ORF 535 Term Project

Ed Stivers and Lu Xin

January 18, 2015

Jennifer Lee Retirement Plan Report

This report highlights the best possible retirement investment plans for Ms. Jennifer Lee. Report created with the following assumptions:

Name	**Jennifer**
Age	35
Sex	F
Salary	175,000
Salary Increase	2%
Employer Matching Contribution	4%
Max Additional DC Contribution	10%
Max Total Contribution	20%
DC Plan Balance	60,000
Non-Retirement Savings Balance	60,000
Minimum Retirement Age	68

Table 1: Jennifer Lee XML Data

Objectives

The primary goal is to find the best possible retirement plan to ensure the highest probably of a fully funded retirement, calculated to be $4.91 million. The secondary object is to minimise the expected amount of time Jennifer needs to wait for retirement if does not meet her funding target by age 68. Analysis of 1000 possible scenarios was conducted to see the distribution of possible paths.

Strategy

Findings from our Financial Planning Tool (FPT) suggests that the best method for Ms. Lee to save and invest is the follow:

Saving strategy:

- Save 14% of annual salary in DC retirement account, earn additional 4% from matched contribution (the retirement account has tax benefits).
- Save 6% of annual salary in taxable savings account (capital gains tax assumed to be constant 10%).
- If a funding ratio of 1.01 is at any time before retirement age, rebalance and immunize entire portfolio with government bonds.
- *Additional consideration: if the stock markets are hit with big downside shocks on multiple occasions, say the 10% worst case scenarios, we allow Jennifer to lower her retirement spending needs by 15%.*

Investment strategy - Adjustable Risk Allocation Policy:

- Start with the best allocation for a standard fixed mix approach. Roughly: stocks 70%, bonds 30%, FTSE overlay 50%, cash 0%.
- Each year, adjust allocation if there is a big shock in the stock market:
  - 10% or more downward shock: increase allocation of stocks by 10% of total wealth, decrease bonds by 10% of total wealth
  - 10% or more upward shock: decrease allocation of stocks by 10% of total wealth, increase bonds by 10% of total wealth
  - Minimum allocation to stocks is 40%, maximum 100%.
  - Always keep FTSE overlay and cash fixed.

The idea of this investment strategy is to take on more risk when the target funding ratio is far away, and reduce risk as we get closer. We mitigate most risk once the the funding ratio is 1.01 or above.

Spending Strategy - upon retirement:

- Jennifer should spend money from her non-retirement savings account before taking from her DC plan account. The tax structure of the two plans makes spending from the non-retirement account beneficial as it minimizes the amount of taxes relative to the gains Jennifer's accounts make during retirement.

Results

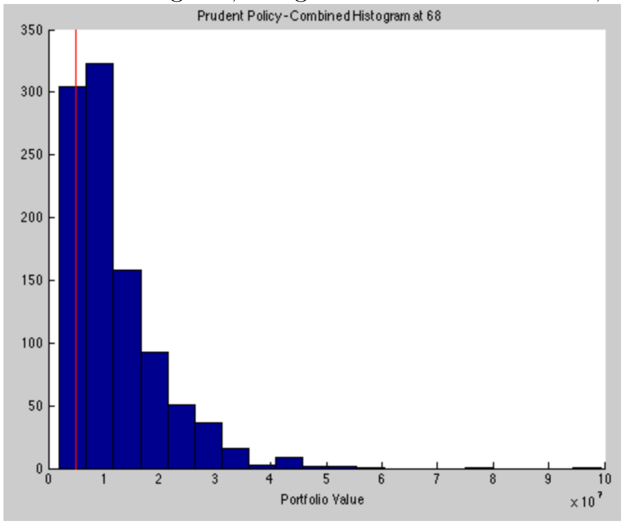
The above-mentioned pension investment strategy yields the best probabilities of achieving our goal:

- Probability of achieving goal at or before 68: **92.60%**
- Expected age if delayed retirement: **70.77 years old**
- Expected age to reach Funding Ratio = 1.01: **55.50 years old**
- Pension plan portfolio max drawdown: 15.54%
- Expected bequeathed amount to family: \$4.16 million
- Probability of running out of funds during retirement: 1.5%

Metrics of Findings

Distribution of Retirement Portfolio Value:

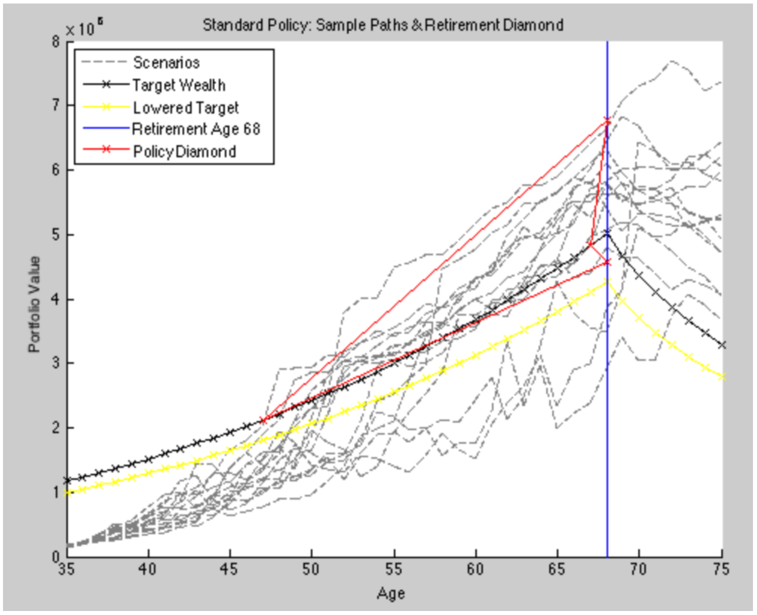
The distribution of values for 1000 scenarios at age 68, using the immunization tactic, is:



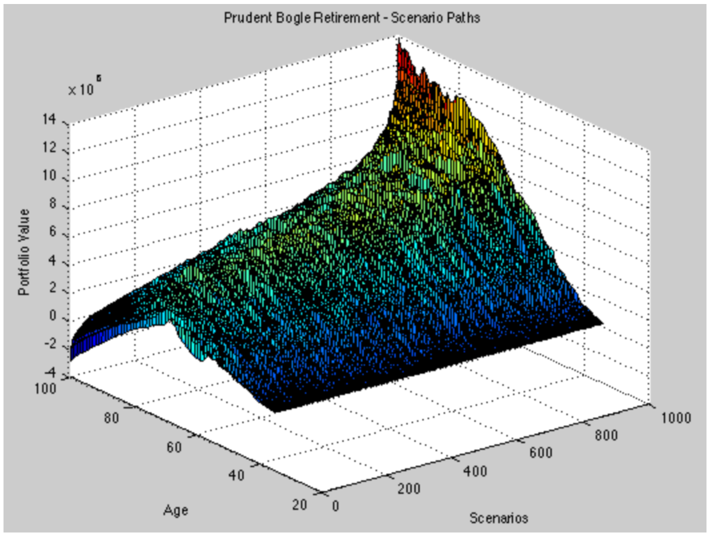
The red line indicates the target level of wealth needed. As we can observe, majority (about 92%) of the distribution of portfolio values are greater than the expected wealth.

Policy Diamond:

Below we plot some sample paths of the pension portfolio of the scenarios in gray. Blue line indicates the retirement age of 68. Black line indicates the target funding ratio of 1.01 (using a 2.5% per year discounting on retirement liabilities, continuous compounding). Yellow line indicates the funding needed for the 15% reduced spending policy if applied. The red 'Diamond' indicates the 80% confidence interval of age reaching a funding ratio of 1.01 along the x-axis, and the 80% confidence range of possible portfolio values at the retirement age of 68.



Wealth paths for all sampled scenarios are visualized below:



The distribution of Jennifer's portfolio value in the DC Pension plan and the Taxable savings account are as follows:

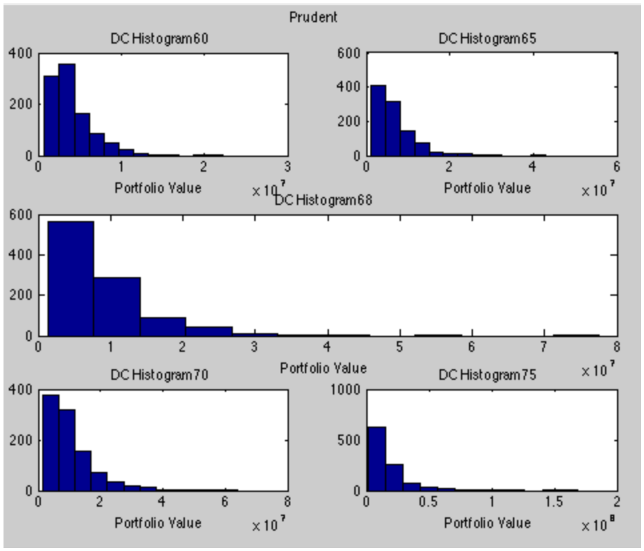


Figure 1: DC Wealth Ages 60,65,68,70,75

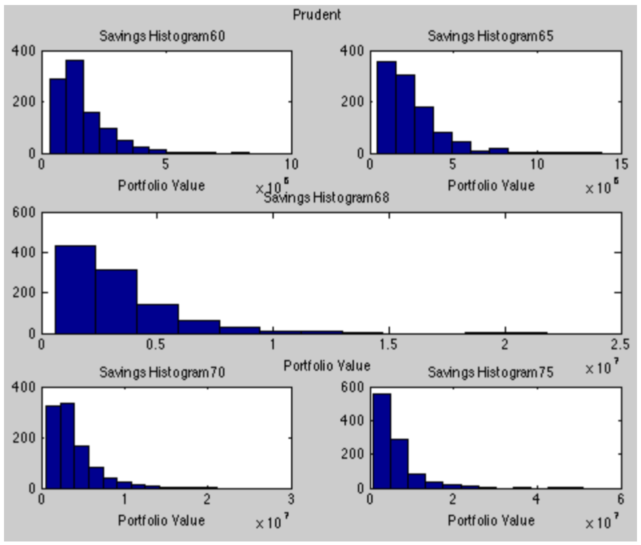
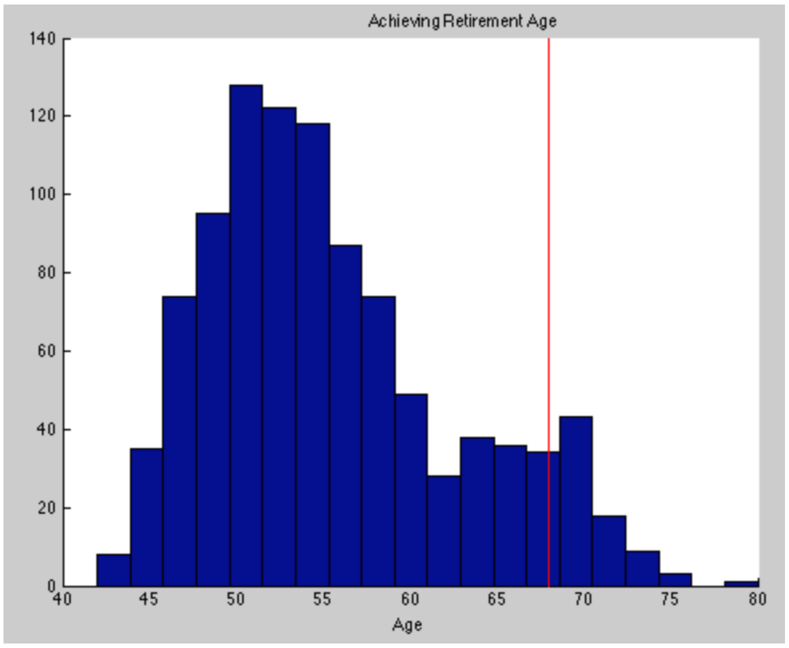


Figure 2: Savings Wealth Ages 60,65,68,70,75

Due to the method in which we allocate our savings, we see the DC pension account is significantly larger than the taxable savings.

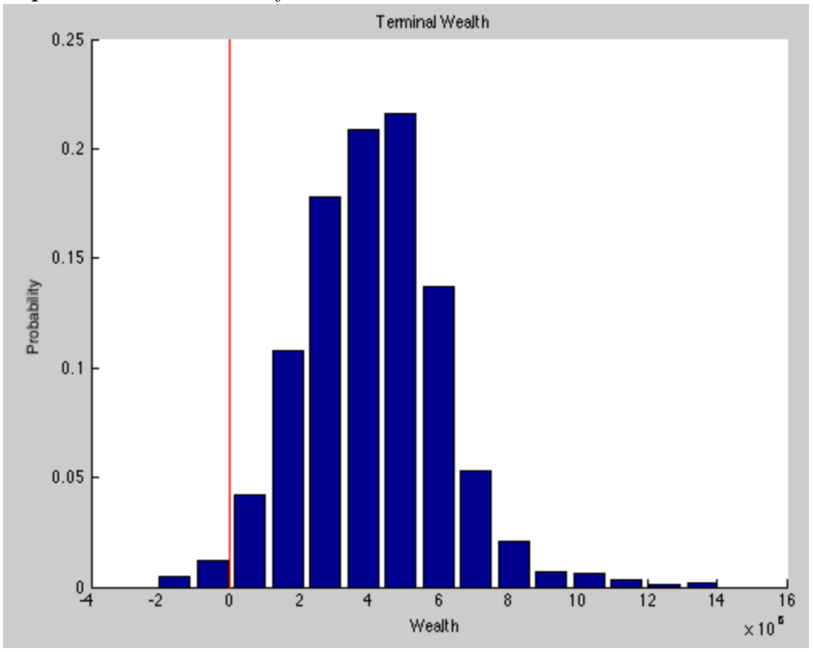
The distribution of the age in which Jennifer acquires her funding ratio target of 1.01 is:



The red line indicates the minimum retirement age for Jennifer, 68.

Longevity

During retirement, given Jennifer's projected spending levels, we simulate the spending and longevity to provide possible amount of assets bequeathed to her family:



Expected bequeathed amount to family: $4.16 million. Probability of running out of funds during retirement: 1.5%.

Possible Improvements

We can improve the 92.60% chance of achieving retirement goal if Jennifer boosts her savings per year to above 20% of her annual salary. For example, some possible considerations are:

Increasing to saving **22%** of annual salary:

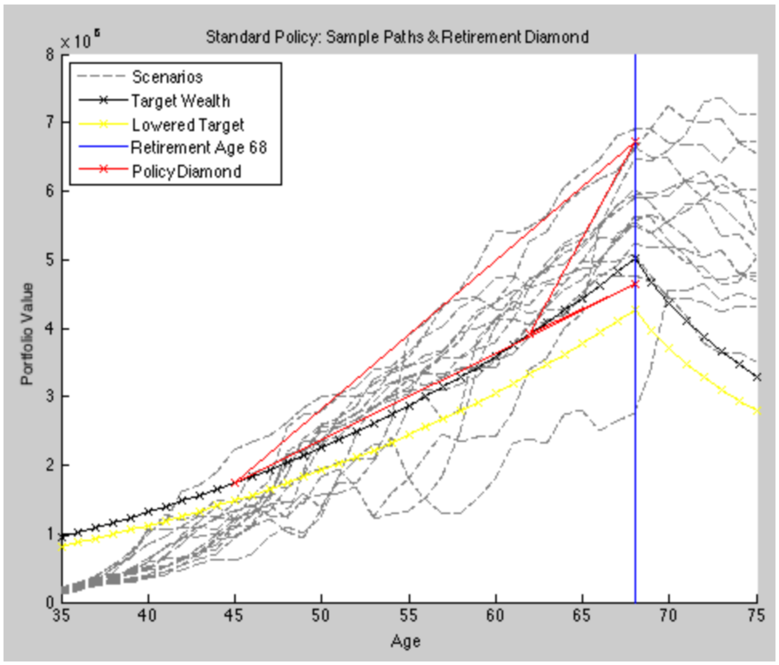
- Probability of achieving goal at or before 68: **93.60%**
- Expected age if delayed retirement: **70.65 years old**
- Expected age to reach Funding Ratio = 1.01: **54.35 years old**
- Pension plan portfolio max drawdown: 13.99%
- Expected bequest amount to family: \$4.46 million
- Probability of running out of funds during retirement: 1.1%

