Simulating Tracking Error in Variable Annuities

Major Qualifying Project
Sponsored by Worcester Polytechnic Institute and Towers Watson

12/17/2010
Jon Abraham, Advisor
Guillaume Briere-Giroux, Liaison

Ian Cahill
Elizabeth Dailey
Blake Kelly
Charlotte McDonnell

Abstract

Our project analyzed the standard deviation of projected mutual fund returns relative to the actual performance of the mutual fund. We performed Monte-Carlo simulations using geometric Brownian motion to obtain the projected mutual funds. Our project tested the effects of a regime-switching model and distribution of manager's alpha by generating many scenarios to draw conclusions about fund mapping accuracy. Using our results, we analyzed the specific effects of these variables to provide conclusions to our sponsor, Towers Watson.

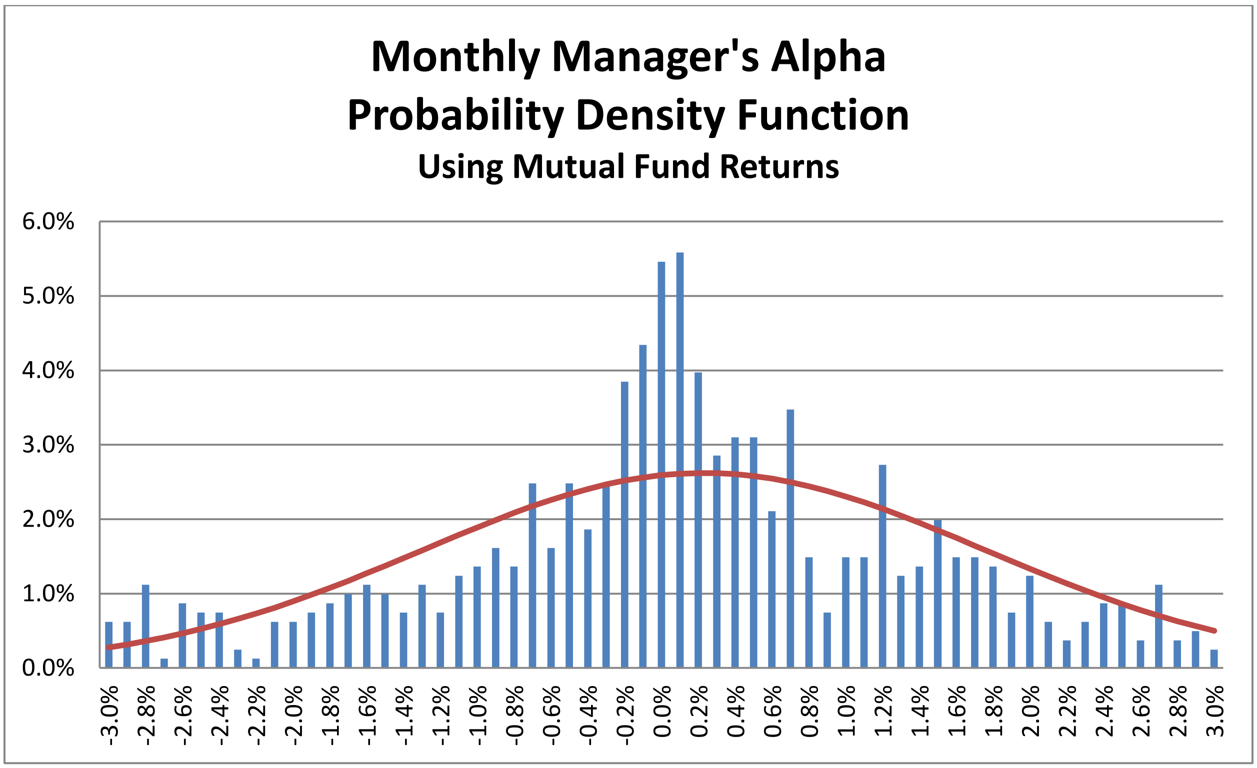
Executive Summary

Variable annuities, a new investment product, are growing rapidly in today's markets and beginning to outsell traditional fixed annuities (Mercado, 2010). This new products provide a safe investment vehicle by creating a portfolio compiled of a mutual fund as well as a hedging program which provide benefit options to the holder. By actively managing mutal funds, insurance companies protect themselves against losses (Stulz, 1985). However, these mutual funds rarely have defined benchmarks and therefore it is difficult to determine the performace of the fund.

In order to predict the performance of investment options which are part of variable annuities, we have created a Monte Carlo simulation which can create a real life scenario for insurance companies. The simulation tests the tracking error of the fund from the proxy over future scenarios which are generaged based on actual past data. By adding complexities into the modeling process, our team has quantified the effect of the distribution of manager's alpha, regression, and switching regimes on the benchmark's tracking error.

Our project focused on four main versions of the Monte Carlo Simulation. The first is the perfect world, where the mapping weights are exact and the MFI returns follow the SPY. The first layer of complexity we added was the use of a regression between the MFI and the RUS and SPY for the Future-Future scenarios. Changes in correlation between these two indicies were also tested, but there was little to no difference in the tracking error when the correlation was modified. Therefore, we determined that the correlation between the indicies does not have a large impact on the tracking error.

We next investigated the distribution of manager's alpha to be used in the model. We observed daily and montly data from Exchange Traded Funds (ETFs) as well as mutual funds against their benchmarks to find sample manager's alpha. Although the manager alpha component was different for each individual ETF or mutual fund, our overall result is that manager's alpha is normally distributed with an annual mean of 0.15% and standard deviation and volatility of 2.25%.



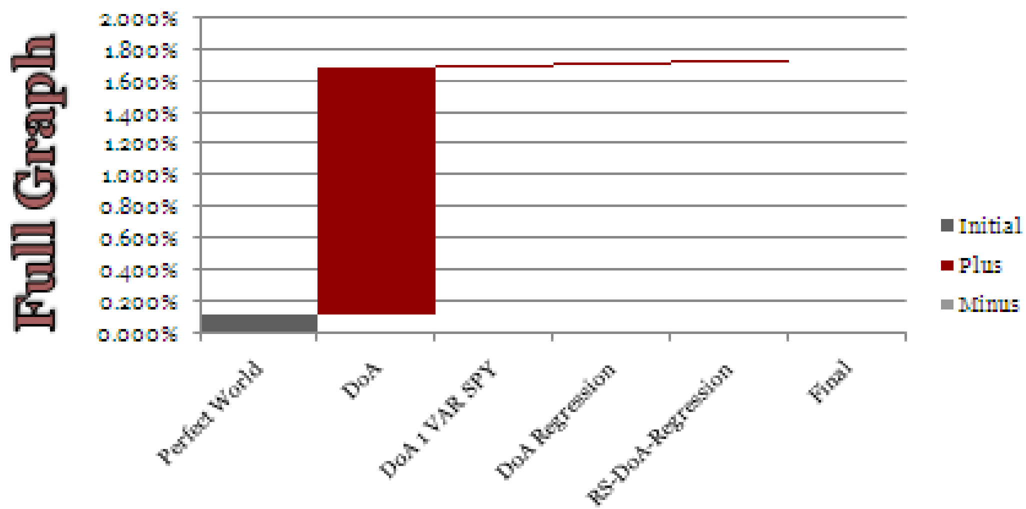
Long term returns often have period of higher growth as well as periods of lower returns. We included this tendency in our model by using a two regime-switching model for the returns of the SPY, RUS and MFI. In order to estimate the high and low returns, we analyzed historical data for the two indicies. Using this data, we also calculated the maximum likelihood estimators for the probabilities of switching between regimes each month or remaining in the same regime.

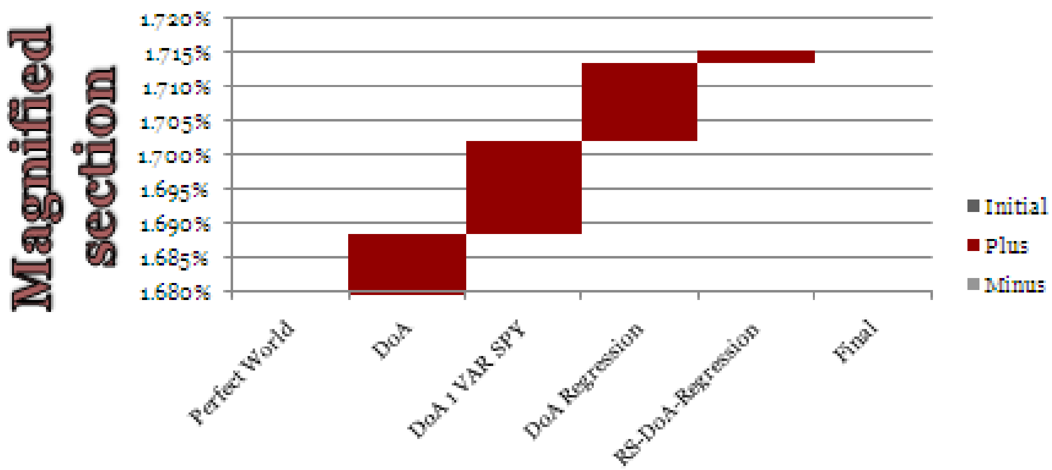
Regime-Switching Parameters

	SPY Parameters	RUS Parameters
$p_{1,2}$ (Probability of switching from Regime 2 to Regime 1)	5.24%	5.24%
$p_{2,1}$ (Probability of switching from Regime 1 to Regime 2)	10.75%	10.75%
μ_1 (Mean Montly Return for Regime 1)	1.31%	1.22%
σ_1 (Montly Standard Deviation for Regime 1)	2.82%	4.56%
μ_2 (Mean Montly Return for Regime 2)	-1.17%	-2.05%
σ_2 (Montly Standard Deviation for Regime 2)	7.19%	10.72%

Notice that the probabilities of switchin are the same for the two indices. We combined the historical returns to create the probability matrix which was the most appropriate for both the SPY and RUS in order for the regimes to switch at the same times in the Monte Carlo simulation.

Using the results from studying the behavior of alpha and the Regime-Switching parameters, we applied all combinations of the modficiations in our model and computed the resulting changes in tracking error. The largest effect was from the change in parameters for the distrubition of alpha. This is because of the large volatility associated with the new parameters; the volatility is about 7.8% annually when the distribution of alpha is used instead of 0.5% volatility without it. Due to this large change in volatility, it was expected that the manager's alpha component of our simulation would cause the biggest change in tracking error. The other two additions, regime-switching and regression, had minor effects on the tracking error. The percent change when introducing these two processes was less than 1%, therefore showing that the fund followed the proxy closely despite introducing a regression or regime-switching model. Our group also experimented by introducing each of the complexities in different orders, but found that this had no significant impact on the percent change of tracking error after adding each complexity.





Overall, the simulation showed that the simulated mutual fund tends to follow the proxy closely despite the complexities that were added. Although we know this is not the case for most real life funds, it is an interesting result and leads to further experiments that could be performed using our model.

Table of Contents

Table of Figures

1. Introduction

Variable annuities are becoming more prevalent in today's market. Due to the current economic state, investors are looking to invest in safer areas and variable annuities offer that safe investment vehicle. The insurers create a portfolio with two separate accounts; one is like a mutual fund and the other is a hedging program. In addition to splitting the funds, variable annuities (as discussed in Section 2.3) also offer the investors benefit options that act as insurance. These benefits include guaranteed minimum withdrawal, income, accumulation, and death benefits. The benefits offer protection to the policy holder, but not to the insurance company. To protect themselves, insurance companies use hedging strategies, which are designed to protect the company in the event of unfavorable market changes

The construction of these hedging portfolios requires managers to use derivative instruments which protect them against losses caused by interest rate changes, realized volatility, and implied volatility costs (Stulz, 1985). The actual investment is comprised of actively managed mutual funds, which do not always have a clear benchmark. To create a benchmark, managers compare the mutual fund to indices, which is known as fund mapping. Fund mapping is used to project the expected mutual fund returns based on the performance of the benchmark.

Due to the complexity of variable annuities, and their increasing demand, it is beneficial to a company if the insurers can predict how the annuities should perform. The intent of this project is to quantify the effect of different simulation techniques on the deviation of variable annuities from their benchmark. This will help the sellers to determine how to price and market variable annuities and will help both the seller and buyer to judge the success of the variable annuity against a benchmark.

In order to develop this benchmark, a simulated mutual fund was created, which was designed to follow the returns of the S&P 500 and Russell 2000 with a level of random error. Monte-Carlo simulations were performed using the Black-Scholes Model to project our mutual fund over a 20 year period. We began with a “perfect world” scenario, where the mutual fund directly followed the S&P 500. Then we implemented different processes into our model such as, regression modeling, regime switching modeling, and a distribution for manager’s alpha. This allowed us to analyze different sources of tracking error and basis risk. With our results we offer the ability to minimize tracking error by identifying its cause.

2. Background

In this section, we present and discuss information regarding various subjects to help the reader fully understand the process and goal of our project. As our project is a continuation of a project completed in 2009, we will begin by providing a brief summary of that project. We will then offer an overview of our sponsor and a simplified description of what a variable annuity consists of. From there we will move on to discuss tracking error, exchange traded funds, and stochastic processes. Lastly, we introduce our Visual Basic for Applications (VBA) tool that we used to complete our project.

2.1 2009's Major Qualifying Project

In the 2009 Major Qualifying Project (MQP), "Analysis of Fund Mapping Techniques for Variable Annuities," the MQP group used a VBA coded macro in Excel to simulate a mutual fund to test the accuracy of fund mapping. In their project, they offered an overview of mutual funds and hedging techniques, as well as information regarding models for equity returns and regression analysis.

2.1.1 Mutual Funds and Hedging Techniques

"A mutual fund is a diversified portfolio funded by various investors, with assets such as stocks, bonds, short-term money market instruments or securities (Mutual Funds, 2007). The investors purchase shares of the mutual fund, which are redeemable, meaning that they can sell the shares back to the fund at any time. When purchasing and selling funds, investors use four main sources: professional financial advisers, employer sponsored retirement plans, fund companies, and fund supermarkets. When individuals are just beginning to invest, most depend on financial advisers to make intelligent decisions for them in order to earn a good return on their investments" (Fredricks, Ingalls, & McAlister, 2010).

There are various types of funds, which can be composed of any of the following;

- **Corporate Bonds:** Bonds issued by corporations, which pay higher rates due to their riskiness (Corporate Bonds, 2009).
- **Growth Funds:** Growth of capital is the main objective of this type of fund, which is composed mostly of common stocks. These funds can be either conservative (investing in large cap) or aggressive (investing in small cap).
- **Sector Funds:** Invest in companies in a specific geographic area or industry.
- **Income Funds:** Provide investors with a high yield by investing in stocks and bonds that make dividend payments to shareholders.
- **Balanced Funds:** Have a conservative investment policy invested in common stock, preferred stock and bonds.
- **US Government Securities Funds:** Invest in securities offered by the U.S. government, such as Treasury bills, notes, and bonds.
- **Money Market Funds:** These investors have a high return and high liquidity, which are high-yield short-term debt securities.
- **International Funds:** Invest in common stocks of foreign countries.

(Finance, 2005) (Fredricks, Ingalls, & McAlister, 2010)

"After the client has chosen their mutual fund contract, the next step is to send the contract to the insurance company. When the insurance company receives this contract, they try to hedge against the funds based on the riskiness of the investments they chose. Insurance companies use many hedging techniques in order to provide them with confidence about the risk that may arise. Furthermore, there are also hedge funds that are used by many investors. Insurance companies hedging for variable annuities do not use hedge funds due to their risky nature; however, typical techniques for hedging can be seen in hedge funds" (Fredricks, Ingalls, & McAlister, 2010).

Hedging funds are ways for managers to reduce risks taken when investing. Managers are constantly trying to develop a trade-off between the risks and rewards. When an investor reduces the risk by investing in an offsetting investment, they are hedging their risk (Chriss, 1996).

One method used for hedging techniques is the use of the Black-Sholes model. If the weights on each investment are kept balanced, then the value of the option and the portfolio are always equal (Chriss, 1996). Managers use this method when hedging variable annuities in order to provide for the benefit chosen by the insured party.

Variable annuities are usually split between two accounts. One account, the variable account, is invested in an actively managed mutual fund. The other account is created to minimize risk, therefore the manager creates an account that will short the market, or increase as the market decreases. Hedge funds work the much like mutual funds, where a diversified portfolio is created using pooled funds. However, unlike mutual fund managers, a hedge fund manager's main objective is reducing the risk within the portfolio. This allows the managers to have much more flexibility when choosing their investments (Wolfinger, 2005).

(Fredricks, Ingalls, & McAlister, 2010)

2.1.2 Models for Equity Returns

The 2009 MQP group considered the method they used for modeling for equity returns as a Monte-Carlo method. “The Monte-Carlo method consists of observing a large number of possible random outcomes in order to predict what the true result or outcome will be” (Fredricks, Ingalls, & McAlister, 2010).

They used their VBA macro in excel to produce simulated funds, each one a scenario representing a possible ‘world’ with future stock prices. These worlds are used to aid in risk management. In generating many scenarios, the distribution of the prices is expected to converge on the actual distribution (M. Crouhy, 2001).

There were two mains steps involved in their Monte-Carlo method. “The first step [involved] finding the risk factors and estimating parameters such as volatility and correlation using the historical data of the stock prices and returns (M. Crouhy, 2001). The second step [was] to use geometric Brownian motion to construct price paths, which [were] created using a normal random number generator” (Fredricks, Ingalls, & McAlister, 2010).

2.1.2.1 Black-Scholes Model for Estimating Volatility

When modeling a single stock or index, the manager must know some historical information about the stock or index. Particularly, the manager has to be able to estimate both the drift rate and the volatility. “The drift rate is the expected return of a stock in a given time period. If a stock price increases 6% per year on average, then the drift rate is 6%. Volatility is a measure of uncertainty about the returns of a stock or index (Hull J. C., 2006)” (Fredricks, Ingalls, & McAlister, 2010). Since volatility depends on the amount of variance over a continuous interval rather than over a set beginning and ending time, it is more difficult to estimate.

Given a set of stock prices from S_0 to S_n, the equation used for estimating volatility from historical data is

$$\sigma = \frac{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}((u_i - \bar{u})^2)}}{\sqrt{\tau}} = \frac{SD(u_i)}{\sqrt{\tau}}$$

Where $u_i = \ln\left(\frac{S_i}{S_{i-1}}\right)$, $\bar{u} = \frac{\sum_{i=1}^{n} u_i}{n}$, $n = number\ of\ observations$, and

$\tau = length\ of\ time\ between\ S_i$ and S_{i+1} in years (Hull J. C., 2009).

In other words, “the estimate of the volatility is the standard deviation of all log returns divided by the square root of the amount of time between each observation (in years)” (Fredricks, Ingalls, & McAlister, 2010). The future stock prices can be simulated once the annual volatility and drift rate are estimated.

2.1.2.2 Geometric Brownian Motion

Geometric Brownian motion is a method used to simulate stock prices over time. (M. Crouhy, 2001). When given the expected return (μ), the annual volatility (σ), the initial stock price (S_t), and a standard normal random number (ε, such that $\varepsilon \sim N(0,1)$), the stock price at time $t + \Delta t$ can be calculated using the following equation,

$$S_{t+\Delta t} = S_t exp\left(\left(\mu - \frac{\sigma^2}{2}\right)\Delta t + \sigma\varepsilon\sqrt{\Delta t}\right).$$

Which is the same as saying $S_{t+\Delta t}$ is lognormal, or

$$\ln(S_{t+\Delta t}) \sim N\left(\ln(S_t) + \left(\mu - \frac{\sigma^2}{2}\right)t, \sigma^2 t\right)$$

(Hull J. C., 2009).

This process is used repeatedly to simulate stock prices at various times from S_t and continuing overΔt. Each time $S_{t+\Delta t}$ becomes the newS_t, a new ε is found, and a new $S_{t+\Delta t}$ is calculated (Fredricks, Ingalls, & McAlister, 2010).

2.1.3 Regression Analysis

Once the Monte-Carlo simulation is completed, the mutual fund is regressed against the chosen market indices to show which market index best elucidates the movement of the mutual fund. The regression analysis helps to portray the relationship between two or more variables. "Linear regressions find the relationship between a dependent variable and an independent variable, whereas multiple regressions have a dependent variable and two or more independent variables (Multiple Regression , 2008)" (Fredricks, Ingalls, & McAlister, 2010).

When determining the relationship between the variables, it does not mean that one variable produces the other, but there is some significant association between the variables (Linear Regression, 1997-1998). Therefore we can see that regressions establish correlation, but not causation.

A valuable numerical measure of the relation between two variables is the correlation coefficient, or β which indicates the strength of the association (Linear Regression, 1997-1998). "When β equals one, it shows that the variables are perfectly correlated and will move along the same path. When β is negative it means that the variables move in opposite directions (Levinson, 2005). When saying that the variables move in opposite directions, it means that when one variable increases, the other is expected to decrease. Furthermore, when β equals zero, it means that there is no relationship between the two variables" (Fredricks, Ingalls, & McAlister, 2010). Usually when analyzing stock regressions, the independent variable represents the market and the dependent variable represents the stock.

2.2 Towers-Watson

Towers-Watson is a professional services and consulting company that is a product of a recent merger between Towers Perrin and Wyatt Watson in January of 2010. The company offers benefits consulting to its clients such as retirement plans, health and group insurance, and technology and administration solutions. Their other main offerings are risk and financial consulting in insurance, investments, and risk management. See their mission statement below.

> *"Towers Watson is a leading global professional services company that helps organizations improve performance through effective people, risk and financial management. With 14,000 associates around the world, we offer solutions in the areas of employee benefits, talent management, rewards, and risk and capital management"* (towerswatson.com, 2010).

2.3 Variable Annuities

A variable annuity is a long-term investment vehicle, under which a given insurance company agrees to make periodic payments to an insured client, based on the performance of the insured's initial investment in a mutual fund. Variable annuities are similar to mutual funds in that they both invest in a combination of stocks, bonds, and money markets (Variable Annuites: What you should know, 2009). However, variable annuities are different than mutual funds because they provide periodic payments, a death benefit, and they are tax-deferrable among other advantages.

Typically, an insured will allocate a certain percentage of their purchase payment to multiple investment options, and let their fund accumulate. For example, 30% of the purchase payments may be invested in bonds and the remaining 70% may be invested in the stock market. The percentage return that an insured gains on their variable annuity depends on the performance of their overall investment. Those who invest in variable annuities are provided with information on their options of investment, which include past performances and overall risk and volatility of the fund (Variable Annuites: What you should know, 2009). After a designated period of time, the fund will begin its payout phase, in which the insured will receive either a lump-sum payment, or a stream of periodic payments.

Variable annuities are considered an insurance product because they can provide a death benefit, or a minimum payment option. If an insured client dies, a beneficiary will receive the greater of the account value of the investment or some guaranteed minimum. The insurance aspect of variable annuities makes them more attractive than other investments because the risk is minimized. Some variable annuities offer a stepped up death benefit, which can be increased at an agreed upon future date if the account value is greater than the original benefit (Variable Annuites: What you should know, 2009). In doing this, insurance companies will have a smaller loss if the investment does not perform well, and clients benefit in case of a spike in their investment's performance, followed by a decline.

2.4 Exchange-Traded Fund (ETF)

Exchange-traded funds (ETFs) are investment funds that are traded on stock exchanges. An ETF is comprised of assets, such as, stocks, commodities, or bonds, much like an index fund. However, since ETFs are traded like stocks, they do not have a net asset value (NAV) calculated daily. This distinguishes ETFs from regular Mutual Funds. An ETF combines the valuation feature of a mutual fund with the trade feature of a close-end fund. This allows for the owner to have the diversification as well as the ability to sell short, buy on margin and purchase as little as one share. Another benefit of ETFs is that most of them track an index like one of the most widely known ETFs, the Spider (SPDR), which tracks the S&P 500 index and trades under the symbol SPY (Investopedia, 2010).

2.5 Tracking Error

2.5.1 Benchmark

Those who invest in Exchange-Traded Funds (ETF) look to “buy the benchmark” of the ETF before they purchase it. A benchmark is essentially an approximation of how much the fund is predicted to grow over a certain period of time. For instance, if the benchmark is up to 10 percent, a $50 investment will become a $55 investment. However, benchmarks are rarely on-point accurate because of tracking error; this is why benchmarks are merely approximations. Tracking error is defined as the difference between the performances of a fund and the performance of its underlying index (Tracking Error In Exchange Traded Funds, 2007). A broader definition is the difference between absolute returns and the indexes benchmark. Generally, all investors want to see as little tracking error as possible, because when they initially purchase an ETF, they are buying it because of the gains that are advertised by the benchmark.

