Jeffrey Rodriguez, Sued Spada, Daniel Sullivan
AMS 315 - Data Analysis Report
1/19/2018

Introduction

By analyzing the stocks of two companies, we may be able to find and predict trends between their stock prices over time. This of course requires some degree of correlation between the two. One way to go about studying this correlation is to study the log-return values of stock prices. Two companies we believe may be highly correlated are Amazon and Netflix. Although Amazon provides a wide variety of services, both of these companies own production studios and provide streaming subscriptions for television shows and movies, along with original series which they have produced. Through statistical analysis, we aim to study the results associated with the linear relationship between the daily log-returns of these two stocks. Ideally, we will be able to use our collected data to predict the actual stock prices for the month of December 2017.

Method

The data required for this analysis was obtained from Yahoo Finance, with results generated using the R programming language and software R Studio (more details in Appendix). Data for stocks from Amazon and Netflix were downloaded with dates between December 2016 and December 2017. However, for much of the analysis, we focus on the data between December 2016 and November 2017 which make up our training data.

First we compute the log-returns of Amazon and Netflix's Adjusted Close Prices column, using the formula $\log - \text{return} = \log(\frac{Y_{i+1}}{Y_i})$, where Y_i denotes the adjusted close price stock of the i^{th} day. Log-returns were computed using a function from the R package, 'stochvol' and stored into a data frame to be used for most of the following computations. Unless noted otherwise, any other computations were performed using built-in functions inside R. For all parts, we work with a level of significance of 0.05.

Next, we test if the expected means of these log-returns are equal to zero by performing a t-test. Next, we perform another t-test to test if there is a significant difference between the two means. This first requires we perform a variance test to see if the variances are equal in order to determine if we should perform a pooled or unpooled t-test. After this, we calculate the correlation between the log-returns of the two stocks, and test for significance.

After performing these tests, we chose Netflix to be our dependent stock, and Amazon as the independent stock (and as training data). We then fit a linear model and found the confidence interval for the slope.

Next the aforementioned linear model is used to predict the log-return of Netflix's stocks for the first day of December 2017. A new linear model was created, with this date included, and this model is used to predict the log-return for the second day. This process is repeated for the rest of the month (20 dates in total).

These values were stored in a vector in R, and converted into a predicted price using the inverse of the log-return formula: $Y_{i+1} = Y_i \times \exp\{\log(\text{return})\}$. These predicted log-returns were also used to compute prediction intervals, with upper and lower bounds converted similarly to price values. Finally, a plot was made with the x-axis corresponding to dates of December 2017, and the y-axis corresponding to the predicted price, actual price, and predicted interval bounds for Netflix. This plot was created using the R package, 'plotly'.

Results

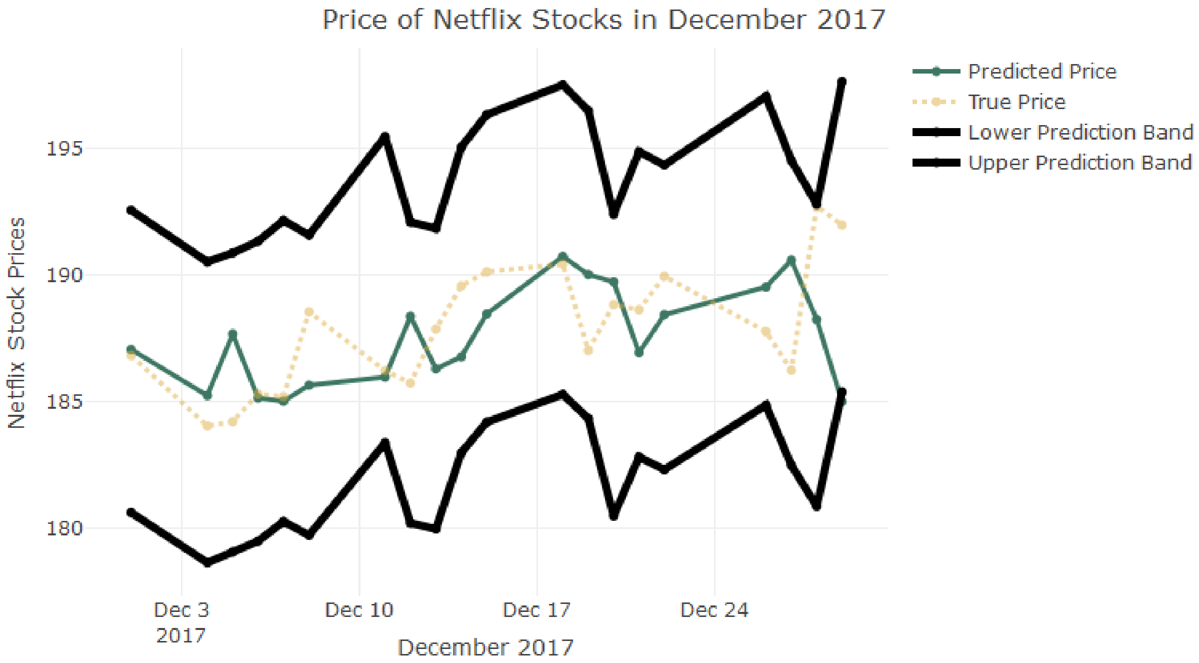
For the t-test of Amazon's log-returns mean, we obtain a sample mean of 0.0018 with $p-\text{value} = 0.0270$. For Netflix, we obtain a sample mean of 0.0019 with $p-\text{value} = 0.0932$. From these, we can conclude that although the sample means are close, only Netflix has an expected log-return mean equal to zero, as the p-value is not less than 0.05.

Next, we perform the variance test to see if their variances are equal. This results in a ratio of $F = 0.5471$, with a very small p-value. So, we conclude that the variances are not equal, and must perform an unpooled t-test. After performing this t-test, we obtain a p-value of 0.9742 and conclude that the difference in means is not significantly different from 0.

The correlation between Amazon and Netflix is then computed to be 0.4023 with the correlation test reporting a very small p-value, so we conclude from this that the true correlation between the two is not zero.

The linear model for the training data is fit as $\hat{y} = 0.0009 + 0.5339x$, with $R^2 = 0.1618$, and the slope has a confidence interval of $(0.3894,\ 0.6984)$.

The following plot was graphed after obtaining the predicted values for Netflix's stocks and the upper and lower prediction interval bands.



Conclusion

Our goal was to study the relationship between two stocks we suspect may be highly correlated. Upon further investigation, we find that the correlation is not as high as would be ideal ($r \approx 0.40$). However it is still significantly different from zero as reported by the correlation test, and so it can be helpful in studying the stock of Netflix. We used the linear model of their log-returns to predict how the stock prices would behave in December 2017 as shown in the above graph. Some points show the prediction to be near accurate, along with some points having a smaller or large distance from values in the same day. Both the prediction and true values stay bounded between the prediction interval, which hints at some degree of accuracy for this procedure. Ideally, a different stock would be compared with Netflix to predict it's behavior. The higher the magnitude of the correlation coefficient, the better, but this model does provide some near equivalent values, as well as some with relatively small differences between points, so we can gain some insight by studying stock data this way.

Appendix

Please see the .csv files to view data downloaded from Yahoo Finance.
All programming was done in the attached .R file. As stated previously, tests were performed using R's built-in functions.
Results for parts A through D can be viewed in a separate results text file, with the log-return calculated using the 'stochvol' package.
Results in part E were not included as they are stored in data structures (20+ entries per column), but by running the script, they will be readily available to view. Part e made use of the 'plotly' package, logret from 'stochvol', and built-in methods to create new data structures used for this part.