

Economic Scenario Generator

A Major Qualifying Project
submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
degree of Bachelor of Science

by
Erica Lee
Rahul Kumar
Hanyi Jiang

Date:
May 1st, 2020

Professors Barry Posterro and Jon Abraham
Worcester Polytechnic Institute

This report represents work of WPI undergraduate students submitted to the faculty as evidence of a degree requirement. WPI routinely publishes these reports on its web site without editorial or peer review. For more information about the projects program at WPI, see <http://www.wpi.edu/Academics/Projects>.

ABSTRACT

The behavior of the stock market is quite unpredictable. When making financial decisions, companies and individuals want to be prepared to face the numerous economic futures that could arise. An Economic Scenario Generator (ESG) uses probabilistic and other mathematical models in order to produce simulations which demonstrate the potential future behavior of stock returns. We built our ESG using Visual Basic for Applications in Microsoft Excel, with the goal of being able to generate simulations of the annual log returns of seven stock indices. The simulations are generated by extrapolating the historical data of these indices, which was analysed with the use of Maximum Likelihood Estimation (MLE). The resulting returns generated by our ESG may help investors to view several potential future economic scenarios before making important financial decisions.

ACKNOWLEDGEMENTS

Our team would like to thank Professor Barry Posterro and Professor Jon Abraham for their guidance on this project. Additionally, the help of Professor Min Wu and Professor Randy Paffenroth aided in the completion of correlation within our project. Thank you to Emily Chen, a graduate student at WPI who helped us obtain accurate historical data for our project.

TABLE OF CONTENTS

ABSTRACT.....	2
ACKNOWLEDGEMENTS.....	3
TABLE OF CONTENTS.....	4
TABLE OF FIGURES.....	6
TABLE OF TABLES.....	6
EXECUTIVE SUMMARY.....	7
Background.....	7
Methodology.....	7
Results.....	8
Final Recommendations.....	9
CHAPTER 1: Introduction.....	10
CHAPTER 2: Background.....	11
2.1 Economic Scenario Generator.....	11
2.2 Markov Chains (Regime-Switching).....	11
2.3 Transition Matrices.....	12
2.4 Steady State of Transition Matrices.....	13
2.5 Random Number Generator.....	15
2.6 Regime-Switching Lognormal Models.....	16
2.7 The Normal Distribution.....	16
2.8 Indices Included in our ESG.....	17
2.9 Calculating Monthly & Annual Log Returns.....	18
2.10 Maximum Likelihood Estimation & Mary Hardy’s Paper.....	18
2.11 Covariance & Correlation.....	19
2.12 Cholesky Decomposition.....	20
CHAPTER 3: Methodology.....	23
3.1 One-Page Input.....	23
3.2 Method for Creating and Using Random Numbers.....	23
3.3 Adding Markov Chains to our ESG in VBA Through Code.....	24
3.4 Gathering of Historical Data.....	26
3.5 Calculating Annual Returns.....	26
3.6 Obtaining Realistic Model Parameters.....	27
3.7 Obtaining Model Parameters for S&P 500 by Maximum Likelihood Estimation.....	27
3.8 Maximum Likelihood Estimation for the Other 6 Indices.....	29
3.9 Choosing Regime 3 Parameters.....	29
3.10 Correlating Our Returns.....	30
3.11 Adding Correlation to our ESG in VBA Through Code.....	31
3.12 Outputting our Results as a CSV File.....	32
CHAPTER 4: Results and Analysis.....	33
4.1 ESG Run Time.....	33
4.2 Maximum Likelihood Estimation Results for Model Parameters.....	33
4.3 Analysis of Maximum Likelihood Estimation Results.....	33

4.4 Steady State Ratios of Results.....	34
4.5 Analysis of our ESG Results.....	35
4.6 Covariance and Correlation Matrix.....	36
CHAPTER 5: Conclusion and recommendations	37
5.1 Code Writing.....	37
5.2 Data Collection.....	37
5.3 Steady State Ratios of Results.....	37
5.4 Correlation and Covariance Matrices	38
5.5 Analysis of Data.....	38
REFERENCES	39
APPENDIX A: ONE-PAGE INPUT (ORANGE IS changeable VALUES; BLUE IS STEADY STATE RATIO)	
.....	40
APPENDIX B: RANDOM NUMBER FROZEN SPREADSHEET	41

TABLE OF FIGURES

Figure 1: Summary of Economic Trends Exhibited by Selected Indices	8
Figure 2: A visual of the weather example where there are two possible regimes: sunny or rainy.	11
Figure 3: Step-by-step process of finding the system of equations necessary for solving the steady state.....	14
Figure 4: A normal distribution with mean 0 and standard deviation 1.	16
Figure 5: Relationship between covariance and correlation.....	20
Figure 6: Brief code diagram of how we set up our Markov chain in VBA.....	25
Figure 7: Brief code diagram of how we set up correlating returns in VBA.	31
Figure 8: The order in which we listed our data in the CSV file.....	32
Figure 9: Results of Maximum Likelihood Estimation for Model Parameters.....	33
Figure 10: Comparison of arbitrary values to the maximum likelihood estimates.	34
Figure 11: Comparison of MLE Parameters (top) with Results of ESG Simulation (bottom).	35
Figure 12: P-Values of Two-Sample Hypothesis Tests used for comparison of MLE & ESG results.	35
Figure 13: Comparison of Regime 1 covariance matrices (top Bloomberg, bottom MLE).....	36
Figure 14: Comparison of Regime 2 covariance matrices (top Bloomberg, bottom MLE).....	36

TABLE OF TABLES

Table 1: Transition Matrix for our Weather Example.....	12
Table 2: Two-time-step Probability Transition Matrix for our Weather Example.....	13
Table 3: Steady State Matrix Solved for Mathematically from Example.....	15
Table 4: Steady State Matrix Solve for with MATLAB from Example	15
Table 5: The stock indices involved in our project, involving a brief description and a few example companies included within each stock.	17
Table 6: Example Correlation Matrix between S&P 500 and NASDAQ	19
Table 7: Example Covariance Matrix between S&P 500 and NASDAQ	21
Table 8: Example Lower Triangle Matrix between S&P 500 and NASDAQ Generated from Cholesky Decomposition.....	21
Table 9: Example Uncorrelated Normal Returns with Variance = 1	22
Table 10: Resulting Correlated Normal Returns	22
Table 11: Cumulative Distribution Function Matrix that Results from our Weather Example	24

EXECUTIVE SUMMARY

One of the primary characteristics of decision making is the desire to minimize risk (of losses) while maximizing expected gains. While our limitations as human beings make it impossible to see the future, technology provides us with the capability to simulate the near infinite possibilities that the future may hold. An economic scenario generator (ESG) is a mathematical model that generates simulated stock returns for a group of correlated stock indices.

The goal of our project was to make an ESG that can be used to help companies and individuals envision possible future economies by extrapolating historical data. The project objectives were to:

1. Have three regimes which represent good, bad, and catastrophic economic years.
2. Program our ESG to generate as many scenarios as the user chooses, each of a user-specified length.
3. Ensure our ESG extrapolates historical data of seven indices, which are obtained through Maximum Likelihood Estimation (MLE).
4. Generate normally distributed stock returns.
5. Establish correlating between the returns of each index which accurately reflects the correlating of the historical data collected.

Background

There are two main types of economic scenario generators. We built a real-world economic scenario generator that utilized real-world data and focused on the future behavior of economic variables. In order to mimic the unpredictability of the stock market, a process known as regime-switching is applied to our ESG by using transition matrices. Regime-switching allows the ESG to switch between good, bad, and catastrophic economic periods, each of which are characterized by their own unique mean and standard deviation. We can also solve for a steady state matrix, which can represent the ratio of each regime we should expect in the future. Correlating our returns through Cholesky decomposition was essential to make sure that our returns reflected the regime we were in, and we needed to understand maximum likelihood estimation for fitting our ESG to accurate data.

Methodology

To achieve the goals for this project, we approached making a user-friendly ESG that anyone could use and understand after reading this paper. Additionally, we wanted to allow readers to follow our methods to create their own ESG to be adjustable to their own needs and situation. We needed to create a system for generating random numbers and use these

numbers to determine how our ESG will switch regimes. We also wanted to project three states: good, bad, and catastrophic. Therefore, we coded the fundamental lines in VBA to create a regime-switching lognormal model with three regimes. To make sure our ESG was as realistic to historical data as possible, we created a formula for maximum likelihood estimation to analyze historical data and create realistic parameters to reflect the seven stock indices. When using random numbers as our returns, we also need to make sure that we reflect the correlation that occurs between the returns of the indices. Therefore, we utilize Cholesky decomposition in creating correlation between the results of our returns for each index. Lastly, we wanted our data to be understandable for any reader to analyze, as well as compress the file size as much as possible. Therefore, we output our final returns as a CSV file so that the large data would not create a huge file and could easily be turned into an Excel sheet.

Results

Our ESG took one minute and 42 seconds to run 10,000 scenarios, each 50 years long. We used our maximum likelihood estimator to find the parameters for our ESG. The transition probabilities p_{12} and p_{21} determined for the S&P 500 index were the same for all seven indices. We also analyzed the parameters for each index that resulted from our maximum likelihood estimator.

"Good" Economic Periods (Regime 1)		
Characteristic	Min	Max
Average Annual Log Returns	10.75% (MSCI EAFE)	15.82% (NASDAQ)
Volatility	27.92% (S&P 500)	56.60% (MSCI (EM))
"Bad" Economic Periods (Regime 2)		
Characteristic	Min	Max
Average Annual Log Returns	-5.17% (MSCI EAFE)	4.56% (NASDAQ)
Volatility	62.39% (S&P 500)	96.97% (NASDAQ)

Figure 1: Summary of Economic Trends Exhibited by Selected Indices

After conducting our maximum likelihood estimation, we determined the probabilities for switching between two regimes, “good” and “bad”. There was a 3.49% chance that the economy would switch from being in a “good” situation to being in a “bad” situation. Similarly, the chance that the economy would switch from being in a “bad” state to a “good” state was 3.60% (Figure 1).