One clear-cut cause of negative tracking error are the fees that are associated with purchasing an ETF. More specifically, managers will charge a client a fee, which is taken directly out of the clients net returns on their investment. So, in most situations, these fees will contribute to tracking error (Tracking Error In Exchange Traded Funds, 2007).

Tracking error is also caused by optimization of an ETF, which is done to try to mimic the index portfolio. Some ETFs will purchase the same stocks at the same weights as the index, which will greatly minimize tracking error, but will raise the costs of trading. Other funds use "optimization techniques," which are essentially buying a subset of the index's stocks in the belief that they will provide similar performance to the full portfolio, at a lower cost to trade (Tracking Error In Exchange Traded Funds, 2007). Trading costs are calculated by finding the percentage difference of the price of the stock before the trade, and the total cost by the purchaser after the trade. Obviously, you want your cost of trade to be as close to zero as possible. The level of optimization can be a contributor these trading costs which, in turn, contribute to tracking error (Tracking Error In Exchange Traded Funds, 2007).

Another form of tracking error can be tied to diversification requirements, enforced by the Securities Exchange Commission. They have two clear cut rules:

- No single security can be more than 25 percent of the portfolio
- Securities with more than a 5 percent share can't make up more than 50 percent of the fund

These rules can keep ETF managers from utilizing their optimization techniques, and stop them from mimicking the index. Since the managers are unable to mimic the index, which generally decreases the amount of tracking error seen, these rules can also contribute to the overall tracking error of an ETF (Tracking Error In Exchange Traded Funds, 2007).

In order to calculate tracking error, the following equation is used:

$$TE = SD\ (r_p - r_b) = \sqrt{VAR\ (r_p - r_b)}$$

Where r_p is equal to the portfolio return, r_b is equal to the fund mapping return and SD is equal to standard deviation. In order to make the tracking error period, the equation would be:

$$TE_n = TE * \sqrt{n}$$

Where n is equal to the number of periods that are being considered. For instance, if you are annualizing the Tracking Error, you would use $\sqrt{12}$ (Fredricks, Ingalls, & McAlister, 2010).

2.5.2 Basis Risk

Hedging can be used to manage the risk of investments, but no strategy can eliminate risk all together. There are many reasons why actual and predicted experience can diverge, and this difference is referred to as basis risk or spread risk. A basis is defined as following;

$\boldsymbol{Basis = Spot\ price\ of\ the\ Asset\ to\ be\ hedged - Future\ price\ of\ the\ contract\ used,}$

where the future price is the "market determined price of the asset at a certain date in the future" and the spot price is the expected price of the asset (Hull J. C., 2009). The basis will be zero when the actual and estimated prices are equal, but often differs because of the uncertainty in predicting a fund’s performance.

An increase or strengthening of the basis can happen because of "interest costs, storage costs, [or] positive handling and transportation costs between the location and the futures delivery point" (Benhamou). Decreases in the basis, or a weakening of the basis risk occurs due to "shortage of local supply on the spot market, positive dividends paid by the underlying asset of the futures contract, [or] known positive cash flows generated by the underlying asset of the futures contract" (Benhamou).

When trading for an insurance company, managers attempt to match the underlying assets directly to the separate account mutual funds, but perfection is impossible. The difference in the estimation of the fund mappings can be attributed to basis risk (Fredricks, Ingalls, & McAlister, 2010). Basis risk can also occur if the fund managers change their investment portfolio in a way that does not match the assets. The manager may also attempt to exceed the returns of the underlying asset, or beat the benchmark, and doing so may be another source of basis risk. Overall the basis risk can be described as the "deviation between the benchmark and the performance the manager is expecting (Fredricks, Ingalls, & McAlister, 2010)."

2.6 Stochastic Processes

"A stochastic process is a family of random variables $\{X(t)|t \in T\}$ defined on a given probability space, indexed by time variable t, where t varies over an index set T" (Trivedi, 2002). Stochastic processes assign a sample function $x(t, s)$ to each outcome s (Trivedi, 2002). In the context of our project, stochastic processes like those described in the following two subsections, will be applied to our model to create a more realistic model for our simulated funds.

2.6.1 Time Series

For investors, time series are the progression of an asset's price over given intervals of time, usually daily, weekly or annually. When modeling future returns, many experts use "regime switching time series models, which are models that allow parameters of the conditional mean and variance to vary according to some finite-valued stochastic process with states or regimes" (Lange & Rahbek, 2009). There are many different types of regime switching models, including Markov, Bayesian or observation models, but we found the most applicable for the SPY and RUS is the Long-Term Stock Return Regime-Switching Model by Mary Hardy.

2.6.2 Regime-Switching Model of Long-Term Stock Returns

Modeling stock returns over a long term could be done using a Black-Scholes approach or a Regime-Switching model. Typically, it is assumed that stock returns follow a lognormal distribution where the stock price at time t + Δt equals $S_{t+\Delta t} = S_t\, e^{\left(\mu - \frac{\sigma^2}{2}\right)\Delta t + \sigma\varepsilon\sqrt{\Delta t}}$ with "drift rate (expected return) μ, annual volatility σ, initial stock price S_t, and a standard normal random number ε ($\varepsilon \sim N(0,1)$)" (Fredricks, Ingalls, & McAlister, 2010). This model is appropriate for modeling short amounts of time, but does not allow for large deviations in price movements in the long run (Hardy M. R., 2001).

The regime-switching model is more appropriate for modeling long-term stock returns because it "more accurately captures the more extreme observed behavior" (Hardy M. R., 2001). The model assumes that the volatility of the stock can be one of K discrete values, and switches between these values randomly and independent of previous behaviors. A two-regime model (K=2) is most often used since the added complexity of higher regime models does not add a great deal of accuracy. For each of the 2 regimes, denoted ρ_t where at each time t, ρ_t can either equal 1 or 2, the stock returns still follow a normal distribution but with different means and standard deviations. Two-regime models include a four by four transition matrix, depicted below, which displays the probabilities of switching regimes or staying within the same regime (Hardy M. R., 2001).

$$Transition\ Matrix = \begin{bmatrix} p_{1,1} = \Pr[\rho_{t+1} = 1|\rho_t = 1] & p_{1,2} = \Pr[\rho_{t+1} = 2|\rho_t = 1] \\ p_{2,1} = \Pr[\rho_{t+1} = 1|\rho_t = 2] & p_{2,2} = \Pr[\rho_{t+1} = 2|\rho_t = 2] \end{bmatrix}$$

These probabilities, as well as the parameters for the normal distributions under each of the two regimes are estimated using maximum likelihood functions (Hardy M. R., 2001).

2.7 Monte Carlo Simulation

In order to generate the future-past, the tool utilizes Monte Carlo simulation. Monte Carlo simulation is used to estimate a distribution by generating a large number of scenarios. Those scenarios are ordered and the empirical distribution that is found is assumed to be the true distribution (Hardy M. R., 2006). Monte Carlo simulation is often used in situations that are too complex to solve analytically and situations that have significant uncertainty. As such, Monte Carlo methods are stochastic techniques that utilize random numbers and probability to investigate possible outcomes (Woller, 1996).

Monte Carlo methods apply to the estimation of drift rate (expected return) mostly because of the randomness associated with stock returns. Many factors involved in the stock market are very difficult to predict, such as the state of the economy and large mergers. Because of these factors, it is very difficult to calculate a single, accurate estimation of the return of a mutual fund over ten years. The estimation of drift rate is also a very complex calculation. There are many factors that need to be incorporated into the calculation of stock returns that make it very difficult to be solved analytically, especially when trying to estimate 10 years of returns. So, instead of trying to calculate a single estimation of how the market and mutual fund *will* act, the tool generates fifty scenarios of how the market and mutual fund *could* act. Then, a distribution is formed, with a random element, to estimate what the market will do based on the fifty scenarios.

3. Methodology

This chapter discusses the procedures we used in order to achieve our goals and objectives. The first section explains the tool we used to generate our data accompanied by subsections discussing how it was created and how it works. The next section goes into more detail regarding the steps taken to obtain our output and the significance of each piece of data. The final section in this chapter provides a look at the VBA code used in the tool and the spreadsheet constructed by the tool.

3.1 Overview of Monte Carlo Simulation

The tool we created uses Monte Carlo simulations to produce simulated funds for SPY, RUS, and our mutual fund, MFI. We called the first part of our simulation the Future-Past (FP). "First, we simulated the SPY and RUS using geometric Brownian motion for ten years. Then, we generated a MFI that followed the SPY simulation with some amount of random error to represent manager alpha" (Fredricks, Ingalls, & McAlister, 2010). The MFI we created used the SPY as its perfect fund mapping. This task was done repeatedly in order to generate a large number of FPs.

The next part of our simulation was called the Future-Future (FF). In this step we performed more stock market simulations, continuing from the end of each FP. Since we wanted a large amount of data to increase credibility, we created 30 FFs for each FP. This gave us a large set of unique FFs for every FP.

3.1.1 Future Past

Before using the tool, the user must input certain factors into to inputs sheet. For the future-past, the user may specify how many FPs to run and how many years each FP will be. Both of these must be whole numbers. The number of FPs is called the number of scenarios, since each FP will contain an entire scenario including FP, regressions, and FPs. For our simulations, we used 30 FPs of ten years each.

"The user may also specify how often the index and MFI values are calculated in the FP, as well as how often they are displayed. Both options may [be specified as] either daily or monthly" (Fredricks, Ingalls, & McAlister, 2010). Clearly, if the values are calculated monthly, then they cannot be displayed daily because that data does not exist. Lastly, the user may choose whether to show the values of the index or the returns during the FP. The simulation is still based on geometric Brownian motion using logarithmic returns, no matter which of these options is chosen. For our simulations, we calculated daily data and displayed monthly data in the FP, and printed the returns for our MFI, RUS and SPY.

Another component in the Excel spreadsheet is the Δt value, which is the proportion of a year between each period in the FP. This is automatically updated by the program based on how often the values in the FP are calculated, and does not require user specification.

3.1.2 Future-Future

Just like in the FP section, the variables in the FF are similar to those in the FP. The user inputs the number of years in each FF, how many FFs to create for each FP, how often to calculate the values in the FF, and how often to display the values in the FF. Again, Δt is calculated based on how often the values in the FF are calculated. The parameters themselves, while similar to those in FP, are completely independent of those in the FP. It does not matter what was chosen in the FP, a user may choose something completely different for the FF.

For our runs, we used the same parameters for the FF as were used in the FP. We simulated 50 FFs for each FP, and the number of years in each FF was 10. The values were calculated daily, but only displayed monthly. We only displayed monthly returns to save space in our spreadsheet. Lastly, we decided to have the Macro print out information all at once rather than running a screen refresh while the macro was running to cut down on processing time.

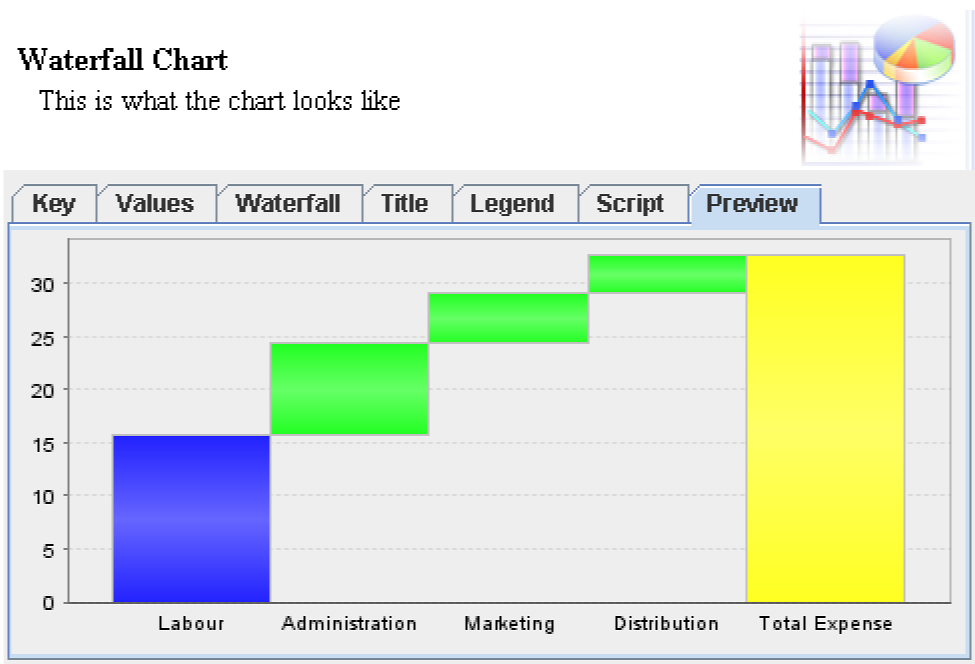
3.1.3 Generation of Random Numbers

The FP and FF in our project were calculated in a very similar way to those of the 2009 projects. One modification we made to the simulation was the random number element. In the 2009 project a random number was added to the general formula using the random number generator in Excel to account for a small amount of differentiation in returns. In our project, however, we generated a spreadsheet of random numbers using Excel's random number generator to account for the random noise in our returns. As we are tracking the change in deviation from the benchmark, we wanted to be able to test specific changes that we made to the formula. While this differentiation in returns is important within a scenario, we did not want it to affect separate scenarios. To rectify this differentiation, we decided to create a sheet of random numbers that will list numbers for each scenario between 0 and 1. These numbers are utilized sequentially in the calculation of each return, where the order in which the numbers are pulled remains the same for each scenario. This will eliminate any deviation in the random numbers between the scenarios, since that would skew our results.

3.2 Composition of Waterfall

When deciding how to display our results, our group settled on using a waterfall graph, an example of which can be seen in Figure 1. This graphic is a good way of showing how an initial value is affected by a series of intermediate positive or negative values. The initial and the final values are represented by whole columns, while the transitional values are represented by floating columns. The graph displays a gradual change from the initial value to the ending value, which is what we are trying to determine in our calculations. Our waterfall graph is comprised of 4 main components; The "Perfect World" scenario, regressions, a simulation where alpha has a predetermined distribution and a simulation with regime switching turned on.

Figure 1 : Waterfall Graph (elixirtech.com)



3.2.1 The Perfect World Scenario

The first component of our waterfall is the "perfect world" component. In this scenario the MFI follows the SPY almost completely in the FP. There is little to no deviation from the SPY other than the noise generated from the random numbers sheet. This perfect world scenario acts our initial value in which to base all of our deviations off of, and is therefore the first column in our waterfall graph.

3.2.2 Calculating 1-Var and 2-Var Regression and Testing Correlation

Following the Perfect World scenario, the Monte Carlo simulator next creates a specified number of FF scenarios based off of each FP. These calculation are different from the Perfect World scenario because they regression to find the mapping weights of each index. Using the FP, the most accurate weighting of the SPY and RUS are used for each return of the MFI for the FF. The weighting used for the mutual fund in the FP was 100% SPY and 0% RUS.

The display in the waterfall distinguishes between the one variable regression (1-Var) and the two variable regressions (2-Var or just Regression). The 1-Var regression denotes the relationship between the SPY and the mutual fund, whereas the 2-Var shows the relationship between the two indices and the mutual fund. The standard deviation of the change in returns is calculated differently for the 1-Var and Regression. For the 1-Var, the result is the standard deviation across all of the periods of the one variable SPY minus the mutual fund. The Regression result is calculated by finding the standard deviation of the two variable SPY return minus the mutual fund return for each period.

Our group tested the effect of the correlation between the SPY and RUS on the results from the simulator. The first simulations set the correlation between the SPY and RUS to be approximately .89. This was based on the calculated correlation from the past data for the two indices. After those simulations, we ran the tool with a correlation of .5 between the two indices to see if it would have any impact on the tracking error results. Our expectation was that the change in correlation may increase the tracking error for the 2-VAR regression because it is based on both indices. Any change in the relationship between these might decrease the accuracy of the regression.

3.2.3 Distribution of Alpha

The Distribution of Alpha is the next component in our simulations. It introduces new parameters for alpha based on a distribution. This distribution is created using historical data from actively managed mutual funds. After comparing data from both large cap mutual funds and actively managed ETFs we decided to use the mutual fund data.

3.2.3.1 ETF - Powershares Active AlphaQ (PQY)

The investment objective of the Powershares Active AlphaQ fund is long-term capital appreciation, through investing at least 95% of its total assets in Nasdaq-listed stocks, which makes the Nasdaq 100 the funds benchmark. The reason why this fund is considered actively managed is because the Nasdaq stocks are screened weekly by fund advisors. The stocks are tracked and rated by the advisors and, a “Master Stock List” is generated, which ranks roughly 3,000 different stocks of companies with market capitalization of over $400 Million that are traded within the United States. The 3000 stocks are then narrowed down to the 100 largest, and then the fund will generally select approximately 50 stocks that are included in the list. Powershares Active AlphaQ is also not an index fund; therefore it doesn’t necessarily look to replicate the index that it is following.

3.2.3.2 Large Cap Mutual Funds

Large Cap Mutual Funds are those mutual funds, which seek capital appreciation by investing primarily in stocks of large companies with above-average prospects for earnings growth. These companies usually have a market capitalization value of more than $10 billion. “Large cap is an abbreviation of the term "large market capitalization". Market capitalization is calculated by multiplying the number of a company's shares outstanding by its stock price per share” (Investopedia, 2010).

3.2.3.2.1 American Funds Growth Fund of America (AGTHX)

Net Assets*:	**151.28B**

(Yahoo Finance)

This fund invests primarily in the common stocks of companies that seem to offer a better opportunity for growth, which is self-explanatory. This fund is managed by a group of portfolio counselors, where the portfolio of the fund is divided into individual segments. Each counselor will individually decide how their respective segments are invested. (American Funds) The success of the fund is dependent on the profession judgment of its advisor, who oversees the individual portfolio counselors. The adviser has a very simple investment strategy; to make long-term investments in attractively valued companies. The advisor will analyze potential companies, which may include meetings with company executives, employees, customers and the company’s competition. If the investment begins to decline in return, the securities will be sold by the adviser. (American Funds)

3.2.3.2.2 Capital World Growth and Income Fund (CWGIX)

Net Assets*:	78.81B

(Yahoo Finance)

This fund looks to invest in common stocks that have the potential to pay dividends and are denominated in U.S dollars or other currencies. Under normal market conditions, the fund will look to invest a large portion of its assets in securities of companies residing outside of the United States. The fund tends to invest in stocks that the adviser believes are relatively stable during declines in the market. The adviser and counselors for the fund all have the same responsibilities and strategies as the fund that is listed above. (American Funds)

3.2.3.2.3 Vanguard Total International Stock Index Fund Investor Shares (VGTSX)

Net Assets*:	39.44B

(Yahoo Finance)

This fund invests most of its assets in the common stocks included in the funds target index, which is the Emerging Markets Index. The Emerging Markets Index includes approximately 1,700 stocks of companies located in 43 different countries. (Vanguard)

3.2.3.2.4 Vanguard Institutional Index mutual fund (VINIX)

Net Assets*:	80.40B

(Yahoo Finance)

The funds investment strategy is to track the S&P 500. The fund looks to replicate the benchmark index by investing nearly all of its assets in the stocks that the index is composed of, at approximately the same weights. (Vanguard Institutional)

3.2.3.2.5 American Funds EuroPacific Growth Fund (AEPGX)

Net Assets*:	**103.22B**

(Yahoo Finance)

The fund looks to invest primarily in common stocks of issuers in Europe and the Pacific that the adviser believes has growth potential. The only difference between this fund and the AGTHX is where the fact that this fund mainly invests abroad. Other than that, the fund adviser and counselors have the same responsibilities and strategies. (American Funds)

3.1.3.3 Developing distributions from ETFs and Mutual Funds

In order to more accurately include the manager's alpha component of the MFI in our model, our group created distributions using either daily or monthly returns from ETFs and mutual funds. Our trials began using daily data for an ETF, followed by daily data for a collection of mutual funds and finally monthly data for the collection of mutual funds.

We started by used the data from the aforementioned ETF and mutual funds to develop a distribution. We compiled the daily returns for the ETF, mutual funds, and their benchmarks and found the difference between the fund and the benchmark to estimate the manager's alpha. There were many days in which the return of the ETF did not change, meaning the manager had not updated the portfolio. Because of this, we chose to exclude any of the days which had a 0% return for the ETF. Also, we excluded the one day following any 0% return in the ETF since this change captures any differences made in the prior two days and would skew the data. We also did a five number summary (Petrucelli, Nandram, & Chen, 1999, p. 60) to find which of the data points could be considered outliers, and excluded these points in our analysis.

We compared the probability distributions from the daily and monthly returns in order to pick the best distribution for our model.

3.1.3.4 Distribution of Alpha in Monte Carlo Simulator

In our Monte Carlo simulator, the mutual fund was designed with the S&P 500 as the benchmark. Therefore, the return for the mutual fund mimics the S&P 500 throughout the simulation with a slight error component. The distribution of manager's alpha that we created from the Power Shares Active Mega Cap ETF is what makes up this error component. At each interval throughout the simulation, the return for the S&P 500 is taken from a normal distribution with given parameters. The return for the mutual fund is calculating by the taking the S&P 500 return and adding a number from a specified normal distribution to that return. This manager's alpha accounts for two things throughout the simulation. For one, it accounts for the inability of a fund manager to perfectly match a benchmark, and it also accounts for the manager attempting to outperform the benchmark.

3.2.4 Monthly Regime Switching Model

Another addition to our modeling process was including a two regime switching model into the predicted returns. Research shows that the two regime model is appropriate for monthly returns. Weekly returns are slightly more accurate with a three regime model, while quarterly returns show no improvement when additional regimes are added (Hardy M. R., 2001). Therefore, our project chose to analyze the returns daily, but switch regimes monthly. This will allow the modeled returns to have periods of higher returns as well as lower returns following the past experience.

3.2.4.1 Parameter Estimation

In order to accurately introduce this variability into the model, we first had to estimate the parameters for the low and high regimes, as well as the transition matrix which described the probability of switching regimes or staying in the same regime. Our group created a spreadsheet which calculated the likelihood function for the log returns, and then estimated the parameters to maximize this function.

As mentioned previously, the data which was used to estimate our regime-switching model parameters was the monthly returns. Therefore, the parameters which were produced were monthly volatilities and log returns.

The creation of the likelihood function had many components. The likelihood function itself was the product of each of the probability densities for each observation given the previous observations and the parameter set Θ (Hardy M. R., 2001). This can be written as

$$L(\Theta) = f(y_1|\Theta)f(y_2|\Theta, y_1)f(y_3|\Theta, y_1, y_2) \dots f(y_n|\Theta, y_1 \dots y_{n-1})$$

Each of these probabilities is the product of three components. The first component is the probability of transition which is taken from the probability matrix. Next is the probability density function for the normal distribution using the given observation in the given regime. The final component is the probably of being in the previous regime, which is calculated using the probability of being in a regime the past recursion over the total probability of the last recursion. Multiplying these three numbers together provides one of the terms for the likelihood function for one of the terms. This equation then needs to be repeated for each of the four combinations of $\rho_t = 1, 2$ and $\rho_{t-1} = 1, 2$ as seen in the following equation (Hardy M. R., 2001).

$$f(y_t|\Theta, y_1 \dots y_{t-1}) = \sum \Pr(\rho_t|\rho_{t-1}, \Theta) * f(y_t|\rho_t, \Theta) * \Pr(\rho_{t-1}|y_{t-1} \dots y_1, \Theta)$$

In order to maximize this likelihood function, it is easiest to take the natural logarithm of each term and add them all together. Our parameter estimation spreadsheet is seen in Figure 2 below. This spreadsheet broke up each of the components for each term, added them together, and then took the natural log of each. The sum of the natural logarithms is at the top of column N, and this is the value which should be maximized to get the most accurate parameters. The values highlighted in orange on the left side of the spreadsheet are the parameters which we are estimating. Using the excel add-in Solver, we maximized the sum of the log likelihood function changing these parameters using certain constraints. These constraints were that $0.001 \leq \rho_{1,2}, \rho_{2,1} \leq 0.999$ as well as $0.001 \leq \sigma_1, \sigma_2$.