Increasing to saving **24%** of annual salary:

- Probability of achieving goal at or before 68: **96.60%**
- Expected age if delayed retirement: **70.79 years old**
- Expected age to reach Funding Ratio = 1.01: **52.56 years old**
- Pension plan portfolio max drawdown: 11.97%
- Expected bequest amount to family: \$4.94 million
- Probability of running out of funds during retirement: 0.7%

From the above, we observe that increasing savings to 24% each each gives a major boost to the probability of being fully funded by age 68.

Policy Diamond - 24% Adjustable Risk Policy



Conclusion

We suggest Jennifer choose to invest 24% of her annual salary into her retirement plan with the allocation of 14% (+4% matched contribution) in her DC pension plan and 10% in taxable savings account. The investment policy should be to take on more risk and invest more in stocks when her portfolios drifts away from her target funding ratio, and reduce risk by investing more in bonds when she is near her funding ratio. Once the funding ratio is reach, Jennifer should put her entire portfolio into bonds.

This retirement policy dominates other policies such as Fixed-Mix, Bogle Rule, and others variants which are discussed in the appendix. With the above-mentioned plan, she can achieve her target 96.60% of the time at or before the age of 68. The downside is also very limited, with an expected delay of 2.79 years should she not reach her retirement goal by 68.

Appendix - Financial Planning Tool (FPT)

1 FPT - Integrated Planning System Design

1.1 Scenario Generator

We first generate 1000 random starting dates (we increased the samples from 50 to 1000 to obtain more accurate results), then calculate the geometric mean returns for these 1000 random 1-year samples. With the values we obtain, the covariance matrix Q is given by:

$$Q = \begin{bmatrix} 0.0304 & -0.0007 & 0.0025 & 0 \\ -0.0007 & 0.0009 & 0.0009 & 0 \\ 0.0025 & 0.0009 & 0.0146 & 0 \\ 0 & 0 & 0 & 0.0001 \end{bmatrix}$$

The correlation matrix is:

$$\Sigma = \begin{bmatrix} 1 & -0.1245 & 0.1189 & 0 \\ -0.1245 & 1 & 0.2510 & 0 \\ 0.1189 & 0.2510 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The Cholesky Factorisation (using Matlab) to get:

$$\begin{aligned} Q &= F^T F \\ F &= \begin{bmatrix} 0.1745 & -0.0038 & 0.0144 & 0 \\ 0 & 0.0302 & 0.0324 & 0 \\ 0 & 0 & 0.1157 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix} \end{aligned}$$

We generate 1000 economic samples by the following:

$$\tilde{r}_t = \bar{r} + F \cdot \epsilon_t$$

where $\epsilon_t \sim N(0, \mathbf{1})$, $\bar{r} = [0.060 \quad 0.0250 \quad 0.050 \quad 0.005]^T$, for SP500, Bonds, FTSE Commodities Index, and Cash respectively.

1.2 Antithetic-Variate - Variance Reduction

We take advantage of Antithetic-Variate approach to variance reduction to produce more accurate Monte Carlo simulations. The method allows us to double the sample size by:

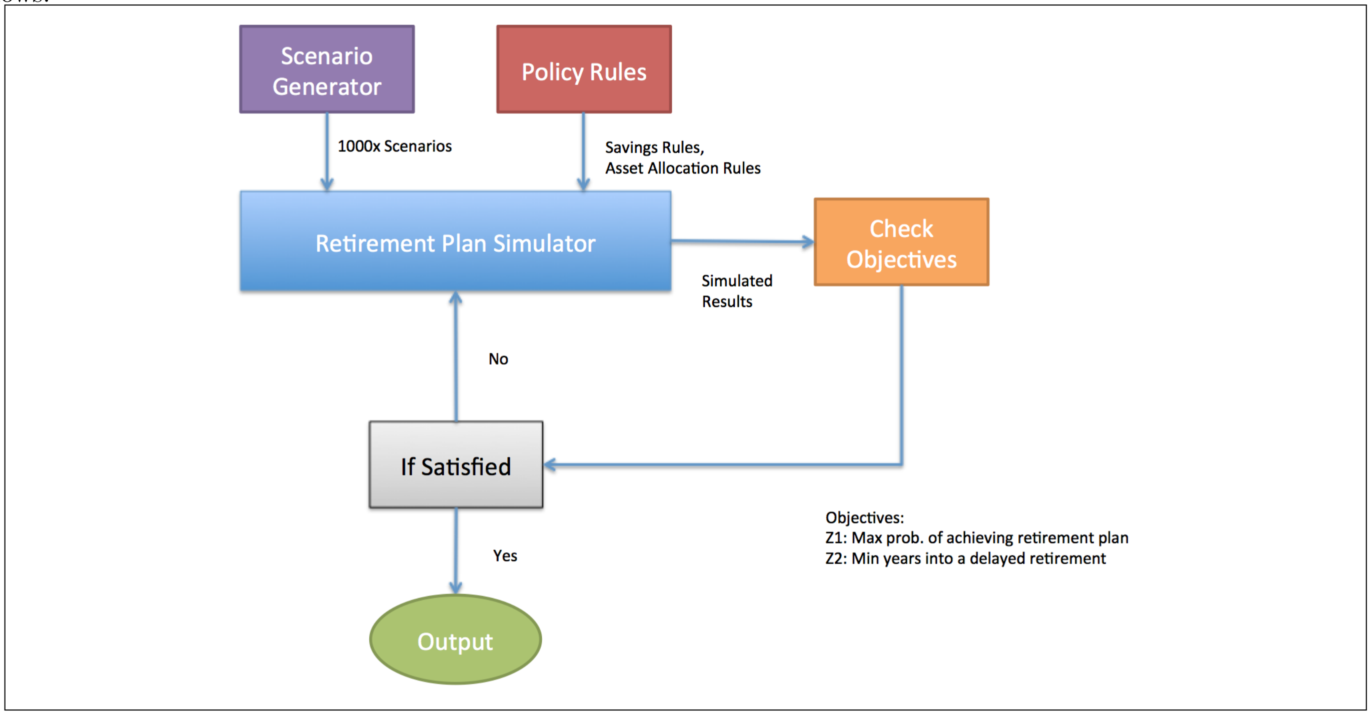
$$\begin{aligned}\tilde{r}_t^+ &= \bar{r} + F \cdot \epsilon_t \\ \tilde{r}_t^- &= \bar{r} - F \cdot \epsilon_t\end{aligned}$$

This does not change the distribution of samples as ϵ_t is just as likely to occur as $-\epsilon_t$. We can hence double the sample size without any extra calculations. The effect on the standard error is:

$$SE_{Antithetic} = \sqrt{\frac{Var(h(\tilde{r}_t^+)) + Var(h(\tilde{r}_t^-)) + 2cov\left(h(\tilde{r}_t^+), h(\tilde{r}_t^-)\right)}{4N}}$$

1.3 Software Design

The FPT tool is designed for flexibility and extensibility. We make use of interfacing techniques borrowed from OOP, such as Java and C++, to construct uniform policy rule functions. As we apply multiple layers of policies, the inputs and results of the functions are all formatted in a consistent way. The flow diagram of how the program works is as follows:


Scenario Generator
Policy Rules
1000x Scenarios
Savings Rules, Asset Allocation Rules
Retirement Plan Simulator
Check Objectives
Simulated Results
No
If Satisfied
Yes
Output
Objectives:
Z1: Max prob. of achieving retirement plan
Z2: Min years into a delayed retirement

1.4 XML Injection

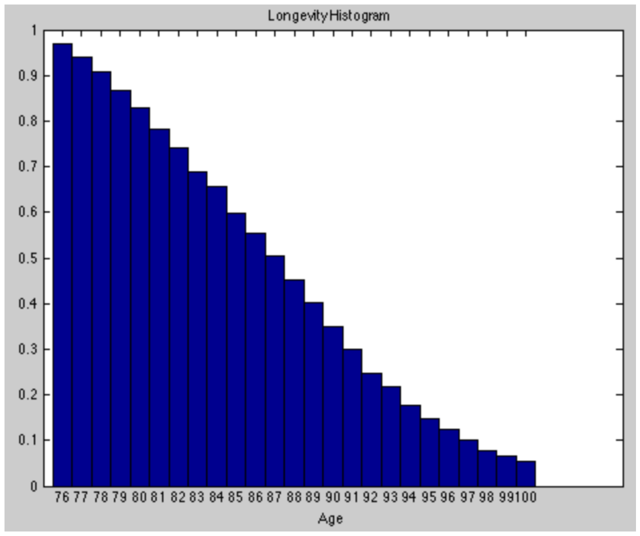
To ensure flexibility and generality, we make use of industry standard XML dependency injection to allow inputs of any person to be entered. The fields required are stated in the example in Section 2. These values are then automatically parsed into our FPT engine and converted MatLab variables. These variables are then used throughout the program in the most general sense so as to keep the code base generic for any user to be evaluated.

1.5 Version Control

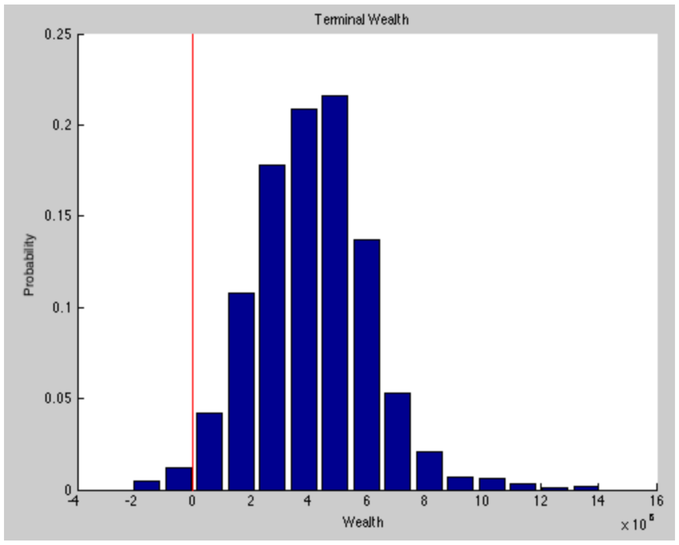
We took advantage of Git as our main source control tool. The enabled concurrent development of the software and warranted the stability of the code base. Versioning is important for large scale development and extensions later on.