The resulting steady state ratios that came from our ESG results were not accurate to the steady state ratios given from multiplying the PDF transition matrix by itself multiple times. This was due to our probability transition values being highly unlikely to switch regimes. Additionally, we were limited by the number of scenarios and years we could run for our ESG, meaning we did not have enough data for convergence.

We also computed statistical model tests to check our results from the ESG. To compare the means, we implemented the two-sample t-test. We also performed the F-test in order to compare standard deviations. Our statistical analysis did not show any significant differences in parameter values between the historical data and the data produced by our simulation. Hence, we were able to conclude that our ESG could successfully extrapolate historical data, generating stock returns that accurately reflect past trends.

Lastly, we checked our covariance matrix obtained through the (historical data) correlation matrix and parameters obtained from the MLE. We compared this covariance matrix to the covariance matrix that directly resulted from the historical data. The accuracy between the covariance matrices indicated that our maximum likelihood estimates of the regime standard deviation accurately represented the historical data we utilized.

Final Recommendations

While we chose to use VBA to write the code, the ESG can be generated through any coding language as long as it can handle large amounts of data, and we encourage readers of this report to use a different coding language if it provides a higher scenario and year count compared to VBA. This will result in your returns more likely converging to the steady state. In addition to the coding platform, the data is extremely important in building a successful ESG. We recommend Bloomberg as a source of data collection because Yahoo Finance was not accurate.

Error may also come about when trying to adjust the values within the covariance matrix. Changing values may cause the matrix to not Cholesky decompose. We recommend always keeping track of the positive-definite property of the matrix to make this process easier for you.

Lastly, every scenario generated should be independently analyzed and compared to the results of the maximum likelihood estimator. Based on the comparison of the results, each scenario can be put in one of the following three categories:

1. Improved Economy - characterized by higher means and lower standard deviations
2. Stable Economy - characterized by parameters equivalent to the MLE values
3. Worsened Economy - characterized by decreased means and increased standard deviations

The user can then base their decisions on the relative frequencies of the three categories. In other words, the user can make their decisions according to which economic future is the most likely to occur.

CHAPTER 1: INTRODUCTION

For a company to make executive decisions that will better ensure its long-term financial success and stability, they will most likely want to have a solid understanding of current economic trends. As there is no way of seeing into the future, the best that companies can do is make decisions based on how they believe the future will turn out. An Economic Scenario Generator (ESG) serves the purpose of producing simulations which demonstrate possible future economies that companies may face.

The goal of our project was to make an ESG that can be used to help companies and individuals envision possible future economies by extrapolating historical data. The project objectives were to:

1. Have three regimes which represent good, bad, and catastrophic economic years.
2. Program our ESG to generate as many scenarios as the user chooses, each of a user-specified length.
3. Ensure our ESG extrapolates historical data of seven indices, which are obtained through Maximum Likelihood Estimation (MLE).
4. Generate normally distributed stock returns.
5. Establish correlation between the returns of each index which accurately reflects the correlation of the historical data collected.

We created our ESG in Visual Basic for Applications (VBA) in Microsoft Excel. This ESG can switch between good, bad, and catastrophic economic years using a method known as “regime-switching”. In order to ensure that our ESG generates simulations of future economies that accurately represent the current economy, an analysis of historical data utilizing maximum likelihood estimation was performed.

The stocks and indices can be changed based on the maximum likelihood estimation spreadsheet we created. With these factors, our ESG can output normally distributed returns for seven different indices. The user can choose both the number of scenarios and the length of each scenario (in years).

In this report, we explain the background knowledge needed in order to understand the mathematical concepts involved, the steps taken, and the results of our ESG. We conclude by making recommendations on how anyone who wants to improve upon our work can proceed.

CHAPTER 2: BACKGROUND

2.1 Economic Scenario Generator

An Economic Scenario Generator (ESG) refers to a computer-based mathematical model that produces simulations which demonstrate the future joint behavior of economic or financial markets. ESG allows the user to assess the risk associated with different economic elements and determine which factors play an important role in financial variability. ESGs can be used for various purposes across many industries and for specific activities. A basic application of ESGs is the simulation of interest rates in order to determine how changes in interest rates affects investment portfolios. A more complex application of ESGs involves using a risk management approach in order to identify internal and external risks that could possibly influence the financial success and stability of an enterprise (Pedersen et al., 2016, p.6-7).

Our ESG is meant to act as a real-world program rather than a risk-neutral program. Risk-neutral ESGs are less concerned about the future behavior of economic variables and are focused more on the mathematical relationship between them. On the contrary, real-world ESGs utilize real-world data, concentrating on the future behavior of economic variables. People can use the data obtained from real-world ESGs for a variety of different tasks. They will generate scenarios in which the economic risk factors and asset returns follow realistic probability distributions. This data can then be used for either short-term or long-term analysis.

2.2 Markov Chains (Regime-Switching)

A Markov chain is a mathematical system in which you transition from one state to another, and this transition is only dependent on the probabilities of transitioning and the starting state. As such, a Markov chain is memoryless, and when looking to solve for a new state, we only need to know the current state and probabilities of transitioning from one state to another. We can consider a simple example in which we have two states for the weather: sunny and rainy. Figure 2 shows a diagram of an example we will use to explain the concept of a Markov chain:

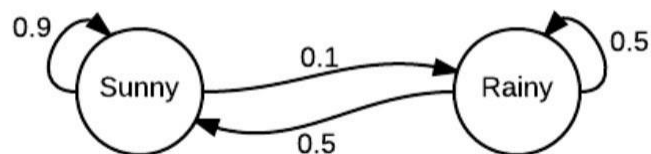


Figure 2: A visual of the weather example where there are two possible regimes: sunny or rainy.

Figure 2 shows arrows either pointing to the opposite regime, or loops around to its own regime. Arrows pointing back to the same circle represent the probability of the next day having the same state, whereas arrows pointing to the opposite circle represent the probability of transitioning to the opposing state. In this example, if we start in a sunny state, then there is a 90% chance of the weather being sunny the next day and a 10% chance of the weather being rainy the next day. If we start in a rainy state, then there is a 50% chance of the weather being sunny the next day and a 50% chance of the weather being rainy the next day.

Assume today is Monday and today is sunny. Then Tuesday has a 90% chance of being sunny, and a 10% chance of being rainy. Either outcome is possible, so let us say that Tuesday happens to be rainy. We can determine the probability of the weather for Wednesday by once again looking at Figure 2. Because a Markov chain is “memoryless”, information regarding the weather of any previous day will not affect the weather probability of tomorrow. Thus, in this example, we do not need to consider the weather on Monday being sunny to find out the probabilities for Wednesday’s weather; we only need to consider Tuesday’s state to determine the probabilities of Wednesday’s weather. Therefore, because Tuesday is rainy, there is a 50% chance of Wednesday being sunny and a 50% chance of Wednesday being rainy.

2.3 Transition Matrices

A transition matrix is a representation of a “Markov chain”. Let t_{ij} represent the probability that starting in state i , you move to state j after a given time period has elapsed. Using our above example of weather on a given day, a transition matrix is illustrated in Table 1. For this example, state 1 is sunny and state 2 is rainy.

Table 1: Transition Matrix for our Weather Example

	new state = sunny	new state = rainy
starting state = sunny	$t_{11} = 0.9$	$t_{12} = 0.1$
starting state = rainy	$t_{21} = 0.5$	$t_{22} = 0.5$

Notice that when we are transitioning from a starting state to a new state, there are only two possible new states: sunny or rainy. Each row of the transition matrix will add up to 1 because there are only two possibilities; in our example, tomorrow will either be sunny or rainy. If we are considering starting on a sunny Monday and are trying to determine the weather for Tuesday, we would look at the row where the starting state is sunny and look at the values in each column. The columns tell us that $t_{11} = 0.9$ and $t_{12} = 0.1$. This aligns with the above statement that each row must add up to 1, or:

$$1 - t_{11} = t_{12}$$

In this example:

$$1 - 0.9 = 0.1$$

Now if we were to continue with our previous example and consider that Tuesday ends up being rainy, we would look at the row where the starting state is rainy. This process can be done recursively to consider, for example, the weather within a given week. Thus, we would look at the transition matrix seven times, with each new day using what was the new state as now the starting state.

2.4 Steady State of Transition Matrices

When using a transition matrix, we have a starting state and transition into a new state. This represents a one-time-step probability transition matrix. For example, the weather transition matrix we used above represents a one-time-step probability transition matrix. To get the two-time-step probability transition matrix, we take the one-time-step probability transition matrix and multiply it by itself. In this case, if M is the one-time-step probability transition matrix given in Table 1, then to solve for the two-time-step probability transition matrix:

$$M * M = \text{two-time-step probability transition matrix}$$

An example of a two-time-step probability transition matrix using our previous weather matrix from Table 1 is shown in Table 2:

Table 2: Two-time-step Probability Transition Matrix for our Weather Example

	new state = sunny	new state = rainy
starting state = sunny	$t_{11} = 0.86$	$t_{12} = 0.14$
starting state = rainy	$t_{21} = 0.7$	$t_{22} = 0.3$

The i^{th} , j^{th} entry indicates the probability that the weather two days from now will be in state j given that the weather is currently in state i . For example, if today is a sunny day, then two days from now, there is an 86% chance that it will be sunny two days from now, and a 14% chance it will be rainy two days from now. To get the k^{th} -time-step probability transition matrix, we would take the one-time-step probability matrix and raise it to the k^{th} power. Eventually, as we get to a higher number for k , the values within the matrix will begin to converge, and at high enough k , the transition matrix will reach equilibrium. This is known as the “steady-state” transition matrix of the Markov chain. The steady-state transition matrix

implies that if we have a matrix M that represents the one-time-step probability transition matrix, and X is the steady-state matrix, then:

$$X * M = X$$

The mathematics with the matrices given in our weather example can be followed in Figure 3:

$$X * M = X \rightarrow \begin{pmatrix} x & y \end{pmatrix} * \begin{pmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{pmatrix} = \begin{pmatrix} 0.9x + 0.5y \\ 0.1x + 0.5y \end{pmatrix} = \begin{pmatrix} x & y \end{pmatrix}$$

Figure 3: Step-by-step process of finding the system of equations necessary for solving the steady state.