Figure 2: Parameter Estimation Spreadsheet

	B	C	E	F	H	I	J	K	L	M	N
1											
2											166.7777338
3			Time	Log Return		based on π_1		based on π_2		$f(y_t\|y_{t-1} \dots y_1,\Theta)$	$\log f(y_t\|y_{t-1} \dots y_1,\Theta)$
4			1	0.007257	$f(y_1\|\Theta)$	8.43212791		2.69413471		11.12626262	1.046349307
5						$\rho_1=1, \rho_2=1$	$\rho_1=2, \rho_2=1$	$\rho_1=1, \rho_2=2$	$\rho_1=2, \rho_2=2$		
6	$p_{1,2}$	0.0179	2	0.000602	$f(y_2\|y_1, \Theta)$	10.2106175722	0.0762462739	0.0844604425	1.4731624245	11.8444867131	1.073516245
7	$p_{2,1}$	0.02295	3	0.024391	$f(y_3\|y_2, y_1, \Theta)$	11.9138951145	0.0421616490	0.0880454511	0.7277821578	12.7718843724	1.106254978
8	μ_1	0.01301	4	-0.016294	$f(y_4\|y_3, y_2, y_1, \Theta)$	7.4707919311	0.0119141089	0.1025820309	0.3821174597	7.9674055306	0.901316923
9	σ_1	0.02594	5	0.030011	$f(y_5\|y_4, y_3, y_2, y_1, \Theta)$	11.4429372587	0.0173235099	0.0912395932	0.3226356683	11.8741360302	1.07460202
10	μ_2	-0.00377	6	-0.009611	$f(y6\|y5, \dots y1, \Theta)$	9.9680409513	0.0084133562	0.1073634982	0.2116638936	10.2954816992	1.012646671
11	σ_2	0.06392	7	0.000585	$f(y7\|y6, \dots y1, \Theta)$	13.0513455622	0.0097541973	0.1079948450	0.1885254298	13.3576200344	1.125729086
12			8	0.033648	$f(y8\|y7, \dots y1, \Theta)$	10.7610906813	0.0057097282	0.0920303066	0.1140566666	10.9728873827	1.040320922
13	$p_{1,1}$	0.9821	9	-0.001415	$f(y9\|y8, \dots y1, \Theta)$	12.6988153051	0.0056808272	0.1095347129	0.1144539568	12.9284848019	1.111547629
14	$p_{2,2}$	0.97705	10	0.017406	$f(y10\|y9, \dots y1, \Theta)$	14.6313043297	0.0060288731	0.1039115572	0.1000109648	14.8412557248	1.171470648
15			11	-0.011758	$f(y11\|y10, \dots y1, \Theta)$	9.4444264758	0.0030751288	0.1093141963	0.0831371169	9.6399529178	0.984074913
16			12	0.007295	$f(y12\|y11, \dots y1, \Theta)$	14.4489574671	0.0068789937	0.1078496505	0.1199327536	14.6836188649	1.166833103
17			13	0.031644	$f(y13\|y12, \dots y1, \Theta)$	11.4874406686	0.0042304320	0.0943295111	0.0811408753	11.6671414870	1.066964465
18		MLE	14	-0.028295	$f(y14\|y13, \dots y1, \Theta)$	4.1879623487	0.0014945412	0.1022167808	0.0852036114	4.3768772821	0.64116437
19	$p_{1,2}$	0.0179	15	-0.054098	$f(y15\|y14, \dots y1, \Theta)$	0.5090632282	0.0005322496	0.0784209045	0.1915164675	0.7795328498	-0.108165579
20	$p_{2,1}$	0.02295	16	0.032980	$f(y16\|y15, \dots y1, \Theta)$	7.3410707612	0.0908827277	0.0619022125	1.7900236966	9.2838793980	0.96772949
21	μ_1	0.01301	17	0.013848	$f(y17\|y16, \dots y1, \Theta)$	12.0855555618	0.0703836422	0.0860921012	1.1711137794	13.4131450847	1.127530622
22	σ_1	0.02594	18	-0.025872	$f(y18\|y17, \dots y1, \Theta)$	4.4517825556	0.0107605959	0.0953599316	0.5383924661	5.0962955491	0.707254606
23	μ_2	-0.00377	19	0.034813	$f(y19\|y18, \dots y1, \Theta)$	9.2890338469	0.0308313275	0.0815270940	0.6320541802	10.0334464486	1.001450137
24	σ_2	0.06392	20	0.029059	$f(y20\|y19, \dots y1, \Theta)$	11.5854396795	0.0207315049	0.0909433773	0.3801194147	12.0772339764	1.08196748
25			21	-0.018269	$f(y21\|y20, \dots y1, \Theta)$	7.0169011389	0.0066560976	0.1046219121	0.2318075845	7.3599867331	0.866877031
26			22	0.019349	$f(y22\|y21, \dots y1, \Theta)$	13.9899699870	0.0156616753	0.0998529251	0.2611036183	14.3665882057	1.157353644
27			23	-0.039361	$f(y23\|y22, \dots y1, \Theta)$	1.9185871706	0.0011556285	0.0932589046	0.1312072060	2.1442089098	0.331267096
28			24	0.020562	$f(y24\|y23, \dots y1, \Theta)$	12.9620228677	0.0354213419	0.0930247918	0.5937745778	13.6842435792	1.136220796

RUS Data | SPY Data | RSLN RUS | RSLN SPY

3.2.4.2 Regime-Switching Model in Monte Carlo Simulator

The major adaptation that our group made to the Monte Carlo simulator that was already established was to integrate the usage of the Regime-Switching Model for the simulation of returns for the S&P 500, the Russell 200, and the mutual fund. In order to follow the Regime Switching Model, our simulation needed to have two "regimes" that represented two different distributions that the stock returns could follow. Because each regime is meant to represent an economic state, it was determined that both indices would be in the same regime at any given time. Also, because the mutual fund is meant to mimic the S&P 500, it follows the same regime pattern as the S&P 500.

It was determined by our group that the indices would have the ability to switch regimes on a monthly basis. This was because the parameters, as developed through Mary Hardy's directions, were monthly parameters, so adapting them to a different time interval might make them less valid. Also, when dealing with the possibility of changing "economic states", it seems highly unrealistic to have that possibly happen on a daily basis.

In the Monte Carlo simulator, a random number between 0 and 1 is generated at the beginning of the month. If the indices were previously in regime 1, then the number is compared to $p_{1,1}$, the probability of remaining in regime 1. If the random number is less than $p_{1,1}$ then the indices remain in regime 1 for the following month, and if the random number is greater than $p_{1,1}$ then the indices switch to regime 2 for the following month. If the indices were in regime 2 in the previous month, the same step is taken, but the random number is compared to $p_{2,2}$, the probability of remaining in regime 2. After the regime is determined, investment return calculations are done daily for that month, and then the process repeats.

4 Analysis and Discussion

In this section, we present and discuss our results for the various simulations and procedures we completed throughout our project. We first discuss our results pertaining to the determination of our distribution for alpha. We also display the data we found regarding our probability matrix in the regime switching model. Then we presented our final waterfall graphs with data from our Monte-Carlo Simulations; and lastly, our analysis of the impact of correlation between the SPY and RUS.

4.1 Distribution of Alpha

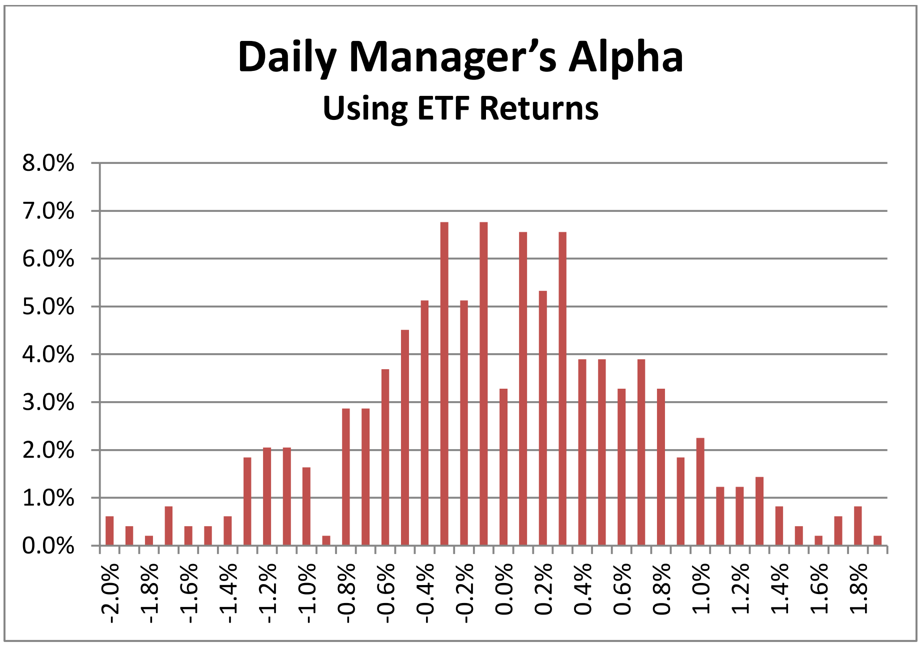
In order to introduce a distribution for manager's alpha into our Monte Carlo simulation, we analyzed the difference between an investment and its benchmark. We rounded the data points to the nearest tenth of a percent order to count the frequency of each return and develop the graph, expected value and standard deviation. The graphs for all three samples appear normally distributed and have sample skewness and kurtosis within 10% of the normal distribution (Miller & Miller, 2004, p. 147). Therefore, we chose to fit our data to a normal distribution.

Table 1: Skewness and Kurtosis of Sample Distributions

Sample Skewness and Kurtosis		
	Measure of Skewness	**Measure of Kurtosis**
Normal Distribution	0	3
Distribution of Manager's Alpha using Daily ETF Returns	-0.085	3.046
Distribution of Manager's Alpha using Daily Mutual Fund Returns	-0.028	3.170
Distribution of Manager's Alpha using Monthly Mutual Fund Returns	0.010	3.212

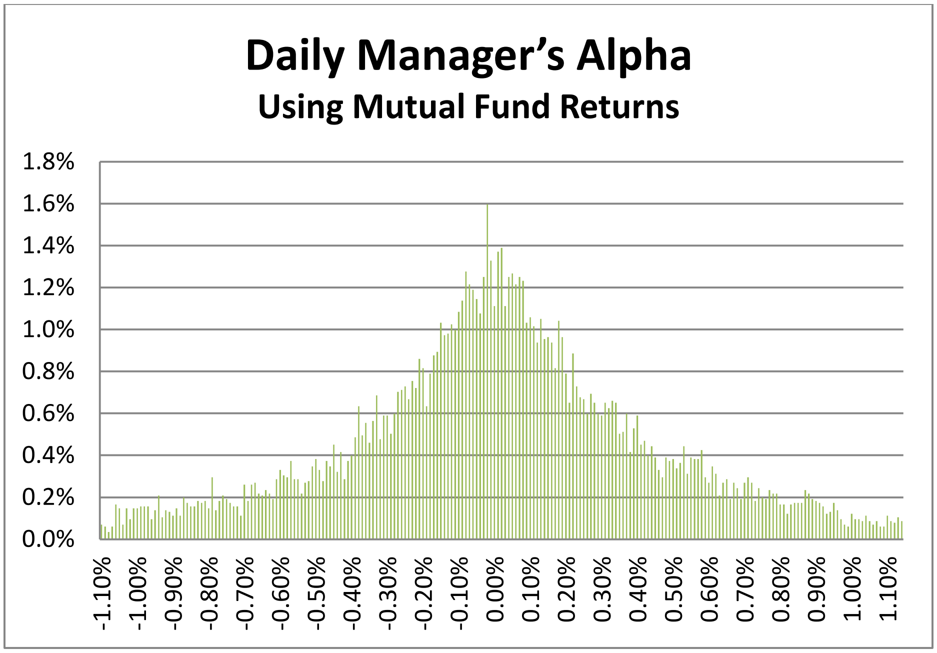
Since the maximum likelihood estimators of a normal distribution are the mean and standard deviation (Miller & Miller, 2004, p. 341), the distribution of manager's alpha using daily ETF returns is normal with a mean of -0.013% and a standard deviation of 0.729%. This equates to annual expected returns of -0.21% and standard deviation of 7.43%

Figure 3: ETF Daily Manager Alpha



Next, we repeated the analysis to find the distribution of manager's alpha using daily returns from large cap mutual funds. Because most ETFs are less than 3 years old and not updated regularly, there are not many reliable data points to create the distribution off of. Mutual funds on the other hand have more historical data to utilize and are updated much more frequently. Our group used the SPY index as a benchmark for all of the individual mutual funds in order to determine manager's alpha (Gruber, 1996). After excluding any outliers, we compiled the probability density function found in Figure 4 below.

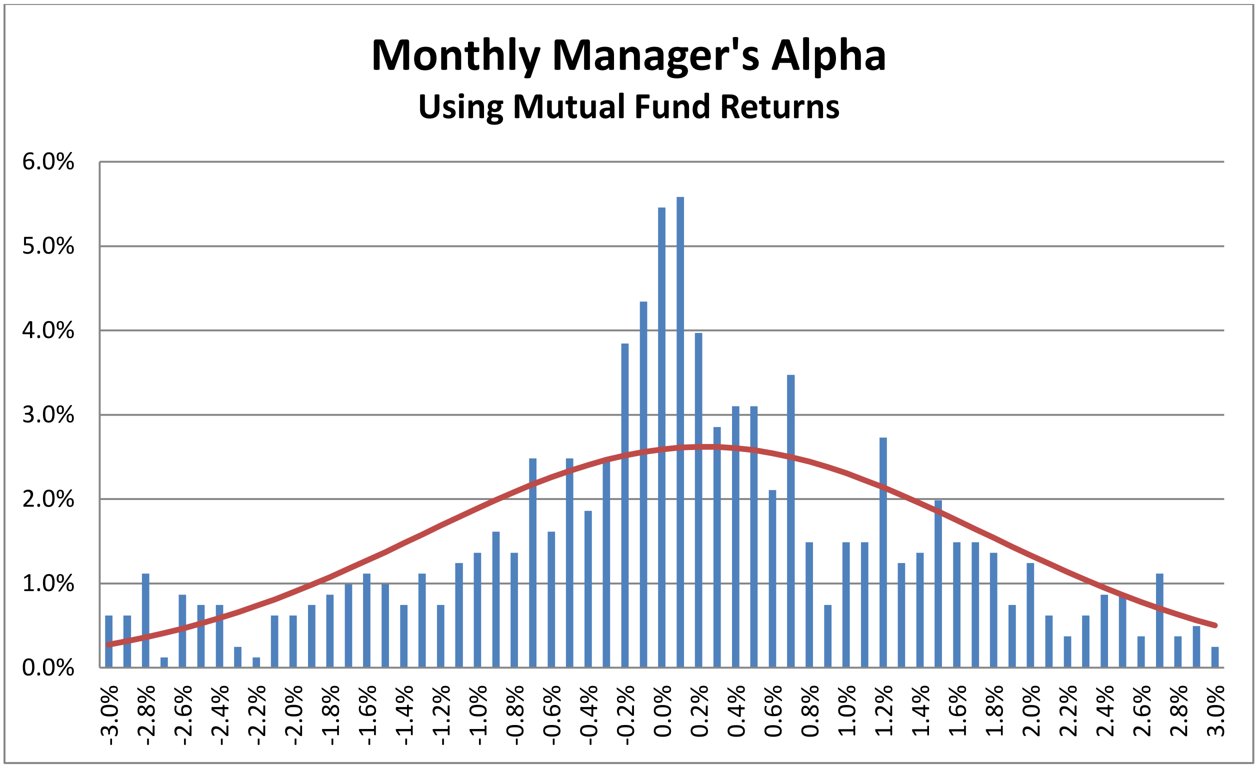
Figure 4: Mutual Fund Daily Manager Alpha



The range of the values for the manager's alpha found using the mutual funds is smaller than that of the ETF, but this is most likely due to the large amount of data points that were included. The distribution appears to be a normal distribution, and the parameters are $\mu = 0.014\%$ and $\sigma = 0.43\%$ daily. These parameters are the equivalent of an annual expected return of 3.6% and an annual standard deviation of 6.9%.

Our group believed that some of the large standard deviation was due to a lag in the mutual fund adjusting its investment strategy. Therefore, we chose to also model the monthly returns, assuming that managers would be more likely to update their portfolio by the end of the month. Using monthly returns for the same five mutual funds against the S&P and removing the outliers, we found the distribution of alpha to be normal with $\mu = 0.18\%$ and $\sigma = 1.52\%$, which result in annual values of $\mu = 2.16\%$ and $\sigma = 5.27\%$. The probability density functions of the actual returns as well as the normal distribution are shown in Figure 5 below.

Figure 5: Mutual Fund Monthly Managers Alpha



Since this result had the lowest standard deviation and a reasonable value for manager's alpha, we choice to use monthly mutual fund results in our model. However, we included the outliers in our calculations to ensure that the data was accurate, which adjusted our mean to 0.15% and standard deviation to 2.25% monthly.

4.2 Regime Switching

Using the spreadsheet from Section 3.1.2.1 (Figure 6), we calculated the following monthly maximum likelihood parameters for the SPY and the RUS based off of historical data.

Figure 6: Initial Regime Switching Results

	SPY Parameters	RUS Parameters
$p_{1,2}$	1.79%	1.37%
$p_{2,1}$	2.30%	16.83%
μ_1	1.30%	0.81%
σ_1	2.59%	5.31%
μ_2	-0.38%	-4.36%
σ_2	6.39%	15.35%

Historically, indices generally change indices at the same time (Hardy M. R., 2001), however the two probability matrices shown above have large differences in the probability of switching from regime 1 into regime 2. Since we wanted our model to simulate the real world, we decided to adjust our parameters so that they had the same probability matrix.

To do this, we took an average of the monthly returns for the SPY and RUS to create combined parameters. These parameters have 50% of the SPY return for each month and 50% of the RUS return over the historical period. These combined returns were then input into the model, and the maximum likelihood parameters are show in the first column of

Figure 7. We then extrapolated by fixing the probability matrix within the model for both the SPY and RUS individual returns, and ran the model again for each to find the maximum likelihood estimators using these probabilities of switching. The adjusted parameters for the SPY and RUS seen in

Figure 7 below are the parameters we chose to use in our model.

Figure 7: Adjusted Regime Switching Parameters

Forcing the Same Probability Matrix

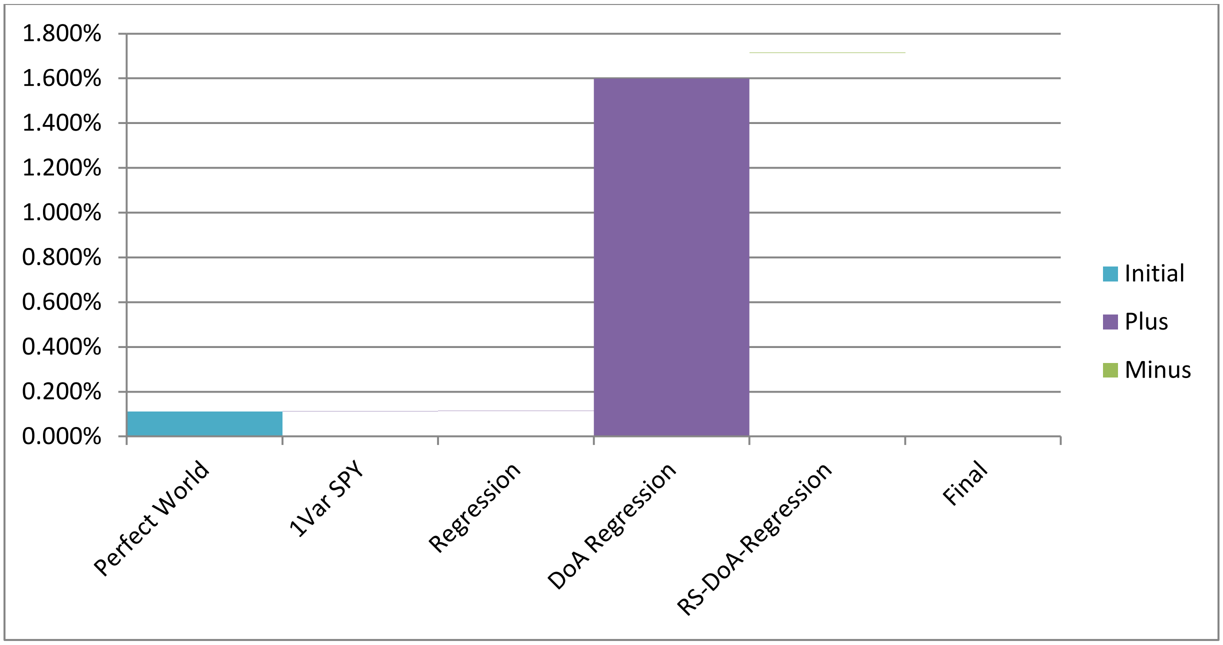
	Combined MLE Parameters	SPY Adjusted Parameters	RUS Adjusted Parameters
$p_{1,2}$	5.24%	5.24%	5.24%
$p_{2,1}$	10.75%	10.75%	10.75%
μ_1	1.12%	1.31%	1.22%
σ_1	3.40%	2.82%	4.56%
μ_2	-1.95%	-1.17%	-2.05%
σ_2	8.94%	7.19%	10.72%

4.3 Waterfall Graph Results

After compiling our data from the simulations we decided to display our results in the form of a waterfall graph. This graphic is a good way of showing how an initial value is affected by a series of intermediate positive or negative values. The initial value represents the "Perfect World"in each graph and the intermediate values represent the effects of regression, regime switching, and the addition of the distribution of alpha. These intermediary components are in different orders depending on the graphic displayed. We chose to try different orders for the simulations to see if there were any discernable changed in the graphs.

In Figure 8, seen below, the y-axis represents the percent change in the standard deviation of the simulated mutual fund from the benchmark. The order in which our components are displayed is "Perfect World", 1Var SPY, Regression, Distribution of Alpha, and Regime Switching. As one can see the Distribution of Alpha had the greatest effect on the simulation.

Figure 8: Waterfall Graph (1)



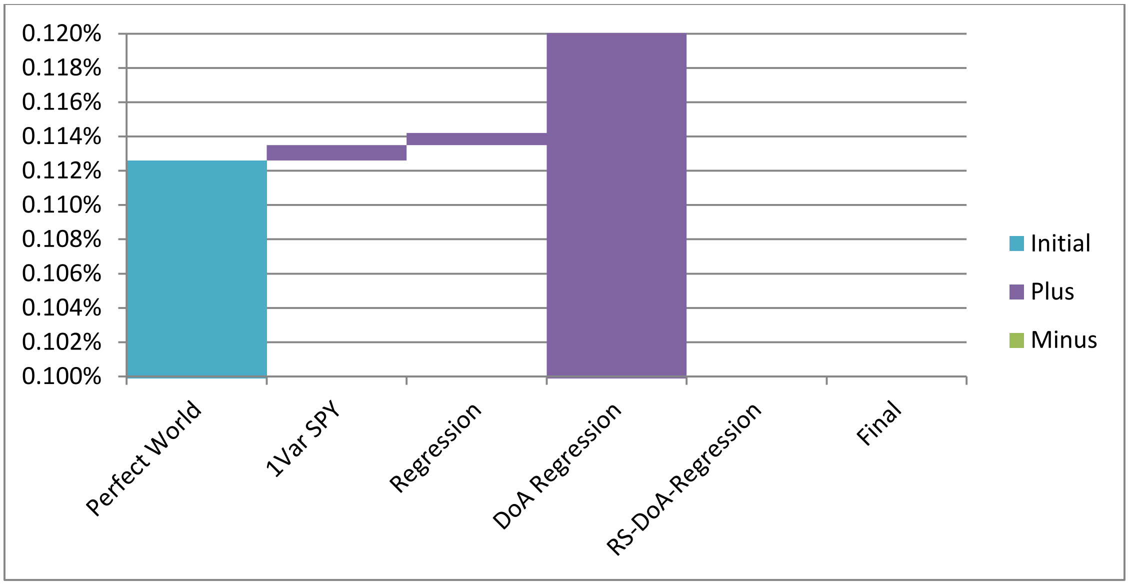
The values associated with each bar on the graph can be seen in Table 2. There is a very small change in the first two components from the "Perfect World" showing that there was little to no effect. The most significant jump in percent change is causes by the introduction of a distribution of alpha.

