**Individual report**

## Introduction

For the Final Group Project, my partner Andrew Gobbi and I used an LSTM neural network to predict day ahead foreign exchange rates. The motivation behind selecting this use-case revolves around our group's working knowledge of time series statistical methods used for forecasting future values based on past observations. This project topic presented an opportunity to use a neural network to identify the underlying generating process which was both important to the subject matter taught during the course and valuable exercise in improving our domain expertise.

## Scope of Individual Work

Andrew and I both selected a currency dataset to divide the initial steps of our work evenly. I chose the Brazilian Real / US dollar as my currency exchange rate to study while Andrew decided to study the Swiss Franc / US dollar exchange rate. Both datasets are supplied by the United States Federal Reserve and can be accessed online. We agreed to individually research the topic but collectively decide on the best experiment design, and after a few weeks of literature review we settled on the LSTM network architecture and the Keras deep learning library. I relied heavily on my partners ability to logically think through problems related to software coding

## Processing and Exploratory Data Analysis

The first step to understanding the Brazilian Real dataset was to use my time series analysis background and implement traditional tests for stationarity, unit roots, and autocorrelations. This process enabled me to recommend a starting point for our search for the optimal number of neurons to use in our LSTM model. Using the ACF of the differenced Brazilian Real dataset and the PACF, it was clear that the AR(1) lag was significant and should be utilized when explaining the model results.

## Training and Testing

The two currency datasets provided ample observations to perform in-sample and out-of-sample training and testing. The Brazilian / US dollar exchange rate data provided daily exchange rate levels for over 1,100 observations to index and fold the data into the necessary train-test-split configurations for implementation.

I wanted to be sure I had an effective understanding of how our network operated so I implemented my own code to explore the dataset and test for the optimal number of neurons for our LSTM model. Admittedly, my process for testing the Brazilian Real dataset for the optimal number of LSTM neurons was crude, but I arrived at 20 neurons. 20 neurons were particularly good for this dataset because the mean squared error of the test set reached a desired level of performance relative to the test sets with additional neurons.

**Validation & Back testing**

My partners code was far superior to mine therefore I utilized his back-testing procedure to confirm the performance of the predictions on our various test sets. Our model initially was trained on the first three years of the data and tested on the fourth year. Out second configuration used less training data and trained using the first two years, while testing on the third. The last configuration back tested by training on the third year of data while testing on the second.

**Results**

The 20 neuron LSTM model performed consistently in each experimental scenario. The model was able to minimize the loss function, or decrease its mean squared error, enough to conclude that the forecast would express the desired level of fit. These results were consistent across both currency datasets and lead to unbiased forecast estimations.

**Conclusion**

The only issue with forecasting one step ahead is that there is not much room for arbitrage. Most models can effectively predict outcomes within one day and therefore a better outcome would be to extended an accurate forecast over multiple steps ahead to have any economic impact.