Indexed Universal Life

A Major Qualifying Project Report
Submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
In partial fulfillment of the requirements
for the Degree of Bachelor of Science
in Actuarial Mathematics
by:

Hairan Li
Andrew St. Jean
Sergey Tsitlenko

Sponsoring Agency: John Hancock Financial Services, Boston, MA

Submitted to:
Project Advisor: Jon Abraham, WPI
Liaison: Stephen O'Brien, Life Product Management, John Hancock

Date: 25 April 2013

Abstract

John Hancock Financial Services is a major provider of various life insurance products in the United States. Due to government statutes, the company can only illustrate each of their insurance products at one interest rate. The goal of this project was to determine an appropriate interest rate for John Hancock to use in illustrating their Indexed Universal Life Insurance policy. This was accomplished by building a model that simulates the policy multiple times and generates a probability distribution of the internal rates of return.

Acknowledgements

Our team would like to extend thanks to several individuals and organizations for their support and contributions to this project. We would like to thank our sponsor, John Hancock Financial Services for providing us with the opportunity to pursue this project. Stephen J. O’Brien, Life Product Management, was especially integral to the development and focus of our study. He guided our modeling process and provided us with insight into analysis techniques.

Furthermore, we would like to thank the Worcester Polytechnic Institute Mathematics Department for supporting this project. Jon Abraham, Actuarial Program Coordinator, was also integral to the development of our study and was a great help with analysis and final compilations.

Executive Summary

John Hancock Financial Services is a leading company in the insurance industry, providing life insurance, mutual funds, and annuities. This project focused on John Hancock's Indexed Universal Life insurance product, specifically trying to determine what interest rate to illustrate to potential customers. By government statute, insurance companies are only allowed to use one interest rate to illustrate the financial growth of an insurance policy to potential customers. Advertising the product with a higher interest rate will produce more business but may leave customers disappointed with the product's final outcome. Our team created a model of this life insurance product and determined internal rates of return for varying scenarios.

The main objective of this project was to determine a reasonable interest rate for John Hancock to use for illustrative purposes. To accomplish our goal, our project team did the following:

- Developed a Random Walk simulation
- Modeled the major features of the Indexed Universal Life policy
- Simulated the policy with varying parameters
- Conducted probability analysis on the simulation outcomes
- Determined optimal interest rates for illustrative purposes

This project required a strong understanding of the various features of an Indexed Universal Life policy. The major features of the model include the Random Walk projection of stock market returns, various cash flow patterns, and mortality based scenarios. These features were implemented into our model using Microsoft Excel and Visual Basic for Applications. Having implemented the various features of the Indexed Universal Life policy, we used Visual Basic for Applications to simulate the life of a policy 10,000 times and determine the probability distribution of the internal rates of return for each scenario.

The purpose of this project was to provide John Hancock with a model that projects stock market returns, cash flow patterns, and mortality scenarios to determine an internal rate of return under certain policy assumptions. Using this model, we were able to calculate a reasonable confidence interval for the IRR. Our hope is that John Hancock can use these recommendations to more accurately advertise their product to potential customers.

Table of Contents

List of Figures

1. Introduction

In today's market, there are many types of life insurance products. Traditionally, life insurance pays out a sum of money to the beneficiary upon death of the policyholder or after a set period. However, new forms of life insurance have been developed in the past 30 years to satisfy different customer needs. One of the features in these new products is flexible premiums, where the customer can choose to deviate from the standard monthly payments of premiums. Another new feature is the option to invest part of the policy into the stock market.

John Hancock's existence can be traced back to 1862, when it was chartered as "John Hancock Mutual Life Insurance Company." In 2004, the company was acquired by Manulife Financial, a Canadian insurance company that operates in Canada, Asia, and the United States. John Hancock offers various financial services, including 401(k), Mutual Funds, Annuities, College Savings, Long Term Care, and Life Insurance. Some of their life insurance products include Variable Universal Life Insurance and Indexed Universal Life Insurance. The Indexed Universal Life product provides the option to invest a part of the policy into the stock market. Unlike a direct investment into the market, the IUL provides a guaranteed minimum return to prevent losses on the investment.

When illustrating the Indexed Universal Life policy to potential customers, John Hancock wants to show how the policy value will grow over time. Due to government statute, the company can only use one interest rate when illustrating the product. The sales agents would like to illustrate the IUL product at a higher interest rate in order to attract more customers, while the actuaries would prefer to use a lower interest rate to prevent overestimating the return. This rate of return will vary under different conditions. Some of the factors that may affect this rate include different market conditions and varying cash flow patterns. Our project aims to determine a reasonable rate of return for the Indexed Universal Life product that John Hancock can use in their illustrations to potential customers.

2. Background

2.1 Variable Universal Life Insurance

Variable Universal Life Insurance (VUL) was first developed and offered to the public in the late 1980s to satisfy customers' needs of both life insurance protection and investment opportunity. Compared to traditional whole life insurance, VUL is known for its flexibility of cash flows; specifically, the premium can be changed by the customer as needed. Furthermore, VUL accounts are similar to mutual funds in that cash value may be divided into separate accounts to be invested into stocks, bonds, and other securities according to the customers' preferences (Variable Life Insurance Policy Definition). Our model assumes that the stock market is the primary investment vehicle.

2.2 Previous MQP in 2012

The WPI Major Qualifying Project "Numerical Investigation for Universal Variable Life Insurance" by Han Yang and Maarja-Liisa Sokk modeled John Hancock's Indexed Universal Life product. The model was created and optimized using various analysis methods to determine what annual rate of return John Hancock should use to illustrate the policy.

The project team researched various techniques to model stock returns. In particular, they compared the differences between using the Regime Switch technique and the Random Walk model. Although the Regime Switch model is more complex than the Random Walk model, the team concluded that both models yield very similar results for the internal rates of return. In our model, we used the Random Walk model for stock market returns because it appeared to be the most practical. Using this modeling technique, our team implemented a dynamic model for the Indexed Universal Life policy with adjustable parameters.

2.3 Indexed Universal Life at John Hancock

In 2011, John Hancock launched the Indexed Universal Life policy (IUL), which is a life insurance policy designed to incorporate investments in the stock market, including protection against stock market downturns. This policy includes choices for the policy-holder about how to conduct investments that emulate the S&P 500's performance. There are two main options for this policy. The Capped Indexed Account has a capped rate of return of 13% and a minimum of 0%, The Uncapped Indexed Account returns 5% less than the S&P500 rate of return and has a minimum of 0%.

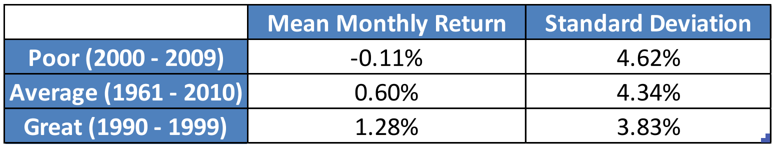
While the IUL reduces the inherent uncertainty of the stock market by using a cap and a floor, there is still a steady amount of variation involved. Depending on the market conditions, policyholders may end up with returns at either extreme. When illustrating this policy to a potential customer, it is important for John Hancock to provide a fair illustration of the policy's rate of return. Due to government statute, John Hancock can only illustrate a single interest rate to potential customers. However, an illustration that allowed for some kind of probability distribution of outcomes would be more meaningful to the John Hancock product development team. Our project aims to provide our sponsor with an estimated rate of return of an Indexed Universal Life policy based on probability distributions under varying conditions.

3. Methodology

3.1 Random Walk Model

Since the Indexed Universal Life product is based on the performance of the stock market, our model began as a Random Walk simulation. The Random Walk is a model that tries to capture the random movements of the stock market; it is widely used by economists and financial managers to simulate the market (Fama 75). Using the Random Walk model, we attempted to predict future stock prices under certain conditions. The model assumes that the stock price moves in normally distributed random increments. The parameters of this distribution are a (monthly) average return and standard deviation. The parameters generally imply certain market conditions; for example, a high mean may imply a really good market, while a low mean could represent a poor market. Once the parameters are chosen, the model generates random price increments using the normal distribution with the given parameters. This allowed us to randomly project future stock prices.

In order to properly use the Random Walk simulation, our team had to determine the relationship between market conditions and Random Walk parameters. To accomplish this, we analyzed historical monthly returns for the S&P 500 index. Similarly to the previous John Hancock MQP group, we chose three historical periods for our analysis: 1990-1999 (commonly associated with high market returns), 2000-2009 (when market returns were poor and unstable), and the 50-year period from 1961 to 2010 (which constitutes average market returns). We then calculated the average monthly returns and the standard deviations for each of the three historical periods. When a user runs a simulation, he can choose to use the parameters for a poor market, an average market, or a great market. The table below shows the parameters obtained from the S&P 500 data.



	Mean Monthly Return	Standard Deviation
Poor (2000 - 2009)	-0.11%	4.62%
Average (1961 - 2010)	0.60%	4.34%
Great (1990 - 1999)	1.28%	3.83%

Figure 1 – Table of Market Performance Parameters

3.2 Simulations Using VBA

Our next step was to build a mechanism that runs this simulation a pre-determined number of times and collects the results. The number of simulations is a user input item. It is beneficial to run many simulations in order to obtain a larger sample for the probability analysis. A large sample of 10,000 simulations provides much more significant results than a small sample of 100 simulations. To accomplish this analysis, our team created a Visual Basic for Applications (VBA) macro. The macro runs a loop that generates new sets of random price increments; each simulation results in a different internal rate of return. In order to conduct our analysis, the results are grouped into bins, the size of which is input by the user. The macro then outputs the results onto an empty sheet and graphed in a histogram, a probability graph, and a cumulative distribution graph.