Figure 3 creates this system of equations:

$$0.9x + 0.5y = x$$

$$0.1x + 0.5y = y$$

However, once simplified, we notice that they are the same equation. Because these two equations are the same, we cannot solve for x and y without a third equation. Therefore, we must recall that:

$$x + y = 1$$

This is because in the steady state, we know that each starting state will have the same probabilities due to reaching equilibrium. Thus, we can now solve for x and y:

$$0.1x - 0.5y = 0$$

$$x + y = 1$$

$$\rightarrow x = 1 - y$$

$$\text{Using Substitution: } 0.1*(1 - y) - 0.5y = 0$$

$$\rightarrow 0.1 - 0.1y - 0.5y = 0$$

$$\rightarrow 0.1 = 0.6y$$

$$\rightarrow y = \frac{1}{6}$$

$$\text{Plugging } y \text{ back into } x + y = 1: x + \frac{1}{6} = 1$$

$$\rightarrow x = \frac{5}{6}$$

Therefore, our steady state is shown in Table 3:

Table 3: Steady State Matrix Solved for Mathematically from Example

0.83333	0.16667
---------	---------

The steady state reflects the long-run daily probability of rain or sun. That is, if you look at the system on some random day, there is an 83.3% chance that it is sunny and a 16.7% chance that it is rainy. While this is a straightforward method with two states, it can become more complex with a higher number of regimes. Therefore, another way to solve for the steady state is to take our one-time-step probability transition matrix and raise it to a very high power until we see each column the same value. When doing this with our example one-time-step transition matrix above, we get the matrix in Table 4:

Table 4: Steady State Matrix Solve for with MATLAB from Example

0.83333	0.16667
0.83333	0.16667

The results reflect the same probability values as Table 3. Either method works to solve for the steady state, but it is easier to find the steady state for many regimes using the method reflected in Table 4. This is because more regimes will create more variables, and the math involved will become more complex than what is shown in Figure 3.

2.5 Random Number Generator

A random number generator is designed to generate a random set of numbers through a deterministic computation program (RNG, 2017). Despite its given name, it is not random; these numbers are pseudo-random numbers. Since a real random number cannot be calculated by mathematical equations and computational algorithms, Microsoft Excel has a '=RAND' function that generates pseudo-random numbers to be used in place of random numbers. The pseudo-random numbers generated by this function will repeat after ten-trillion random numbers. However, these pseudo-random numbers are enough to fulfill most of the tasks in practice. Specifically, the function '=RAND' generates a number between 0 and 1, which is useful in the use of probability and random outcomes. Random number generators can be used in the application of statistical sampling and computer simulation.

2.6 Regime-Switching Lognormal Models

To accurately represent the unpredictability of stock market behavior, we used a regime-switching lognormal model to generate the simulated stock returns for each of the seven indices. The model can switch between regimes at random. Each regime follows a normal distribution and is characterized by its own unique mean and standard deviation. The characterization of our model as lognormal is because log returns are normally distributed. Markov chains are used in order to determine which regime the model is in at any given time. Stock returns are generated by applying the parameters corresponding to the current regime and generating normally distributed returns. The regime-switching process mimics the behavior of the actual stock market, switching between good (high returns, low volatility) and bad (low returns, high volatility) periods.

2.7 The Normal Distribution

The normal distribution, also known as the Gaussian distribution, is the most important and commonly used probability distribution in statistical analysis. Numerous response variables (dependent variables) are normally distributed including but not limited to height, blood pressure and IQ scores. The normal distribution has two parameters μ and σ . As with all other probability distributions, we can use the acronym CUSS (Center, Unusual Features, Shape, Spread) in order to describe the characteristics of the normal distribution. Variables that follow a Gaussian distribution are centered about the mean μ . Unusual features include outliers, gaps, or clusters in the data. The normal distribution is a symmetric distribution and is graphically represented by what is known as the bell curve (Figure 4). The spread of the normal distribution is described by its standard deviation σ .

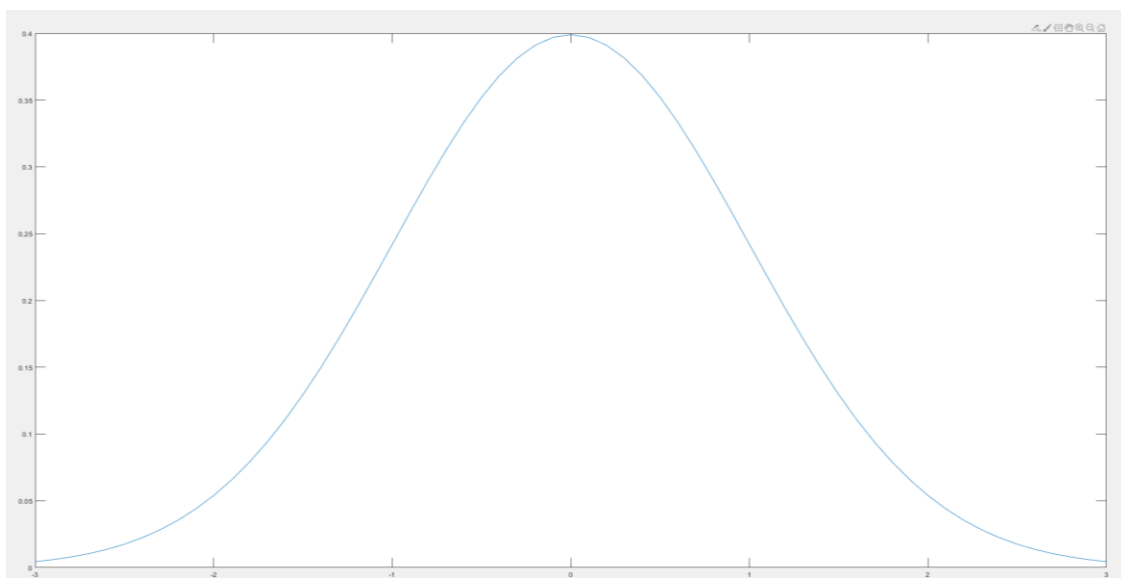


Figure 4: A normal distribution with mean 0 and standard deviation 1.

2.8 Indices Included in our ESG

We wanted our ESG to extrapolate historical data of seven specific stock indices: S&P 500, NASDAQ, Russell 2000, MSCI EAFE, MSCI EM, Russell 1000 Growth, and Russell 1000 Value. Table 5 provides a brief description of each of the seven stock indices used in our ESG and lists a few of the constituent companies.

Table 5: The stock indices involved in our project, involving a brief description and a few example companies included within each stock.

Name	Description	Companies
S&P 500	The S&P 500 index is a collection of the 500 largest stocks in the United States. This index combines about 80% of the U.S. market's capitalization. It is the best indicator of how large stocks are performing, and therefore it is one of the most followed indices by investors.	Apple Microsoft Amazon AT&T
NASDAQ	The NASDAQ is one of the most-followed indices in the U.S. stock market. The NASDAQ composite index contains over 3,300 stocks listed on NASDAQ stock exchange. It is a broad-based market index mainly weighted toward technology companies.	Apple Microsoft Netflix Starbucks
Russell 2000	A small-cap stock market index that measures 2000 of the bottom companies in another index, Russell 3000, which includes the 3000 largest U.S. stocks. It is used as a way for investors to see a wide measurement of the overall performance of small-cap stocks.	Cheese Factory INC Children's Place Retail Coca Cola Bottling
MSCI EAFE	MSCI EAFE index stands for Morgan Stanley Capital International and represents the performances of 21 developed markets, excluding the U.S. and Canada. This index is widely used as a benchmark and the basis of index-linked financial goods.	Nestle Toyota Motor CORP HSBC Holdings (GB)
MSCI EM	MSCI EM index also stands for Morgan Stanley Capital International and measures the performance of stocks in the global emerging market. It includes mid-cap and large-cap stocks of 12 emerging market countries. This index represents over 13% of the global market capitalization.	China Mobile Samsung Electronics Co Alibaba Group Hldg Adr
Russell 1000 Growth	The Russell 1000 is a leader index of large cap investing which includes 1000 largest companies in the U.S. stock market. The Russell 1000 Growth index represents those companies included in the Russell 1000 index and exhibits a probability of growth.	Target Corp Microsoft Corp Visa Inc
Russell 1000 Value	The Russell 1000 is a leader index of large cap investing which includes 1000 largest companies in the U.S. stock market. The Russell 1000 Value index represents those companies included in the Russell 1000 index and exhibits a value probability.	Johnson & Johnson Bank Of America Corp Disney Walt Co

2.9 Calculating Monthly & Annual Log Returns

Monthly log returns can be calculated from given stock prices by applying the following formula:

$$\text{log return} = \ln(\text{current price} / \text{previous price})$$

For example, if the price of a stock in April 1998 is \$30 and \$37 in May 1998, then the log return for May 1998 is $\ln(37/30) \approx 0.209720531$. In order to obtain the annual log returns, multiply the values calculated for monthly log returns by 12. The corresponding annual log return for our example would be $12 * \ln(37/30) \approx 2.516646372$.