Table 2: Waterfall Values (1)

	Standard Deviation
Perfect World	0.1126%
1Var SPY	0.1135%
Regression	0.1142%
DoA Regression	1.7136%
RS-DoA-Regression	1.7131%

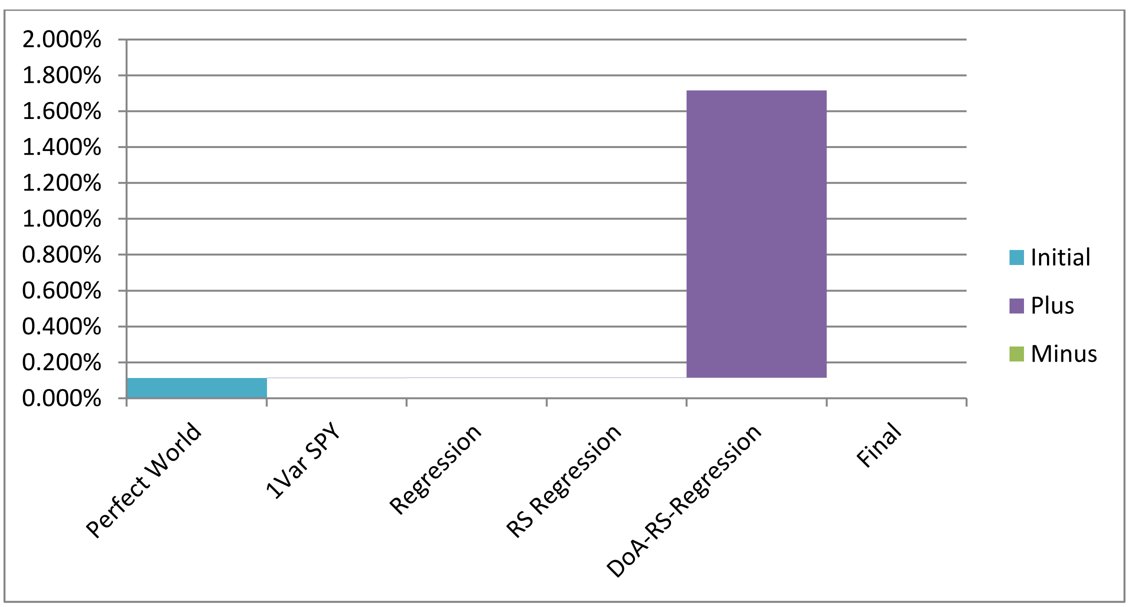
Due to the size of the jump with the distribution of alpha the other components are difficult to see in the large graphic. A magnified version of the graph in Figure 9 shows a clearer look at the percent change for the other components.

Figure 9: Magnified Waterfall Graph (1)



The subsequent graphs, Figure 10 through Fig 14 show the different orders and the values of each component within the graph. The order of the components in Figure 10 is "Perfect World", 1Var Spy, Regression, Regime Switching, and Distribution of Alpha.

Figure 10: Waterfall Graph (2)



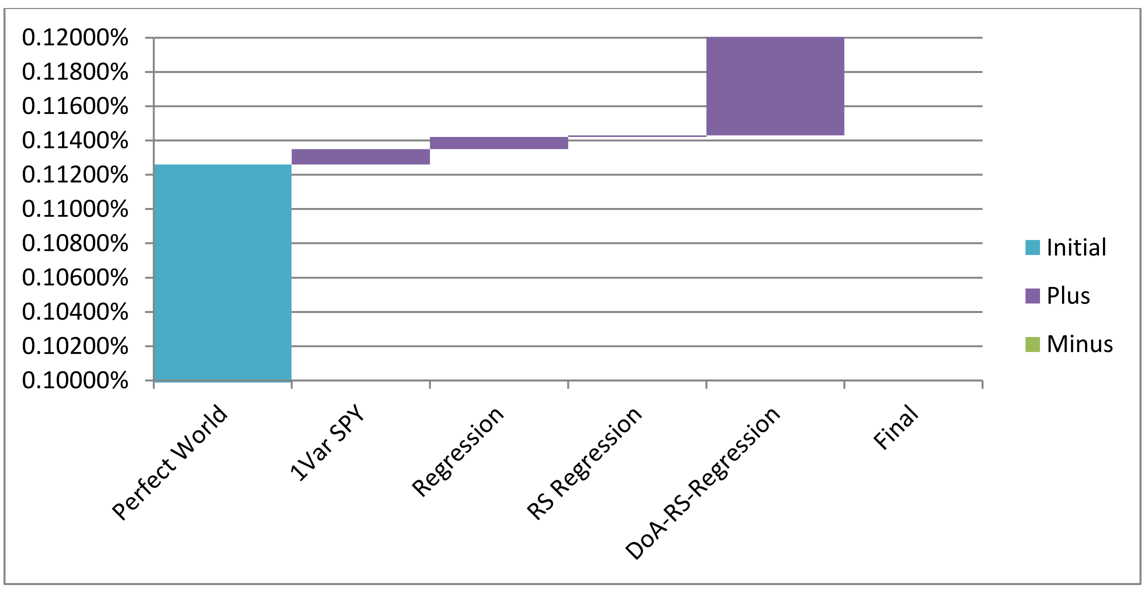
The distribution of alpha component again had the greatest affect on the simulation, resulting in a jump of more than 1.5%.

Table 3: Waterfall Values (2)

	Standard Deviation
Perfect World	0.1126%
1Var SPY	0.1135%
Regression	0.1142%
RS Regression	0.1143%
DoA-RS-Regression	1.7151%

In the magnified version of this graph, Figure 11, one can see that regime switching had the smallest effect on the simulation causing a change of only a ten thousandth of a percent.

Figure 11: Magnified Waterfall Graph (2)



The final order we chose to try was Distribution of Alpha, 1Var SPY, Regression, and Regime Switching, as seen in Figure 12. Distribution of Alpha once again had the greates impact on the simulation and Regime Switching again had the littlest impact.

Figure 12: Waterfall Graph (3)

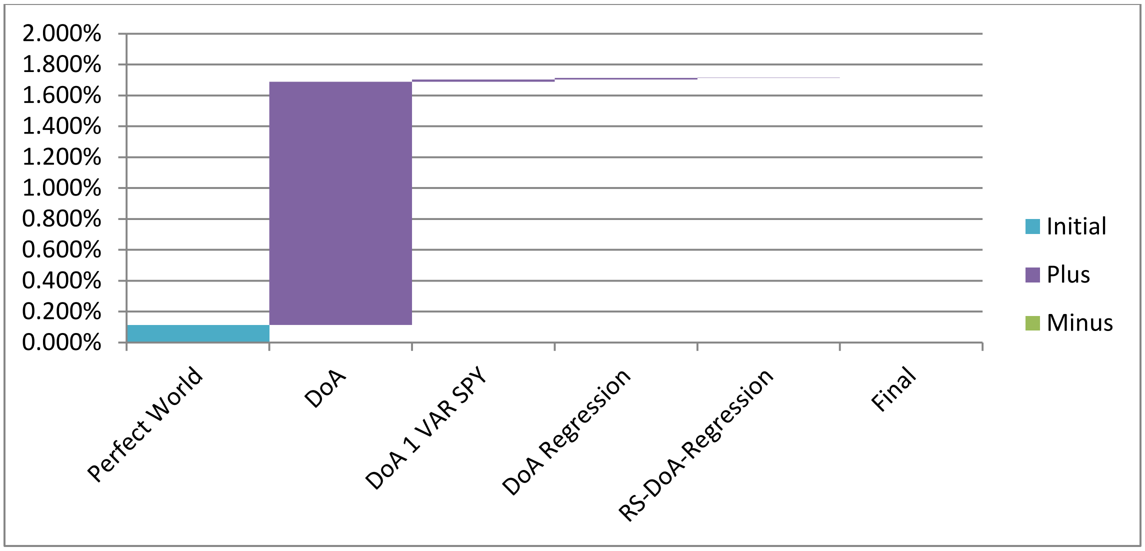
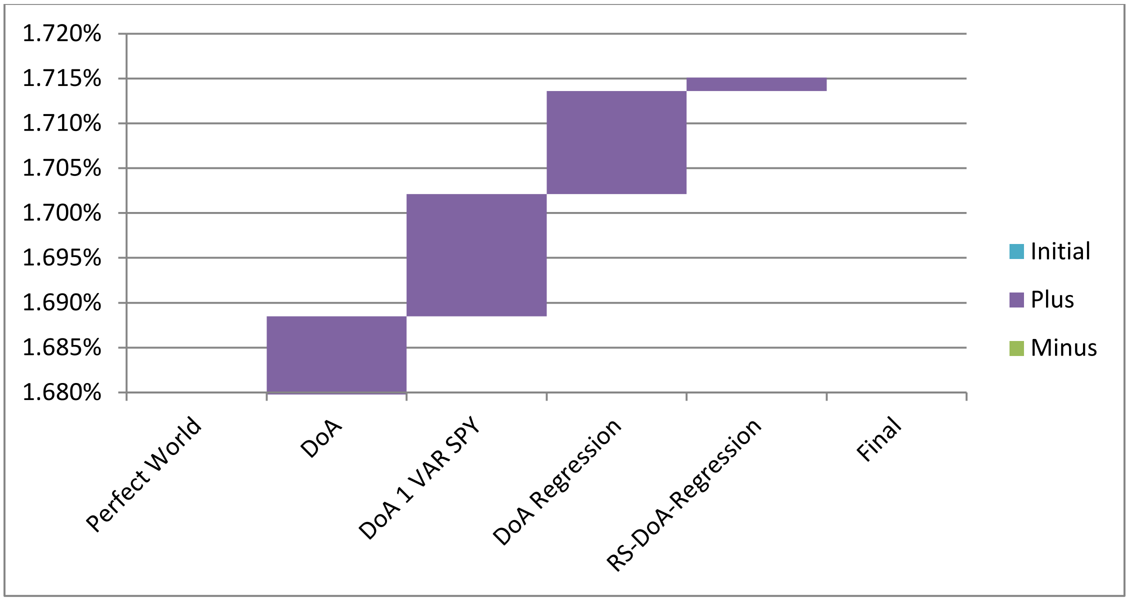


Table 4: Waterfall Values (3)

	Standard Deviation
Perfect World	0.1126%
DoA	1.6885%
DoA 1 VAR SPY	1.7021%
DoA Regression	1.7136%
RS-DoA-Regression	1.7151%

Figure 13: Magnified Waterfall Graph (3)



4.4 Impact of Correlation

After we ran the simulator with the realized correlation from the historical data, our group ran the simulator with a significantly lower correlation. In doing so, we hoped to see what kind of impact the correlation between the two indices had on the tracking error. However, the change in correlation had nearly no impact on the tracking error. This was not quite the result we expected since the regression was based on both indices. We expected the relationship between the SPY and RUS to pertain to the tracking error. However, the 2-VAR regression still relied overwhelmingly on the SPY, so the effect of the RUS on the regression was minimal. In our simulator, the mutual fund was based solely on the SPY, but it would be interesting to see the impact of correlation in a simulator where the mutual fund is based on multiple indices.

5 Conclusions and Recommendations

Overall, from our waterfall graphs, we found that the addition of the Distribution of Alpha had the greatest impact on the simulation and Regime Switching had the most minimal effect. The percent changes also held fairly constant regardless of which order we ran the simulations in. We found that the approximate effect of each component was about;

Table 5: Results

Component	Effect (+ or –)
1Var	80 basis points (bps)
Regression	62 bps
Regime Switching	9 bps
Distribution of Alpha	140,000 bps

The Distribution of Alpha that we used made a significant change and for future projects it could be interesting to see if using another distribution would minimize the effect.

In our model, both the SPY and the RUS carry their respective parameters from the future-past to the future-future. Our group had the intent of changing this, if there was suitable time, by allowing the simulator to run the future past with one set of parameters. Then the regressions would be created based on the future-past, as the tool does now. However, after the regressions are complete, a new set of parameters would be entered for the future-future. This would be used in order to simulate an economic change. By basing the regression off one set of expected returns, and then changing the expected return in the future-future, the simulator would show how inaccurate the regression could be if expected returns were to change. This is a closer simulation of the real world, because currently the tool has the same expected return for a 20 year period. This would be a good change to the tool for future project groups to focus on.

It is difficult to find an ETF or a Mutual Fund with an easy to follow benchmark. For instance, it was an educated guess to make the benchmark the S&P 500 for the Mutual Funds that we chose. For the exchange traded funds it was much more difficult to find an easy to follow benchmark in general, since there were many days where it seemed that data wasn't being recorded by the fund manager. One possible way to remedy this would be to personally contact the fund manager of the ETF or Mutual Fund that is of interest. This would help to fill any gaps and answer any unknowns about the fund's benchmark and the daily recorded data for the fund.

Once a reliable ETF or MF is found, it becomes tough to properly fit a non-normal distribution to Manager Alpha. There are a couple of possibilities that can be explored, given our time constraints; we were not able to look into. The Skew-Normal distribution seemed that it could be a possible fit. However, we just did not have the time to estimate the parameters for the distribution once it was determined as a possible fit. Another possibility would be to use a Chi-Squared distribution, which is an all positive distribution, because it is essentially a normal distribution squared. So in order to make this disitribution a possible fit, our suggestion would be to add some constant to make the furthermost negative point 0. For instance, if the lowest x-value on the graph was -8%, then you would add 8% to that point, and all other points in order to shift the graph accordingly. We found that a Chi-Squared distribution with 5 degrees of freedom seemed to be the best possible fit for the data, given that it is possible to shift all xvalues with a constant, and still retain an accurate model. However, once again, there was not enough time to explore this possibility further.

6 Bibliography

(2010). Retrieved from towerswatson.com: http://www.towerswatson.com/about/

American Funds. (n.d.). Retrieved December 11, 2010, from AmericanFunds: https://www.americanfunds.com/pdf/mfgepr-905_gfap.pdf

American Funds. (n.d.). Retrieved December 11, 2010, from American Fund: https://www.americanfunds.com/pdf/mfgepr-933_wgip.pdf

American Funds. (n.d.). Retrieved December 11, 2010, from AmericanFunds: https://www.americanfunds.com/pdf/mfgepr-916_eupacp.pdf

Benhamou, E. (n.d.). *Eric Benhamou Home Page.* Retrieved October 6, 2010, from Publications: http://www.ericbenhamou.net/documents/Encyclo/Basis%20risk.pdf

Chriss, N. A. (1996). *Black Scholes and Beyond : Option Pricing Models.* The McGraw-Hill Companies.

Corporate Bonds. (2009). Retrieved November 20, 2010, from Investor Words: http://www.investorwords.com/1131/corporate_bond.html

elixirtech.com. (n.d.). Retrieved December 1, 2010, from Report Elements: Charts: http://www.elixirtech.com/release/Rep7.0.1/Report-Designer/ch04s17.html

Finance, B. I. (2005). *Boston Institute of Finance Mutual Fund Advisor Course : Series 6 and Series 63 Test Prep.* Boston: John Wiley & Sons, Incorporated.

Fredricks, G., Ingalls, E., & McAlister, A. (2010). *Analysis of Fund Mapping Techniques for Variable Annuities.* Worcester: Worcester Polytechnic Institute.

Gruber, M. J. (1996). Another Puzzle: The Growth in Actively Managed Mutual Funds. *The Journal of Finance , 51* (3), 783-810.

Hardy, M. R. (2001). A Regime-Switching Model of Long-Term Stock Returns. *North American Actuarial Journal/2001/april/naaj0104_4.pdf* , 41-53.

Hardy, M. R. (2006). Introduction to Risk Measures for Actuarial Applications. *Construction and Evaluation of Actuarial Models Study Note* , 1-31.

Hull, J. C. (2006). *Options, Futures, and Other Derivatives.* Upper Saddle River: Pearson Education, Inc.

Hull, J. C. (2009). *Options, Futures, and Other Derivatives.* Upper Saddle River, New Jersey: Pearson Education, Inc.

Investopedia. (2010). Retrieved September 2010, from http://www.investopedia.com/terms/e/etf.asp

Lange, T., & Rahbek, A. (2009). An Introduction to Regime Switching. *Handbook of Financial Time Series* , 871-887.

Levinson, M. (2005). *Guide to Financial Markets.* Profile Books Limited.

Linear Regression. (1997-1998). Retrieved November 20, 2010, from The Department of Statistics at Yale University: http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm

M. Crouhy, D. G. (2001). *Risk Management.* The McGraw-Hill Companies.

Miller, I., & Miller, M. (2004). *John E. Freund's Mathematical Statistics with Applications.* Upper Saddle River, NJ: Pearson Prentice Hall.

Multiple Regression . (2008). Retrieved December 29, 2009, from Electronic Textbook StatSoft: http://www.statsoft.com/TEXTBOOK/stmulreg.html

Mutual Funds. (2007, May 14). Retrieved October 2, 2009, from U.S. Securities and Exchange Commision: http://www.sec.gov/answers/mutfund.htm

Petrucelli, J. D., Nandram, B., & Chen, M. (1999). *Applied Statistics for Engineers and Scientists.* Upper Saddle River, NJ: A Pearson Education Company.

Stulz, C. W. (1985). The Determinants of Firms' Hedging Policies. *Journal of Financial and Quantitative Analysis, 20* , 391-405.

Tracking Error In Exchange Traded Funds. (2007, May 30). Retrieved November 9, 2010, from Seeking Alpha: http://seekingalpha.com/article/36875-tracking-error-in-exchange-traded-funds

Trivedi, K. S. (2002). *Probability, Statistics with Reliability, Queueing and Computer Science Applications, 2nd ed.* Wiley.

Vanguard. (n.d.). Retrieved December 11, 2010, from Vanguard: www.vanguard.com/pub/Pdf/p113.pdf

Vanguard Institutional. (n.d.). Retrieved December 11, 2010, from Vanguard: https://institutional.vanguard.com/iippdf/pdfs/IO854.pdf

Variable Annuites: What you should know. (2009, April 28). Retrieved November 9, 2010, from U.S. Securities and Exchange Commision: http://www.sec.gov/investor/pubs/varannty.htm

Wolfinger, M. (2005). *Create Your Own Hedge Fund: Increase Profits and Reduce Risk with ETF's and Options.* John Wiley & Sons, Inc.

Woller, J. (1996, Spring). The Basics of Monte Carlo Simulations. Lincoln, Nebraska, United States of America.

Yahoo Finance. (n.d.). Retrieved December 11, 2010, from Yahoo: http://finance.yahoo.com/q?s=AGTHX&ql=1

Yahoo Finance. (n.d.). Retrieved December 11, 2010, from Yahoo: http://finance.yahoo.com/q?s=CWGIX&ql=1

Yahoo Finance. (n.d.). Retrieved December 11, 2010, from Yahoo: http://finance.yahoo.com/q?s=VGTSX&ql=1

Yahoo Finance. (n.d.). Retrieved December 11, 2010, from Yahoo: http://finance.yahoo.com/q?s=VINIX&ql=1

Yahoo Finance. (n.d.). Retrieved December 11, 2010, from Yahoo: http://finance.yahoo.com/q?s=AEPGX&ql=1