In addition to running the simulations, our model uses VBA for several auxiliary purposes. While the simulations are being run, our macro displays a progress bar. This bar shows the progress of the simulation, allowing the user to estimate the time of completion. The progress bar slowly increases in size as the simulations run, reaching 100% when all the iterations are completed and the graphs are created.

One of the issues with our simulation macro was that it overwrote previous results, including the output and all of the graphs. To address this, we created another VBA macro that archives the results of the last simulation. The macro copies the inputs from the simulation and the cumulative distribution graph onto a new sheet. This allows the user to easily save the results of a simulation and compare them to results from other runs.

3.3 Twelve Monthly Segments

Having set up a basic simulation mechanism, we attempted to implement some of the features of the Indexed Universal Life. One of the key features of this policy is the 12-month guarantee, which differentiates Indexed Universal Life from a traditional investment in the stock market. The guarantee ensures that the policyholder's account will be credited at some minimum interest rate (the "floor") for a particular year even if the market return for that year was unfavorable. On the other hand, the guarantee caps the return at a certain rate (the “cap”). If the market performance in a particular year was higher than the cap, the policyholder receives an amount of interest based on the cap, not the market return.

Due to the 12-month guarantee, our model had to take into account the timing of cash flows. If the policyholder deposits $1,000 in January of 2012, that deposit earns an annual interest rate based on the period from January 2012 to January 2013 (the January “segment”). For simplicity, all deposits were assumed to have occurred on the 1st of the month. If the policyholder then makes another deposit in February of 2012, that deposit will be subject to a different interest rate guarantee than the January deposit, because that rate is obtained from the February 2012 – February 2013 segment. To properly represent these varying conditions, we divided the policy into 12 monthly segments. The model keeps track of 12 different balances, each corresponding to a monthly segment. The segments total to the policy’s account value at any given time.

Segmenting the policy into 12 sections created complications for the withdrawal patterns. Following the actual policy provisions, when a withdrawal occurs in our model, the money is taken from the current month’s segment. However, if the withdrawal exceeds the amount in the segment, the remaining amount is then taken pro rata (proportionally) from the remaining segments. For example, if the withdrawal is $2,000 in January, but there’s only $900 in the January segment, the January segment is emptied, and 1100/11 = $100 is then withdrawn from each of the remaining 11 segments.

3.4 Withdrawal/Deposit Patterns

During the life of a policy, the policyholder deposits various amounts of money into the policy. While in a real-life scenario the policyholder may alter his premium/deposit patterns, our model assumes only two options. The first option is a single premium deposit, where the policyholder deposits a lump sum at the beginning of the policy. The second option is to make level monthly deposits that stop when the withdrawals begin. According to our sponsor, the two options provided in our model closely represent real-life situations.

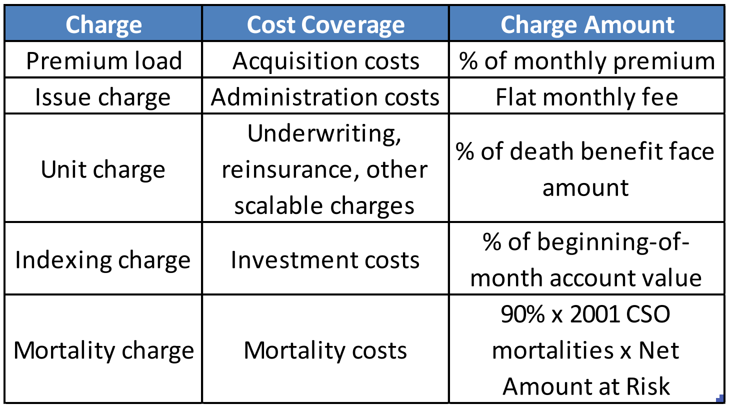
Once the deposits are finished, the policyholder in our model begins to withdraw money out of the account. The withdrawals begin once the policyholder reaches a certain age and end after a certain number of years. The starting age and the length of the withdrawal period are input by the user. In our model, we analyzed withdrawal patterns that reflect common real-life scenarios: withdrawing for 30 years starting at age 60, withdrawing for 20 years starting at age 70, and withdrawing for 15 years starting at age 90.

The amounts of the withdrawals vary between scenarios. Assuming a policyholder completes his withdrawal period before death, our model adjusts the withdrawals to leave money in the policy at the end of the planned withdrawal period. This sum is left in the policy to pay off the charges and keep the policy from lapsing after the withdrawal period ends. If the policyholder dies before the end of the planned withdrawal period, the withdrawal amounts are greater because the policy terminates at death. As a result, our withdrawal patterns are inherently deterministic, because they are dependent on when the policyholder dies. In a real life scenario, the policyholder's death age is unknown, and the withdrawal amounts are independent of the death age. However, this issue is beyond the scope of our project and will be discussed in the Future Development section of this report.

3.5 Charges and Death Benefit

Like most insurance products, the Indexed Universal Life includes several monthly charges that the policyholder must pay to keep the policy from lapsing. There are five charges that the model takes into account. The premium load is a percentage of the monthly premium, which covers acquisition costs. The issue charge is a flat monthly charge that pays for the administration costs of the policy. The unit charge is a percentage of the face amount of the death benefit. The unit charge pays for underwriting costs, reinsurance, and other scalable charges. The indexing charge is a percentage of the beginning-of-month account value (prior to taking out the other charges). This charge covers investment costs. Finally, the mortality charge is based on the 2001 CSO mortality tables and the net amount at risk. The mortality charge pays for the expected mortality, which increases with attained age. The mortality charge decreases the CSO mortalities by 90% to "show the effect of conservative underwriting, conservative tables, and mortality improvement since 2000" (S.

O'Brien, personal communication, January 23, 2013). See Figure 2 for a summary of the charges.



Charge	Cost Coverage	Charge Amount
Premium load	Acquisition costs	% of monthly premium
Issue charge	Administration costs	Flat monthly fee
Unit charge	Underwriting, reinsurance, other scalable charges	% of death benefit face amount
Indexing charge	Investment costs	% of beginning-of-month account value
Mortality charge	Mortality costs	90% x 2001 CSO mortalities x Net Amount at Risk

Figure 2 – Table of Charges

The death benefit is the amount received by the beneficiary of the policy when the policyholder dies. The beneficiary receives the initial death benefit (set by the user) plus the accumulated account value in the policy; this type of death benefit is referred as an "Option 2" death benefit in the insurance industry. The amount of the initial death benefit is determined by the user. It is important to ensure that the death benefit is reasonable relative to the deposit amounts because these are guidelines for the product to be considered "insurance." For single premium scenarios, we set the initial death benefit to be four times the single premium amount.

In our model, the policyholder has a probability of dying during each year of the policy. In some simulations, the policyholder dies after their scheduled withdrawal period has ended. Therefore, the policyholder must leave enough money in the policy at the end of the withdrawal period to cover the policy charges after his withdrawals end and until his death. To determine this amount, our model calculates the present value of these charges at the withdrawal end age. This present value is calculated using a monthly discount rate. In the insurance industry, companies typically use their own proprietary rate.

3.6 Order of Decrements

During the simulation, it was important for the model to apply the various decrements in a consistent order. Every month (after the first year), the current monthly segment accrues interest from the previous year. This interest and the monthly deposit (if any) are added to the account value. After the incoming cash flows are added, the charges are applied to the account. The indexing charge, which is a percentage of the account value, is applied first. The premium load, issue charge, and unit charge are applied next. The mortality charge is applied last. After the charges are applied, the withdrawal (if any) is processed.

3.7 Mortality Scenarios

Every year during the life of the policy, the policyholder has a certain probability of dying based on the 2001 CSO mortality tables. Using these probabilities, we implemented a mortality scenario that makes the policyholder die at random. For every year during the policy's projected lifespan, the simulation determines whether or not the policyholder died in that year. Upon death, the beneficiary receives the initial death benefit and the account value of his policy. The four mortality tables we used can be found on the Society of Actuaries website (http://www.soa.org/research/experience-study/ind-life/tables/intl-2001-cso-preferred-class-structure-mortality-tables.aspx). The tables we used were:

- 2001 CSO Select and Ultimate ANB Male Nonsmoker
- 2001 CSO Select and Ultimate ANB Male Smoker
- 2001 CSO Select and Ultimate ANB Female Nonsmoker
- 2001 CSO Select and Ultimate ANB Female Smoker

3.8 Calculating the IRR

Once all the parameters are correctly entered, our model calculates the internal rate of return for the given scenario. The IRR is calculated based on the cash flows of the policy. The positive cash flows included in the IRR are the deposits and the accrued interest. The negative cash flows included in the IRR are the withdrawals and the death benefit. The policy charges are not considered cash flows when calculating the IRR. However, the charges have an impact on the account balance and the final withdrawal (death benefit), which directly affects the IRR.

In our model, there is a small chance that the policyholder will die very soon after the policy is issued. In this scenario, the IRR calculation yields an abnormally large result. To simplify our probability analysis, we capped the IRR at 50%. This did not affect the integrity of our analysis.

4. Analysis and Discussion

In our analysis, we decided to keep several parameters constant. These parameters were assumed to stay constant across all possible scenarios. All of the charges fall into this category, because we assumed that the charges do not vary across different IUL policies. Our analysis used the following values for the charges: 5% premium load, $10/month issue charge, $0.10 unit charge per $1,000 of death benefit coverage, and a 0.01% (of Account Value) indexing charge. The mortality charge adjusts the mortality rate from the CSO table by a factor of 90% and multiplies that by the net amount at risk.