2.10 Maximum Likelihood Estimation & Mary Hardy's Paper

In order to define the maximum likelihood estimator, we first must define our data sample. Suppose that $\{A_1, \dots, A_n\}$ is our sample of size n , where A_j is the value of the j^{th} observation. Also assume that A_j is the observed value of the random variable X_j . The random variables $\{X_1, \dots, X_n\}$ are not required to be identically distributed. However, their distributions should all depend on the same parameter vector θ . The final assumption is that the random variables are independent.

The goal of maximum likelihood estimation is to determine the parameter value that maximizes the probability of obtaining the observed sample. Since it is assumed that the observations are independent, the likelihood function is:

$$L(\theta) = Pr(X_1 \in A_1, \dots, X_n \in A_n | \theta) = \prod_{j=1}^n Pr(X_j \in A_j | \theta)$$

The natural logarithm of the likelihood function is:

$$\ln[L(\theta)] = \ln(\theta) \sum_{i=1}^n x_i + [n - \sum_{i=1}^n x_i] \ln(1 - \theta)$$

Since it is easier to maximize the natural logarithm of the likelihood function, the log-likelihood function is used instead of the likelihood function in order to find the maximum likelihood estimator of θ (Klugman et al., 2012, p.260-261).

For example, suppose that we have a sample of Bernoulli distributed random variables all dependent on the parameter θ . The likelihood function is:

$$L(\theta) = \prod_{i=1}^n \theta^{x_i} (1 - \theta)^{1-x_i} = \theta^{\sum_{i=1}^n x_i} (1 - \theta)^{n - \sum_{i=1}^n x_i}$$

The corresponding log-likelihood function is:

$$\ln[L(\theta)] = \ln(\theta) \sum_{i=1}^n x_i + [n - \sum_{i=1}^n x_i] \ln(1 - \theta)$$

We then perform the following steps in order to obtain the maximum likelihood estimator of the parameter θ .

$$\begin{aligned} \text{Set } \frac{d}{d\theta} \ln[L(\theta)] &= \frac{1}{\theta} \sum_{i=1}^n x_i - \frac{1}{1-\theta} [n - \sum_{i=1}^n x_i] = 0 \\ \frac{1}{\theta} \sum_{i=1}^n x_i &= \frac{1}{1-\theta} [n - \sum_{i=1}^n x_i] \Rightarrow \frac{n\bar{x}}{\theta} = \frac{n - n\bar{x}}{1-\theta} \Rightarrow \frac{n\bar{x}}{\theta} = \frac{n(1-\bar{x})}{1-\theta} \\ &\Rightarrow \frac{\bar{x}}{\theta} = \frac{1-\bar{x}}{1-\theta} \Rightarrow \bar{x} - \theta\bar{x} \\ &\theta = \bar{x} \end{aligned}$$

Therefore, we can see that the maximum likelihood estimator of the parameter θ is the mean of our sample.

“A Regime-Switching Model of Long-Term Stock Returns” by Mary Hardy provided a detailed explanation of how to obtain the likelihood function needed to utilize maximum likelihood estimation in order to find the parameters of our model.

2.11 Covariance & Correlation

When we generate random numbers and use them to generate our log-normally distributed returns for each index, we may have uncorrelated results. However, this is not accurate to the real world because many stocks are correlated, and we would expect that in each economic state, the stock returns would move in a similar way. For example, the S&P 500 and the NASDAQ share similar stocks (Apple, Microsoft), so if the S&P 500 returns move in a direction, we would expect the NASDAQ returns to move in the same direction. Therefore, we account for the correlation between two indices by considering their correlation and covariance. For each index in relation to another, we have a correlation value. With these values, we can create a correlation matrix. A simple example of a correlation matrix between two indices is shown in Table 6:

Table 6: Example Correlation Matrix between S&P 500 and NASDAQ

	S&P 500	NASDAQ
S&P 500	1	0.95
NASDAQ	0.95	1

these random variables before multiplying by lower triangle L: the random variables must have a variance of 1, and they must be uncorrelated. Therefore, we can create random variables with the variances necessary to have the correlated values we wanted them to have in relation to each other. This allows us to create random returns between stocks that includes the necessary variance needed to return data accurate to historical data.

To reflect this method, suppose we have an example covariance matrix reflecting to covariance between the S&P500 and the NASDAQ. For reference, the values in this table were generated from using Table 6 as the correlation matrix, then setting $\sigma = 0.05$ for the S&P500 and $\sigma = 0.06$ for the NASDAQ. Using the formula shown in Figure 5, we generated the values for the covariance matrix. An example of the usage of this method is shown below to find the covariance between the S&P500 and the NASDAQ:

$$\begin{aligned} Cov(S\&P500, NASDAQ) &= Corr(S\&P500, NASDAQ) * \sigma_{S\&P500} * \sigma_{NASDAQ} \\ 0.95 * 0.05 * 0.06 &= 0.00285 = Cov(S\&P500, NASDAQ) \end{aligned}$$

The final table representing the covariance matrix is shown in Table 7:

Table 7: Example Covariance Matrix between S&P 500 and NASDAQ

	S&P 500	NASDAQ
S&P 500	0.0025	0.00285
NASDAQ	0.00285	0.0036

Because this matrix is symmetric and positive definite, we can take the Cholesky decomposition of this matrix (Table 8). Note that the covariance between a random variable with itself is equivalent to the random variable's variance.

Table 8: Example Lower Triangle Matrix between S&P 500 and NASDAQ Generated from Cholesky Decomposition

0.05	0
0.057	0.0187

Notice this matrix is a lower triangle matrix. We will refer to this matrix as L. If we took the transpose of L and multiplied it by L, we would return the values given in Table 7 ($C = L * L^t$). Now suppose we want to use this lower triangle matrix to correlate random variables. In our project, we are returning normal returns. Suppose we are trying to correlate these random arbitrary values that follow the normal distribution:

Table 9: Example Uncorrelated Normal Returns with Variance = 1

S&P 500 Normal Inverse Return	1.213
NASDAQ Normal Inverse Return	0.453

The normal inverse returns for the S&P500 and the NASDAQ given in Table 9 follow Normal(0, 1) with no correlation between the two. However, we want to reflect correlation between the returns because we know that the S&P500 and the NASDAQ have a correlation of 0.95. To correlate these returns, we simply must multiply the lower triangle L with the uncorrelated random return from Table 9. After this multiplication, the following returns are:

Table 10: Resulting Correlated Normal Returns

S&P 500 Correlated Normal Return	0.0607
NASDAQ Correlated Normal Return	0.0776

Notice that the values for the returns in Table 10 have been adjusted to be a lot closer in value. In addition to this, the variances of these returns are not 1 anymore. Instead, they now reflect the standard deviations that contributed in making the covariance matrix and the Cholesky decomposition matrix.

CHAPTER 3: METHODOLOGY

To achieve the goal of our project, we needed to build an ESG that considered any number of years and scenarios as input, consider the correlation between our chosen stock indices, reflect historical data from our stock indices through MLE, and output a simulation of a possible future outcome. We followed these steps in creating an ESG that would accomplish these goals:

1. Make the program user-friendly by creating a one-page input excel sheet to manage the program easily.
2. Create a system for generating random numbers and use these numbers to determine how our ESG will switch regimes.
3. Code the fundamental lines in VBA to create a regime-switching lognormal model with three regimes.
4. Create a formula for maximum likelihood estimation to analyze historical data and create realistic parameters to reflect the seven stock indices.
5. Utilize Cholesky decomposition in creating correlation between the results of our returns for each index.
6. Output our final returns as a CSV file so that the large data will not create a huge file and can easily be turned into an Excel sheet.

3.1 One-Page Input

When developing our code, we created a “One Page Input” Excel sheet. This is meant to act as an organized way to display our numbers. We labeled each cell with a “name” that is referred to in our code, and thus our code is not bound to specific cells. Therefore, for anyone that wishes to use our ESG, they will be able to easily change the starting state, number of scenarios, number of years per scenario, values of means or standard deviations, transition probability values, and correlation matrices. As a result, the code reads the values of the named cells, and will run the code with these new numbers. A visual of this one-page input can be found in Appendix A.

3.2 Method for Creating and Using Random Numbers

To prevent the code from carrying out too many operations, we avoided using “RAND()” directly in our code, as this would need to generate a random number each time we needed it. Considering we needed thousands of random numbers for the Markov chain, as well as to create our returns, our team instead created a second sheet within the Excel file specifically made for random numbers. We generated a list of random numbers depending on how many random numbers we needed. This list is broken into multiple columns: the first column is for the Markov chain process, and the remaining columns are for each index. The Markov chain random numbers are necessary to determine how to switch between each regime, and the index random numbers are necessary to simulate a future return. The number of random numbers needed for each column can be found through the multiplication of the input values

for the number of years and the number of scenarios, as each column needs only one random number for every year generated. Once we generated a list of random values using RAND() on this sheet, we copied and pasted the values of these random numbers so that Excel would not continuously refresh the values. With this process completed, we could now refer to the values in the list within our code to have the random numbers we needed. A visual of this random number spreadsheet can be found in Appendix B.

An example of how we used a random number in the Markov chain is as follows. Suppose we are considering the weather example with probabilities given in Figure 2. In this demonstration, suppose we are starting in the “sunny” state. This means that there is a 90% chance of tomorrow being sunny and a 10% chance of tomorrow being rainy. We determined the state of tomorrow by generating a random number and using that number as the determinant of the regime switch. For the sunny state, we looked at the value in the first column of the matrix (0.9) and chose the range of 0 to 0.9 to be the sunny state range. Then, we looked at the second column. The value in the second column of the matrix (0.1) was added to the upper bound of the previous range (0.9) to determine the upper bound of the second range. The lower bound of the second range would start at where the previous range ended (0.9). Therefore, the second range was chosen to be 0.9 to 1. We repeated this process for the case where we are starting in the rainy case. These ranges were made a cumulative distribution function (CDF), as shown in Table 11:

Table 11: Cumulative Distribution Function Matrix that Results from our Weather Example

	new state = sunny	new state = rainy
starting state = sunny	$t_{11} = 0.9$	$t_{11} + t_{12} = 1$
starting state = rainy	$t_{21} = 0.5$	$t_{21} + t_{22} = 1$

With the CDF, we can use the random numbers generated from RAND() to decide what the next state will be. Suppose we are starting in a sunny state and we generated a random value of 0.93212. In this case, the program will first check to see if 0.93212 is less than or equal to 0.9. If it is, then we determine that the new state is sunny. If the equation is not true, we move onto the next column to check. The program determines that the new state is not sunny, so we look at the next CDF value. The program now checks to see if 0.93212 is less than or equal to one. Because this statement is true, the program determines that the new state is rainy.

3.3 Adding Markov Chains to our ESG in VBA Through Code

For each scenario, we wanted the ESG to transition between three regimes at random, generating returns based on the parameters and inputs provided. We denoted “1” as being in a good state, “2” as being in a bad state, and “3” as being in a catastrophic state. We created a

for-loop that starts at year 1 and ends at the number of years inputted within the one page input sheet. We set the “current state” equal to the “new state” at the beginning of the for-loop. This means that for each year we use the Markov chain, we are considering the previous year’s state.

Within the for-loop, we coded the Markov chain process as an if-statement so that each year has a transition. For the Markov chain if-statement, our code will consider the “current state” of the year and use a random number from our frozen list of random numbers to determine the “new state.” Specifically, for the first year of each scenario, our code takes the inputted starting state from the one page input sheet and makes it the first state of the scenario and the “new state” for the first year of the Markov chain. A visual of the code format for our Markov chain is shown in Figure 6.

```
If current state = 1 Then
    If random number is less than or equal to CDF transition11 Then
        new state = 1
    Elself random number is less than or equal to CDF transition12 Then
        new state = 2
    Else
        new state = 3
    End If
If current state = 2 Then
    If random number is less than or equal to CDF transition21 Then
        new state = 1
    ...
End If
If current state = 3 Then
    If random number is less than or equal to CDF transition31 Then
        new state = 1
    ...
End If
End If
```

Figure 6: Brief code diagram of how we set up our Markov chain in VBA.

We will demonstrate this code with an example. Suppose we are starting in state 1. The code will start in year one and determine that starting state = new state (placed right before the for-loop). Now, new state is equal to 1. However, as mentioned previously, we have a current state = new state at the beginning of the for loop. Now, current state = 1. We receive a random number from the frozen random number spreadsheet. Because current state = 1, we must look at the if-statement that says current state = 1. Going in order, we will check to see if the random number is less than the CDF t_{11} value (method explained in previous section, **3.2 Method for Creating and Using Random Numbers**). If it is, we determine that the new state = 1. Otherwise, we move onto the Elseif statement to check the condition. If the random number is less than $t_{11} + t_{12}$, new state = 2. Otherwise, we know that the new state must be 3 (Else statement).

Next, when we are in year 2 and trying to determine the state of year 3, the current state = new state at the beginning of the for-loop will turn the resulting new state from year 1 into the current state of year 2. This process will repeat for each year until all years are complete and the states are determined. Then, the for-loop will move onto the next scenario, where we once again start with the starting state as 1, and we will generate as many scenarios as inputted by the one-page input sheet.

3.4 Gathering of Historical Data

With the assistance of Emily Chen, a graduate student in WPI's Financial Mathematics program, the monthly returns for the seven indices were obtained by using the Bloomberg Terminal. The monthly returns are given as prices in U.S. dollars and data is available from as far back as 1987, possibly even earlier.

3.5 Calculating Annual Returns

The monthly log returns were calculated from the given returns by applying the following formula:

$$\text{log return} = \ln(\text{current price} / \text{previous price})$$

For example, if the price of a stock in April 1998 is \$30 and \$37 in May 1998, then the log return for May 1998 is $\ln(37/30) \approx 0.209720531$. We then multiplied the values calculated for monthly log returns by 12 in order to obtain the annual log returns. The corresponding annual log return for our example would be $12 * \ln(37/30) \approx 2.516646372$. With these values, we were able to find realistic parameter values for our seven indices (mean and standard deviation) and for the regime-switching component of our model (transition probabilities) using maximum likelihood estimation.

3.6 Obtaining Realistic Model Parameters

To figure out how we would obtain realistic values for our model parameters, we referred to “*A Regime-Switching Model of Long-Term Stock Returns*,” a paper written by Mary Hardy. In her essay, Mary Hardy states that we can solve for the model parameters we desire by applying maximum likelihood estimation. The Likelihood Function is able to express the validity of different parameter values for a given sample of data, such as the historical monthly stock prices of the S&P 500 and the Toronto Stock Exchange (TSE 300) from 1956 to 1999. Once we complete the maximum likelihood estimation process, we have found the model parameters that best fit our data.

In her paper, Mary Hardy used historical data of the S&P 500 and TSE 300 indices to illustrate the results of applying maximum likelihood estimation to monthly log returns. For our ESG, we wanted our parameters to be based on annual log returns. While Mary Hardy’s model only involves two indices, our ESG involves seven: S&P 500, NASDAQ, Russell 2000, Russell 1000 Growth, Russell 1000 Value, MSCI EAFE and MSCI EM. We decided that regime 1 would be the “good” economic state characterized by high returns and low volatility and regime 2 would be the “bad” economic state characterized by low returns and high volatility.

3.7 Obtaining Model Parameters for S&P 500 by Maximum Likelihood Estimation

In order to perform maximum likelihood estimation, we first had to calculate the probability density function (PDF) for each observation. Then by taking the natural logarithm of the PDFs and summing them up, we were able to determine the Log-Likelihood function. Finally, the optimization of the Log-Likelihood function was accomplished with the help of Microsoft Excel’s solver add-in.

We started off the process of maximum likelihood estimation by first defining the random variable y_t to be the log return at time t . Also used was the random variable ρ_t , defined by Mary Hardy to be the regime the model is in at time t . Considering the historical log returns as our data sample, we then set out to determine the probability density functions for each of our data points.

Since each log return came from either regime 1 or regime 2, we determined that the probability density function for y_t , the observed log return at time t , can be expressed as the following:

$$f(y_t) = Pr(\rho_t = 1)f(y_t|\rho_t = 1) + Pr(\rho_t = 2)f(y_t|\rho_t = 2)$$

$Pr(\rho_t = 1)$ is the probability that the model is in regime 1 at time t . Likewise, $Pr(\rho_t = 2)$ is the probability that the model is in regime 2 at time t . $f(y_t|\rho_t = 1)$ is the probability density function of the observed log return at time t , given that the model is in regime 1 at time t . Similarly, $f(y_t|\rho_t = 2)$ is the probability density function of the observed log return at time t , given that the model is in regime 2 at time t .

Because log returns are normally distributed and each regime is characterized by its own unique set of parameters, the conditional probability density functions of our observations can be described as follows:

$$f(y_t|\rho_t = 1) \sim \text{Normal}(\mu_1, \sigma_1^2)$$

$$f(y_t|\rho_t = 2) \sim \text{Normal}(\mu_2, \sigma_2^2).$$

As Markov chains play a critical role in determining which regime the model is in at any time, our next step was to derive recursive formulas for $Pr(\rho_t = 1)$ and $Pr(\rho_t = 2)$.

$$Pr(\rho_t = 1) = Pr(\rho_{t-1} = 1|y_{t-1})p_{11} + Pr(\rho_{t-1} = 2|y_{t-1})p_{21}$$

$$Pr(\rho_t = 2) = Pr(\rho_{t-1} = 1|y_{t-1})p_{12} + Pr(\rho_{t-1} = 2|y_{t-1})p_{22}$$

$Pr(\rho_{t-1} = 1|y_{t-1})$ is the probability that the model was in regime 1 at time $t - 1$, given that y_{t-1} was observed. Comparably, $Pr(\rho_{t-1} = 2|y_{t-1})$ is the probability that the model was in regime 2 at time $t - 1$, given that y_{t-1} was observed. The formulas for these probabilities can be obtained by applying Bayes' theorem, yielding the following:

$$Pr(\rho_{t-1} = 1|y_{t-1}) = \frac{f(y_{t-1}|\rho_{t-1} = 1)Pr(\rho_{t-1} = 1)}{f(y_{t-1})}$$

$$Pr(\rho_{t-1} = 2|y_{t-1}) = \frac{f(y_{t-1}|\rho_{t-1} = 2)Pr(\rho_{t-1} = 2)}{f(y_{t-1})}.$$

If you want the probabilities at time t , replace the $t - 1$ subscripts with t .

p_{11} and p_{22} are the probabilities that the model stays in its current regime. p_{12} is the probability the model switches from regime 1 to regime 2 and p_{21} is the probability that the model switches from regime 2 to regime 1. Since the model either stays in its current regime or switches to the other regime, we know that $p_{11} + p_{12} = 1$ and $p_{21} + p_{22} = 1$. As we were also aiming to find the maximum likelihood estimators of p_{12} and p_{21} , $p_{11} = 1 - p_{12}$ and $p_{22} = 1 - p_{21}$ were used when necessary.

As our data starts at time $t = 1$, the recursive formulas cannot be used. For this special case, we consulted with Mary Hardy's paper in which she suggests we use the following formulas:

$$Pr(\rho_t = 1) = \pi_1 = \frac{p_{21}}{p_{12} + p_{21}}$$

$$Pr(\rho_t = 2) = \pi_2 = \frac{p_{12}}{p_{12} + p_{21}}$$

Here, π_1 and π_2 represent the steady state probabilities of being in either of the two regimes. Therefore, we can rewrite the probability density function for the observed log return at time $t = 1$ as,

$$f(y_1) = \pi_1 f(y_1|\rho_1 = 1) + \pi_2 f(y_1|\rho_1 = 2)$$

After determining the probability density function for each of observations, we take their natural logarithms and sum them together to yield the Log-Likelihood Function, which is given by the following expression:

$$\sum_{t=1}^n \ln[f(y_t)]$$

To wrap up this procedure, we created an Excel spreadsheet model using these formulas and used the Solver add-in in order to determine the maximum likelihood estimators of our model parameters.

3.8 Maximum Likelihood Estimation for the Other 6 Indices

As the most important of the seven indices used, we decided to use the S&P 500 index as a benchmark for the regime-switching aspect of our model. While computing the maximum likelihood estimators for the S&P 500, we added a column to the Excel spreadsheet that would keep track of what regime the model is in at time t .

We decided that if $Pr(\rho_t = 1|y_t) > Pr(\rho_t = 2|y_t)$, then the model is in regime 1 and otherwise it is in regime 2. After obtaining our maximum likelihood estimators for the S&P 500 index, we were able to see its historical regime-switching pattern. The reason why this was important is that we wanted to ensure that at any time t , all seven indices are in the same regime. As a result, the maximum likelihood estimators for the transition probabilities only had to be calculated once. This also allows us to simplify the formulas needed in order to find the maximum likelihood estimators of the parameters for the other six indices.

Since we now know what regime the model is in $\forall t$, the probability density function for y_t can be described as follows:

$$\text{If } \rho_t = 1, f(y_t) = f(y_t|\rho_t = 1), \text{ otherwise } f(y_t) = f(y_t|\rho_t = 2).$$

The rest of the maximum likelihood estimation process was carried out in an identical manner as that for the S&P 500 index.

3.9 Choosing Regime 3 Parameters

The goal for regime 3 was to make an economic state that is worse than what our economy has previously faced. We made a specific assumption about this regime; because we expect that when it is a very bad economic state, that every index will face a similar amount of devastation within their returns, we made our returns all have a correlation of 1. Transitively, this means that all our indices had the same means and standard deviations for this regime.

We obtained our data to create these parameters from Yahoo Finance. We looked at the S&P 500 and downloaded the CSV file for the daily closing returns from January 1st, 2007 to January 1st, 2009. With these values, we took the 250-day ratio to represent the length of a

trading year, and then we looked at the mean, standard deviation, maximum, and minimum. This ratio was meant to roughly represent a realistic time where the economy was facing a very bad state. We then took the mean and standard deviation and adjusted these numbers to represent an even worse situation. Our chosen values were arbitrarily adjusted to reflect a catastrophic scenario, and thus regime 3 transition probabilities are given as 0 to exclude it from our results. Additionally, users will most likely wish to change these values to their own findings, and thus inputting different values for the transition probabilities and the parameters for the S&P 500 will automatically adjust the ESG for these input changes.

3.10 Correlating Our Returns

We obtained the data of correlation between indices by taking historical data, splitting the data into either regime 1 or regime 2 depending on the return, and using the “CORREL(…)” Excel function to find the correlation between the indices. Using these correlation matrices, we converted them into covariance matrices using the parameters we estimated from our maximum likelihood estimator. We then used the covariance matrices and Cholesky decomposed them using MATLAB to find a lower triangle for regime 1 and regime 2. With these lower triangles, we were able to consider the different correlations depending on the economic state the year was in. To change the random returns from our random numbers spreadsheet into returns that correlate between indices, we needed to convert the variance and standard deviation of the random numbers. RAND() generates random numbers from the distribution Uniform(0, 1). The formula for variance for Uniform(a, b) is:

$$(b-a)^2/12$$

Thus, we used this formula to solve for the variance of Uniform(0,1), and evaluate if we needed to change the random variables to have unit variance. Plugging in the values, we find that the variance of a Uniform(0,1) distribution is equal to 1/12. Therefore, to adjust the variance and standard deviation, we used the “NORM.INV(…)” function in Excel, which uses inverse normal transformation. Specifically, we applied a mean of 0 and a standard deviation of 1 so that the random numbers would have the characteristics necessary to be correlated. With this, when we multiply our lower triangle L from the Cholesky decomposition with our random variables of mean 0 and standard deviation 1, we now have correlated random numbers. These correlated random numbers follow Normal(0, σ^2). To adjust the random numbers to have the means we are looking for, we add the mean of that regime to the value of the random numbers, turning our random numbers into Normal(μ , σ^2). Therefore, with this method, we have generated our normal returns for our ESG.

3.11 Adding Correlation to our ESG in VBA Through Code

Once we determined a method of correlating our returns, we implemented our findings into code to produce correlated normal returns. Within the for-loop that goes through each year, we added code structured similarly to Figure 7.

```
S&P 500 Normal Inverse = NORMINV(S&P 500 random number, 0, 1)
NASDAQ Normal Inverse = NORMINV(NASDAQ random number, 0, 1)
...
random number = random number + 1
Array(S&P 500 Normal Inverse, NASDAQ Normal Inverse, ...)
If current state = 1
    Multiply Array and Cholesky Lower Triangle for regime 1 = Correlated Array
    Add means for regime 1 to each respective return in the Correlated Array
Elseif current state = 2
    Multiply Array and Cholesky Lower Triangle for regime 2 = Correlated Array
    Add means for regime 2 to each respective return in the Correlated Array
Else
...
End If
```

Figure 7: Brief code diagram of how we set up correlating returns in VBA.

As mentioned in the **3.10 Correlating Our Returns** section of our paper, we needed to account for the random numbers starting in Uniform(0, 1) not having unit variance. We therefore took the random numbers and used the “=NORMINV(…)” function to return these random numbers as normal returns with mean 0 and standard deviation 1. We include random number = random number + 1 after creating these normal returns because it will move onto the next numbers in the random number spreadsheet in the for-loop. Next, we take all the normal returns and add them into an array. Once we have this set up, we will check which regime we are in. Depending on the regime, we will correlate the array by multiplying the array with the Cholesky lower triangle for the respective state we are in. Once we have our new correlated array, we can add the means for that state to each index return, and these results

will be outputted as our final returns. After the state switches and we move onto the next year, this process is repeated.

3.12 Outputting our Results as a CSV File

Once we had our returns processed within our code, we wanted a method to output the results in an organized manner so that a reader could look at the data and understand the output. We listed the output results in this order: scenario number, year number, the state number, the S&P 500 return, the NASDAQ return, the Russell 2000 return, the MSCI EAFE return, the MSCI EM return, the Russell 1000 Growth return, and the Russell 1000 Value return (Figure 8). Every line of data outputted will be in this order, and thus we can tell the specific scenario and year each return belongs to. The output file format we chose was a “comma-separated value” (CSV) file. This file can be uploaded into a text program and will separate each consecutive number in a line with a comma. If the user instead wanted to open this file into Excel, the file can be opened and instead of being separated by commas, each number per row will be separated into individual cells. Thus, scenario number would be column one, year number would be column two, etc.

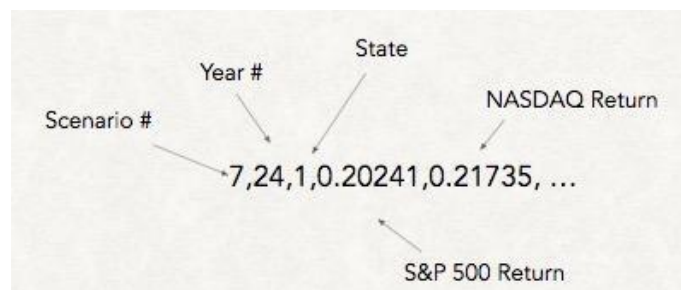


Figure 8: The order in which we listed our data in the CSV file.

CHAPTER 4: RESULTS AND ANALYSIS

4.1 ESG Run Time

Our ESG took one minute and 42 seconds to run 10,000 scenarios, each 50 years long. Overall, this is very efficient as we intended to design our code to have as few operations as possible. This should give the user flexibility in running many scenarios in a relatively short period of time.

4.2 Maximum Likelihood Estimation Results for Model Parameters

After building our Excel spreadsheet models and utilizing the Solver add-in, we were able to determine the maximum likelihood estimates for our model parameters. Figure 9 shows our results. Note that the regime-switching process was based on the S&P 500 index. Therefore, the transition probabilities p_{12} and p_{21} determined for the S&P 500 index are essentially the same for all seven indices.

Parameter	MLE w/ Annual Log Returns						
	S&P 500	NASDAQ	Russell 2000	Russell 1000 Growth	Russell 1000 Value	MSCI EAFE	MSCI EM
μ_1	0.1308	0.1582	0.1284	0.1388	0.1183	0.1075	0.1462
σ_1	0.2792	0.3938	0.4249	0.3051	0.2850	0.3866	0.5660
μ_2	0.0311	0.0456	0.0307	0.0339	0.0226	-0.0517	-0.0032
σ_2	0.6239	0.9697	0.8056	0.7313	0.6257	0.7181	0.9618
p_{12}	0.0349						
p_{21}	0.0360						

Figure 9: Results of Maximum Likelihood Estimation for Model Parameters

4.3 Analysis of Maximum Likelihood Estimation Results

By looking at the results of our maximum likelihood estimation, we can summarize the economic trends exhibited by the seven indices. Our seven indices were chosen by our project advisors in such a way that they do a fairly good job at representing the economy.

Figure 1 shows a summary of the economic trends exhibited by selected indices. There was a 3.49% chance that the economy would switch from being in a “good” situation to being in a “bad” situation. Similarly, the chance that the economy would switch from being in a “bad” state to a “good” state was 3.60%. While calculating the MLEs for the S&P 500 index, we were keeping track of which regime the model was in at time t . The historical data we collected consisted of 385 observations for each index, starting from December 1987 and ending at January 2020. From this, we concluded that historically, our economy spent 198 months (\approx 51.43% of the time) in “good” financial conditions and spent 187 months (\approx 48.57% of the time) in “bad” financial conditions. These were relatively close to our steady state probabilities of being in either of the two states: 50.8% (regime 1) and 49.2% (regime 2).