7 Appendix

Monte Carlo VBA Code

```
Sub RSMonteCarlo()

'
' SPMonteCarlo Macro
'

'Turn screen refresh off
Application.ScreenUpdating = False

'Start time
Dim starttime As Date

starttime = Now
Worksheets("Output").Range("B3").Value = starttime

'clear output tab
Worksheets("Output").Range("I1:IV65536").Value = ""

''''''''''''''''''''''''''''''''
'Start Varible Declaration'
''''''''''''''''''''''''''''''''

'Declare Variables
Dim scenarios As Long
Dim projYears As Long
Dim projDays As Long
Dim projMonths As Long
Dim timeinterval As Double
Dim a11 As Double
Dim a21 As Double
Dim a22 As Double
Dim randnorm As Double
Dim randnum As Double
Dim FPcalculate As Boolean
Dim FPdailyDisplay As Boolean
Dim FPprintType As Boolean
Dim periodsPerYear As Integer
Dim regimeSwitching As Boolean
Dim p11 As Double
```

```
Dim p12 As Double
Dim p21 As Double
Dim p22 As Double
Dim randRegime As Double
Dim regime As Integer
Dim displayPeriods As Integer
Dim dOfa As Boolean

'Declare S&P Variables
Dim sigma As Double
Dim mu As Double
Dim sigma1 As Double
Dim mu1 As Double
Dim sigma2 As Double
Dim mu2 As Double
Dim startvalue As Double
Dim correlation As Double
Dim newvalue As Double
Dim oldvalue As Double
Dim lnreturn As Double
Dim FPfinal As Double
Dim SPYreturn As Double
Dim SPYmonthEnd As Double
Dim SPYaverage As Double

'Declare Mutual Fund Variables
Dim SPYalpha As Double
Dim mfstartvalue As Double
Dim mfnewvalue As Double
Dim mfoldvalue As Double
Dim mflnreturn As Double
Dim FPmfFinal As Double
Dim MFreturn As Double
Dim MFmonthEnd As Double
Dim MFaverage As Double
Dim mfrandnorm As Double
Dim mfrandnum As Double
Dim alpha As Double
Dim alphaMu As Double
Dim alphaSigma As Double

'Declare Russell 2000 Variables
Dim russigma As Double
Dim rusmu As Double
```

```
Dim russigma1 As Double
Dim rusmu1 As Double
Dim russigma2 As Double
Dim rusmu2 As Double
Dim russtartvalue As Double
Dim rusrandnorm As Double
Dim rusrandnum As Double
Dim rusnewvalue As Double
Dim rusoldvalue As Double
Dim ruslnreturn As Double
Dim FPrusFinal As Double
Dim RUSreturn As Double
Dim RUSmonthEnd As Double
Dim RUSaverage As Double

'Declare Regression Variables
Dim scenarioCounter As Long
Dim regressionOffset As Long
Dim currentRegression As Long
Dim maxRegression As Long
Dim regression As Long
Dim twoIntercept As Double
Dim SP2VarCoeff As Double
Dim RUS2VarCoeff As Double
Dim SPintercept As Double
Dim SP1VarCoeff As Double
Dim RUSintercept As Double
Dim RUS1VarCoeff As Double
Dim regressDaily As Boolean
Dim regressLN As Boolean
Dim yValues() As Double
Dim xValues() As Double
Dim x1values() As Double
Dim x2values() As Double

'Declare Future Future Variables
Dim FFtimeinterval As Double
Dim FFmfiB As Double
Dim FFmfiBnewvalue As Double
Dim FFmfiBoldvalue As Double
Dim FFmfiBlnreturn As Double
Dim FFmfiC As Double
Dim FFmfiClnreturn As Double
Dim FFmfiCnewvalue As Double
```

```
Dim FFmfiColdvalue As Double
Dim FFmfiD As Double
Dim FFmfiDlnreturn As Double
Dim FFmfiDnewvalue As Double
Dim FFmfiDoldvalue As Double
Dim FFprojYears As Long
Dim FFprojDays As Long
Dim FFprojMonths As Long
Dim FFscenarios As Long
Dim FFrowOutput As Long
Dim FFheaderOutput As Long
Dim FFcalculate As Boolean
Dim FFdailyDisplay As Boolean
Dim FFprintType As Boolean
Dim FFperiodsPerYear As Integer
Dim FFdisplayPeriods As Integer

'Set Starting Values for Variables
scenarios = Worksheets("Input").Range("B4").Value
scenarioCounter = 0
sigma1 = Worksheets("Input").Range("B8").Value
mu1 = Worksheets("Input").Range("C8").Value
sigma2 = Worksheets("Input").Range("F8").Value
mu2 = Worksheets("Input").Range("G8").Value
correlation = Worksheets("Input").Range("E8").Value
projYears = Worksheets("Input").Range("C4").Value
projDays = projYears * 252
projMonths = projYears * 12
newvalue = Worksheets("Input").Range("D8").Value
timeinterval = Worksheets("Input").Range("G4").Value
alpha = Worksheets("Input").Range("B12").Value
SPYalpha = Worksheets("Input").Range("C12").Value
mfnewvalue = Worksheets("Input").Range("D12").Value
rusnewvalue = Worksheets("Input").Range("D16").Value
rusmu1 = Worksheets("Input").Range("C16").Value
russigma1 = Worksheets("Input").Range("B16").Value
rusmu2 = Worksheets("Input").Range("F16").Value
russigma2 = Worksheets("Input").Range("E16").Value
a11 = Worksheets("Input").Range("B19").Value
a21 = Worksheets("Input").Range("B20").Value
a22 = Worksheets("Input").Range("C20").Value
regression = WorksheetFunction.Min(projYears, Worksheets("Input").Range("B24").Value)
Worksheets("Input").Range("B24").Value = regression
FFtimeinterval = Worksheets("Input").Range("G28").Value
```

```
FFprojYears = Worksheets("Input").Range("C28").Value
FFprojDays = FFprojYears * 252
FFprojMonths = FFprojYears * 12
FFscenarios = Worksheets("Input").Range("B28").Value
p11 = Worksheets("Transition Matrix").Range("B4").Value
p12 = Worksheets("Transition Matrix").Range("C4").Value
p21 = Worksheets("Transition Matrix").Range("B5").Value
p22 = Worksheets("Transition Matrix").Range("C5").Value
SPYmonthEnd = newvalue
MFmonthEnd = mfnewvalue
RUSmonthEnd = rusnewvalue
SPYaverage = 0
RUSaverage = 0
MFaverage = 0
alphaMu = Worksheets("Input").Range("B12").Value
alphaSigma = Worksheets("Input").Range("C12").Value

'Sets Regime Switching variable
If Worksheets("Input").Range("E12").Value = "Yes" Then
   'Regime Switching means true
   dOfa = True
   Else
   dOfa = False
End If

'Sets Regime Switching variable
If Worksheets("Input").Range("H4").Value = "Yes" Then
   'Regime Switching means true
   regimeSwitching = True
   Else
   regimeSwitching = False
End If

'Sets Parameters for Non-Regime Switching
If regimeSwitching = False Then
   'Set the mean and standard deviation for SPY
   sigma = sigma1
   mu = mu1 - 0.5 * sigma ^ 2

   'Set the mean and standard deviation for RUS
   russigma = russigma1
   rusmu = rusmu1 - 0.5 * russigma ^ 2
End If
```

```
'Sets Future Past Calculate variable
If Worksheets("Input").Range("D4").Value = "Daily" Then
   'Daily calculation of returns means True
   FPcalculate = True
   Else
   If Worksheets("Input").Range("D4").Value = "Monthly" Then
      'Monthly calculation of returns means False
      FPcalculate = False
      Else
      'When Neither 'Daily' or 'Monthly' is entered, assume 'Daily'
      Worksheets("Input").Range("D4").Value = "Daily"
      FPcalculate = True
   End If
End If

'If calculating by month, then Proj Days becomes Proj Months
If (FPcalculate = False) Then
   projDays = 12 * projYears
End If

'Sets Future Past Display variable
If Worksheets("Input").Range("E4").Value = "Daily" Then
   'Daily display of data means True
   FPdailyDisplay = True
   periodsPerYear = 252
   Else
   If Worksheets("Input").Range("E4").Value = "Monthly" Then
      'Monthly display of data means False
      FPdailyDisplay = False
      periodsPerYear = 12
      Else
      'When Neither 'Daily' or 'Monthly' is entered, assume 'Daily'
      Worksheets("Input").Range("E4").Value = "Daily"
      FPdailyDisplay = True
      periodsPerYear = 252
   End If
End If

'Test
If FPcalculate = False Then
   If FPdailyDisplay = True Then
      'Scenarios can't be run with monthly calculations but outputting values daily
      'So, switch to monthly output.
```

```
      Worksheets("Input").Range("E4").Value = "Monthly"
      FPdailyDisplay = False
      periodsPerYear = 12
   End If
End If

'Set displayPeriods
displayPeriods = projYears * periodsPerYear

'Sets Future Past Print variable
If Worksheets("Input").Range("F4").Value = "LN Return" Then
   'LN Return output means True
   FPprintType = True
   Else
   If Worksheets("Input").Range("F4").Value = "Values" Then
      'Values output means False
      FPprintType = False
      Else
      'When Neither 'LN Return' or 'Values' is entered, assume 'LN Return'
      Worksheets("Input").Range("F4").Value = "LN Return"
      FPprintType = True
   End If
End If

'Sets Regression Calculating Variable
If Worksheets("Input").Range("C24").Value = "Daily" Then
   'Regressing daily means True
   regressDaily = True
   Else
   If Worksheets("Input").Range("C24").Value = "Monthly" Then
      'Regressing monthly means False
      regressDaily = False
      Else
      'When Neither 'Daily' or 'Monthly is entered, assume 'Daily'
      Worksheets("Input").Range("C24").Value = "Daily"
      regressDaily = True
   End If
End If

'Test
If FPdailyDisplay = False Then
   If regressDaily = True Then
      'Regressions can't be done daily if there is only monthly data for the scenarios
      'So, switch to monthly regression.
```

```
      Worksheets("Input").Range("C24").Value = "Monthly"
      regressDaily = False
   End If
End If

'Sets whether to regress the ln returns or index values
If Worksheets("Input").Range("D24").Value = "LN Return" Then
   'LN Regression means true
   regressLN = True
   Else
   If Worksheets("Input").Range("D24").Value = "Values" Then
      'Values Regression means false
      regressLN = False
      Else
      'When Neither 'LN Return' or 'Values' is entered, assume 'LN Return'
      Worksheets("Input").Range("D24").Value = "LN Return"
      regressLN = True
   End If
End If

'Sets Future Future Calculate variable
If Worksheets("Input").Range("D28").Value = "Daily" Then
   'Daily calculation of returns means True
   FFcalculate = True
   Else
   If Worksheets("Input").Range("D28").Value = "Monthly" Then
      'Monthly calculation of returns means False
      FFcalculate = False
      Else
      'When Neither 'Daily' or 'Monthly' is entered, assume 'Daily'
      Worksheets("Input").Range("D28").Value = "Daily"
      FFcalculate = True
   End If
End If

'Sets Future Future Display variable
If Worksheets("Input").Range("E28").Value = "Daily" Then
   'Daily display of data means True
   FFdailyDisplay = True
   FFperiodsPerYear = 252
   Else
   If Worksheets("Input").Range("E28").Value = "Monthly" Then
      'Monthly display of data means False
      FFdailyDisplay = False
```

```
      FFperiodsPerYear = 12
      Else
      'When Neither 'Daily' or 'Monthly' is entered, assume 'Daily'
      Worksheets("Input").Range("E28").Value = "Daily"
      FFdailyDisplay = True
      FFperiodsPerYear = 252
   End If
End If

'Test
If FFcalculate = False Then
   If FFdailyDisplay = True Then
      'Scenarios can't be run with monthly calculations but outputting values daily
      'So, switch to monthly output.
      Worksheets("Input").Range("E28").Value = "Monthly"
      FFdailyDisplay = False
      FFperiodsPerYear = 12
   End If
End If

'Set displayPeriods
FFdisplayPeriods = FFprojYears * FFperiodsPerYear

'If Future Future Display is false (data is shown monthly), then FFprojDays is actually
FFprojMonths
If FFdailyDisplay = False Then
   FFprojDays = FFprojYears * 12
End If

'Sets Future Future Print variable
If Worksheets("Input").Range("F28").Value = "LN Return" Then
   'LN Return output means True
   FFprintType = True
   Else
   If Worksheets("Input").Range("F28").Value = "Values" Then
      'Values output means False
      FFprintType = False
      Else
      'When Neither 'LN Return' or 'Values' is entered, assume 'LN Return'
      Worksheets("Input").Range("F28").Value = "LN Return"
      FFprintType = True
   End If
End If
```

```
''''''''''''''''''''''''
'End Varible Declaration'
''''''''''''''''''''''''

'Backup old "Scenario Output" Worksheet
Dim oldoutputs As Integer
Dim sheet As Integer
sheet = 1
Do While sheet < Worksheets.Count + Charts.Count + 1
   If Sheets(sheet).Name = "Scenario Output" Then
      oldoutputs = 1
      sheet = 1
      Do While sheet < Worksheets.Count + Charts.Count + 1
         If Sheets(sheet).Name = "Scenario Output Old" & oldoutputs Then
            oldoutputs = oldoutputs + 1
            sheet = 0
         End If
         sheet = sheet + 1
      Loop
      Worksheets("Scenario Output").Name = "Scenario Output Old" & oldoutputs
   End If
   sheet = sheet + 1
Loop


'New "Scenario Outputs" Worksheet
Sheets.Add Before:=Worksheets("Output")
ActiveSheet.Name = "Scenario Output"
ActiveWindow.Zoom = 85

'''''''''''''''''''''''''
'Formatting New Worksheet'
'''''''''''''''''''''''''

'Column Height
Worksheets("Scenario Output").Columns.ColumnWidth = 8.57
Worksheets("Scenario Output").Columns(1).ColumnWidth = 13.51

'Number Formatting
Worksheets("Scenario Output").Columns.NumberFormat = "0.0000"
If FPprintType = True Then
   For I = 1 To displayPeriods + 2
      'If printing LN Returns, use percentage
      Worksheets("Scenario Output").Rows(I).NumberFormat = "0.00%"
```

```
   Next I
Else
   For I = 1 To displayPeriods + 2
      'If printing values, use number
      Worksheets("Scenario Output").Rows(I).NumberFormat = "0.00"
   Next I
End If
If FFprintType = False Then
   'If printing values, use number
   For I = 1 To FFscenarios * 2
      For j = 1 To (FFdisplayPeriods + 2) * 2
      Worksheets("Scenario Output").Rows(7 + j + 25 + displayPeriods + (I - 1) * 2 *
(FFdisplayPeriods + 2)).NumberFormat = "0.00"
      Next j
   Next I
Else
   'If printing LN Returns, use value
   For I = 1 To FFscenarios
      For j = 1 To (FFdisplayPeriods + 2) * 2
      Worksheets("Scenario Output").Rows(7 + j + 25 + displayPeriods + (I - 1) * 2 *
(FFdisplayPeriods + 2)).NumberFormat = "0.00%"
      Next j
   Next I
End If
Worksheets("Scenario Output").Columns(1).NumberFormat = "0"

'Bolding Stuff
Worksheets("Scenario Output").Columns(1).Font.FontStyle = "Bold"
Worksheets("Scenario Output").Rows("1:2").Font.FontStyle = "Bold"
Worksheets("Scenario Output").Range("A1").Font.FontStyle = "Unbold"
For I = 1 To FFscenarios * 2
   Worksheets("Scenario Output").Rows(31 + displayPeriods + (I - 1) * (FFdisplayPeriods +
2)).Font.FontStyle = "Bold"
   Worksheets("Scenario Output").Rows(32 + displayPeriods + (I - 1) * (FFdisplayPeriods +
2)).Font.FontStyle = "Bold"
Next I

'Enlarging Fonts
Worksheets("Scenario Output").Rows("1:2").Font.Size = 14
Worksheets("Scenario Output").Range("A1").Font.Size = 10
For I = 1 To FFscenarios * 2
   Worksheets("Scenario Output").Rows(31 + displayPeriods + (I - 1) * (FFdisplayPeriods +
2)).Font.Size = 14
```

```
   Worksheets("Scenario Output").Rows(32 + displayPeriods + (I - 1) * (FFdisplayPeriods +
2)).Font.Size = 14
Next I

'Borders
With Worksheets("Scenario Output").Rows(2).Borders(xlEdgeBottom)
   .LineStyle = xlContinuous
   .ColorIndex = xlAutomatic
   .TintAndShade = 0
   .Weight = xlThin
End With
With Worksheets("Scenario Output").Rows(3).Borders(xlEdgeTop)
   .LineStyle = xlContinuous
   .ColorIndex = xlAutomatic
   .TintAndShade = 0
   .Weight = xlThin
End With
With Worksheets("Scenario Output").Columns(1).Borders(xlEdgeRight)
   .LineStyle = xlContinuous
   .ColorIndex = xlAutomatic
   .TintAndShade = 0
   .Weight = xlThin
End With
With Worksheets("Scenario Output").Columns(2).Borders(xlEdgeLeft)
   .LineStyle = xlContinuous
   .ColorIndex = xlAutomatic
   .TintAndShade = 0
   .Weight = xlThin
End With
For I = 1 To FFscenarios * 2
   With Worksheets("Scenario Output").Rows(5 + 25 + displayPeriods + (I - 1) *
(FFdisplayPeriods + 2)).Borders(xlEdgeBottom)
      .LineStyle = xlContinuous
      .ColorIndex = xlAutomatic
      .TintAndShade = 0
      .Weight = xlThick
   End With
   With Worksheets("Scenario Output").Rows(6 + 25 + displayPeriods + (I - 1) *
(FFdisplayPeriods + 2)).Borders(xlEdgeTop)
      .LineStyle = xlContinuous
      .ColorIndex = xlAutomatic
      .TintAndShade = 0
      .Weight = xlThick
   End With
```

```
   With Worksheets("Scenario Output").Rows(7 + 25 + displayPeriods + (I - 1) *
(FFdisplayPeriods + 2)).Borders(xlEdgeBottom)
       .LineStyle = xlContinuous
       .ColorIndex = xlAutomatic
       .TintAndShade = 0
       .Weight = xlThin
   End With
   With Worksheets("Scenario Output").Rows(8 + 25 + displayPeriods + (I - 1) *
(FFdisplayPeriods + 2)).Borders(xlEdgeTop)
       .LineStyle = xlContinuous
       .ColorIndex = xlAutomatic
       .TintAndShade = 0
       .Weight = xlThin
   End With
Next I
With Worksheets("Scenario Output").Rows(5 + 25 + displayPeriods + FFscenarios * 2 *
(FFdisplayPeriods + 2)).Borders(xlEdgeBottom)
   .LineStyle = xlContinuous
   .ColorIndex = xlAutomatic
   .TintAndShade = 0
   .Weight = xlThick
End With
With Worksheets("Scenario Output").Rows(6 + 25 + displayPeriods + FFscenarios * 2 *
(FFdisplayPeriods + 2)).Borders(xlEdgeTop)
   .LineStyle = xlContinuous
   .ColorIndex = xlAutomatic
   .TintAndShade = 0
   .Weight = xlThick
End With
With Worksheets("Scenario Output").Rows(5 + 25 + displayPeriods + FFscenarios *
(FFdisplayPeriods + 2)).Borders(xlEdgeBottom)
   .LineStyle = xlContinuous
   .ColorIndex = xlAutomatic
   .TintAndShade = 0
   .Weight = xlThick
End With
With Worksheets("Scenario Output").Rows(6 + 25 + displayPeriods + FFscenarios *
(FFdisplayPeriods + 2)).Borders(xlEdgeTop)
   .LineStyle = xlContinuous
   .ColorIndex = xlAutomatic
   .TintAndShade = 0
   .Weight = xlThick
End With
```

```
'Display Period Numbers
If FPdailyDisplay = False Then
   If FPcalculate = True Then
      If FPprintType = True Then
         'For Monthly periods with daily LN Return calculations
         Worksheets("Scenario Output").Range("A2").Value = "Months (LN Returns, calculated
Daily)"
      Else
         'For Monthly periods with monthly Value calculations
         Worksheets("Scenario Output").Range("A2").Value = "Months (Index Values, calculated
Daily)"
      End If
   Else
      If FPprintType = True Then
         'For Monthly periods with monthly LN Return calculations
         Worksheets("Scenario Output").Range("A2").Value = "Months (LN Returns, calculated
Monthly)"
      Else
         'For Monthly periods with monthly Value calculations
         Worksheets("Scenario Output").Range("A2").Value = "Months (Index Values, calculated
Monthly)"
      End If
   End If
Else
   If FPprintType = True Then
      'For Daily periods with daily LN Return calculations
      Worksheets("Scenario Output").Range("A2").Value = "Days (LN Returns, calculated Daily)"
   Else
      'For Daily periods with daily Value calculations
      Worksheets("Scenario Output").Range("A2").Value = "Days (Index Values, calculated
Daily)"
   End If
End If

For I = 1 To displayPeriods
Worksheets("Scenario Output").Cells(I + 2, 1).Value = I
Next I

'Print Regression Years on Left Side
If regressLN = True Then
   If regressDaily = True Then
      Worksheets("Scenario Output").Cells(displayPeriods + 4, 1).Value = "Regressions (LN
Returns, calculated Daily)"
   Else
```

```
        Worksheets("Scenario Output").Cells(displayPeriods + 4, 1).Value = "Regressions (LN
Returns, calculated Monthly)"
    End If
Else
    If regressDaily = True Then
        Worksheets("Scenario Output").Cells(displayPeriods + 4, 1).Value = "Regressions (Index
Values, calculated Daily)"
    Else
        Worksheets("Scenario Output").Cells(displayPeriods + 4, 1).Value = "Regressions (Index
Values, calculated Monthly)"
    End If
End If
Worksheets("Scenario Output").Cells(displayPeriods + 4, 1).Font.Size = 14

Worksheets("Scenario Output").Cells(displayPeriods + 5, 1).Value = regression & " YEAR"

''''''''''''''''''''''''''''
'Formatting Stuff Ends Here'
''''''''''''''''''''''''''''

''''''''''''''''''''''''''''
'Scenario Stuff Starts Here'
''''''''''''''''''''''''''''

'Start Scenario Loop
Do While scenarioCounter < scenarios

    'Randomize
    Randomize

    'Output Scenario Heading
    Worksheets("Scenario Output").Cells(1, scenarioCounter * 4 + 2).Value = "Scenario " &
scenarioCounter + 1
    Worksheets("Scenario Output").Cells(2, scenarioCounter * 4 + 2).Value = "MFI"
    Worksheets("Scenario Output").Cells(2, scenarioCounter * 4 + 3).Value = "SPY"
    Worksheets("Scenario Output").Cells(2, scenarioCounter * 4 + 4).Value = "RUS"

    'Reset Variables
    startvalue = Worksheets("Input").Range("D8").Value
    mfstartvalue = Worksheets("Input").Range("D12").Value
    russtartvalue = Worksheets("Input").Range("D16").Value
    newvalue = startvalue
    rusnewvalue = russtartvalue
    mfnewvalue = mfstartvalue
```

```
    If regimeSwitching = True Then
      randRegime = Rnd()
      If randRegime < (p21 / (p12 + p21)) Then
        regime = 1
        Else
        regime = 2
      End If
      Else
      regime = 1
    End If

    '''''''''''''''''''''''''''
    'Future Past Stuff Starts Here'
    '''''''''''''''''''''''''''

    'Start Daily Projection Loop for Future Past
    For currentmonth = 0 To projMonths - 1
          'set value as month end
          SPYmonthEnd = newvalue
          MFmonthEnd = mfnewvalue
          RUSmonthEnd = rusnewvalue

      If regimeSwitching = True Then
          'Evaluate regime

          'Reset Regime Random Number
          randRegime = Worksheets("Random Numbers").Cells(currentmonth + 2 +
scenarioCounter * 120, 2).Value

          'Set the Regime for the month
            If (regime = 1) Then
              If (randRegime < p11) Then
                  regime = 1
                Else
                  regime = 2
              End If
              Else
                If (randRegime < p22) Then
                  regime = 2
                  Else
                    regime = 1
                End If
            End If
```

```
        'Set the mean and standard deviation for SPY
        If (regime = 1) Then
          sigma = sigma1
          mu = mu1 - 0.5 * sigma ^ 2
          Else
          sigma = sigma2
          mu = mu2 - 0.5 * sigma ^ 2
        End If

        'Set the mean and standard deviation for RUS
        If (regime = 1) Then
          russigma = russigma1
          rusmu = rusmu1 - 0.5 * russigma ^ 2
          Else
          russigma = russigma2
          rusmu = rusmu2 - 0.5 * russigma ^ 2
        End If
      End If

      If (FPcalculate = True) And (FPdailyDisplay = True) Then
      'This top part of the If statement is used when the indexes are calcuated daily and
outputted daily

      ' Calculate and print values for the current month
        For I = 1 To 21

          'Previous New Values become Old Values
          oldvalue = newvalue
          rusoldvalue = rusnewvalue
          mfoldvalue = mfnewvalue

          'Generate "random" numbers
          randnum = Worksheets("Random Numbers").Cells(currentmonth * 21 + I +
scenarioCounter * projDays + 1, 6).Value
          rusrandnum = Worksheets("Random Numbers").Cells(currentmonth * 21 + I +
scenarioCounter * projDays + 1, 7).Value
          mfrandnum = Worksheets("Random Numbers").Cells(currentmonth * 21 + I +
scenarioCounter * projDays + 1, 8).Value

          randnorm = Application.WorksheetFunction.NormSInv(randnum)
          rusrandnorm = Application.WorksheetFunction.NormSInv(rusrandnum)
          mfrandnorm = Application.WorksheetFunction.NormSInv(mfrandnum)
```

```
            'Calculate returns
            SPYreturn = mu * timeinterval + sigma * randnorm * timeinterval ^ 0.5
            RUSreturn = rusmu * timeinterval + russigma * ((correlation * randnorm) + (1 -
correlation ^ 2) ^ 0.5 * rusrandnorm) * timeinterval ^ 0.5

            'Calculate new values
            newvalue = Exp(SPYreturn) * newvalue
            rusnewvalue = Exp(RUSreturn) * rusnewvalue

            'Calculate ln of Returns
            lnreturn = Log(newvalue / oldvalue)