2 Longevity Issues

For the analysis we assume that Jennifer will live to at least the age of 75, after which we will use mortality tables in our simulations to determine how she should plan for retirement. The mortality tables change based on the gender of the individual. We choose to implement the 2010 valuation mortality table to perform our calculations. The sex field in the XML file allows us to select the correct mortality table. For Jennifer, our simulated life/death scenarios for after the age of 75 are as follows:



In order to compute Jennifer's age distribution, we used Markov transition matrices calculating sample scenarios of Jennifer's age of death. The entries in the transition matrix tells us the probability that Jennifer will live to the next year given that she is already a certain age. We average 30 simulations per scenario to find Jennifer's longevity in each case while guaranteeing that she lives to at least 75 in every scenario. Once we know Jennifer's longevity, we can calculate the amount of wealth Jennifer bequeaths to her sister. One possible distribution of the wealth Jennifer passes on to her sister is given below.



3 Retirement Plan Objectives

3.1 Maximize Probability of Funded Retirement

Our primary goal is to assist Jennifer in saving for her retirement so that she can retire and maintain her lifestyle without worrying about future expenses. Ideally, with our guidance Jennifer will implement a strategy that fits her risk appetite and also steadily grows her retirement wealth throughout her career. On a very high level, we hope to maximize the probability that Jennifer achieves these goals and minimize her worrying along the way to a fully funded retirement at her desired retirement age.

3.2 Minimize Expected Retirement Delay

Ideally, we hope that Jennifer would be able to save enough to retire at 68 years old in every single scenario. Unfortunately, there are instances where the market performance does not allow this to be the case. Our objective is to minimize the expected retirement delay Jennifer experiences, given that she did not meet her target retirement wealth by the age of 68. We can assist Jennifer in this regard through investment policy rules and through savings policy rules, by implementing an optimal policy corresponding to her level of risk aversion.

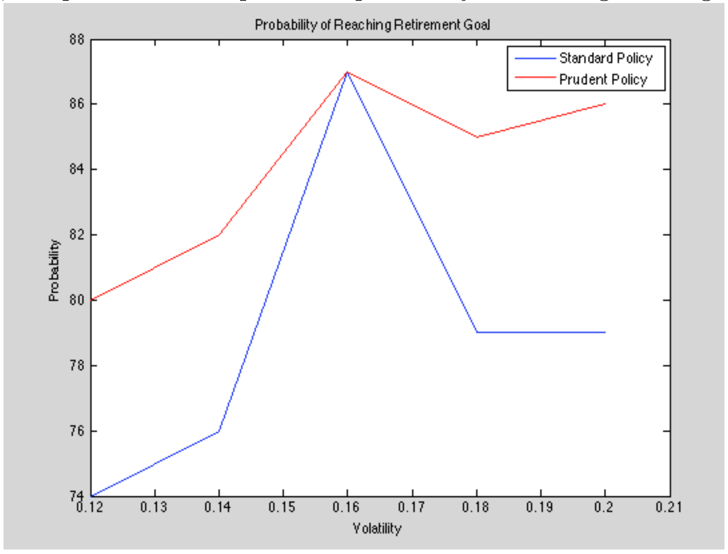
4 Investment Policy Rules

Policy	Average Return	Average Vol	Average Sharpe Ratio	Average Max DD
Fixed-Mix	6.48%	14.95%	0.4334	38.75%
Variable Fixed Mix	6.40%	14.14%	0.4529	33.60%
Bogle Rule	6.34%	14.52%	0.4364	38.67%
Minimise Max Drawdown	5.57%	9.37%	0.5944	21.63%
Adjustable Risk Rule	7.87%	17.25%	0.4560	38.04%

4.1 Fixed-Mix Rule

A possible simple policy rule is to implement a strategy with a fixed amount of total capital in each of the three possible assets (cash, stock, and bonds) and with a fixed amount in the overlay commodities index. This rule can be good for long term investing, but does not take into account Jennifer's age or how close she is to attaining her goals. In this policy, we run our simulations and perform a maximization across them to find the highest average return applying the same weights to every year in each simulation. This method is particularly challenging because it is blind to the retirement account's growth and evolution through time. Ultimately, the strategy is too simple for Jennifer and will produce too much volatilty for her expected return.

Optimization: When performing the Fixed-Mix strategy we constrain the volatility. In optimizing the strategy, we maximized the return across a variety of volatility constraints, ranging from 12% to 20%. We find the optimal volatility restriction to be 16%. Additionally, the plot below compares the probabilty of reaching the target wealth across volatilities.