4.4 Steady State Ratios of Results

We created a state counter to compare the ratios of each regime to the steady state probabilities. Using a base case of 10,000 scenarios, each 50 years long, we tested our ESG using the maximum likelihood estimates of transition probabilities, as well as with arbitrary values for transition probabilities (Figure 10).

Note: All results were found using the same frozen list of random numbers for the Markov Chain.							
Arbitrary Transition Values, 3 Regimes:				Probabilities from MLE:			
	Regime 1:	Regime 2:	Regime 3:		Regime 1:	Regime 2:	Regime 3:
State Count:	237877	168420	93703	State Count:	315618	184382	0
Ratio of Total:	0.475754	0.33684	0.187406	Ratio of Total:	0.631236	0.368764	0
Steady State:	0.4753	0.3416	0.1931	Steady State:	0.508	0.492	0 (From MATLAB)
Transitions:	(PDF)			Transitions:	(PDF)		
	1	2	3		1	2	3
1	0.6	0.3	0.1	1	0.965112	0.034888106	0
2	0.5	0.3	0.2	1	0.036026	0.963973666	0
3	0.1	0.5	0.4	3	0	0	0

Figure 10: Comparison of arbitrary values to the maximum likelihood estimates.

We noticed that when we utilized our maximum likelihood estimates, the ratios of each regime did not match the steady state for its transition matrix, but when we used arbitrary values, the resulting ratios were much closer to its corresponding steady state. Based on these results, we concluded that the code worked because the arbitrary transition probability values were able to reach the steady state. We also ran our ESG utilizing the maximum likelihood estimates of transition probability values for 20,000 scenarios instead of 10,000 scenarios to see if the regime 1 and regime 2 ratios would move towards their steady state values. The ratios of regime 1 and regime 2 were closer to 0.508 and 0.492, respectively, meaning we needed more scenarios for the ratios to converge to the steady state probabilities.

4.5 Analysis of our ESG Results

We analyzed the simulation to determine how the ESG performed regarding generating simulated stock returns that accurately reflect historical economic trends (Figure 11).

MLE Parameter Values							
Parameter	S&P 500	NASDAQ	Russell 2000	Russell 1000 Growth	Russell 1000 Value	MSCI EAFE	MSCI EM
μ_1	0.1308	0.1582	0.1284	0.1388	0.1183	0.1075	0.1462
σ_1	0.2792	0.3938	0.4249	0.3051	0.2850	0.3866	0.5660
μ_2	0.0311	0.0456	0.0307	0.0339	0.0226	-0.0517	-0.0032
σ_2	0.6239	0.9697	0.8056	0.7313	0.6257	0.7181	0.9618

ESG Simulation Results							
Parameter	S&P 500	NASDAQ	Russell 2000	Russell 1000 Growth	Russell 1000 Value	MSCI EAFE	MSCI EM
μ_1	0.1302	0.1575	0.1280	0.1384	0.1176	0.1079	0.1461
σ_1	0.2801	0.3946	0.4261	0.3058	0.2852	0.3860	0.5657
μ_2	0.0321	0.0459	0.0313	0.0348	0.0239	-0.0505	0.0011
σ_2	0.6247	0.9712	0.8059	0.7327	0.6257	0.7195	0.9635

Figure 11: Comparison of MLE Parameters (top) with Results of ESG Simulation (bottom).

In order to compare the results of our ESG simulation to the historical trends exhibited by our seven indices, we conducted a series of two-sample hypothesis tests at the 5% significance level ($\alpha = 0.05$). To compare the means, we implemented the two-sample t-test. We also performed the F-test in order to compare standard deviations. For each comparison, the null hypothesis was that the historical parameter value (obtained by MLE) was equal to the parameter value generated by our ESG simulation. Figure 12 shows the resulting p-values from the hypothesis tests.

P-Values of Two-Sample Hypothesis Tests							
Parameter	S&P 500	NASDAQ	Russell 2000	Russell 1000 Growth	Russell 1000 Value	MSCI EAFE	MSCI EM
μ_1	0.9784	0.9802	0.9904	0.9836	0.9754	0.9890	0.9969
σ_1	0.9745	0.9932	0.9830	0.9913	0.9868	0.9497	0.9658
μ_2	0.9817	0.9975	0.9909	0.9857	0.9770	0.9807	0.9514
σ_2	0.9914	0.9963	0.9783	0.9989	0.9718	0.9969	0.9990

Figure 12: P-Values of Two-Sample Hypothesis Tests used for comparison of MLE & ESG results.

The p-values of all the hypothesis tests are far greater than our chosen α . This means that we did not have enough statistical evidence to reject any of our null hypotheses. In other words, our statistical analysis did not show any significant differences in parameter values between the historical data and the data produced by our simulation. Hence, we were able to conclude that our ESG can successfully extrapolate historical data, generating stock returns that accurately reflect past trends.

4.6 Covariance and Correlation Matrix

We were able to analyze how accurate our maximum likelihood estimates of regime 1 and regime 2 standard deviations were based on the correlation and covariance matrices formed from the Bloomberg historical data. Starting with the correlation matrix, we were able to use the standard deviations of each index to form a covariance matrix. With the covariance matrix we formed, we could compare the values to the covariance matrix that was formed directly using the Covariance function from Excel’s Analysis ToolPak on the Bloomberg data.

Actual	NASDAQ	S&P 500	Russell 2000	Russell 1000 Growth	Russell 1000 Value	MSCI EAFE	MSCI EM
NASDAQ	0.155	0.088	0.139	0.108	0.076	0.058	0.098
S&P 500	0.088	0.076	0.088	0.078	0.073	0.053	0.071
Russell 2000	0.139	0.088	0.181	0.095	0.091	0.062	0.099
Russell 1000 Growth	0.108	0.078	0.095	0.093	0.066	0.051	0.072
Russell 1000 Value	0.076	0.073	0.091	0.066	0.081	0.054	0.072
MSCI EAFE	0.058	0.053	0.062	0.051	0.054	0.149	0.109
MSCI EM	0.098	0.071	0.099	0.072	0.072	0.109	0.32

MLE	NASDAQ	S&P 500	Russell 2000	Russell 1000 Growth	Russell 1000 Value	MSCI EAFE	MSCI EM
NASDAQ	0.155	0.089	0.139	0.108	0.076	0.058	0.098
S&P 500	0.089	0.078	0.089	0.08	0.074	0.054	0.072
Russell 2000	0.139	0.089	0.181	0.095	0.091	0.062	0.099
Russell 1000 Growth	0.108	0.08	0.095	0.093	0.066	0.051	0.072
Russell 1000 Value	0.076	0.074	0.091	0.066	0.081	0.054	0.072
MSCI EAFE	0.058	0.054	0.062	0.051	0.054	0.149	0.109
MSCI EM	0.098	0.072	0.099	0.072	0.072	0.109	0.32

Figure 13: Comparison of Regime 1 covariance matrices (top Bloomberg, bottom MLE).

Actual	NASDAQ	S&P 500	Russell 2000	Russell 1000 Growth	Russell 1000 Value	MSCI EAFE	MSCI EM
NASDAQ	0.94	0.524	0.676	0.66	0.418	0.476	0.661
S&P 500	0.524	0.407	0.43	0.447	0.377	0.367	0.453
Russell 2000	0.676	0.43	0.649	0.485	0.405	0.421	0.581
Russell 1000 Growth	0.66	0.447	0.485	0.535	0.376	0.401	0.511
Russell 1000 Value	0.418	0.377	0.405	0.376	0.391	0.344	0.422
MSCI EAFE	0.476	0.367	0.421	0.401	0.344	0.516	0.542
MSCI EM	0.661	0.453	0.581	0.511	0.422	0.542	0.925

MLE	NASDAQ	S&P 500	Russell 2000	Russell 1000 Growth	Russell 1000 Value	MSCI EAFE	MSCI EM
NASDAQ	0.94	0.513	0.676	0.66	0.418	0.476	0.661
S&P 500	0.513	0.389	0.42	0.437	0.369	0.359	0.443
Russell 2000	0.676	0.42	0.649	0.485	0.405	0.421	0.581
Russell 1000 Growth	0.66	0.437	0.485	0.535	0.376	0.401	0.511
Russell 1000 Value	0.418	0.369	0.405	0.376	0.391	0.344	0.422
MSCI EAFE	0.476	0.359	0.421	0.401	0.344	0.516	0.542
MSCI EM	0.661	0.443	0.581	0.511	0.422	0.542	0.925

Figure 14: Comparison of Regime 2 covariance matrices (top Bloomberg, bottom MLE).

The highlighted columns of Figure 13 and Figure 14 represent the values that differ from the covariance values from the Bloomberg data. We noticed that these differences were contained only within the S&P 500. This could be due to error within the formulas used to obtain our MLEs, but it provided us with accurate values for every other index, and the matrix was able to decompose. Overall, the accuracy between the covariance matrices indicated that our maximum likelihood estimates of the regime standard deviation accurately represented the historical data we utilized.

CHAPTER 5: CONCLUSION AND RECOMMENDATIONS

5.1 Code Writing

Based on our project, VBA in Microsoft Excel was able to run our 10,000 scenarios, each 50 years long. This was effective for our goal, but we are limited to 1,048,576 random numbers, meaning our ESG is limited in how many scenarios it may run. While we chose to use VBA to write the code, the ESG can be generated through any coding language as long as it can handle large amounts of data, and we encourage readers of this report to use a different coding language if it provides a higher scenario and year count compared to VBA.

