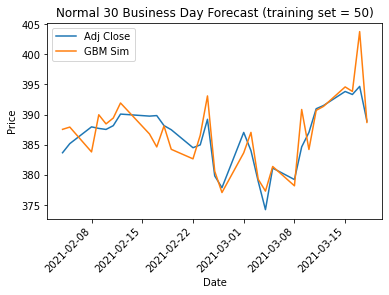
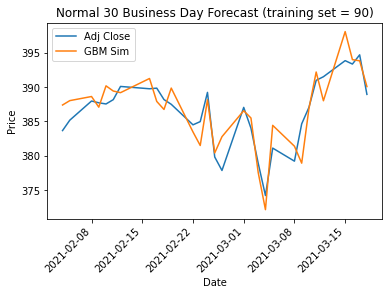


Running the t-test on these sets of training data, the 50-day training set’s difference is statistically significant from all other training sizes. Once again, there is going to be a compromise between RMSE and P(Correct Direction), but the 50-day set has the third highest probability, so the impact will be minimal:



However, when looking at the normalized RMSE (nRSME), it’s approximately 2 to 2.5 times smaller than AMD’s nRMSE. This seems to be expected given that ETFs by definition would have less variance than a single stock option since the variance is based on multiple stock options. The smaller variance would result in more predictable and consistent behavior to forecast due to the tighter bounds.

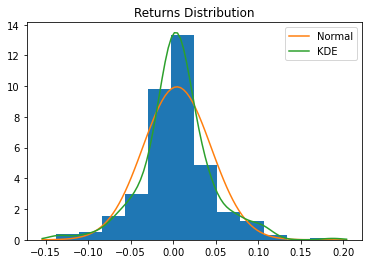
### Sample 30 Business Day Forecasts:

## Bitcoin (BTC):

### Analysis of Returns:

While the general shape and the Q-Q plot can be interpreted as generally normal, the normal distribution does not cover the peaks of the returns enough. Kernel density estimation can be employed to create a distribution that would capture this behavior better.

 Chart, line chart

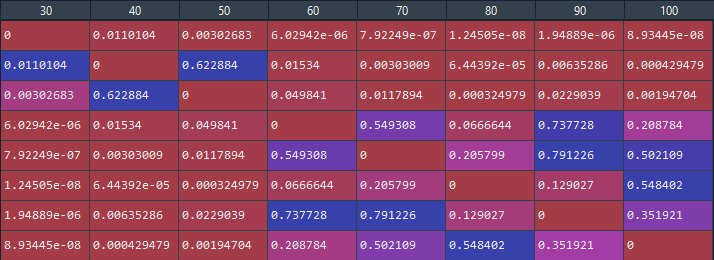
Description automatically generated

### Forecasting Predictions:

Training size of 80-days results in the best expected RMSE and MAPE. An interesting note when looking at the accuracy of directional change is that this produces the most accurate results amongst the three asset classes, with the most accurate resulting from a 50-day training set with an experimental probability of 63.6%. This is a bit unexpected given that cryptocurrencies are more volatile than traditional financial assets, and the assumption of normality by the geometric Brownian motion seems to be incorrect based on the returns distribution.



When running the t-test against the expected RMSE of each permutation, the training sizes of 60 to 100 are not statistically significant against the 80-day set. Given that the 60-day set produces the second most accurate P(Correct Direction), that should be the best choice of a training set size with minimal compromises in predicting the directional change.



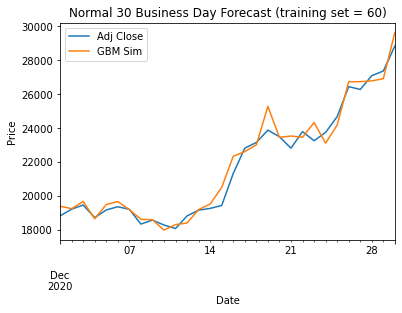
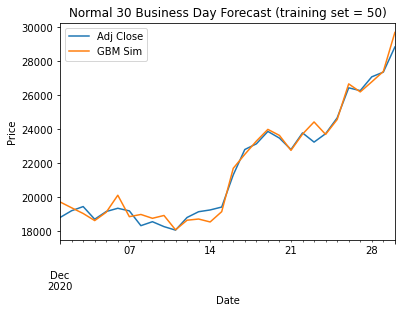
Looking at the expected nRMSE values show expected results given the volatile nature of cryptocurrencies. The expected nRMSE are higher than that of AMD’s, and three times higher than that of the S&P500’s nRMSE. This means that the forecasting capabilities of standard geometric Brownian motion isn’t as accurate in comparison to its applications to traditional assets where a normal distribution assumption is more acceptable.

When attempting the same procedure but with a KDE to model the change in price, the results are surprisingly very similar, albeit a bit less accurate. Interestingly enough, **[INSERT RESEARCH CITATION**] had a similar result even when it was noted that the Cauchy distribution fit better than a normal distribution.

Perform t-test to ensure that there is no difference.



### Sample 30 Business Day Forecasts:

# Conclusion: