Time Series Final Project Report

Purpose:

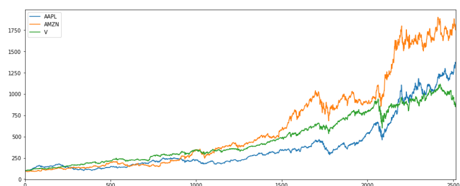
Our purpose is to model stock data as a time series. Through this project, we aim to perform arima modeling on stock data and see how well this kind of model fits the data. We also will perform forecasting and compare the forecasts obtained from the time series model to models commonly used in stock analysis.

Data:

Our data is the adjusted closing price for the stock price of Apple, Amazon, and Visa, every business day for the past 10 years. The data was taken from yahoo finance.

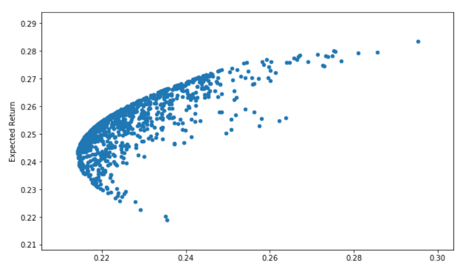
Initial Analysis:

First, we did some initial analysis on the data to understand it. Here we plot the three stock data normalized to 100, so we can see how they perform relative to each other.



A large increase can be seen around halfway in, marking the time of the technology boom. A second large increase can be seen for Amazon during the last 2 years. This jump can be explained by the effects of the pandemic on everyday life, resulting in a higher dependence on Amazon services.

Next, we create a portfolio for the three stocks. If we were to invest in all three stocks, what should be the weights? In order to answer this question, we performed a markowitz portfolio optimization. Below is the efficient frontier.



Using this curve, we calculate the weights with the highest return with risk between 0.235 and 0.245.
portfolio weights: AAPL: 0.38411, AMZN: 0.56330, V: 0.052589
Expected log annual return: 0.27140. Expected Volatility: 0.24485

Measuring Stationarity, Differencing:

We split each series in half to check if different parts of the whole series have a similar mean and variance between each half. If it's at least similar for both mean and variance we can support that the series is probably stationary and oscillates around a position that won't drift into infinity or negative infinity. After running the mean and variance of the different series we can easily see that mean and variance increase rapidly from the first half of the split and the second. This supports that we should test all three series for stationarity. The code attached shows the graphs after differencing occurs as well as the other ADF Tests for stationarity.

```
aapl_1.mean()                amzn_1.mean()                visa_1.mean()
20.770856614160717           396.9265558512329            52.74437878918055

aapl_2.mean()                amzn_2.mean()                visa_2.mean()
71.24023096979325            2045.4031147273436           158.54766016136733

aapl_1.var()                 amzn_1.var()                 visa_1.var()
25.56560451882553            32038.55292371845            295.94198819980414

aapl_2.var()                 amzn_2.var()                 visa_2.var()
1560.3096976977772           768451.3155908189            2222.1202429051723
```

We wrote a custom function that takes a series and supports if a series is stationary from the Augmented Dickey Fuller Test (ADF-Test). For all critical value benchmarks, all three series subsequently failed to be stationary by a wide margin. This is probably to be expected, stock data is normally a random walk with unpredictable jumps between time periods. The primary method of amending this problem is first differencing. The differencing will subtract the current position minus a specific shift of the same series (1 in this case). Re-running each of the first differenced series in the ADF-Tests support that each of these series are indeed now stationary so we do not need to re-run the differencing procedure further, as it could hinder analysis to be over-differenced. This would be inefficient due to our concept of parsimony. With this portion of procedures developed, we can proceed to perform more thorough analysis, unlike if it remained non-stationary. An example of the test ADF for a stock is below.

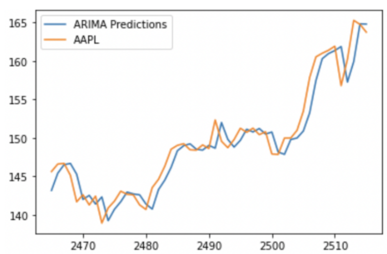
```
Augmented Dickey-Fuller Test Results:
ADF Test Statistic            2.736165
P-Value                       0.999088
# Lags Used                  27.000000
# Observations Used        2488.000000
Critical Value (1%)          -3.432981
Critical Value (5%)          -2.862702
Critical Value (10%)         -2.567389
dtype: float64
Is the time series stationary? False
```

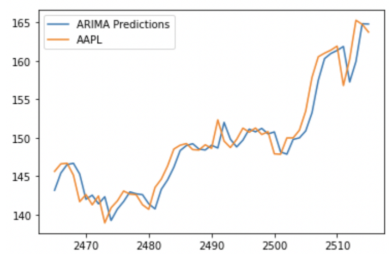
ARIMA analysis

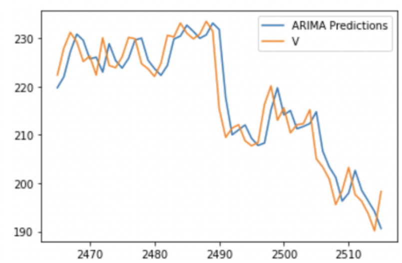
Due to the stochastic nature of daily stock price fluctuations, it is generally hard to choose an ARIMA model that will accurately predict stock prices. Stock prices are theoretically speaking assumed to follow a random walk, based on Brownian motion. In our case though, one autoregressive term in the ARIMA model of each stock greatly increased the model's performance. After analyzing the ACF, and PACF plots for all the stocks, we decided to choose an ARIMA(1,1,0) model for Apple , an ARIMA(1,1,1) model for Amazon and an ARIMA(1,1,1) model for Visa. The ACF and PACF plots clearly recommended the addition of an autoregressive term, but the moving average term was determined using AIC as a selection criterion. No seasonal patterns were observed in the securities' time series data, and both the ACF and PACF plots further confirmed the lack of seasonality in the time series data.

Training and prediction

Based on the order we trained the model. We checked the result to see whether the model trained appropriately. We can see from these graphs that our model fits the data pretty well.



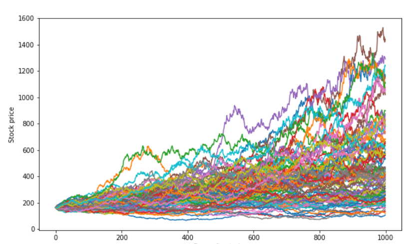


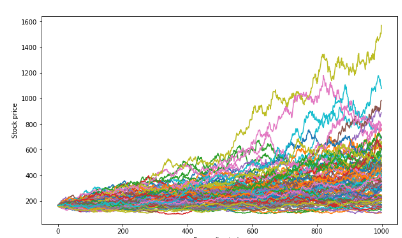


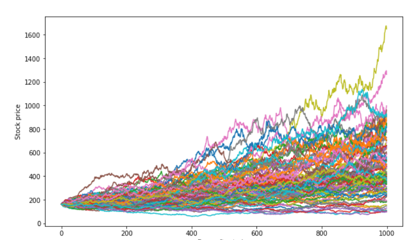
Monte Carlo

Monte Carlo simulations are often used in stock analysis. Our goal is to compare the Arima model to this simulation. We performed a monte carlo simulation for the next 1000 days.

AAPL AMZN V







These graphs displayed a wide variety of predictions. Monte Carlo simulations can account for many more possible random futures than arima can, which is useful because stock data rarely changes as predicted. Accounting for many possibilities is often more useful than following a singular model..

Authors: Jeffrey Dean 161009064, Kunal Jangam 186002760, Jinwoo Lee 213001413, Michael Gleyzer 185001920