4.2 Variable Fixed-Mix Rule

This policy is a slightly more complex version of the Fixed-Mix rule discussed above. The rule performs the same Fixed-Mix analysis to calculate the weights allocated to the various assets, but does so over smaller time intervals rather than the entire data set. For example, it optimizes over the next n years of simulations and then provides an optimal allocation for those n years. This strategy adds some variablility over the Fixed-Mix rule, but through our simulations produces generally the same weights for each time subset, with small deviations in the expected return. Our implementation of the rule changes the allocation every 10 years.

4.3 Bogle Variant Rule

The Bogle variant method would have Jennifer invest 100 minus her age in stocks and the remainder of her savings in bonds and cash (as well as the overlay). This rule is a simplistic way to have Jennifer invest more conservatively as she approaches retirement. This rule is superior to the two prior rules in that it incorporates Jennifer's current age and allows her to be more conservative as she approaches retirement. However, if Jennifer is not projected to reach her goal at a given date, placing more and more money into bonds will decrease her probability of reaching her retirement goal. We feel that this rule is overly simplistic as it does not allow Jennifer the upside potential of investing in stocks if she is falling short of her target retirement wealth.

4.4 Capped Max Drawdown Rule

The drawdown rule calculates the max drawdown over each simulated path and finds the average across these paths. The policy rule then maximizes weights in assets in the Fixed-Mix method while ensuring that the average maximum drawdown does not exceed our investor specified maximum drawdown threshold. Through the life of this investment policy, the investor enjoys the peace of mind to know that her retirement accound does not suffer large losses, though with these conservative investment weights the investor may not reach her goals. We recommend this policy for investors planning to save a large percentage of total salary each year, in addition to those who are particularly risk averse.

4.5 Adjustable Risk Rule

The adjustable risk rule increases weight in the highest volatility strategy, which in Jennifer's case is the stock allocation, when stocks perform below a specified threshold in the prior year. This strategy increases risk when the retirement account performs poorly hoping to capture higher returns and help the terminal wealth projection recover from losses. Additionally when the accounts see large gains, the asset allocation changes so that there is less weight in the risky stock index, thereby increasing Jennifer's probability of maintaining large gains for her retirement. We find that for investors with at least a moderate risk appetite, this policy rule can be quite successful, producing the highest probability of being funded for retirement of any rule presented here.

Optimization: We optimize the size of the drawdown and the size of the allocation increase to maximize Jennifer's probability of reaching her target. We vary the allocation increase from 1% to 15% and the drawdown from 5% to 30% testing each combination and find that using the 10% drawdown as the threshold and increasing the allocation by 10% produces the maximum return.

5 Saving Policy Rules

5.1 Account Allocation Rule

Each year, Jennifer must decide her allocation between her DC plan and her non-retirement savings account. This decision is largely tax based as Jennifer weighs the implications of paying a tax immediately versus paying taxes when she removes her money for retirement. With Jennifer's tax structure, she pays 30 % on the retirement account when she removes the money at retirement. For her non-retirement account, she pays a 30% tax immediately and then 10% each year on returns (with a tax advantage on losses). As Jennifer's wealth grows in this scenario, she will repeatedly pay a capital gains tax on her non retirement account and will save all of her returns in the retirement account. Since the tax rate of both accounts is equal, Jennifer should place the maximum amount possible in her retirement account to minimize her tax exposure.

5.2 Standard Savings Rule

In the standard savings rule, Jennifer wants to avoid all volatility in the total wealth of her accounts once she reaches her retirement age. At age 68, Jennifer keeps her wealth entirely in cash to avoid the volatility of the stock index and government bonds. Once age 68 passes, Jennifer's accounts no longer gain positive returns and decrease by the amount she spends each year. This rule is suboptimal as it does not take into account Jennifer's funding ratio throughout time or increasing her account balance after retirement.

5.3 Prudent Funding Ratio Rule

This method takes into account Jennifer's funding ratio and uses it as a metric to assess her asset allocation. As soon as Jennifer's funding ratio exceeds one (beginning with a 1.01 ratio), we assume that she will want to invest more conservatively to be more certain that she will reach her goals. We optimize the optimal funding ratio before immunizing the portfolio and find that though 1.00 produces the highest probability of Jennifer achieving her goals, the worst case scenarios with the 1.01 funding rule are superior to those with a funding ratio of 1. Taking Jennifer's worst case scenario into account we chose to use a funding ratio of 1.01. Once Jennifer's funding ratio reaches 1.01, her asset allocation is redistributed to be entirely in government bonds, which earn lower returns but with a lower variance. Since we assume Jennifer will work until she is at least 68 years old, immunizing her retirement wealth can be extremely advantageous and protects her against large losses that could force her to work beyond her desired retirement age. This strategy significantly improves Jennifer's probability of retiring at 68 years of age. Figures 3 and 4 below juxtapose the standard savings rule with the prudent funding ratio policy. The retirement diamond of the prudent policy has been shifted left meaning the 80 percent confidence interval of Jennifer's age of reaching her funding goal occurs earlier in her career. Additionally, we see the target wealth before 68 is a function of time for the prudent policy, since the immunization places the total wealth into government bonds rather than cash. Therefore, Jennifer can realize her target wealth by meeting a target wealth threshold where immunizing into government bonds will allow her to reach her target with more certainty.