            ruslnreturn = Log(rusnewvalue / rusoldvalue)
            mflnreturn = lnreturn + alpha / ((projDays / projYears) ^ 0.5) * mfrandnorm

            'Calculate Mutual Fund
            mfnewvalue = Exp(mflnreturn) * mfoldvalue

            'Print Values
            If FPprintType = True Then
               'Output ln of Returns
               Worksheets("Scenario Output").Cells(currentmonth * 21 + I + 2, scenarioCounter * 4
+ 2).Value = mflnreturn
               Worksheets("Scenario Output").Cells(currentmonth * 21 + I + 2, scenarioCounter * 4
+ 3).Value = lnreturn
               Worksheets("Scenario Output").Cells(currentmonth * 21 + I + 2, scenarioCounter * 4
+ 4).Value = ruslnreturn
            Else
               'Output Index Values
               Worksheets("Scenario Output").Cells(currentmonth * 21 + I + 2, scenarioCounter * 4
+ 2).Value = mfnewvalue
               Worksheets("Scenario Output").Cells(currentmonth * 21 + I + 2, scenarioCounter * 4
+ 3).Value = newvalue
               Worksheets("Scenario Output").Cells(currentmonth * 21 + I + 2, scenarioCounter * 4
+ 4).Value = rusnewvalue
            End If
         Next I
      ElseIf (FPcalculate = False) And (FPdailyDisplay = False) Then
      'This part of the If statement is used when the indexes are calcuated monthly and
outputted monthly

         ' Calculate and print values for the entire current month

         'Previous New Values become Old Values
         'oldvalue = newvalue
```

```
        'rusoldvalue = rusnewvalue
        'mfoldvalue = mfnewvalue

        'Generate "random" numbers
        'randnorm = Worksheets("Random Numbers").Cells(currentmonth + scenarioCounter *
projDays + 2, 6).Value
        'rusrandnorm = Worksheets("Random Numbers").Cells(currentmonth + scenarioCounter
* projDays + 2, 7).Value
        'mfrandnorm = Worksheets("Random Numbers").Cells(currentmonth + scenarioCounter
* projDays + 2, 8).Value

        'Calculate Returns for Regime Switching
        'SPYreturn = Application.WorksheetFunction.NormInv(randnorm, mu, sigma)
        'RUSreturn = Application.WorksheetFunction.NormInv(rusrandnorm, rusmu, russigma)
        'MFreturn = Application.WorksheetFunction.NormInv(mfrandnorm, mu, sigma)

        'Calculate new values
        'newvalue = Exp(SPYreturn) * newvalue
        'rusnewvalue = Exp(RUSreturn) * rusnewvalue
        'mfnewvalue = Exp(MFreturn) * mfnewvalue

        'Calculate ln of Returns
        'lnreturn = Log(newvalue / oldvalue)
        'ruslnreturn = Log(rusnewvalue / rusoldvalue)
        'mflnreturn = Log(mfnewvalue / mfoldvalue)

        'Print Values
        'If FPprintType = True Then
           'Output ln of Returns
        '   Worksheets("Scenario Output").Cells(currentmonth + 3, scenarioCounter * 4 +
2).Value = mflnreturn
        '   Worksheets("Scenario Output").Cells(currentmonth + 3, scenarioCounter * 4 +
3).Value = lnreturn
        '   Worksheets("Scenario Output").Cells(currentmonth + 3, scenarioCounter * 4 +
4).Value = ruslnreturn
        'Else
           'Output Index Values
        '   Worksheets("Scenario Output").Cells(currentmonth + 3, scenarioCounter * 4 +
2).Value = mfnewvalue
        '   Worksheets("Scenario Output").Cells(currentmonth + 3, scenarioCounter * 4 +
3).Value = newvalue
        '   Worksheets("Scenario Output").Cells(currentmonth + 3, scenarioCounter * 4 +
4).Value = rusnewvalue
        'End If
```

```
        ElseIf (FPcalculate = True) And (FPdailyDisplay = False) Then
        'This part of the If statement is used when the indexes are calcuated daily and outputted
monthly

        ' Calculate and values for the current month
          For I = 1 To 21

            'Previous New Values become Old Values
            oldvalue = newvalue
            rusoldvalue = rusnewvalue
            mfoldvalue = mfnewvalue

            'Generate "random" numbers
            randnum = Worksheets("Random Numbers").Cells(currentmonth * 21 + I +
scenarioCounter * projDays + 1, 6).Value
            rusrandnum = Worksheets("Random Numbers").Cells(currentmonth * 21 + I +
scenarioCounter * projDays + 1, 7).Value
            mfrandnum = Worksheets("Random Numbers").Cells(currentmonth * 21 + I +
scenarioCounter * projDays + 1, 8).Value

            randnorm = Application.WorksheetFunction.NormSInv(randnum)
            rusrandnorm = Application.WorksheetFunction.NormSInv(rusrandnum)
            mfrandnorm = Application.WorksheetFunction.NormSInv(mfrandnum)

            'Calculate Returns
            SPYreturn = mu * timeinterval + sigma * randnorm * timeinterval ^ 0.5
            RUSreturn = rusmu * timeinterval + russigma * ((correlation * randnorm) + (1 -
correlation ^ 2) ^ 0.5 * rusrandnorm) * timeinterval ^ 0.5

            'Calculate new values
            newvalue = Exp(SPYreturn) * newvalue
            rusnewvalue = Exp(RUSreturn) * rusnewvalue

            'Calculate ln of Returns
            lnreturn = Log(newvalue / oldvalue)
            ruslnreturn = Log(rusnewvalue / rusoldvalue)

            If dOfa = True Then
              mflnreturn = lnreturn + (alphaMu + alphaSigma * mfrandnorm) / 21
              Else
              mflnreturn = lnreturn + alphaMu * mfrandnorm / 21
            End If

            'Calculate Mutual Fund
```

```
            mfnewvalue = Exp(mflnreturn) * mfoldvalue

          Next l

            'Calculate ln of Returns for month
            lnreturn = Log(newvalue / SPYmonthEnd)
            ruslnreturn = Log(rusnewvalue / RUSmonthEnd)
            mflnreturn = Log(mfnewvalue / MFmonthEnd)

            If FPprintType = True Then
              'Output ln of Returns
              Worksheets("Scenario Output").Cells(currentmonth + 3, scenarioCounter * 4 +
2).Value = mflnreturn
              Worksheets("Scenario Output").Cells(currentmonth + 3, scenarioCounter * 4 +
3).Value = lnreturn
              Worksheets("Scenario Output").Cells(currentmonth + 3, scenarioCounter * 4 +
4).Value = ruslnreturn
              Worksheets("Scenario Output").Cells(currentmonth + 3, scenarioCounter * 4 +
5).Value = regime
            Else
              'Output Index Values
              Worksheets("Scenario Output").Cells(currentmonth + 3, scenarioCounter * 4 +
2).Value = mfnewvalue
              Worksheets("Scenario Output").Cells(currentmonth + 3, scenarioCounter * 4 +
3).Value = newvalue
              Worksheets("Scenario Output").Cells(currentmonth + 3, scenarioCounter * 4 +
4).Value = rusnewvalue
            End If
        End If
    Next currentmonth

    'Preserve Values
    FPmfFinal = mfnewvalue
    FPfinal = newvalue
    FPrusFinal = rusnewvalue

    'Print average return
    MFaverage = (FPmfFinal / mfstartvalue) ^ (1 / projYears) - 1
    SPYaverage = (FPfinal / startvalue) ^ (1 / projYears) - 1
    RUSaverage = (FPrusFinal / russtartvalue) ^ (1 / projYears) - 1

    Worksheets("Output").Cells(3, 9 + scenarioCounter).Value = MFaverage
    Worksheets("Output").Cells(2, 9 + scenarioCounter).Value = SPYaverage
    Worksheets("Output").Cells(4, 9 + scenarioCounter).Value = RUSaverage
```

```
   'Print Borders
   With Worksheets("Scenario Output").Columns(scenarioCounter * 4 + 5).Borders(xlEdgeRight)
      .LineStyle = xlContinuous
      .ColorIndex = xlAutomatic
      .TintAndShade = 0
      .Weight = xlThin
   End With
   With Worksheets("Scenario Output").Columns(scenarioCounter * 4 + 2).Borders(xlEdgeLeft)
      .LineStyle = xlContinuous
      .ColorIndex = xlAutomatic
      .TintAndShade = 0
      .Weight = xlThin
   End With

   'Color Columns
   With Worksheets("Scenario Output").Range(Cells(1, scenarioCounter * 4 + 2),
Cells(displayPeriods + 2, scenarioCounter * 4 + 2)).Interior
      .Pattern = xlSolid
      .PatternColorIndex = xlAutomatic
      .Color = 5296274
      .TintAndShade = 0
      .PatternTintAndShade = 0
   End With
   With Worksheets("Scenario Output").Range(Cells(1, scenarioCounter * 4 + 3),
Cells(displayPeriods + 2, scenarioCounter * 4 + 3)).Interior
      .Pattern = xlSolid
      .PatternColorIndex = xlAutomatic
      .Color = 14136213
      .TintAndShade = 0
      .PatternTintAndShade = 0
   End With
   With Worksheets("Scenario Output").Range(Cells(1, scenarioCounter * 4 + 4),
Cells(displayPeriods + 2, scenarioCounter * 4 + 4)).Interior
      .Pattern = xlSolid
      .PatternColorIndex = xlAutomatic
      .Color = 49407
      .TintAndShade = 0
      .PatternTintAndShade = 0
   End With

   'Merge Top Row
   With Worksheets("Scenario Output").Range(Cells(1, scenarioCounter * 4 + 2), Cells(1,
scenarioCounter * 4 + 4))
```

```
        .Merge
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlBottom
        .Interior.Color = 8421631
    End With

    '''''''''''''''''''''''''''''
    'Future Past Stuff Ends Here'
    '''''''''''''''''''''''''''''

    ''''''''''''''''''''''''''''''
    'Regression Stuff Starts Here'
    ''''''''''''''''''''''''''''''

    regressionOffset = displayPeriods

        '''''''''''''''''''''''''''''
        'POPULATING ARRAYS WITH DATA'
        '''''''''''''''''''''''''''''

        'This top part of the If statement is used when the regression is performed on all outputted
values
        If (FPdailyDisplay = True And regressDaily = True) Or (FPdailyDisplay = False And
regressDaily = False) Then

            ReDim yValues(1 To displayPeriods - (projYears - regression) * periodsPerYear) As Double
            ReDim x1values(1 To displayPeriods - (projYears - regression) * periodsPerYear) As
Double
            ReDim x2values(1 To displayPeriods - (projYears - regression) * periodsPerYear) As
Double
            ReDim xValues(1 To 2, 1 To displayPeriods - (projYears - regression) * periodsPerYear) As
Double

            'Checks if the scenarios are outputted in the same way (ln or values) that the regression
will be done
            If (FPprintType = True And regressLN = True) Or (FPprintType = False And regressLN =
False) Then

                'Populate the Arrays with the correct values
                For I = 1 To displayPeriods - (projYears - regression) * periodsPerYear
                    yValues(I) = Worksheets("Scenario Output").Cells((projYears - regression) *
periodsPerYear + I + 2, scenarioCounter * 4 + 2).Value
                    x1values(I) = Worksheets("Scenario Output").Cells((projYears - regression) *
periodsPerYear + I + 2, scenarioCounter * 4 + 3).Value
```

```
            x2values(I) = Worksheets("Scenario Output").Cells((projYears - regression) *
periodsPerYear + I + 2, scenarioCounter * 4 + 4).Value
            xValues(1, I) = x1values(I)
            xValues(2, I) = x2values(I)
        Next I

      Else
        'This is used if the scenarios are outputted as values, but the regression will be done
on ln
        If regressLN = True And FPprintType = False Then

            'Populate the Arrays with the correct values
            If regression = projYears Then
                yValues(1) = Log(Worksheets("Scenario Output").Cells(3, scenarioCounter * 4 +
2).Value / mfstartvalue)
                x1values(1) = Log(Worksheets("Scenario Output").Cells(3, scenarioCounter * 4 +
3).Value / startvalue)
                x2values(1) = Log(Worksheets("Scenario Output").Cells(3, scenarioCounter * 4 +
4).Value / russtartvalue)
            Else
                yValues(1) = Log(Worksheets("Scenario Output").Cells((projYears - regression) *
periodsPerYear + 3, scenarioCounter * 4 + 2).Value / Worksheets("Scenario
Output").Cells((projYears - regression) * periodsPerYear + 2, scenarioCounter * 4 + 2).Value)
                x1values(1) = Log(Worksheets("Scenario Output").Cells((projYears - regression) *
periodsPerYear + 3, scenarioCounter * 4 + 3).Value / Worksheets("Scenario
Output").Cells((projYears - regression) * periodsPerYear + 2, scenarioCounter * 4 + 3).Value)
                x2values(1) = Log(Worksheets("Scenario Output").Cells((projYears - regression) *
periodsPerYear + 3, scenarioCounter * 4 + 4).Value / Worksheets("Scenario
Output").Cells((projYears - regression) * periodsPerYear + 2, scenarioCounter * 4 + 4).Value)
            End If

            xValues(1, 1) = x1values(1)
            xValues(2, 1) = x2values(1)

            For I = 2 To displayPeriods - (projYears - regression) * periodsPerYear
                yValues(I) = Log(Worksheets("Scenario Output").Cells((projYears - regression) *
periodsPerYear + I + 2, scenarioCounter * 4 + 2).Value / Worksheets("Scenario
Output").Cells((projYears - regression) * periodsPerYear + I + 1, scenarioCounter * 4 + 2).Value)
                x1values(I) = Log(Worksheets("Scenario Output").Cells((projYears - regression) *
periodsPerYear + I + 2, scenarioCounter * 4 + 3).Value / Worksheets("Scenario
Output").Cells((projYears - regression) * periodsPerYear + I + 1, scenarioCounter * 4 + 3).Value)
                x2values(I) = Log(Worksheets("Scenario Output").Cells((projYears - regression) *
periodsPerYear + I + 2, scenarioCounter * 4 + 4).Value / Worksheets("Scenario
Output").Cells((projYears - regression) * periodsPerYear + I + 1, scenarioCounter * 4 + 4).Value)
```

```
            xValues(1, I) = x1values(I)
            xValues(2, I) = x2values(I)
          Next I

        Else
        'This is used if the scenarios are outputted as ln, but the regression will be done on values

          'Populate the Arrays with the correct values
          If regression = projYears Then
            yValues(1) = mfstartvalue * Exp(Worksheets("Scenario Output").Cells(3, scenarioCounter * 4 + 2).Value)
            x1values(1) = startvalue * Exp(Worksheets("Scenario Output").Cells(3, scenarioCounter * 4 + 3).Value)
            x2values(1) = russtartvalue * Exp(Worksheets("Scenario Output").Cells(3, scenarioCounter * 4 + 4).Value)
          Else
            yValues(1) = mfstartvalue * Exp(Application.WorksheetFunction.Sum(Worksheets("Scenario Output").Range(Cells(3, scenarioCounter * 4 + 2), Cells((projYears - regression) * periodsPerYear + 3, scenarioCounter * 4 + 2)).Value))
            x1values(1) = startvalue * Exp(Application.WorksheetFunction.Sum(Worksheets("Scenario Output").Range(Cells(3, scenarioCounter * 4 + 3), Cells((projYears - regression) * periodsPerYear + 3, scenarioCounter * 4 + 3)).Value))
            x2values(1) = russtartvalue * Exp(Application.WorksheetFunction.Sum(Worksheets("Scenario Output").Range(Cells(3, scenarioCounter * 4 + 4), Cells((projYears - regression) * periodsPerYear + 3, scenarioCounter * 4 + 4)).Value))
          End If

          xValues(1, 1) = x1values(1)
          xValues(2, 1) = x2values(1)

          For I = 2 To projDays - (projYears - regression) * periodsPerYear
            yValues(I) = yValues(I - 1) * Exp(Worksheets("Scenario Output").Cells((projYears - regression) * periodsPerYear + 2 + I, scenarioCounter * 4 + 2).Value)
            x1values(I) = x1values(I - 1) * Exp(Worksheets("Scenario Output").Cells((projYears - regression) * periodsPerYear + 2 + I, scenarioCounter * 4 + 3).Value)
            x2values(I) = x2values(I - 1) * Exp(Worksheets("Scenario Output").Cells((projYears - regression) * periodsPerYear + 2 + I, scenarioCounter * 4 + 4).Value)
            xValues(1, I) = x1values(I)
            xValues(2, I) = x2values(I)
          Next I
```

```
            End If

         End If

      'This portion is used when the indexes are outputted daily, but regression is done monthly
      Else

      ReDim yValues(1 To (displayPeriods - (projYears - regression) * periodsPerYear) / 21) As
Double
      ReDim x1values(1 To (displayPeriods - (projYears - regression) * periodsPerYear) / 21) As
Double
      ReDim x2values(1 To (displayPeriods - (projYears - regression) * periodsPerYear) / 21) As
Double
      ReDim xValues(1 To 2, 1 To (displayPeriods - (projYears - regression) * periodsPerYear) /
21) As Double

         'Checks if the scenarios are outputted in the same way (ln or values) that the regression
will be done
         If (FPprintType = True And regressLN = True) Or (FPprintType = False And regressLN =
False) Then

            'Populate the Arrays with the correct values (i*21 because 21 trading days per month)
            For I = 1 To (projDays - (projYears - currentRegression) * periodsPerYear) / 21
               yValues(I) = Worksheets("Scenario Output").Cells((projYears - regression) *
periodsPerYear + I * 21 + 2, scenarioCounter * 4 + 2).Value
               x1values(I) = Worksheets("Scenario Output").Cells((projYears - regression) *
periodsPerYear + I * 21 + 2, scenarioCounter * 4 + 3).Value
               x2values(I) = Worksheets("Scenario Output").Cells((projYears - regression) *
periodsPerYear + I * 21 + 2, scenarioCounter * 4 + 4).Value
               xValues(1, I) = x1values(I)
               xValues(2, I) = x2values(I)
            Next I

         Else
            'This is used if the scenarios are outputted as values, but the regression will be done
on ln
            If regressLN = True And FPprintType = False Then

               'Populate the Arrays with the correct values
               If currentRegression = projYears Then
                  yValues(1) = Log(Worksheets("Scenario Output").Cells(23, scenarioCounter * 4 +
2).Value / mfstartvalue)
```

```
            x1values(1) = Log(Worksheets("Scenario Output").Cells(23, scenarioCounter * 4 +
3).Value / startvalue)
            x2values(1) = Log(Worksheets("Scenario Output").Cells(23, scenarioCounter * 4 +
4).Value / russtartvalue)
          Else
            yValues(1) = Log(Worksheets("Scenario Output").Cells((projYears - regression) *
periodsPerYear + 23, scenarioCounter * 4 + 2).Value / Worksheets("Scenario
Output").Cells((projYears - regression) * periodsPerYear + 2, scenarioCounter * 4 + 2).Value)
            x1values(1) = Log(Worksheets("Scenario Output").Cells((projYears - regression) *
periodsPerYear + 23, scenarioCounter * 4 + 3).Value / Worksheets("Scenario
Output").Cells((projYears - regression) * periodsPerYear + 2, scenarioCounter * 4 + 3).Value)
            x2values(1) = Log(Worksheets("Scenario Output").Cells((projYears - regression) *
periodsPerYear + 23, scenarioCounter * 4 + 4).Value / Worksheets("Scenario
Output").Cells((projYears - regression) * periodsPerYear + 2, scenarioCounter * 4 + 4).Value)
          End If

          xValues(1, 1) = x1values(1)
          xValues(2, 1) = x2values(1)

          For I = 2 To (projDays - (projYears - regression) * periodsPerYear) / 21
            yValues(I) = Log(Worksheets("Scenario Output").Cells((projYears - regression) *
periodsPerYear + I * 21 + 2, scenarioCounter * 4 + 2).Value / Worksheets("Scenario
Output").Cells((projYears - regression) * periodsPerYear + (I - 1) * 21 + 2, scenarioCounter * 4 +
2).Value)
            x1values(I) = Log(Worksheets("Scenario Output").Cells((projYears - regression) *
periodsPerYear + I * 21 + 2, scenarioCounter * 4 + 3).Value / Worksheets("Scenario
Output").Cells((projYears - regression) * periodsPerYear + (I - 1) * 21 + 2, scenarioCounter * 4 +
3).Value)
            x2values(I) = Log(Worksheets("Scenario Output").Cells((projYears - regression) *
periodsPerYear + I * 21 + 2, scenarioCounter * 4 + 4).Value / Worksheets("Scenario
Output").Cells((projYears - regression) * periodsPerYear + (I - 1) * 21 + 2, scenarioCounter * 4 +
4).Value)
            xValues(1, I) = x1values(I)
            xValues(2, I) = x2values(I)
          Next I

        Else
        'This is used if the scenarios are outputted as ln, but the regression will be done on
values

          'Populate the Arrays with the correct values
          If regression = projYears Then
```

```
                yValues(1) = mfstartvalue *
Exp(Application.WorksheetFunction.Sum(Worksheets("Scenario Output").Range(Cells(3,
scenarioCounter * 4 + 2), Cells(23, scenarioCounter * 4 + 2)).Value))
                x1values(1) = startvalue *
Exp(Application.WorksheetFunction.Sum(Worksheets("Scenario Output").Range(Cells(3,
scenarioCounter * 4 + 3), Cells(23, scenarioCounter * 4 + 3)).Value))
                x2values(1) = russtartvalue *
Exp(Application.WorksheetFunction.Sum(Worksheets("Scenario Output").Range(Cells(3,
scenarioCounter * 4 + 4), Cells(23, scenarioCounter * 4 + 4)).Value))
            Else
                yValues(1) = mfstartvalue *
Exp(Application.WorksheetFunction.Sum(Worksheets("Scenario Output").Range(Cells(3,
scenarioCounter * 4 + 2), Cells((projYears - currentRegression) * periodsPerYear + 23,
scenarioCounter * 4 + 2)).Value))

                x1values(1) = startvalue *
Exp(Application.WorksheetFunction.Sum(Worksheets("Scenario Output").Range(Cells(3,
scenarioCounter * 4 + 3), Cells((projYears - currentRegression) * periodsPerYear + 23,
scenarioCounter * 4 + 3)).Value))
                x2values(1) = russtartvalue *
Exp(Application.WorksheetFunction.Sum(Worksheets("Scenario Output").Range(Cells(3,
scenarioCounter * 4 + 4), Cells((projYears - currentRegression) * periodsPerYear + 23,
scenarioCounter * 4 + 4)).Value))
            End If

            xValues(1, 1) = x1values(1)
            xValues(2, 1) = x2values(1)

            For I = 2 To (projDays - (projYears - regression) * periodsPerYear) / 21
                yValues(I) = yValues(I - 1) *
Exp(Application.WorksheetFunction.Sum(Worksheets("Scenario
Output").Range(Cells((projYears - regression) * periodsPerYear + 3 + (I - 1) * 21,
scenarioCounter * 4 + 2), Cells((projYears - regression) * periodsPerYear + 2 + I * 21,
scenarioCounter * 4 + 2)).Value))
                x1values(I) = x1values(I - 1) *
Exp(Application.WorksheetFunction.Sum(Worksheets("Scenario
Output").Range(Cells((projYears - regression) * periodsPerYear + 3 + (I - 1) * 21,
scenarioCounter * 4 + 3), Cells((projYears - regression) * periodsPerYear + 2 + I * 21,
scenarioCounter * 4 + 3)).Value))
                x2values(I) = x2values(I - 1) *
Exp(Application.WorksheetFunction.Sum(Worksheets("Scenario
Output").Range(Cells((projYears - regression) * periodsPerYear + 3 + (I - 1) * 21,
scenarioCounter * 4 + 4), Cells((projYears - regression) * periodsPerYear + 2 + I * 21,
scenarioCounter * 4 + 4)).Value))
```

```
                xValues(1, I) = x1values(I)
                xValues(2, I) = x2values(I)
            Next I

        End If

    End If

End If

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'POPULATING ARRAYS WITH DATA IS DONE. NOW, REGRESS THE ARRAYS'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'2 Variable Regression
Worksheets("Scenario Output").Range(Cells(regressionOffset + 9, scenarioCounter * 4 + 2), Cells(regressionOffset + 13, scenarioCounter * 4 + 4)).Value = WorksheetFunction.LinEst(yValues, xValues, True, True)

'Formatting for 2 Variable Regression
Worksheets("Scenario Output").Cells(regressionOffset + 6, scenarioCounter * 4 + 2).Value = "2-VAR"
Worksheets("Scenario Output").Cells(regressionOffset + 7, scenarioCounter * 4 + 2).Value = "SPY"
Worksheets("Scenario Output").Cells(regressionOffset + 8, scenarioCounter * 4 + 2).Value = "RUS"

twoIntercept = Worksheets("Scenario Output").Cells(regressionOffset + 9, scenarioCounter * 4 + 4).Value
SP2VarCoeff = Worksheets("Scenario Output").Cells(regressionOffset + 9, scenarioCounter * 4 + 3).Value
RUS2VarCoeff = Worksheets("Scenario Output").Cells(regressionOffset + 9, scenarioCounter * 4 + 2).Value