Although our model includes the option to vary the market condition between "poor," "average," and "great," our analysis assumes "average" market conditions. Our team made this assumption because we cannot predict future stock market conditions with certainty. However, another user of the model may choose a different market preset if he prefers. Our parameters for the mean and standard deviation are based on our historical analysis of the S&P 500 returns. Another user may choose to conduct his own analysis and use his own custom parameters instead of our presets.

There are several other parameters that we held constant. We assumed the customer to be a 40 year old, non-smoker male, which kept our mortality rates the same for each iteration. The initial death benefit remained constant at $1,000,000 for most of our scenarios. If the insured dies before the end of the withdrawal period, the death benefit is set to be $1,000,000 plus the account balance. The discount rate for charges after the end of the withdrawal period is held constant at 0.21% a month. This rate is used to determine how much money to leave in the policy at the end of the withdrawal period, in which case the policyholder dies after the withdrawals end; thus, the policy can remain in force, and the policyholder can avoid adverse tax consequences.

In our analysis, we vary three key parameters and compare the results. Our comparisons include the following: varying deposit patterns, varying withdrawal patterns, and cap/floor assumptions. In each of these comparisons, we vary one of the parameters while keeping the other two constant. Each comparison provides us with several graphs that allow us to draw conclusions about the internal rates of return. This will allow us to gauge how changing one of the key parameters affects the internal rate of return.

4.1 Deposit Patterns

We began our analysis by comparing the effects of different deposit patterns on the cumulative distributions of the internal rates of return. The two types of deposit patterns that our model is capable of simulating are monthly deposits and a single premium deposits. The monthly deposits involve a relatively small amount of money being deposited into the policy every month until the beginning of the withdrawal period. The single premium is a relatively large lump sum that is deposited in the first month of the policy; after the first month, no more premiums are to be paid. The amounts we used were $2,000 for the monthly deposits and $250,000 for the single premium. The initial death benefit was $1,000,000 for both scenarios.

Our initial hypothesis was that the different premium patterns would result in very different cumulative probability distributions of the IRR's. It seemed that depositing a lump sum at the beginning would yield higher returns than monthly deposits, since the lump sum is much larger than individual monthly deposits and would therefore earn more interest. However, these premium patterns resulted in very similar IRR distributions. As illustrated in the graph below, the distributions follow each other very closely. The single premium deposit has a slightly higher distribution than the monthly deposits, but the difference is negligible. Our results show that these premium patterns do not have a significant effect on the IRR of the policy.

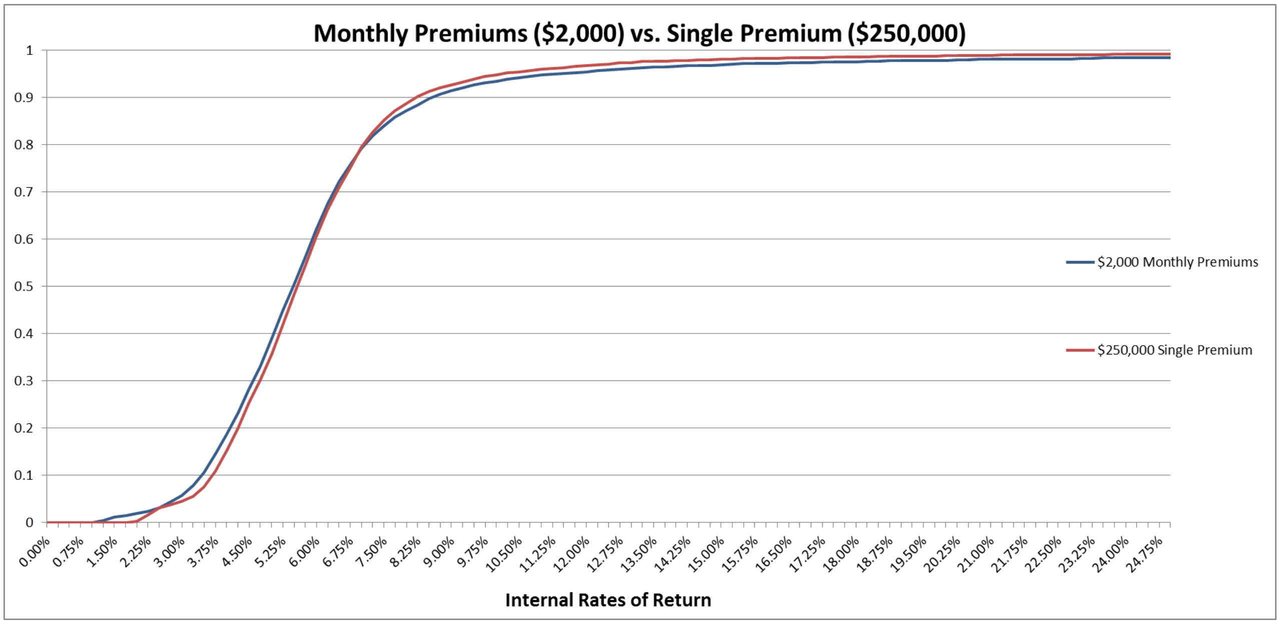


Figure 3 – Monthly Deposits vs. Small Single Premium

Having reviewed these results, we decided to run another scenario to confirm our findings. We changed the single premium amount from $250,000 to $1,000,000 to see if a larger single premium would yield different results. We adjusted the death benefit accordingly, changing it from $1,000,000 to $4,000,000 to maintain reasonability. We then compared our $2,000 monthly deposits scenario to the new $1,000,000 single premium scenario. Our results can be seen in Figure 4 below.

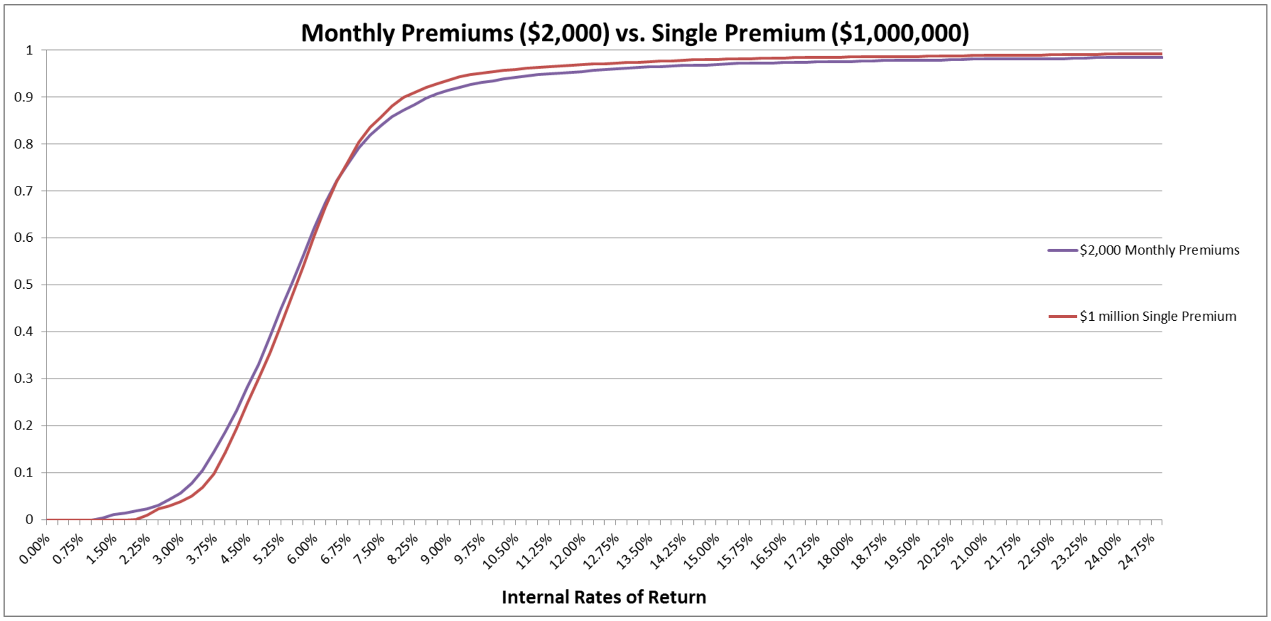


Figure 4 – Monthly Deposits vs. Large Single Premium

The resulting IRR distributions were very similar. There was a slight discrepancy at the extremes of the graph, but overall the IRR distributions under these two scenarios were very close. The increase in the single premium amount did not have a significant impact on the IRR distribution. This supports our results from the first comparison. Based on these results, we concluded that varying the deposit patterns within a reasonable range does not have a significant impact on the IRR of an Indexed Universal Life policy.

4.2 Withdrawal Patterns

Having conducted analysis on deposit patterns, we decided to measure the effect of varying the withdrawal patterns on the IRR. The withdrawal patterns that we looked at are: starting at age 60 for 30 years (60/30), starting at age 70 for 20 years (70/20), and starting at age 90 for 15 years (90/15). As with the deposit patterns, we expected that different withdrawal patterns would yield very different IRR distributions. Below is a graphical representation of the IRR distributions generated by these three withdrawal scenarios.