5.2 Data Collection

Based on the data collection process described in our methodology section, we recommend using the Bloomberg Terminal to gather financial data. It provided us with the monthly total returns data we needed for maximum likelihood estimation and allowed us to save time as we did not have to check the data against other sources in order to verify its reliability. Yahoo Finance was not as reliable because the data retrieved from the website sometimes presented incremental data incorrectly. In our case, when we were getting the monthly returns, each return was given one month later than the actual corresponding return month. This created many issues with our data, so we recommend Bloomberg as a more reputable and accurate data source.

5.3 Steady State Ratios of Results

Based on the observation that utilizing our MLE probabilities caused the ESG to not match the steady state values, we needed to figure out why this difference was occurring despite arbitrary values working. We were able to deduce that the MLE transition probabilities resulted in regime 1 and regime 2 not converging to the steady state ratios for two reasons:

1. The likelihood of changing states given the MLE transition probabilities are very low, meaning convergence would occur if we ran many more scenarios; convergence is extremely slow.
2. An Excel file can only hold so many random numbers; we could generate at most 1,048,576 random numbers, meaning we can only run 20,971 scenarios (each 50 years long).

We recommend trying to generate as many scenarios as possible to get as close to the steady state probabilities. As mentioned about the programming language, if you can use a different language to support generating many more scenarios than VBA, it might be best to use that language so you can achieve results that more closely resemble the steady state.

5.4 Correlation and Covariance Matrices

When generating a covariance matrix to Cholesky decompose separately using MATLAB, a user may find that their covariance matrix will not decompose. We recommend that the user be wary of the two requirements for Cholesky decomposition for a matrix: symmetrical and positive definite. Specifically, little incremental changes in values of the covariance matrix may pose issues due to the covariance matrix values being so small. Therefore, we recommend users of the program to take notice of what changes they are making to the standard deviation values, as well as the correlation values, as this may make the resulting covariance matrix unable to decompose.

5.5 Analysis of Data

For practical utilization of our ESG, we believe that every scenario generated should be independently analyzed and compared to the results of the maximum likelihood estimator. Based on the comparison of the results, each scenario can be put in one of the following three categories:

4. Improved Economy - characterized by higher means and lower standard deviations
5. Stable Economy - characterized by parameters equivalent to the MLE values
6. Worsened Economy - characterized by decreased means and increased standard deviations

The user can then base their decisions on the relative frequencies of the three categories. In other words, the user can make their decisions according to which economic future is the most likely to occur.

REFERENCES

- Klugman, S. A., Panjer, H. H., & Willmot, G. E. (2012). *Loss Models: From Data to Decisions* (4th ed.). John Wiley & Sons, Inc.
- Hardy, M. R. (2001). A Regime-Switching Model of Long-Term Stock Returns. *North American Actuarial Journal*, 5(2), 12.
- Pedersen, H., Campbell, M.P., Christiansen, S.L., Cox, S.H., Finn, C.D., Griffin, K., ... Suchar, C. (2016). *Economic Scenario Generators - A Practical Guide*, Society of Actuaries. Committee on Finance Research Research Expanding Boundaries Pool.
- Random Number Generator (RNG) (2017, February 1). Techopedia.
<https://www.techopedia.com/definition/9091/random-number-generator-rng>
- Stephanie (September 7, 2015). Maximum Likelihood and Maximum Likelihood Estimation, Statistics How To. <https://www.statisticshowto.datasciencecentral.com/maximum-likelihood-estimation/>

APPENDIX B: RANDOM NUMBER FROZEN SPREADSHEET

A	B	C	D	E	F	G	H	I
State Transitions	S&P 500	NASDAQ	Russell 2000	MSCI EAFE	MSCI EM	Russell 1000 Growth	Russell 1000 Value	
0.105399418	0.49496	0.423086	0.834322426	0.18630453	0.200736	0.723546251	0.320324816	
0.547473694	0.808542	0.907349	0.383069233	0.03493894	0.582403	0.022278172	0.439595432	
0.965633667	0.697483	0.481269	0.200385918	0.4712589	0.790103	0.893820812	0.148645087	
0.119674835	0.435601	0.986617	0.926980552	0.21050408	0.358077	0.615096691	0.372214238	
0.139240593	0.533544	0.457253	0.381693935	0.40488627	0.514969	0.785904578	0.131152637	
0.243673738	0.240199	0.783859	0.533077595	0.61019455	0.256359	0.530109018	0.048732053	
0.321835705	0.372403	0.452857	0.951001739	0.69656351	0.766808	0.973181036	0.2009646	
0.133972405	0.085362	0.287229	0.770891561	0.27111452	0.546442	0.546002265	0.380232668	
0.259363501	0.073329	0.070674	0.035276083	0.84620373	0.755816	0.644573187	0.926895117	
0.125541803	0.212727	0.636425	0.188167528	0.48743078	0.263131	0.660577613	0.695101663	
0.910147042	0.332727	0.865526	0.645758255	0.38303306	0.258201	0.794247686	0.902472296	
0.643728086	0.176432	0.020844	0.864440907	0.03539232	0.033673	0.420454048	0.950873321	
0.935945303	0.895989	0.157941	0.249321766	0.10548458	0.977848	0.013333765	0.909509564	
0.453390702	0.976048	0.853889	0.311049662	0.76775015	0.032035	0.307159145	0.272630758	
0.200872491	0.315545	0.820386	0.524172579	0.14630231	0.390115	0.605875617	0.717917523	
0.250187391	0.587892	0.979718	0.403104189	0.29427671	0.482283	0.311086962	0.153439223	
0.984046597	0.73557	0.528833	0.024555614	0.89306101	0.436584	0.536956721	0.289070022	
0.229996528	0.648875	0.074116	0.691508801	0.6881555	0.28853	0.958185429	0.468040154	
0.323690006	0.361407	0.883102	0.014086848	0.08024768	0.45248	0.482602427	0.931235391	
0.186884725	0.214976	0.293376	0.992755066	0.04162896	0.149475	0.900651684	0.932837695	
0.647302381	0.477126	0.073181	0.72051539	0.77112203	0.319394	0.160621742	0.339152602	
0.251575411	0.175003	0.14974	0.726702556	0.43444789	0.601447	0.21071964	0.377459837	
0.397380443	0.567126	0.01258	0.926452682	0.60950775	0.767324	0.59795979	0.629254462	
0.631139405	0.946326	0.337719	0.896541835	0.40196003	0.456004	0.311300772	0.991119769	
0.98321583	0.123134	0.624396	0.444147376	0.41155331	0.680484	0.648540034	0.091201394	
0.971547768	0.3871	0.147549	0.046608976	0.7211237	0.541603	0.799918199	0.336791318	
0.691480004	0.844997	0.983028	0.726056857	0.45523004	0.193738	0.996392403	0.951047409	
0.907832474	0.726806	0.514469	0.681987875	0.05832241	0.716371	0.98901121	0.464447614	
0.552542092	0.09563	0.593674	0.530130664	0.09369297	0.627647	0.711601074	0.656554879	
0.802856415	0.145226	0.196728	0.277986767	0.00314787	0.304297	0.394608499	0.08294235	
0.012688993	0.178471	0.724449	0.199245782	0.1070059	0.008281	0.48333164	0.773028001	
0.989365761	0.22481	0.457034	0.377265933	0.39255691	0.524237	0.098220409	0.513458759	
0.542564004	0.627914	0.027409	0.082874811	0.61561827	0.131928	0.557750435	0.14771336	
0.436105421	0.436838	0.31538	0.733278834	0.08339834	0.088113	0.695708139	0.927984934	
0.685904512	0.267469	0.924087	0.608290766	0.66830698	0.943572	0.066531123	0.377856318	
0.704164844	0.400246	0.554071	0.759515108	0.4891989	0.188881	0.309380399	0.624625165	
0.297630413	0.095409	0.529336	0.481325872	0.02206823	0.315426	0.094665097	0.209777561	
0.721353244	0.037415	0.104322	0.271406542	0.76564951	0.93626	0.31886988	0.583247318	
0.916704525	0.705848	0.789298	0.458608053	0.67377772	0.195092	0.815993433	0.431629542	
0.430495332	0.489999	0.085947	0.030020166	0.05028074	0.699339	0.436790136	0.781759131	
0.636134272	0.381277	0.669321	0.594066749	0.4066212	0.258952	0.406360605	0.986389624	
0.877543551	0.900417	0.29483	0.274741661	0.82401289	0.793271	0.280933455	0.647666542	
0.993822937	0.796428	0.881947	0.162108599	0.33263093	0.888812	0.565403163	0.034400587	
0.664336873	0.604813	0.411713	0.949428343	0.62783588	0.957381	0.775376664	0.439809526	
0.021527725	0.012415	0.336405	0.388582624	0.06163544	0.913258	0.648654817	0.971379107	
0.770481401	0.998917	0.851966	0.270486775	0.26808136	0.323748	0.077505128	0.787054015	
0.276814661	0.474475	0.992173	0.470762272	0.91364119	0.563681	0.469022276	0.050245161	
0.545268689	0.178433	0.248961	0.57653347	0.19153031	0.91114	0.480853266	0.737527464	
0.77078845	0.571184	0.218587	0.9595893	0.89442253	0.766278	0.410605746	0.868529218	
0.115249744	0.722014	0.735931	0.056331523	0.34424142	0.793002	0.946653128	0.743550556	
0.156410985	0.286238	0.967393	0.934753567	0.50128788	0.087667	0.162610754	0.317987497	
0.690635009	0.426681	0.488595	0.726318045	0.78052769	0.580286	0.065117874	0.664985907	
0.013134992	0.061358	0.466883	0.224775897	0.5654211	0.314718	0.021615282	0.992376942	
0.831362897	0.442897	0.926451	0.395770621	0.05702603	0.566187	0.963398627	0.953869961	