Worksheets("Scenario Output").Cells(regressionOffset + 6, scenarioCounter * 4 + 3).Value = twoIntercept
Worksheets("Scenario Output").Cells(regressionOffset + 7, scenarioCounter * 4 + 3).Value = SP2VarCoeff
Worksheets("Scenario Output").Cells(regressionOffset + 8, scenarioCounter * 4 + 3).Value = RUS2VarCoeff

Worksheets("Scenario Output").Range(Cells(regressionOffset + 6, scenarioCounter * 4 + 2), Cells(regressionOffset + 8, scenarioCounter * 4 + 3)).Font.FontStyle = "Bold"
```

```
    Worksheets("Scenario Output").Range(Cells(regressionOffset + 11, scenarioCounter * 4 +
4), Cells(regressionOffset + 13, scenarioCounter * 4 + 4)).Value = ""
    Worksheets("Scenario Output").Cells(regressionOffset + 12, scenarioCounter * 4 +
2).NumberFormat = "0.00"
    Worksheets("Scenario Output").Cells(regressionOffset + 12, scenarioCounter * 4 +
3).NumberFormat = "0"
    If regressLN = False Then
      Worksheets("Scenario Output").Cells(regressionOffset + 13, scenarioCounter * 4 +
2).NumberFormat = "0"
      Worksheets("Scenario Output").Cells(regressionOffset + 13, scenarioCounter * 4 +
3).NumberFormat = "0"
    End If

    With Worksheets("Scenario Output").Rows(regressionOffset + 13).Borders(xlEdgeBottom)
      .LineStyle = xlContinuous
      .ColorIndex = xlAutomatic
      .TintAndShade = 0
      .Weight = xlThin
    End With
    With Worksheets("Scenario Output").Rows(regressionOffset + 14).Borders(xlEdgeTop)
      .LineStyle = xlContinuous
      .ColorIndex = xlAutomatic
      .TintAndShade = 0
      .Weight = xlThin
    End With

    'Output Formula for 2 Variable Regression
    Worksheets("Scenario Output").Cells(regressionOffset + 5, scenarioCounter * 4 + 2).Value =
"MFIB= " & Round(twoIntercept, 4) & " + " & Round(SP2VarCoeff, 4) & "*SPY + " &
Round(RUS2VarCoeff, 4) & "*RUS"
    Worksheets("Scenario Output").Cells(regressionOffset + 5, scenarioCounter * 4 +
2).Font.FontStyle = "Bold"

    'SPY Regression
    Worksheets("Scenario Output").Range(Cells(regressionOffset + 17, scenarioCounter * 4 +
2), Cells(regressionOffset + 21, scenarioCounter * 4 + 3)).Value =
WorksheetFunction.LinEst(yValues, x1values, True, True)

    'Formatting for SPY Regression
    Worksheets("Scenario Output").Cells(regressionOffset + 15, scenarioCounter * 4 + 2).Value
= "1-VAR"
    Worksheets("Scenario Output").Cells(regressionOffset + 16, scenarioCounter * 4 + 2).Value
= "SPY"
```

```
    SPintercept = Worksheets("Scenario Output").Cells(regressionOffset + 17, scenarioCounter
* 4 + 3).Value
    SP1VarCoeff = Worksheets("Scenario Output").Cells(regressionOffset + 17,
scenarioCounter * 4 + 2).Value

    Worksheets("Scenario Output").Cells(regressionOffset + 15, scenarioCounter * 4 + 3).Value
= SPintercept
    Worksheets("Scenario Output").Cells(regressionOffset + 16, scenarioCounter * 4 + 3).Value
= SP1VarCoeff

    Worksheets("Scenario Output").Range(Cells(regressionOffset + 15, scenarioCounter * 4 +
2), Cells(regressionOffset + 16, scenarioCounter * 4 + 3)).Font.FontStyle = "Bold"
    Worksheets("Scenario Output").Cells(regressionOffset + 20, scenarioCounter * 4 +
2).NumberFormat = "0.00"
    Worksheets("Scenario Output").Cells(regressionOffset + 20, scenarioCounter * 4 +
3).NumberFormat = "0"
    If regressLN = False Then
      Worksheets("Scenario Output").Cells(regressionOffset + 21, scenarioCounter * 4 +
2).NumberFormat = "0"
      Worksheets("Scenario Output").Cells(regressionOffset + 21, scenarioCounter * 4 +
3).NumberFormat = "0"
    End If

    With Worksheets("Scenario Output").Rows(regressionOffset + 21).Borders(xlEdgeBottom)
      .LineStyle = xlContinuous
      .ColorIndex = xlAutomatic
      .TintAndShade = 0
      .Weight = xlThin
    End With
    With Worksheets("Scenario Output").Rows(regressionOffset + 22).Borders(xlEdgeTop)
      .LineStyle = xlContinuous
      .ColorIndex = xlAutomatic
      .TintAndShade = 0
      .Weight = xlThin
    End With

    'Output Formula for SPY Regression
    Worksheets("Scenario Output").Cells(regressionOffset + 14, scenarioCounter * 4 + 2).Value
= "MFIC = " & Round(SPintercept, 4) & " + " & Round(SP1VarCoeff, 4) & "*SPY"
    Worksheets("Scenario Output").Cells(regressionOffset + 14, scenarioCounter * 4 +
2).Font.FontStyle = "Bold"

    'RUS Regression
```

```
        Worksheets("Scenario Output").Range(Cells(regressionOffset + 25, scenarioCounter * 4 +
2), Cells(regressionOffset + 29, scenarioCounter * 4 + 3)).Value =
WorksheetFunction.LinEst(yValues, x2values, True, True)

        'Formatting for RUS Regression
        Worksheets("Scenario Output").Cells(regressionOffset + 23, scenarioCounter * 4 + 2).Value
= "1-VAR"
        Worksheets("Scenario Output").Cells(regressionOffset + 24, scenarioCounter * 4 + 2).Value
= "RUS"

        RUSintercept = Worksheets("Scenario Output").Cells(regressionOffset + 25,
scenarioCounter * 4 + 3).Value
        RUS1VarCoeff = Worksheets("Scenario Output").Cells(regressionOffset + 25,
scenarioCounter * 4 + 2).Value

        Worksheets("Scenario Output").Cells(regressionOffset + 23, scenarioCounter * 4 + 3).Value
= RUSintercept
        Worksheets("Scenario Output").Cells(regressionOffset + 24, scenarioCounter * 4 + 3).Value
= RUS1VarCoeff

        Worksheets("Scenario Output").Range(Cells(regressionOffset + 23, scenarioCounter * 4 +
2), Cells(regressionOffset + 24, scenarioCounter * 4 + 3)).Font.FontStyle = "Bold"
        Worksheets("Scenario Output").Cells(regressionOffset + 28, scenarioCounter * 4 +
2).NumberFormat = "0.00"
        Worksheets("Scenario Output").Cells(regressionOffset + 28, scenarioCounter * 4 +
3).NumberFormat = "0"
        If regressLN = False Then
          Worksheets("Scenario Output").Cells(regressionOffset + 29, scenarioCounter * 4 +
2).NumberFormat = "0"
          Worksheets("Scenario Output").Cells(regressionOffset + 29, scenarioCounter * 4 +
3).NumberFormat = "0"
        End If

        With Worksheets("Scenario Output").Rows(regressionOffset + 29).Borders(xlEdgeBottom)
          .LineStyle = xlContinuous
          .ColorIndex = xlAutomatic
          .TintAndShade = 0
          .Weight = xlMedium
        End With
        With Worksheets("Scenario Output").Rows(regressionOffset + 30).Borders(xlEdgeTop)
          .LineStyle = xlContinuous
          .ColorIndex = xlAutomatic
          .TintAndShade = 0
          .Weight = xlMedium
```

```
        End With

        'Output Formula for RUS Regression
        Worksheets("Scenario Output").Cells(regressionOffset + 22, scenarioCounter * 4 + 2).Value
= "MFRUS = " & Round(RUSintercept, 4) & " + " & Round(RUS1VarCoeff, 4) & "*RUS"
        Worksheets("Scenario Output").Cells(regressionOffset + 22, scenarioCounter * 4 +
2).Font.FontStyle = "Bold"

    'Fill with Color
    Worksheets("Scenario Output").Range(Cells(displayPeriods + 4, scenarioCounter * 4 + 2),
Cells(25 + displayPeriods + 4, scenarioCounter * 4 + 5)).Interior.Color = 8421631

    ''''''''''''''''''''''''''
    'Regression Stuff Ends Here'
    ''''''''''''''''''''''''''

    ''''''''''''''''''''''''''''''
    'Future Future Stuff Starts Here'
    ''''''''''''''''''''''''''''''

    'Start Scenario Worksheet Loop
     For FFcurrScenario = 0 To FFscenarios - 1

        'Display Period Numbers
        If scenarioCounter = 0 Then
            '(Only do it once)
            FFheaderOutput = 25 + displayPeriods + (FFdisplayPeriods + 2) * FFcurrScenario * 2
            If FFdailyDisplay = False Then
                If FFcalculate = True Then
                    If FFprintType = True Then
                        'For Monthly periods with daily LN Return calculations
                        Worksheets("Scenario Output").Cells(7 + FFheaderOutput, 1).Value = "Months (LN
Returns, calculated Daily)"
                    Else
                        'For Monthly periods with monthly Value calculations
                        Worksheets("Scenario Output").Cells(7 + FFheaderOutput, 1).Value = "Months
(Index Values, calculated Daily)"
                    End If
                Else
                    If FFprintType = True Then
                        'For Monthly periods with monthly LN Return calculations
                        Worksheets("Scenario Output").Cells(7 + FFheaderOutput, 1).Value = "Months (LN
Returns, calculated Monthly)"
                    Else
```

```
                    'For Monthly periods with monthly Value calculations
                    Worksheets("Scenario Output").Cells(7 + FFheaderOutput, 1).Value = "Months
(Index Values, calculated Monthly)"
                End If
            End If
        Else
            If FFprintType = True Then
                'For Daily periods with daily LN Return calculations
                Worksheets("Scenario Output").Cells(7 + FFheaderOutput, 1).Value = "Days (LN
Returns, calculated Daily)"
            Else
                'For Daily periods with daily Value calculations
                Worksheets("Scenario Output").Cells(7 + FFheaderOutput, 1).Value = "Days (Index
Values, calculated Daily)"
            End If
        End If
        For I = 0 To FFdisplayPeriods - 1
            Worksheets("Scenario Output").Cells(8 + I + FFheaderOutput, 1).Value = I + 1
        Next I
    End If

    'Display Period Numbers for Regression MF's
    If scenarioCounter = 0 Then
        '(Only do it once)
            FFheaderOutput = 30 + displayPeriods + (FFdisplayPeriods + 2) * FFcurrScenario * 2 +
(FFdisplayPeriods + 2)
            If FFdailyDisplay = False Then
                If FFcalculate = True Then
                    If FFprintType = True Then
                        'For Monthly periods with daily LN Return calculations
                        Worksheets("Scenario Output").Cells(2 + FFheaderOutput, 1).Value = "Months
(LN Returns, calculated Daily)"
                    Else
                        'For Monthly periods with monthly Value calculations
                        Worksheets("Scenario Output").Cells(2 + FFheaderOutput, 1).Value = "Months
(Index Values, calculated Daily)"
                    End If
                Else
                    If FFprintType = True Then
                        'For Monthly periods with monthly LN Return calculations
                        Worksheets("Scenario Output").Cells(2 + FFheaderOutput, 1).Value = "Months
(LN Returns, calculated Monthly)"
                    Else
                        'For Monthly periods with monthly Value calculations
```

```
                    Worksheets("Scenario Output").Cells(2 + FFheaderOutput, 1).Value = "Months
(Index Values, calculated Monthly)"
                End If
            End If
        Else
            If FFprintType = True Then
                'For Daily periods with daily LN Return calculations
                Worksheets("Scenario Output").Cells(2 + FFheaderOutput, 1).Value = "Days (LN
Returns, calculated Daily)"
            Else
                'For Daily periods with daily Value calculations
                Worksheets("Scenario Output").Cells(2 + FFheaderOutput, 1).Value = "Days (Index
Values, calculated Daily)"
            End If
        End If
        For j = 1 To FFdisplayPeriods
            Worksheets("Scenario Output").Cells(2 + j + FFheaderOutput, 1).Value = j
        Next j
    End If

    'Output Scenario Heading
    FFheaderOutput = 25 + displayPeriods + (FFdisplayPeriods + 2) * FFcurrScenario * 2
    Worksheets("Scenario Output").Cells(6 + FFheaderOutput, scenarioCounter * 4 + 2).Value =
"Scenario " & scenarioCounter + 1 & "." & FFcurrScenario + 1
    Worksheets("Scenario Output").Cells(7 + FFheaderOutput, scenarioCounter * 4 + 2).Value =
"MFI"
    Worksheets("Scenario Output").Cells(7 + FFheaderOutput, scenarioCounter * 4 + 3).Value =
"SPY"
    Worksheets("Scenario Output").Cells(7 + FFheaderOutput, scenarioCounter * 4 + 4).Value =
"RUS"

    'Reset Variables
    newvalue = FPfinal
    rusnewvalue = FPrusFinal
    mfnewvalue = FPmfFinal
    FFmfiBnewvalue = FPmfFinal
    FFmfiCnewvalue = FPmfFinal
    FFmfiDnewvalue = FPmfFinal

    'Color Columns
    FFheaderOutput = 25 + displayPeriods + (FFdisplayPeriods + 2) * FFcurrScenario * 2
    With Worksheets("Scenario Output").Range(Cells(6 + FFheaderOutput, scenarioCounter * 4
+ 2), Cells(7 + FFheaderOutput + FFdisplayPeriods, scenarioCounter * 4 + 2)).Interior
        .Pattern = xlSolid
```

```
            .PatternColorIndex = xlAutomatic
            .Color = 5296274
            .TintAndShade = 0
            .PatternTintAndShade = 0
        End With
        With Worksheets("Scenario Output").Range(Cells(6 + FFheaderOutput, scenarioCounter * 4
+ 3), Cells(7 + FFheaderOutput + FFdisplayPeriods, scenarioCounter * 4 + 3)).Interior
            .Pattern = xlSolid
            .PatternColorIndex = xlAutomatic
            .Color = 14136213
            .TintAndShade = 0
            .PatternTintAndShade = 0
        End With
        With Worksheets("Scenario Output").Range(Cells(6 + FFheaderOutput, scenarioCounter * 4
+ 4), Cells(7 + FFheaderOutput + FFdisplayPeriods, scenarioCounter * 4 + 4)).Interior
            .Pattern = xlSolid
            .PatternColorIndex = xlAutomatic
            .Color = 49407
            .TintAndShade = 0
            .PatternTintAndShade = 0
        End With

        'Merge Top Row
        With Worksheets("Scenario Output").Range(Cells(6 + FFheaderOutput, scenarioCounter * 4
+ 2), Cells(6 + FFheaderOutput, scenarioCounter * 4 + 5))
            .Merge
            .HorizontalAlignment = xlCenter
            .VerticalAlignment = xlBottom
            .Interior.Color = 8421631
        End With

            'Output Scenario Heading
            FFheaderOutput = displayPeriods + 30 + (FFdisplayPeriods + 2) + FFcurrScenario * 2 *
(FFdisplayPeriods + 2)
            Worksheets("Scenario Output").Cells(FFheaderOutput + 1, scenarioCounter * 4 +
2).Value = "Scenario " & scenarioCounter + 1 & "." & FFcurrScenario + 1 & " (" & regression & "-
Year Regression)"
            Worksheets("Scenario Output").Cells(FFheaderOutput + 2, scenarioCounter * 4 +
2).Value = "MFI 2Var"
            Worksheets("Scenario Output").Cells(FFheaderOutput + 2, scenarioCounter * 4 +
3).Value = "MFI SPY"
            Worksheets("Scenario Output").Cells(FFheaderOutput + 2, scenarioCounter * 4 +
4).Value = "MFI RUS"
```

```
        'Color Columns
        With Worksheets("Scenario Output").Range(Cells(2 + FFheaderOutput, scenarioCounter
* 4 + 2), Cells(2 + FFheaderOutput + FFdisplayPeriods, scenarioCounter * 4 + 2)).Interior
            .Pattern = xlSolid
            .PatternColorIndex = xlAutomatic
            .Color = 5296274
            .TintAndShade = 0
            .PatternTintAndShade = 0
        End With
        With Worksheets("Scenario Output").Range(Cells(2 + FFheaderOutput, scenarioCounter
* 4 + 3), Cells(2 + FFheaderOutput + FFdisplayPeriods, scenarioCounter * 4 + 3)).Interior
            .Pattern = xlSolid
            .PatternColorIndex = xlAutomatic
            .Color = 8575688
            .TintAndShade = 0
            .PatternTintAndShade = 0
        End With
        With Worksheets("Scenario Output").Range(Cells(2 + FFheaderOutput, scenarioCounter
* 4 + 4), Cells(2 + FFheaderOutput + FFdisplayPeriods, scenarioCounter * 4 + 4)).Interior
            .Pattern = xlSolid
            .PatternColorIndex = xlAutomatic
            .Color = 11855054
            .TintAndShade = 0
            .PatternTintAndShade = 0
        End With

        'Merge Top Row
        With Worksheets("Scenario Output").Range(Cells(1 + FFheaderOutput, scenarioCounter
* 4 + 2), Cells(1 + FFheaderOutput, scenarioCounter * 4 + 5))
            .Merge
            .HorizontalAlignment = xlCenter
            .VerticalAlignment = xlBottom
            .Interior.Color = 8421631
        End With

    'Start Projection Loop
    For FFcurrentPeriod = 0 To FFprojMonths - 1
        'set value as month end
        SPYmonthEnd = newvalue
        MFmonthEnd = mfnewvalue
        RUSmonthEnd = rusnewvalue

        If regimeSwitching = True Then
            'Evaluate Regime
```

```
            'Reset Regime Random Number
            randRegime = Worksheets("Random Numbers").Cells(FFcurrentPeriod + 2 + 120 *
FFcurrScenario, 3).Value

            'Set the Regime for the month
            If (regime = 1) Then
               If (randRegime < p11) Then
                     regime = 1
                  Else
                     regime = 2
               End If
               Else
                  If (randRegime < p22) Then
                     regime = 2
                     Else
                        regime = 1
                  End If
            End If

            'Set the mean and standard deviation for SPY
            If (regime = 1) Then
               sigma = sigma1
               mu = mu1 - 0.5 * sigma ^ 2
               Else
               sigma = sigma2
               mu = mu2 - 0.5 * sigma ^ 2
            End If

            'Set the mean and standard deviation for RUS
            If (regime = 1) Then
               russigma = russigma1
               rusmu = rusmu1 - 0.5 * russigma ^ 2
               Else
               russigma = russigma2
               rusmu = rusmu2 - 0.5 * russigma ^ 2
            End If
         End If

         If (FFcalculate = True) And (FFdailyDisplay = True) Then
         'This top part of the If statement is used when the indexes are calcuated daily and
outputted daily

         ' Calculate and print values for the current month
```

```
For I = 1 To 21

    'Previous New Values become Old Values
    oldvalue = newvalue
    rusoldvalue = rusnewvalue
    mfoldvalue = mfnewvalue

    'Generate "random" numbers
    randnorm = Worksheets("Random Numbers").Cells(FFcurrentPeriod * 21 + I +
FFcurrScenario * projDays + 1, 6 + 5 * scenarioCounter).Value
    rusrandnorm = Worksheets("Random Numbers").Cells(FFcurrentPeriod * 21 + I +
FFcurrScenario * projDays + 1, 7 + 5 * scenarioCounter).Value
    mfrandnorm = Worksheets("Random Numbers").Cells(FFcurrentPeriod * 21 + I +
FFcurrScenario * projDays + 1, 8 + 5 * scenarioCounter).Value

    'Calculate Returns for Regime Switching
    SPYreturn = Application.WorksheetFunction.NormInv(randnorm, mu, sigma)
    RUSreturn = Application.WorksheetFunction.NormInv(rusrandnorm, rusmu,
russigma)
    MFreturn = Application.WorksheetFunction.NormInv(mfrandnorm, mu, sigma)

    'Calculate new values
    newvalue = Exp(SPYreturn) * newvalue
    rusnewvalue = Exp(RUSreturn) * rusnewvalue
    mfnewvalue = Exp(MFreturn) * mfnewvalue

    'Calculate ln of Returns
    lnreturn = Log(newvalue / oldvalue)
    ruslnreturn = Log(rusnewvalue / rusoldvalue)
    mflnreturn = Log(mfnewvalue / mfoldvalue)

    'Output Values
    FFrowOutput = 8 + 25 + displayPeriods + FFcurrentPeriod + FFcurrScenario *
(FFdisplayPeriods + 2) * 2
    If FFprintType = True Then
      'Output ln of Returns
      Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 +
2).Value = mflnreturn
      Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 +
3).Value = lnreturn
      Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 +
4).Value = ruslnreturn
    Else
      'Output Index Values
```

```
                Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 +
2).Value = mfnewvalue
                Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 +
3).Value = newvalue
                Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 +
4).Value = rusnewvalue
              End If
            Next I
          ElseIf (FFcalculate = False) And (FFdailyDisplay = False) Then
          'This part of the If statement is used when the indexes are calcuated monthly and
outputted monthly

            ' Calculate and print values for the entire current month

            'Previous New Values become Old Values
            'oldvalue = newvalue
            'rusoldvalue = rusnewvalue
            'mfoldvalue = mfnewvalue

            'Generate "random" numbers
            'randnorm = Worksheets("Random Numbers").Cells(FFcurrentPeriod + FFcurrScenario
* projDays + 2, 6 + 5 * scenarioCounter).Value
            'rusrandnorm = Worksheets("Random Numbers").Cells(FFcurrentPeriod +
FFcurrScenario * projDays + 2, 7 + 5 * scenarioCounter).Value
            'mfrandnorm = Worksheets("Random Numbers").Cells(FFcurrentPeriod +
FFcurrScenario * projDays + 2, 8 + 5 * scenarioCounter).Value

            'Calculate Returns for Regime Switching
            'SPYreturn = Application.WorksheetFunction.NormInv(randnorm, mu, sigma)
            'RUSreturn = Application.WorksheetFunction.NormInv(rusrandnorm, rusmu, russigma)
            'MFreturn = Application.WorksheetFunction.NormInv(mfrandnorm, mu, sigma)

            'Calculate new values
            'newvalue = Exp(SPYreturn) * newvalue
            'rusnewvalue = Exp(RUSreturn) * rusnewvalue
            'mfnewvalue = Exp(MFreturn) * mfnewvalue

            'Calculate ln of Returns
            'lnreturn = Log(newvalue / oldvalue)
            'ruslnreturn = Log(rusnewvalue / rusoldvalue)
            'mflnreturn = Log(mfnewvalue / mfoldvalue)

            'Output Values
```

```
            'FFrowOutput = 8 + 25 + displayPeriods + FFcurrentPeriod + FFcurrScenario *
(FFdisplayPeriods + 2) * 2
            'If FFprintType = True Then
            '   'Output ln of Returns
            '   Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 + 2).Value
= mflnreturn
            '   Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 + 3).Value
= lnreturn
            '   Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 + 4).Value
= ruslnreturn
            'Else
               'Output Index Values
            '   Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 + 2).Value
= mfnewvalue
            '   Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 + 3).Value
= newvalue
            '   Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 + 4).Value
= rusnewvalue
            'End If

        ElseIf (FFcalculate = True) And (FFdailyDisplay = False) Then
        'This part of the If statement is used when the indexes are calcuated daily and outputted
monthly

        ' Calculate and values for the current month
            For I = 1 To 21

            'Previous New Values become Old Values
            oldvalue = newvalue
            rusoldvalue = rusnewvalue
            mfoldvalue = mfnewvalue

            'Generate "random" numbers
            randnum = Worksheets("Random Numbers").Cells(FFcurrentPeriod * 21 + I +
scenarioCounter * projDays + 1, 11 + FFcurrScenario * 5).Value
            rusrandnum = Worksheets("Random Numbers").Cells(FFcurrentPeriod * 21 + I +
scenarioCounter * projDays + 1, 12 + FFcurrScenario * 5).Value

            mfrandnum = Worksheets("Random Numbers").Cells(FFcurrentPeriod * 21 + I +
scenarioCounter * projDays + 1, 13 + FFcurrScenario * 5).Value

            randnorm = Application.WorksheetFunction.NormSInv(randnum)
            rusrandnorm = Application.WorksheetFunction.NormSInv(rusrandnum)
            mfrandnorm = Application.WorksheetFunction.NormSInv(mfrandnum)
```

```
            'Calculate Returns
            SPYreturn = mu * FFtimeinterval + sigma * randnorm * FFtimeinterval ^ 0.5
            RUSreturn = rusmu * FFtimeinterval + russigma * ((correlation * randnorm) + (1 -