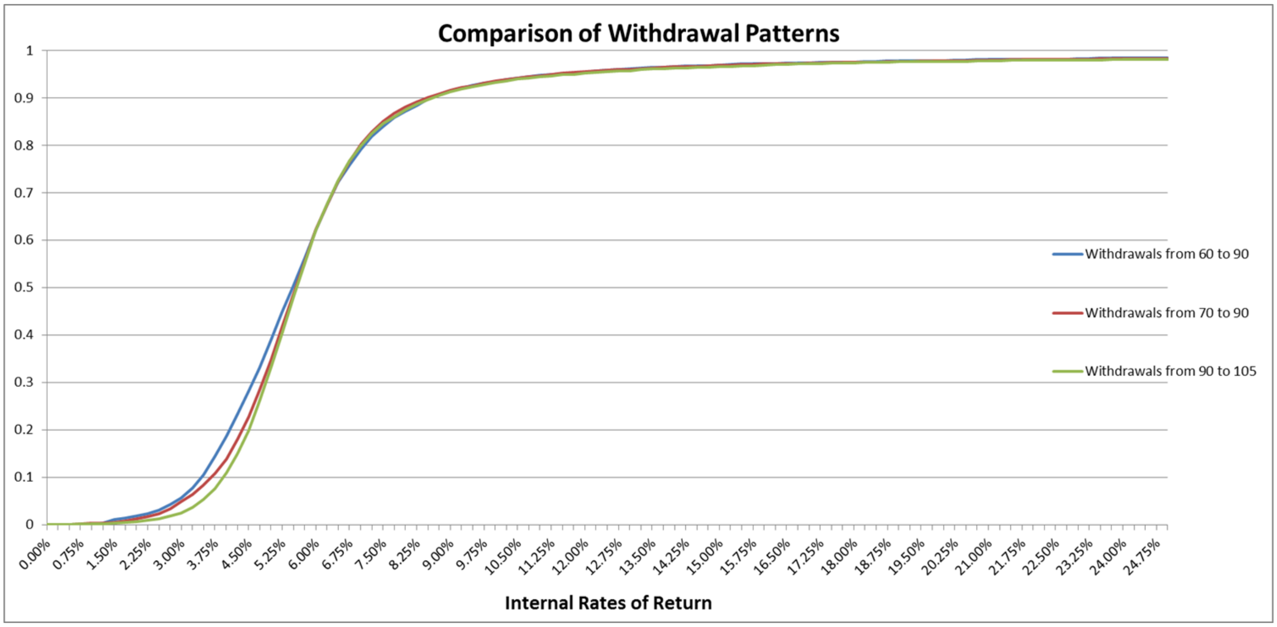


Figure 5 – Analysis of Withdrawal Patterns

Contrary to our expectations, the IRR distributions were not drastically different. At the lower extreme of the graph, there is some variation between the three withdrawal patterns. The 90/15 pattern generates higher IRR's in the lower half of the graph, roughly until P = 0.5. This result is valid because the 90/15 withdrawal pattern allows the account value more time to accumulate interest. The 60/30 pattern, on the other hand, begins to withdraw money earlier, thus accumulating less interest and yielding a lower IRR. However, the higher IRR's in the 90/15 scenario can also be attributed to the mortality feature of the model. In the model, many policyholders die before the age of 90. In the 90/15 scenario, these policyholders die before their withdrawal period begins. Since they did not withdraw any money from the policy, their account value had accumulated more interest and yielded a higher IRR distribution. Although there are slight differences between the distributions, they are not significant enough to be considered crucial. Based on these findings, we determined that the withdrawal patterns do not significantly affect the distribution of the IRR's.

4.3 Cap/Floor Assumptions

The next aspect of the policy that we looked into was the effect of having a cap and a floor on the interest that the policy accrues. We conducted analysis on three scenarios. In the first scenario, the monthly return was a flat 5% regardless of stock market conditions. This scenario reflects the fixed account option of the IUL policy, which allows the policyholder to invest without incurring risk. The second scenario is the 0% floor, 13% cap that the IUL currently offers. The third scenario removes the floor and cap entirely and closely resembles an investment into the stock market. This scenario is not a viable option in the IUL policy, but we decided to analyze it as a reference point. The IRR distributions that resulted from these scenarios are shown in Figure 6 below.

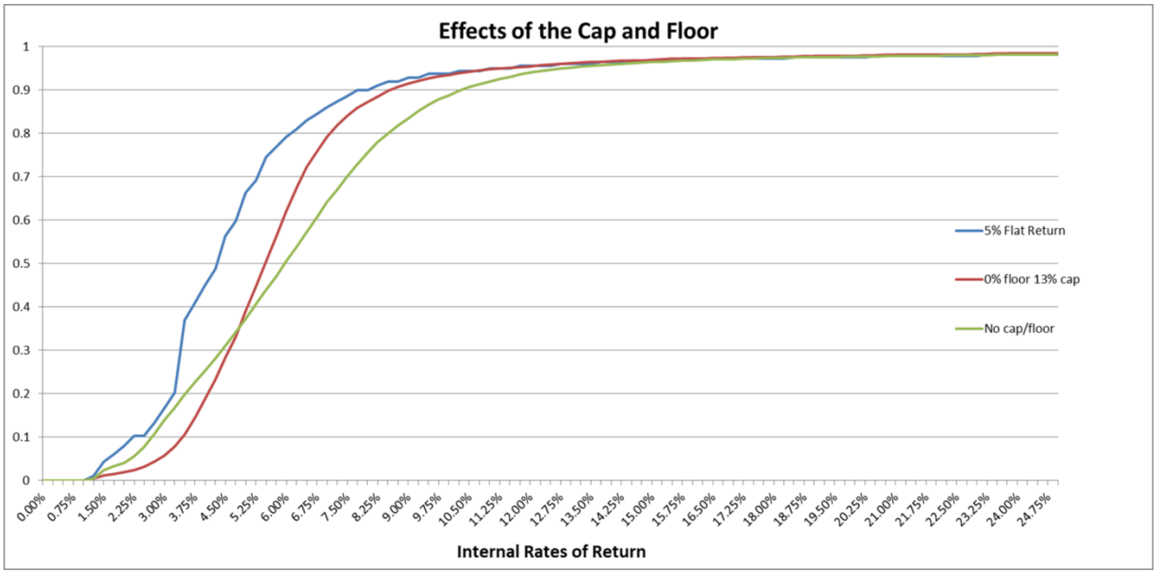


Figure 6 – Analysis of Cap/Floor Effects

This analysis provided us with interesting results. The jaggedness of the flat return graph can be explained by the mortality scenarios. In our model, the policyholder can only die at the beginning of the year, making the deaths discretely distributed. Since they cannot die at any time during the year, there is only one possible IRR for each death age in the case of a flat 5% return. Our IRR distribution is based on 0.25% intervals between each possible IRR; therefore, some of our potential IRR's have a probability of 0 because they are mathematically impossible to achieve.

The 5% flat return scenario yielded a much lower IRR distribution than the other two scenarios. This shows that an investment into the indexed account of the IUL is more profitable than the fixed 5% account, despite the higher degree of uncertainty. The no cap, no floor option appears to be more profitable than the other two, judging by the upper half of the distribution graph. However, taking out the floor removes the protection against market downturns that the IUL is supposed to provide. This option would not qualify as an insurance product, and would be subject to heavy taxation. In addition, this scenario's IRR cumulative distribution is significantly more volatile than the other two, which can be seen from the slope of the graph. Based on these findings, we decided to focus on the 0% floor, 13% cap scenario, which more accurately represents the IUL policy and yields more favorable IRR's than the 5% flat return option.

4.4 Recommendations

After analyzing three major factors, we formulated recommendations for illustrating the IUL policy. Our null hypothesis was that varying the premium patterns and withdrawal patterns would lead to noticeable variations in the probability distributions of the internal rates of return. However, our analysis showed that different patterns of premium deposits and cash withdrawals have little effect on the IRR distributions. Our recommendation is based on the scenario with:

- $2,000 monthly premiums
- Withdrawals starting at age 60 and lasting for 30 years
- 0% floor and 13% cap

Since the variation in the resulting IRR distributions is minimal, the following recommendation applies across multiple scenarios.

Based on our findings, we recommend that John Hancock use an interest rate between 4.5% and 7% for their illustrations of the Indexed Universal Life policy. Our findings show that there is a 50% chance the IRR will be in that range. Our results showed that the most probable interest rate is 5.75%, which is the midpoint of this confidence interval. Figure 7 (below) shows the cumulative probability distribution of IRR's and a graphical representation of the recommended confidence interval. The John Hancock product development team and future WPI project teams can use our model to conduct further analysis.

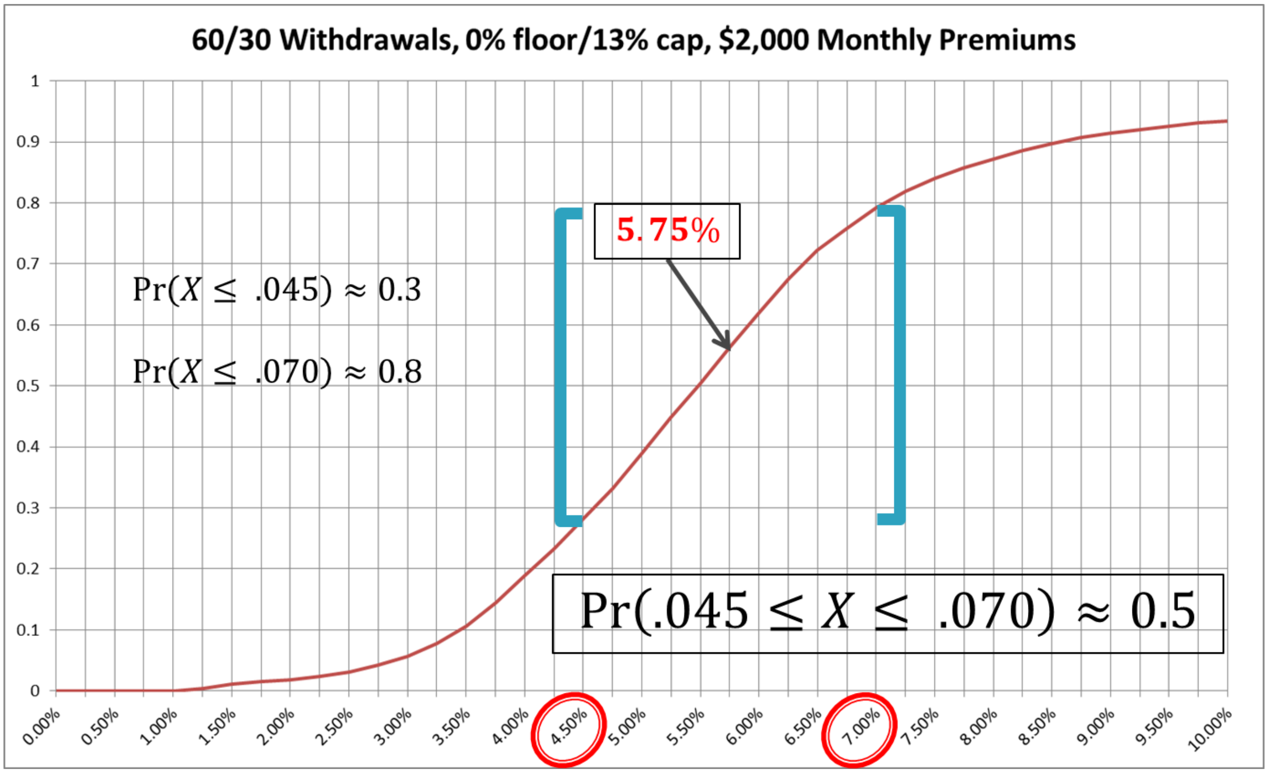


Figure 7 – Recommended Confidence Interval

4.5 Further Development

There are many possible developments for this model that go beyond the scope of this project. Due to our limited time and resources, we could only implement so many features into the model. Our analysis was similarly limited by these constraints. Discussions with our sponsor uncovered many opportunities for enhancing our model. These avenues can be explored by the product development team at John Hancock and/or future WPI MQP teams.

One aspect of our model that can be enhanced is the mechanism for developing randomized market returns. Our model currently generates random interest rates for each simulation, although the distribution of the interest rates is kept the same. This randomness causes a small amount of noise in the resulting probability distributions. When comparing simulation runs, some of the differences in IRR distributions may be attributed to this noise rather than actual differences in results. The noise issue can be addressed by implementing random seeding. Random seeding generates a number of random interest rate scenarios, which are then used in future simulations instead of generating new interest rates for each run. This eliminates the noise while keeping the random factor.

Another potential development for the model is a mechanism to determine the appropriate cap and floor for the policy. Our model allows the user to input the cap and floor manually. However, it may be useful to make the model calculate the optimal cap and floor. This can be done using the Black-Scholes partial differential equation and other market analysis techniques. This development can help John Hancock determine the optimal cap and floor for the Indexed Universal Life policy and make improvements to the product.

The analysis we conducted compared a policy fully allocated into a fixed interest account with a policy fully invested into an indexed account. While this provides some insight into these two options, it may prove beneficial to implement a partial allocation mechanism into the model. Such a mechanism would allow the user to allocate x % into the fixed account and (1-x) % into the indexed account. The user could then compare different allocations and determine an optimal split between the two accounts.

The model could also take into account various changes in customer behavior. A customer can use his life insurance policy in a number of ways. Some customers may be more interested in maximizing the death benefit, while others may be aiming to receive a steady income stream during retirement. These goals may have different assumptions; the first customer may be expecting an earlier death, while the second customer may be expecting to live long and prosper. The model can be developed to choose between these assumptions and base the analysis on them. Our model is capable of analyzing the differences between withdrawal patterns starting at different ages and with different lengths. However, it can be improved to accommodate other customer behaviors, such as retirement planning or college savings funds. The model can also be improved to implement varying degrees of persistency. This would account for situations where the policyholder changes their mind about their financial goals and decides to withdraw their account value, thus lapsing their life insurance policy.

The withdrawal mechanism in this model is deterministic in that the withdrawal amounts depend on the death age of the policyholder. If the policyholder dies after the end of the withdrawal period, the monthly withdrawal amounts are reduced in order to leave enough money in the policy to pay off the charges between the end of the withdrawal period and the death of the policyholder. If the policyholder dies before the end of the withdrawal period, the withdrawals aim to reduce the account value to zero by the end of the withdrawal period. It may be beneficial to remove the deterministic element from this mechanism by making the withdrawal amounts independent of the policyholder's age of death.

Another potential improvement for the model is the implementation of the death benefit. Currently, the initial death benefit amount is input by the user. It is up to the user's discretion to ensure that this amount is appropriate when compared against the account value of the policy. Generally, the death benefit should exceed the account value by 100% at all times. An improved mechanism would calculate an appropriate death benefit based on other model parameters, such as deposit patterns and amounts. In addition, the model could be improved to offer different death benefit options. Our model uses an Option 2 benefit, where the death benefit equals the initial death benefit plus the account value. Other options, such as the Option 1 (death benefit = face amount) could be offered as well.

Our analysis focused on the optimal IRR to be used for illustrative purposes. However, this model can be used to analyze other interesting topics. With a few modifications, our model could be used to determine an optimal premium under varying market and mortality assumptions. The model also has the potential to analyze withdrawals and determine the limits on withdrawals in order to not jeopardize the policyholder's plans for the product. These topics could be researched by future project teams.

Appendix A – Summary of Model

Worksheets

The model workbook is comprised of 10 spreadsheets; 1 contains the model of the policy, 4 are used for analysis of the simulation, 1 contains historical S&P 500 data, and 4 contain the 2001 CSO Select and Ultimate Mortality Tables. The other spreadsheets (highlighted in bright yellow) are archived results used in our analysis. Below is a summary of the worksheets in the workbook.

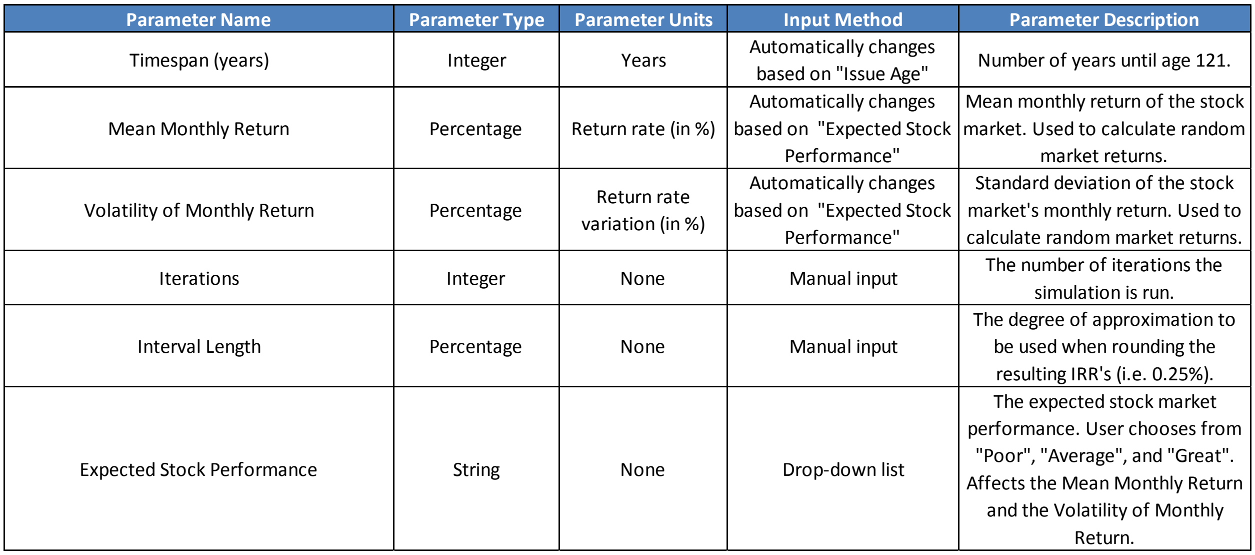
Worksheet Name	Worksheet Purpose
Model	Contains the model of the IUL policy
Output	Contains the IRR's obtained from the last simulation
Histogram	Contains the histogram constructed from the rounded/binned IRR's
P(x)	Contains the probability distribution of the rounded/binned IRR's
F(x)	Contains the cumlative distribution of the rounded/binned IRR's
Stock Performance Data	Contains historical S&P 500 return data, with means and volatilities for different historical periods
2001 CSO S&U ANB Male NS	Contains the 2001 CSO Select and Ultimate ANB Male Nonsmoker Mortality Table
2001 CSO S&U ANB Male S	Contains the 2001 CSO Select and Ultimate ANB Male Smoker Mortality Table
2001 CSO S&U ANB Female NS	Contains the 2001 CSO Select and Ultimate ANB Female Nonsmoker Mortality Table
2001 CSO S&U ANB Female S	Contains the 2001 CSO Select and Ultimate ANB Female Smoker Mortality Table

Figure 8 – Summary of Worksheets

Parameters

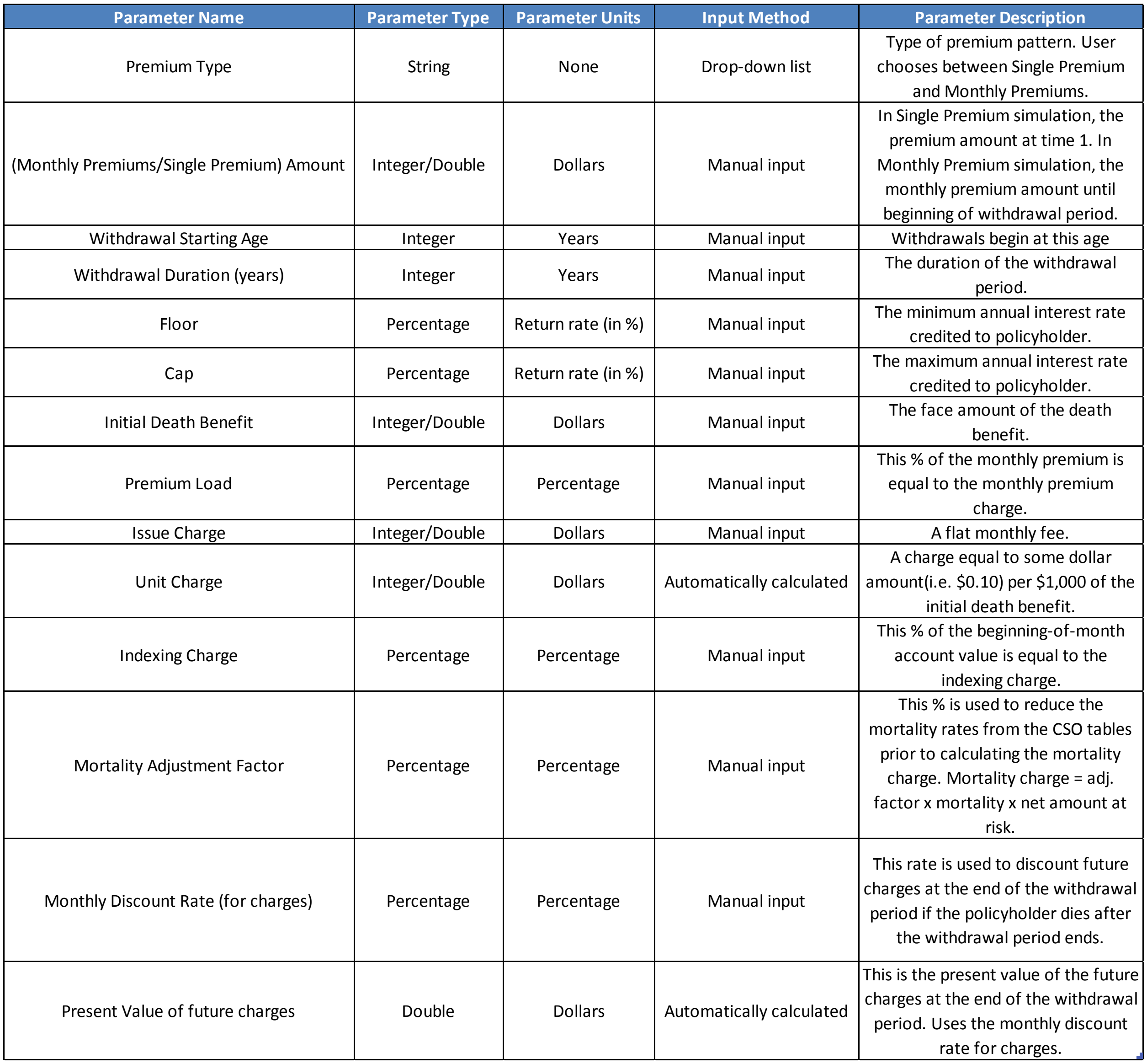
The model uses a number of parameters to simulate the Indexed Universal Life policy, located at the top of the “Model” worksheet. Some of these parameters are input by the user, while others are calculated automatically. Once all the parameters are appropriately entered, the user can begin the simulation by clicking the “Simulate” button in the top left corner of the “Model” worksheet. This will call the “Simulate” Macro, which runs the simulation as many times as the user wants and constructs probability and cumulative distribution graphs. These results can then be archived into a new sheet using the “Archive” button in the “F(x)” worksheet. This button calls the “Archive” macro that copies the current model parameters, probability distribution, cumulative distribution, and the corresponding graphs into a newly created sheet.

The following figures briefly summarize the model’s parameters.



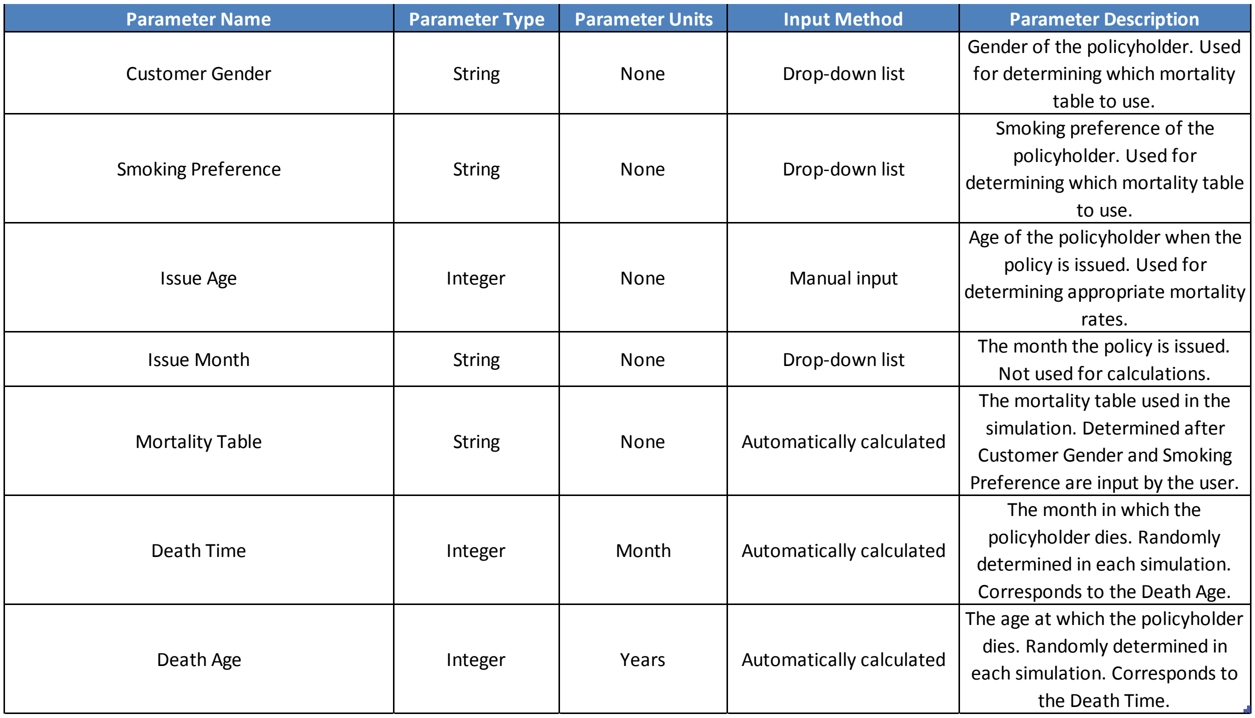
Parameter Name	Parameter Type	Parameter Units	Input Method	Parameter Description
Timespan (years)	Integer	Years	Automatically changes based on "Issue Age"	Number of years until age 121.
Mean Monthly Return	Percentage	Return rate (in %)	Automatically changes based on "Expected Stock Performance"	Mean monthly return of the stock market. Used to calculate random market returns.
Volatility of Monthly Return	Percentage	Return rate variation (in %)	Automatically changes based on "Expected Stock Performance"	Standard deviation of the stock market's monthly return. Used to calculate random market returns.
Iterations	Integer	None	Manual input	The number of iterations the simulation is run.
Interval Length	Percentage	None	Manual input	The degree of approximation to be used when rounding the resulting IRR's (i.e. 0.25%).
Expected Stock Performance	String	None	Drop-down list	The expected stock market performance. User chooses from "Poor", "Average", and "Great". Affects the Mean Monthly Return and the Volatility of Monthly Return.

Figure 9 – Summary of Model Parameters (part 1)



Parameter Name	Parameter Type	Parameter Units	Input Method	Parameter Description
Premium Type	String	None	Drop-down list	Type of premium pattern. User chooses between Single Premium and Monthly Premiums.
(Monthly Premiums/Single Premium) Amount	Integer/Double	Dollars	Manual input	In Single Premium simulation, the premium amount at time 1. In Monthly Premium simulation, the monthly premium amount until beginning of withdrawal period.
Withdrawal Starting Age	Integer	Years	Manual input	Withdrawals begin at this age
Withdrawal Duration (years)	Integer	Years	Manual input	The duration of the withdrawal period.
Floor	Percentage	Return rate (in %)	Manual input	The minimum annual interest rate credited to policyholder.
Cap	Percentage	Return rate (in %)	Manual input	The maximum annual interest rate credited to policyholder.
Initial Death Benefit	Integer/Double	Dollars	Manual input	The face amount of the death benefit.
Premium Load	Percentage	Percentage	Manual input	This % of the monthly premium is equal to the monthly premium charge.
Issue Charge	Integer/Double	Dollars	Manual input	A flat monthly fee.
Unit Charge	Integer/Double	Dollars	Automatically calculated	A charge equal to some dollar amount(i.e. $0.10) per $1,000 of the initial death benefit.
Indexing Charge	Percentage	Percentage	Manual input	This % of the beginning-of-month account value is equal to the indexing charge.
Mortality Adjustment Factor	Percentage	Percentage	Manual input	This % is used to reduce the mortality rates from the CSO tables prior to calculating the mortality charge. Mortality charge = adj. factor x mortality x net amount at risk.
Monthly Discount Rate (for charges)	Percentage	Percentage	Manual input	This rate is used to discount future charges at the end of the withdrawal period if the policyholder dies after the withdrawal period ends.
Present Value of future charges	Double	Dollars	Automatically calculated	This is the present value of the future charges at the end of the withdrawal period. Uses the monthly discount rate for charges.

Figure 10 – Summary of Model Parameters (part 2)



Parameter Name	Parameter Type	Parameter Units	Input Method	Parameter Description
Customer Gender	String	None	Drop-down list	Gender of the policyholder. Used for determining which mortality table to use.
Smoking Preference	String	None	Drop-down list	Smoking preference of the policyholder. Used for determining which mortality table to use.
Issue Age	Integer	None	Manual input	Age of the policyholder when the policy is issued. Used for determining appropriate mortality rates.
Issue Month	String	None	Drop-down list	The month the policy is issued. Not used for calculations.
Mortality Table	String	None	Automatically calculated	The mortality table used in the simulation. Determined after Customer Gender and Smoking Preference are input by the user.
Death Time	Integer	Month	Automatically calculated	The month in which the policyholder dies. Randomly determined in each simulation. Corresponds to the Death Age.
Death Age	Integer	Years	Automatically calculated	The age at which the policyholder dies. Randomly determined in each simulation. Corresponds to the Death Time.

Figure 11 – Summary of Model Parameters (part 3)

Appendix B – VBA Macro Code

```
' This code defines the Simulation() subroutine and the Archive() subroutines. Also defined are several
helper functions.

'These variables are public because they are used by more than one function
Public iterations As Long        ' number of iterations
Public timespan As Integer       ' timespan of the simulation (in months)
Public binsize As Double         ' size of the ranges used in the histogram (interval length)
Public maxbin As Double          ' the largest IRR interval (based on the largest result obtained)

'These are public declarations of the worksheet variables, to be set and used later in the subroutines
Public model As Worksheet
Public output As Worksheet
Public hist As Worksheet
Public px As Worksheet
Public fx As Worksheet

' The Simulation() subroutine runs the simulation in the "Model" worksheet for a given number of
‘ iterations. In each simulation, it calculates and stores the annualized IRR in the output spreadsheet. It
‘ then "bins" (rounds) each final IRR down to the nearest interval (based on the "Interval Length"
‘ parameter) and calls functions that create a histogram, probability distribution, and the cumulative
' distribution of these ranges.

Public Sub Simulation()

'Used to time how long the macro runs for
Dim tstart, ttotal As Single
tstart = Timer

'Turns off the automatic screen update in order to speed up the subroutine
Application.ScreenUpdating = False

'Initiates/sets the worksheet variables
Set model = Worksheets("Model")
Set output = Worksheets("Output")
Set hist = Worksheets("Histogram")
Set px = Worksheets("P(x)")
Set fx = Worksheets("F(x)")

'Initializes the parameters provided by the user in the "Model" worksheet
timespan = 12 * model.Range("F5")     ' timespan is provided in years and needs to be converted into
months
iterations = model.Range("I5")
binsize = model.Range("J5")
```

```
'These are internal variables used in the subroutine
Dim irrs() As Double      'an array that stores the annualized IRR after each iteration
Dim iternum() As Integer     'an array of integers from 1 to the number of iterations
Dim i As Long               ' loop counter

'Set the size of the two arrays defined above
ReDim irrs(1 To iterations)
ReDim iternum(1 To iterations)

'Calls helper function to clear the "Output", "Histogram", "P(x)", and "F(x)" worksheets
Call clearsheets

'Calls helper function that checks if the Analysis Toolpak (used to create the histogram) is installed on this
computer
Call checkAnalysisToolpak

'Adds labels to the "Output" sheet
output.Range("B1") = "Iteration #"
output.Range("C1") = "Annual IRR"
output.Range("D1") = "Bins"
output.Range("B1", "D1").Font.Bold = True

'Progress bar code
Dim percentcomplete As Single   ' determines percentage of progress bar to be shown
Dim begwidth As Double        ' stores the progress bar's initial width
begwidth = ProgressBar.LabelProgress.Width

Load ProgressBar    ' loads the progress bar form
   With ProgressBar
      .LabelProgress.Width = 0
      .Show vbModeless
   End With


'Begin at iteration 1 and continue until the input number of iterations is reached
'In each iteration, we are calculating the IRR based on the cash flows in that particular simulation
'Each IRR is stored in the irrs array

'Start loop counter at iteration 1
i = 1

While i <= iterations
   model.Range("F28", "F" & timespan + 28).Calculate  ' recalculate random stock price

   iternum(i) = i     ' stores the iteration number in the iternum() array

   'While the IRR calculation returns an error, recalculate
   While WorksheetFunction.IsError(model.Range("L25"))
```

```
    model.Range("L25").Calculate
    Wend

  'Stores the annualized IRR for the current iteration in the irrs() array
  irrs(i) = model.Range("L25")

  'Progress bar code
  percentcomplete = WorksheetFunction.Min(i / iterations, 0.97)     ' calculates the percentage completed (leaves 3% for the rest of the functions)
  With ProgressBar
    .LabelProgress.Width = begwidth * percentcomplete
    .Caption = "Processing... " & Format(percentcomplete, "0%")
    End With
    DoEvents
  'End progress bar code

  i = i + 1
  Wend

output.Range("B2", "B" & iterations + 1) = WorksheetFunction.Transpose(iternum)    'outputs the iteration numbers into the output spreadsheet
output.Range("C2", "C" & iterations + 1) = WorksheetFunction.Transpose(irrs)       'outputs the annual IRR's into the output spreadsheet

'Creates a histogram and frequency chart of the rounded/binned IRR's
Call histogram

'Progress bar code (increment after histogram is done)
Dim incr As Single
incr = (1 - percentcomplete) / 3

With ProgressBar
  .LabelProgress.Width = begwidth * (percentcomplete + incr)
  .Caption = "Processing... " & Format(percentcomplete + incr, "0%")
  End With
  DoEvents
'End progress bar code

'Creates a graph of the probability distribution of the rounded/binned IRR's
Call pdf

'Progress bar code (increment after probability graph is done)
   With ProgressBar
    .LabelProgress.Width = begwidth * (percentcomplete + 2 * incr)
    .Caption = "Processing... " & Format(percentcomplete + incr, "0%")
    End With
    DoEvents
  'End progress bar code
```

```
'Creates a graph of the cumulative distrubution of the bins
Call cdf

'Activates the "Random Walk" worksheet
model.Activate

'Progress bar code (completion)
With ProgressBar
    .LabelProgress.Width = begwidth
    .Caption = "Processing... " & Format((i - 1) / iterations, "0%")
    End With
    DoEvents
Unload ProgressBar
'End progress bar code

'Determines how long the macro took to run and outputs the result in a popup window
ttotal = Timer - tstart
MsgBox ("Runtime: " & (ttotal \ 60) & " minutes, " & (ttotal Mod 60) & " seconds")

model.Range("E26") = Date & "; " & Time

'Turns the screen update back on to avoid any errors
Application.ScreenUpdating = True

Application.CutCopyMode = False
End Sub
```

```
'Creates a histogram of the rounded/binned IRR's
Private Function histogram()

'Determines the highest IRR obtained from the simulations
Dim max As Double
max = WorksheetFunction.max(output.Range("C2", "C" & iterations + 1))

'Rounds the maximum IRR down to the nearest bin size (or interval length)
maxbin = Int((max + binsize) / binsize) * binsize

Dim i As Double


'Creates an array of bins and another arrays of their bin labels
Dim bins() As Double
ReDim bins(1 To Int(maxbin / binsize) + 1)
Dim binlabels() As String
ReDim binlabels(1 To Int(maxbin / binsize) + 1)
```

```
i = binsize
While i <= maxbin + binsize
  bins(i / binsize) = i
  binlabels(i / binsize) = i - binsize
  i = i + binsize
  Wend

'Outputs the rounded/binned IRR's into the "Output" worksheet
output.Range("D2", "D" & maxbin / binsize + 1) = WorksheetFunction.Transpose(bins)

'Uses the Analysis Toolpak to create a histogram of the bins
Application.Run "ATPVBAEN.XLAM!Histogram", output.Range("$C$1", "$C$" & iterations + 1),
hist.Range("$B$4"), output.Range("$D$1", "$D$" & maxbin / binsize + 1) _
    , False, False, False, True

'Outputs the binlabels into the "Histogram" worksheet
hist.Range("B5", "B" & maxbin / binsize + 5) = WorksheetFunction.Transpose(binlabels)

hist.Range("B" & maxbin / binsize + 5, "C" & maxbin / binsize + 5).Clear

hist.Activate

'Creates a histogram bar graph
ActiveSheet.Shapes.AddChart.Select
ActiveChart.ChartType = xlColumnClustered
ActiveChart.SetSourceData Source:=Range("$B$5:$C$" & maxbin / binsize + 4)
ActiveChart.SeriesCollection(1).Delete
ActiveChart.SeriesCollection(1).name = "=Histogram!$C$4"
ActiveChart.SeriesCollection(1).XValues = "=Histogram!$B$5:$B$" & maxbin / binsize + 4
ActiveChart.HasTitle = True
ActiveChart.ChartTitle.Text = "Annual IRR's"
ActiveChart.Axes(xlCategory).TickLabels.Orientation = 45
ActiveChart.Axes(xlCategory).AxisBetweenCategories = False
ActiveChart.SetElement (msoElementPrimaryCategoryGridLinesMajor)

With ActiveChart.Parent
  .Width = 800
  .Height = 310
  .Top = 60
  .Left = 200
  End With

hist.Range("B4", "D4").Select

'Formatting cleanup
Selection.ClearFormats
Selection.Font.Bold = True
```

```
output.Columns("D").Hidden = True

hist.Activate

End Function
```

```
'Creates a probability distribution graph of the rounded/binned IRR's and their frequencies
Private Function pdf()

Dim i As Integer

'Copies the table from the "Histogram" worksheet and pastes it into "P(x)" (values and formatting only)
hist.Range("B4", "C" & maxbin / binsize + 4).Copy
px.Range("B4").PasteSpecial Paste:=xlPasteValues
px.Range("B4").PasteSpecial Paste:=xlPasteFormats

'Adds a label to the probability output
px.Range("D4") = "P(x)"
px.Range("D4").Font.Bold = True

'Creates array of probabilities
Dim probs() As Double
ReDim probs(1 To maxbin / binsize)

'Calculates and stores the probability of a bin (probability = frequency/number of iterations)
For i = 5 To maxbin / binsize + 4
    probs(i - 4) = px.Range("C" & i) / iterations
    Next i

'Outputs the probabilities
px.Range("D5", "D" & maxbin / binsize + 4) = WorksheetFunction.Transpose(probs)

px.Activate

'Creates the probability graph
ActiveSheet.Shapes.AddChart.Select
    ActiveChart.ChartType = xlColumnClustered
    ActiveChart.SetSourceData Source:=Range("'P(x)'!$B$5:$D$" & maxbin / binsize + 4)
    ActiveChart.SeriesCollection(1).Delete
    ActiveChart.SeriesCollection(2).Delete
    ActiveChart.SeriesCollection(1).name = "='P(x)'!$D$4"
    ActiveChart.SeriesCollection(1).XValues = "='P(x)'!$B$5:$B$" & maxbin / binsize + 4
    ActiveChart.SeriesCollection(1).Values = "='P(x)'!$D$5:$D$" & maxbin / binsize + 4

    ActiveChart.HasTitle = True
    ActiveChart.ChartTitle.Text = "Probability"
    ActiveChart.Axes(xlCategory).TickLabels.Orientation = 45
    ActiveChart.Axes(xlCategory).AxisBetweenCategories = False
    ActiveChart.SetElement (msoElementPrimaryCategoryGridLinesMajor)

With ActiveChart.Parent
    .Width = 750
    .Height = 340
    .Top = 70
    .Left = 275
    End With
Range("A1").Select

End Function
```