correlation ^ 2) ^ 0.5 * rusrandnorm) * FFtimeinterval ^ 0.5

            'Calculate new values
            newvalue = Exp(SPYreturn) * newvalue
            rusnewvalue = Exp(RUSreturn) * rusnewvalue

            'Calculate ln of Returns
            lnreturn = Log(newvalue / oldvalue)
            ruslnreturn = Log(rusnewvalue / rusoldvalue)

            If dOfa = True Then
               mflnreturn = lnreturn + (alphaMu + alphaSigma * mfrandnorm) / 21
               Else
               mflnreturn = lnreturn + alphaMu * mfrandnorm / 21
            End If

            'Calculate Mutual Fund
            mfnewvalue = Exp(mflnreturn) * mfoldvalue
            Next I

               'Calculate ln of Returns for month
               lnreturn = Log(newvalue / SPYmonthEnd)
               ruslnreturn = Log(rusnewvalue / RUSmonthEnd)
               mflnreturn = Log(mfnewvalue / MFmonthEnd)

               'Output Values
               FFrowOutput = 8 + 25 + displayPeriods + FFcurrentPeriod + FFcurrScenario *
(FFdisplayPeriods + 2) * 2
               If FFprintType = True Then
                  'Output ln of Returns
                  Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 +
2).Value = mflnreturn
                  Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 +
3).Value = lnreturn
                  Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 +
4).Value = ruslnreturn
               Else
                  'Output Index Values
                  Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 +
2).Value = mfnewvalue
```

```
                Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 +
3).Value = newvalue
                Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 +
4).Value = rusnewvalue
            End If

        End If

        'Set Starting Values for Variables

        FFmfiBoldvalue = FFmfiBnewvalue
        FFmfiColdvalue = FFmfiCnewvalue
        FFmfiDoldvalue = FFmfiDnewvalue

            'Reset Variables
            regressionOffset = displayPeriods
            twoIntercept = Worksheets("Scenario Output").Cells(regressionOffset + 6,
scenarioCounter * 4 + 3).Value
            SP2VarCoeff = Worksheets("Scenario Output").Cells(regressionOffset + 7,
scenarioCounter * 4 + 3).Value
            RUS2VarCoeff = Worksheets("Scenario Output").Cells(regressionOffset + 8,
scenarioCounter * 4 + 3).Value
            SPintercept = Worksheets("Scenario Output").Cells(regressionOffset + 15,
scenarioCounter * 4 + 3).Value
            SP1VarCoeff = Worksheets("Scenario Output").Cells(regressionOffset + 16,
scenarioCounter * 4 + 3).Value
            RUSintercept = Worksheets("Scenario Output").Cells(regressionOffset + 23,
scenarioCounter * 4 + 3).Value
            RUS1VarCoeff = Worksheets("Scenario Output").Cells(regressionOffset + 24,
scenarioCounter * 4 + 3).Value

            'Calculate New value of Mutual Funds
            mfnewvalue = Exp(mflnreturn) * mfoldvalue

            'If regressions were done on LN...
            If regressLN = True Then
                FFmfiBlnreturn = twoIntercept + SP2VarCoeff * lnreturn + RUS2VarCoeff *
ruslnreturn
                FFmfiClnreturn = SPintercept + SP1VarCoeff * lnreturn
                FFmfiDlnreturn = RUSintercept + RUS1VarCoeff * ruslnreturn
                FFmfiBnewvalue = Exp(FFmfiBlnreturn) * FFmfiBoldvalue
                FFmfiCnewvalue = Exp(FFmfiClnreturn) * FFmfiColdvalue
                FFmfiDnewvalue = Exp(FFmfiDlnreturn) * FFmfiDoldvalue
```

```
            'Or if regressions were done on values...
            Else
               FFmfiBnewvalue = twoIntercept + SP2VarCoeff * newvalue + RUS2VarCoeff *
rusnewvalue
               FFmfiCnewvalue = SPintercept + SP1VarCoeff * newvalue
               FFmfiDnewvalue = RUSintercept + RUS1VarCoeff * rusnewvalue
               FFmfiBlnreturn = Log(FFmfiBnewvalue / FFmfiBoldvalue)
               FFmfiClnreturn = Log(FFmfiCnewvalue / FFmfiColdvalue)
               FFmfiDlnreturn = Log(FFmfiDnewvalue / FFmfiDoldvalue)
            End If

            'Output Values
            FFrowOutput = 8 + 25 + displayPeriods + (FFdisplayPeriods + 2) + FFcurrentPeriod +
(FFdisplayPeriods + 2) * FFcurrScenario * 2
            If FFprintType = True Then
               'Output ln of Returns
               Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 + 2).Value =
FFmfiBlnreturn
               Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 + 3).Value =
FFmfiClnreturn
               Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 + 4).Value =
FFmfiDlnreturn

            Else
               'Output Index Values
               Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 + 2).Value =
FFmfiBnewvalue
               Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 + 3).Value =
FFmfiCnewvalue
               Worksheets("Scenario Output").Cells(FFrowOutput, scenarioCounter * 4 + 4).Value =
FFmfiDnewvalue
            End If

      Next FFcurrentPeriod

   Next FFcurrScenario

   ''''''''''''''''''''''''''''''
   'Future Future Stuff Ends Here'
   ''''''''''''''''''''''''''''''

   'Advance Counter Variable
   scenarioCounter = scenarioCounter + 1
```

```
Loop

''''''''''''''''''''''''''
'Scenario Stuff Ends Here'
''''''''''''''''''''''''''

''''''''''''''''''''''''''''''
'Formatting Stuff Starts Here'
''''''''''''''''''''''''''''''

'Status Update
Worksheets("Scenario Output").Range("A1").Value = "Finalizing..."

'Put black bar between Future Past and Regressions
Worksheets("Scenario Output").Rows(displayPeriods + 3).RowHeight = 9.75
With Worksheets("Scenario Output").Rows(displayPeriods + 3).Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .Color = 0
    .TintAndShade = 0
    .PatternTintAndShade = 0
End With
With Worksheets("Scenario Output").Rows(displayPeriods + 3).Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .ColorIndex = xlAutomatic
        .TintAndShade = 0
        .Weight = xlThin
End With
With Worksheets("Scenario Output").Rows(displayPeriods + 3).Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .ColorIndex = xlAutomatic
        .TintAndShade = 0
        .Weight = xlThin
End With

'Put black bar between Regressions and Future Future
Worksheets("Scenario Output").Rows(25 + displayPeriods + 5).RowHeight = 9.75
With Worksheets("Scenario Output").Rows(25 + displayPeriods + 5).Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .Color = 0
    .TintAndShade = 0
    .PatternTintAndShade = 0
End With
```

```
With Worksheets("Scenario Output").Rows(25 + displayPeriods + 5).Borders(xlEdgeBottom)
	.LineStyle = xlContinuous
	.ColorIndex = xlAutomatic
	.TintAndShade = 0
	.Weight = xlThin
End With
With Worksheets("Scenario Output").Rows(25 + displayPeriods + 5).Borders(xlEdgeTop)
	.LineStyle = xlContinuous
	.ColorIndex = xlAutomatic
	.TintAndShade = 0
	.Weight = xlThin
End With

''''''''''''''''''''''''''''''''''
'Formatting Stuff Ends Here'
''''''''''''''''''''''''''''''''''
'Data Output
Run (OutputTab(regimeSwitching, dOfa, scenarios, displayPeriods, FFdisplayPeriods, projYears,
projDays, FPcalculate, FPdailyDisplay, FPprintType, sigma1, mu1, sigma2, mu2, startvalue,
correlation, alphaMu, alphaSigma, mfstartvalue, russigma1, rusmu1, russigma2, rusmu2,
russtartvalue, a11, a21, a22, regression, regressDaily, regressLN, FFscenarios, FFprojYears,
FFprojDays, FFcalculate, FFdailyDisplay, FFprintType))

'Status Update
Worksheets("Scenario Output").Range("A1").Value = ""

'End Time
Dim endtime As Date

endtime = Now
Worksheets("Output").Range("C3").Value = endtime

End Sub
Sub Cholesky_Decomp()
'
' Cholesky Decomposition Algorithm
'

' Declare Variables
Dim a
Dim L() As Double
Dim s As Double
Dim n As Integer
```

```
' Form Cosigma Matrix A and number of columns
a = Worksheets("Cosigma Matrix").Range("C3:D4").Value
n = Worksheets("Cosigma Matrix").Range("B2").Value

' Perform Cholesky Algorithm
ReDim L(1 To n, 1 To n)
For j = 1 To n
   s = 0
   For k = 1 To j - 1
      s = s + L(j, k) ^ 2
   Next k
   L(j, j) = a(j, j) - s
   If L(j, j) <= 0 Then Exit For
   L(j, j) = Sqr(L(j, j))

   For I = j + 1 To n
      s = 0
      For k = 1 To j - 1
         s = s + L(I, k) * L(j, k)
      Next k
      L(I, j) = (a(I, j) - s) / L(j, j)
   Next I
Next j

' Print Cholesky Decomposed Matrix
Worksheets("Cosigma Matrix").Range("C11:D12").Value = L

End Sub

Private Function OutputTab(regimeSwitching As Boolean, dOfa As Boolean, scenarios As Long,
displayPeriods As Integer, FFdisplayPeriods As Integer, projYears As Long, projDays As Long,
FPcalculate As Boolean, FPdailyDisplay As Boolean, FPprintType As Boolean, sigma1 As Double,
mu1 As Double, sigma2 As Double, mu2 As Double, startvalue As Double, correlation As Double,
alphaMu As Double, alphaSigma As Double, mfstartvalue As Double, russigma1 As Double,
rusmu1 As Double, russigma2 As Double, rusmu2 As Double, russtartvalue As Double, a11 As
Double, a21 As Double, a22 As Double, regression As Long, regressDaily As Boolean, regressLN
As Boolean, FFscenarios As Long, FFprojYears As Long, FFprojDays As Long, FFcalculate As
Boolean, FFdailyDisplay As Boolean, FFprintType As Boolean)

Worksheets("Output").Range("B7:F7").Value = ""
Worksheets("Output").Range("B11:E11").Value = ""
Worksheets("Output").Range("B15:D15").Value = ""
Worksheets("Output").Range("B19:D19").Value = ""
Worksheets("Output").Range("B22:C23").Value = ""
```

```
Worksheets("Output").Range("B27:D27").Value = ""
Worksheets("Output").Range("B31:F31").Value = ""

'Reprint the Input tab in the Output tab
Worksheets("Output").Range("B7").Value = scenarios
Worksheets("Output").Range("C7").Value = projYears
If FPcalculate = True Then
   Worksheets("Output").Range("D7").Value = "Daily"
Else
   Worksheets("Output").Range("D7").Value = "Monthly"
End If
If FPdailyDisplay = True Then
   Worksheets("Output").Range("E7").Value = "Daily"
Else
   Worksheets("Output").Range("E7").Value = "Monthly"
End If
If FPprintType = True Then
   Worksheets("Output").Range("F7").Value = "LN Return"
Else
   Worksheets("Output").Range("F7").Value = "Values"
End If
If regimeSwitching = True Then
   Worksheets("Output").Range("G7").Value = "Yes"
Else
   Worksheets("Output").Range("G7").Value = "No"
End If

Worksheets("Output").Range("B11").Value = sigma1
Worksheets("Output").Range("C11").Value = mu1
Worksheets("Output").Range("D11").Value = startvalue
Worksheets("Output").Range("E11").Value = correlation
If regimeSwitching = True Then
   Worksheets("Output").Range("F11").Value = sigma2
   Worksheets("Output").Range("G11").Value = mu2
   Else
   Worksheets("Output").Range("F11").Value = ""
   Worksheets("Output").Range("G11").Value = ""
End If

Worksheets("Output").Range("B15").Value = alphaMu
Worksheets("Output").Range("C15").Value = alphaSigma
Worksheets("Output").Range("D15").Value = mfstartvalue
Worksheets("Output").Range("E15").Value = dOfa
```

```
Worksheets("Output").Range("B19").Value = russigma1
Worksheets("Output").Range("C19").Value = rusmu1
Worksheets("Output").Range("D19").Value = russtartvalue
If regimeSwitching = True Then
   Worksheets("Output").Range("E19").Value = russigma2
   Worksheets("Output").Range("F19").Value = rusmu2
   Else
   Worksheets("Output").Range("E19").Value = ""
   Worksheets("Output").Range("F19").Value = ""
End If

Worksheets("Output").Range("B22").Value = a11
Worksheets("Output").Range("B23").Value = a21
Worksheets("Output").Range("C22").Value = 0
Worksheets("Output").Range("C23").Value = a22

Worksheets("Output").Range("B27").Value = regression
If regressDaily = True Then
   Worksheets("Output").Range("C27").Value = "Daily"
Else
   Worksheets("Output").Range("C27").Value = "Monthly"
End If
If regressLN = True Then
   Worksheets("Output").Range("D27").Value = "LN Return"
Else
   Worksheets("Output").Range("D27").Value = "Values"
End If

Worksheets("Output").Range("B31").Value = FFscenarios
Worksheets("Output").Range("C31").Value = FFprojYears
If FFcalculate = True Then
   Worksheets("Output").Range("D31").Value = "Daily"
Else
   Worksheets("Output").Range("D31").Value = "Monthly"
End If
If FFdailyDisplay = True Then
   Worksheets("Output").Range("E31").Value = "Daily"
Else
   Worksheets("Output").Range("E31").Value = "Monthly"
End If
If FFprintType = True Then
   Worksheets("Output").Range("F31").Value = "LN Return"
Else
   Worksheets("Output").Range("F31").Value = "Values"
```

```
End If

'Analyze Data
Dim holdingArray() As Double
Dim holdingArray2() As Double
'If FP is outputted as Values
If FPprintType = False Then

   For I = 1 To scenarios
       Worksheets("Output").Cells(1, 8 + I).Value = "Scenario " & I
       Worksheets("Output").Cells(2, 8 + I).Value = (Worksheets("Scenario
Output").Cells(displayPeriods + 2, I * 4 - 1).Value / startvalue) ^ (1 / projYears) - 1
       Worksheets("Output").Cells(3, 8 + I).Value = (Worksheets("Scenario
Output").Cells(displayPeriods + 2, I * 4 - 2).Value / mfstartvalue) ^ (1 / projYears) - 1
       Worksheets("Output").Cells(4, 8 + I).Value = (Worksheets("Scenario
Output").Cells(displayPeriods + 2, I * 4).Value / russtartvalue) ^ (1 / projYears) - 1
       'Tracking Error (Realized alpha)
       'ReDim holdingArray(1 To projDays) As Double
       'holdingArray(1) = (Worksheets("Scenario Output").Cells(3, i * 4 - 1).Value / startvalue - 1) -
(Worksheets("Scenario Output").Cells(3, i * 4 - 2).Value / mfstartvalue - 1)
       'For j = 2 To projDays
       '    holdingArray(j) = (Worksheets("Scenario Output").Cells(2 + j, i * 4 - 1).Value /
Worksheets("Scenario Output").Cells(1 + j, i * 4 - 1).Value - 1) - (Worksheets("Scenario
Output").Cells(2 + j, i * 4 - 2).Value / Worksheets("Scenario Output").Cells(1 + j, i * 4 - 2).Value -
1)
       'Next j
       'Worksheets("Output").Cells(5, 8 + i).Value =
Application.WorksheetFunction.stdev(holdingArray()) * Sqr(projDays / projYears)
       'Realized Correlation
       'ReDim holdingArray2(1 To projDays) As Double
       'holdingArray(1) = Worksheets("Scenario Output").Cells(3, i * 4 - 1).Value / startvalue - 1
       'holdingArray2(1) = Worksheets("Scenario Output").Cells(3, i * 4).Value / russtartvalue - 1
       'For j = 2 To projDays
       '    holdingArray(j) = Worksheets("Scenario Output").Cells(2 + j, i * 4 - 1).Value /
Worksheets("Scenario Output").Cells(1 + j, i * 4 - 1).Value - 1
       '    holdingArray2(j) = Worksheets("Scenario Output").Cells(2 + j, i * 4).Value /
Worksheets("Scenario Output").Cells(1 + j, i * 4).Value - 1
       'Next j
       'Worksheets("Output").Cells(6, 8 + i).Value =
Application.WorksheetFunction.Correl(holdingArray(), holdingArray2())
   Next I

'If FP is outputted as LN
Else
```

```
    For I = 1 To scenarios
        Worksheets("Output").Cells(1, 8 + I).Value = "Scenario " & I
        'Tracking Error (Realized alpha)
        ReDim holdingArray(1 To displayPeriods) As Double
        For j = 1 To displayPeriods
            holdingArray(j) = Worksheets("Scenario Output").Cells(2 + j, I * 4 - 2).Value -
Worksheets("Scenario Output").Cells(2 + j, I * 4 - 1).Value
        Next j
        Worksheets("Output").Cells(5, 8 + I).Value =
Application.WorksheetFunction.StDev(holdingArray()) * Sqr(displayPeriods / projYears)
        'Realized Correlation
        ReDim holdingArray2(1 To displayPeriods) As Double
        For j = 1 To displayPeriods
            holdingArray(j) = Worksheets("Scenario Output").Cells(2 + j, I * 4 - 1).Value
            holdingArray2(j) = Worksheets("Scenario Output").Cells(2 + j, I * 4).Value
        Next j
        Worksheets("Output").Cells(6, 8 + I).Value =
Application.WorksheetFunction.Correl(holdingArray(), holdingArray2())
    Next I

End If
'FF tracking error regression
If FFprintType = True Then
    'Format sheet for 2 VAR
    Worksheets("Tracking Error").Range("A1").Value = "2 VAR"
    Worksheets("Tracking Error").Range("A2").Value = "FF Scen"
    Worksheets("Tracking Error").Range("B1").Value = "FP Scen"
    For I = 1 To scenarios
        Worksheets("Tracking Error").Cells(1, 2 + I).Value = I
    Next I
    For I = 1 To FFscenarios
        Worksheets("Tracking Error").Cells(1 + I, 2).Value = I
    Next I

    'Format sheet for 1 VAR SPY
    Worksheets("Tracking Error").Cells(4 + FFscenarios, 1).Value = "1 VAR SPY"
    Worksheets("Tracking Error").Cells(5 + FFscenarios, 1).Value = "FF Scen"
    Worksheets("Tracking Error").Cells(4 + FFscenarios, 2).Value = "FP Scen"
    For I = 1 To scenarios
        Worksheets("Tracking Error").Cells(4 + FFscenarios, 2 + I).Value = I
    Next I
    For I = 1 To FFscenarios
        Worksheets("Tracking Error").Cells(4 + FFscenarios + I, 2).Value = I
```

```
    Next I


    For I = 1 To scenarios
        For j = 0 To FFscenarios - 1
            ReDim holdingArray(1 To FFdisplayPeriods) As Double
            For k = 1 To FFdisplayPeriods
            holdingArray(k) = (Worksheets("Scenario Output").Cells(32 + displayPeriods +
(FFdisplayPeriods + 2) * (2 * j + 1) + k, I * 4 - 2).Value - Worksheets("Scenario Output").Cells(32
+ displayPeriods + (FFdisplayPeriods + 2) * 2 * j + k, I * 4 - 2).Value) * (12 ^ 0.5)
            Next k
            Worksheets("Tracking Error").Cells(2 + j, 2 + I).Value =
Application.WorksheetFunction.StDev(holdingArray)
        Next j
    Next I

    For I = 1 To scenarios
        For j = 0 To FFscenarios - 1
            ReDim holdingArray(1 To FFdisplayPeriods) As Double
            For k = 1 To FFdisplayPeriods
            holdingArray(k) = (Worksheets("Scenario Output").Cells(32 + displayPeriods +
(FFdisplayPeriods + 2) * (2 * j + 1) + k, I * 4 - 1).Value - Worksheets("Scenario Output").Cells(32
+ displayPeriods + (FFdisplayPeriods + 2) * 2 * j + k, I * 4 - 2).Value) * (12 ^ 0.5)
            Next k
            Worksheets("Tracking Error").Cells(5 + FFscenarios + j, 2 + I).Value =
Application.WorksheetFunction.StDev(holdingArray)
        Next j
    Next I
    'Average/Standard Deviation
    Worksheets("Output").Cells(1, 10 + scenarios).Value = "Average"
    Worksheets("Output").Cells(1, 11 + scenarios).Value = "StDev"
    ReDim holdingArray(1 To scenarios)
    For I = 1 To 5
        For j = 1 To scenarios
            holdingArray(j) = Worksheets("Output").Cells(I + 1, 8 + j).Value
        Next j
        Worksheets("Output").Cells(I + 1, 10 + scenarios).Value =
Application.WorksheetFunction.Average(holdingArray())
        Worksheets("Output").Cells(I + 1, 11 + scenarios).Value =
Application.WorksheetFunction.StDev(holdingArray())
    Next I
End If

'Analyze Regressions
```

```
    Worksheets("Output").Range("I8").Value = (regression) & "-year Regression"

    '2-Var Intercept Avg.
    ReDim holdingArray(1 To scenarios) As Double
    For j = 1 To scenarios
        holdingArray(j) = Worksheets("Scenario Output").Cells(displayPeriods + 9, j * 4).Value
    Next j
    Worksheets("Output").Cells(9, 9).Value =
Application.WorksheetFunction.Average(holdingArray())

    '2-Var Intercept Avg. Error
    For j = 1 To scenarios
        holdingArray(j) = Worksheets("Scenario Output").Cells(displayPeriods + 10, j * 4).Value
    Next j
    Worksheets("Output").Cells(10, 9).Value =
Application.WorksheetFunction.Average(holdingArray())

    '2-Var SPY Coef. Avg.
    For j = 1 To scenarios
        holdingArray(j) = Worksheets("Scenario Output").Cells(displayPeriods + 9, j * 4 - 1).Value
    Next j
    Worksheets("Output").Cells(11, 9).Value =
Application.WorksheetFunction.Average(holdingArray())

    '2-Var SPY Coef. Avg. Error
    For j = 1 To scenarios
        holdingArray(j) = Worksheets("Scenario Output").Cells(displayPeriods + 10, j * 4 - 1).Value
    Next j
    Worksheets("Output").Cells(12, 9).Value =
Application.WorksheetFunction.Average(holdingArray())

    '2-Var RUS Coef. Avg.
    For j = 1 To scenarios
        holdingArray(j) = Worksheets("Scenario Output").Cells(displayPeriods + 9, j * 4 - 2).Value
    Next j
    Worksheets("Output").Cells(13, 9).Value =
Application.WorksheetFunction.Average(holdingArray())

    '2-Var RUS Coef. Avg. Error
    For j = 1 To scenarios
        holdingArray(j) = Worksheets("Scenario Output").Cells(displayPeriods + 10, j * 4 - 2).Value
    Next j
    Worksheets("Output").Cells(14, 9).Value =
Application.WorksheetFunction.Average(holdingArray())
```

```
    '2-Var R Squared
    For j = 1 To scenarios
        holdingArray(j) = Worksheets("Scenario Output").Cells(displayPeriods + 11, j * 4 - 2).Value
    Next j
    Worksheets("Output").Cells(15, 9).Value =
Application.WorksheetFunction.Average(holdingArray())

    '1-Var SPY Intercept Avg.
    For j = 1 To scenarios
        holdingArray(j) = Worksheets("Scenario Output").Cells(displayPeriods + 17, j * 4 - 1).Value
    Next j
    Worksheets("Output").Cells(16, 9).Value =
Application.WorksheetFunction.Average(holdingArray())

    '1-Var SPY Intercept Avg. Error
    For j = 1 To scenarios
        holdingArray(j) = Worksheets("Scenario Output").Cells(displayPeriods + 18, j * 4 - 1).Value
    Next j
    Worksheets("Output").Cells(17, 9).Value =
Application.WorksheetFunction.Average(holdingArray())

    '1-Var SPY Coef. Avg.
    For j = 1 To scenarios
        holdingArray(j) = Worksheets("Scenario Output").Cells(displayPeriods + 17, j * 4 - 2).Value
    Next j
    Worksheets("Output").Cells(18, 9).Value =
Application.WorksheetFunction.Average(holdingArray())

    '1-Var SPY Coef. Avg. Error
    For j = 1 To scenarios
        holdingArray(j) = Worksheets("Scenario Output").Cells(displayPeriods + 18, j * 4 - 2).Value
    Next j
    Worksheets("Output").Cells(19, 9).Value =
Application.WorksheetFunction.Average(holdingArray())

    '1-Var SPY R Squared
    For j = 1 To scenarios
        holdingArray(j) = Worksheets("Scenario Output").Cells(displayPeriods + 19, j * 4 - 2).Value
    Next j
    Worksheets("Output").Cells(20, 9).Value =
Application.WorksheetFunction.Average(holdingArray())

    '1-Var RUS Intercept Avg.
```

```
   For j = 1 To scenarios
      holdingArray(j) = Worksheets("Scenario Output").Cells(displayPeriods + 25, j * 4 - 1).Value
   Next j
   Worksheets("Output").Cells(21, 9).Value =
Application.WorksheetFunction.Average(holdingArray())

   '1-Var RUS Intercept Avg. Error
   For j = 1 To scenarios
      holdingArray(j) = Worksheets("Scenario Output").Cells(displayPeriods + 26, j * 4 - 1).Value
   Next j
   Worksheets("Output").Cells(22, 9).Value =
Application.WorksheetFunction.Average(holdingArray())

   '1-Var RUS Coef. Avg.
   For j = 1 To scenarios
      holdingArray(j) = Worksheets("Scenario Output").Cells(displayPeriods + 25, j * 4 - 2).Value
   Next j
   Worksheets("Output").Cells(23, 9).Value =
Application.WorksheetFunction.Average(holdingArray())

   '1-Var RUS Coef. Avg. Error
   For j = 1 To scenarios
      holdingArray(j) = Worksheets("Scenario Output").Cells(displayPeriods + 26, j * 4 - 2).Value
   Next j
   Worksheets("Output").Cells(24, 9).Value =
Application.WorksheetFunction.Average(holdingArray())

   '1-Var RUS R Squared
   For j = 1 To scenarios
      holdingArray(j) = Worksheets("Scenario Output").Cells(displayPeriods + 27, j * 4 - 2).Value
   Next j
   Worksheets("Output").Cells(25, 9).Value =
Application.WorksheetFunction.Average(holdingArray())

End Function
```