___

```
'Creates a cumulative distribution graph of the rounded/binned IRR's
Private Function cdf()

Dim i As Long           'loop counter
Dim cdfun() As Double   'array of cdf values
ReDim cdfun(1 To maxbin / binsize)

'Copies all the bin labels from the probability sheet to the cdf sheet
px.Range("B4", "D" & maxbin / binsize + 4).Copy
fx.Range("B4").PasteSpecial xlPasteAll
px.Range("B4").Copy
fx.Range("B4", "E4").PasteSpecial xlPasteFormats
fx.Activate

'Labels the cumulative distribution output
fx.Range("E4") = "F(x)"

cdfun(1) = fx.Range("D5")

'Cdf is calculated by adding up the probabilities of all previous bins and the current bin (bins are in
ascending order)
For i = 6 To maxbin / binsize + 4
    cdfun(i - 4) = cdfun(i - 5) + fx.Range("D" & i)
    Next i

'Outputs the value of the cdf for each bin
fx.Range("E5", "E" & maxbin / binsize + 4) = WorksheetFunction.Transpose(cdfun)

'Creates a cumulative distribution graph
ActiveSheet.Shapes.AddChart.Select
    ActiveChart.ChartType = xlLine
    ActiveChart.SetSourceData Source:=Range("'F(x)'!$B$5:$E$" & maxbin / binsize + 4)
    ActiveChart.SeriesCollection(1).Delete
    ActiveChart.SeriesCollection(1).Delete
```

```
   ActiveChart.SeriesCollection(1).Delete
   ActiveChart.SeriesCollection(1).name = "='F(x)'!$E$4"
   ActiveChart.SeriesCollection(1).XValues = "='F(x)'!$B$5:$B$" & maxbin / binsize + 4
   ActiveChart.HasTitle = True
   ActiveChart.ChartTitle.Text = "Cumulative Distribution"
   ActiveChart.Axes(xlCategory).AxisBetweenCategories = False
   ActiveChart.Axes(xlCategory).TickLabels.Orientation = 45
   ActiveChart.SetElement (msoElementPrimaryCategoryGridLinesMajor)

With ActiveChart.Parent
   .Width = 750
   .Height = 340
   .Top = 70
   .Left = 275
   End With

Range("A1").Select

End Function
```

```
'Clears all the cells and deletes all the charts in the "Output", "Histogram", "P(x)", and "F(x)" worksheets
Private Function clearsheets()

Dim ws As Worksheet
Dim cht As ChartObject

For Each ws In Worksheets
   If ws.name = "Output" Or ws.name = "Histogram" Or ws.name = "P(x)" Or ws.name = "F(x)" Then
      ws.Cells.Clear
      For Each cht In ws.ChartObjects
         cht.Delete
         Next
   End If
   Next

End Function
```

```
'Checks if the Analysis Toolpak and its VBA complement are installed; installs them if not
Private Function checkAnalysisToolpak()

If Not Application.AddIns("Analysis ToolPak").Installed Then
   Application.AddIns.Add Filename:=Application.LibraryPath & "\analysis\analys32.xll"
   Application.RegisterXLL "Analys32.xll"
   AddIns("Analysis ToolPak").Installed = True
   End If
```

```
If Not Application.AddIns("Analysis ToolPak - VBA").Installed Then
   Application.AddIns.Add Filename:=Application.LibraryPath & "\analysis\ATPVBAEN.XLAM"
   AddIns("Analysis ToolPak - VBA").Installed = True
   End If

End Function
```

```
' Archives the probability graph and the cumulative distribution graph from the last set of simulations
Sub archive()

Dim newsheet, model As Worksheet
Dim max As Double
Dim name As String
Dim i As Integer

Set newsheet = Worksheets.Add
Set model = Worksheets("Model")

'Determines a name for the archived worksheet that isn't already in use
i = 1
If model.Range("L5") = "Single Premium" Then

   name = model.Range("K5") & " SP " & model.Range("N5") & "_" & model.Range("O5")

   While SheetExists(name)
      name = name & " (" & i & ")"
      i = i + 1
      Wend
   newsheet.name = name
Else
   name = model.Range("K5") & " MD " & model.Range("N5") & "_" & model.Range("O5")

   While SheetExists(name)
      name = name & " (" & i & ")"
      i = i + 1
      Wend
   newsheet.name = name
   End If

If model.Range("K5") = "Great" Then

   newsheet.Tab.Color = 5296274
   newsheet.Tab.TintAndShade = 0
ElseIf model.Range("K5") = "Average" Then

   newsheet.Tab.Color = 65535
   newsheet.Tab.TintAndShade = 0
```

```
ElseIf model.Range("K5") = "Poor" Then

    newsheet.Tab.ThemeColor = xlThemeColorAccent2
    newsheet.Tab.TintAndShade = 0.399975585192419
    End If

'Copies data and graph from the "F(x)" worksheet
Worksheets("F(x)").Activate

ActiveSheet.Range("A1:Z500").Copy
newsheet.Range("A26:Z525").PasteSpecial

ActiveSheet.ChartObjects(1).Activate
ActiveChart.ChartArea.Copy
newsheet.Paste

newsheet.ChartObjects(1).Activate

iterations = Worksheets("Model").Range("I5")
binsize = Worksheets("Model").Range("J5")
max = WorksheetFunction.max(Worksheets("Output").Range("C2", "C" & iterations + 1))
maxbin = Int((max + binsize) / binsize) * binsize

ActiveChart.SeriesCollection(1).XValues = "='" & newsheet.name & "'!$B$30:$B$" &maxbin / binsize +29
ActiveChart.SeriesCollection(1).Values = "='" & newsheet.name & "'!$E$30:$E$" & maxbin / binsize + 29
ActiveChart.SeriesCollection(1).name = "='" & newsheet.name & "'!$E$29"

With ActiveChart.Parent
    .Height = 350
    .Width = 400
    .Left = 250
    .Top = 400
    End With

'Copies data and graph from the "P(x)" worksheet
Worksheets("P(x)").Activate

ActiveSheet.ChartObjects(1).Activate
ActiveChart.ChartArea.Copy
newsheet.Paste

newsheet.ChartObjects(2).Activate

ActiveChart.SeriesCollection(1).XValues = "='" & newsheet.name & "'!$B$30:$B$"&maxbin / binsize + 29
ActiveChart.SeriesCollection(1).Values = "='" & newsheet.name & "'!$D$30:$D$" & maxbin / binsize + 29
ActiveChart.SeriesCollection(1).name = "='" & newsheet.name & "'!$D$29"
```

```
With ActiveChart.Parent
  .Height = 350
  .Width = 400
  .Left = 250
  .Top = 760
  End With

'Formatting cleanup
Worksheets("Model").Range("F1:O25").Copy

newsheet.Range("A1:J26").PasteSpecial Paste:=xlPasteValues
newsheet.Range("A1:J26").PasteSpecial Paste:=xlPasteFormats
newsheet.Range("A1:J26").PasteSpecial Paste:=xlPasteColumnWidths

newsheet.Rows("2:4").Hidden = True
newsheet.Rows("7:18").Hidden = True
newsheet.Rows("22:23").Hidden = True

newsheet.Columns("A:Z").EntireColumn.AutoFit

End Sub
```

```
'Checks if a worksheet with the given name exists (used in Archive() subroutine)
Private Function SheetExists(wsname As String) As Boolean

Dim ws As Worksheet

SheetExists = False

For Each ws In Worksheets

  If ws.name = wsname Then
    SheetExists = True
  End If

  Next

End Function
```

References

Fama, E. F. (1995). Random Walks in Stock Market Prices. *Financial Analysts Journal*, *51*(1), 75-80. Retrieved April 11, 2013, from http://www.jstor.org/stable/4479810

Variable Life Insurance Policy Definition. (n.d.). *Investopedia*. Retrieved April 11, 2013, from http://www.investopedia.com/terms/v/variablelifeinsurancepolicy.asp