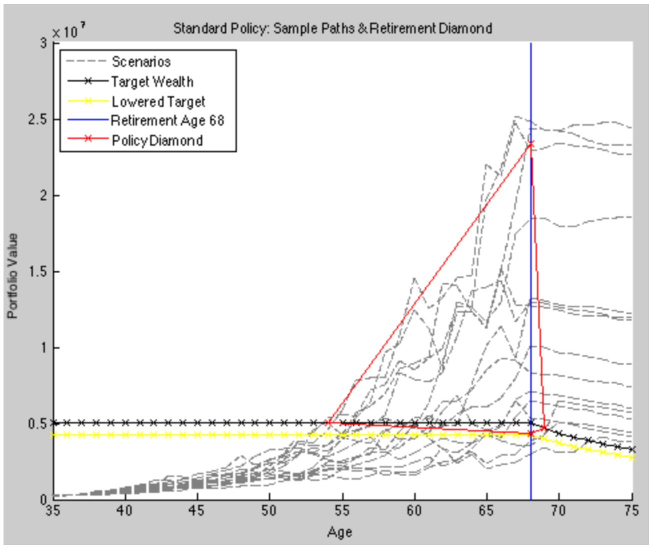


Figure 3: Standard + Adjustable Risk Policies

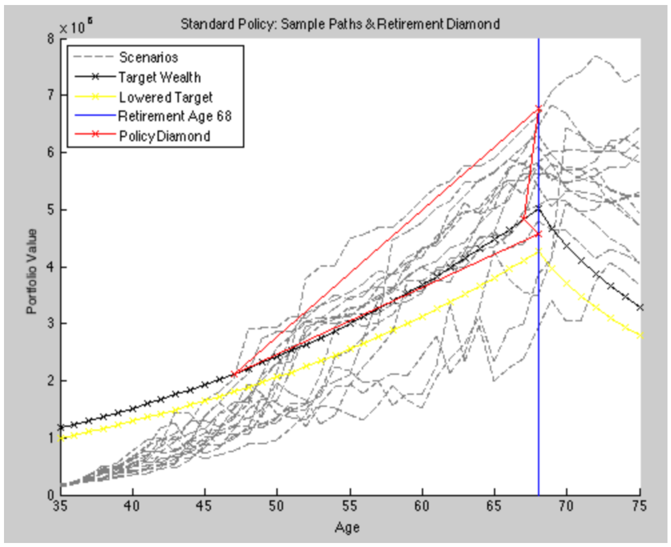


Figure 4: Prudent + Adjustable Risk Policies

5.4 The Contingent Spending Rule

This particular allows Jennifer to readjust her spending appetite after recessions. We simulate across scenarios and find the worst 10% of stock market performances. In these cases Jennifer decides that her target wealth goal may be unrealistic and that she may need to adopt a slightly lower target in order to adjust for the market downturn. For these market crash scenarios, Jennifer readjusts her expectations by deducting 15% from her overall spending target for retirement. Through this method, Jennifer maintains realistic expectations, similar to the "worst case rule" implemented by Princeton University. On average, implementing this rule increases Jennifer's probability of reaching her target by roughly 2%.

5.5 Retirement Spending Decision

Spending from the taxable savings account prior to spending from the DC pension plan allows Jennifer to gain some marginal benefits due to the tax structure. Our simulations showed a minor increase in expected wealth and probability of not depleting her funds before she passes. However, this benefit is not significant, it is even less pronounced in the case of the standard savings rule where Jennifer puts all her money into cash.

6 Comparison of Policies

The five different investment policies produce different returns and variances and present Jennifer with different benefits and drawbacks for her retirement investment. For each investment policy, we always implement the contingent spending savings rule, but have the chioce between the standard savings rule and the prudent funding ratio rule. With five investment policy choices and two savings policy choices we have a total of ten potential choices for Jennifer's optimal policy. We assume that the base policy is the Fixed-Mix saving rule. The Fixed-Mix policy rule selects a fixed allocation that maximized returns based on analyzing simulated scenarios. For example, suppose

Jennifer selects the Fixed-Mix, prudent policy for her savings accounts. The distribution of Jennifer's wealth and savings as she approaches and hits retirement are given by Figures 5 and 6 below.

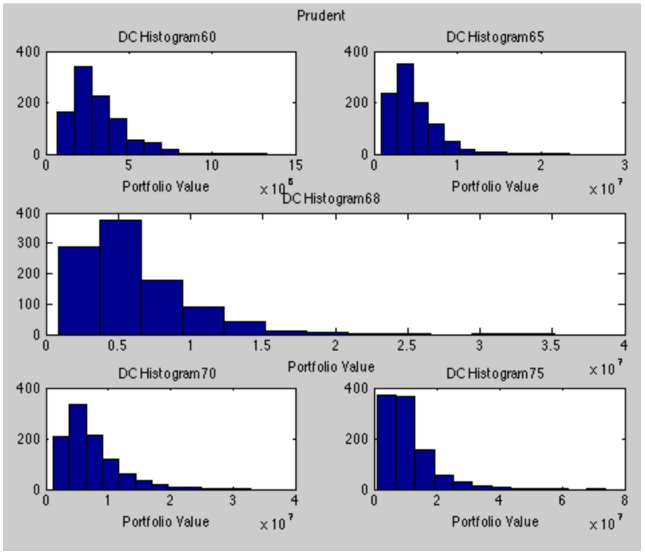


Figure 5: DC Wealth Ages 60,65,68,70,75

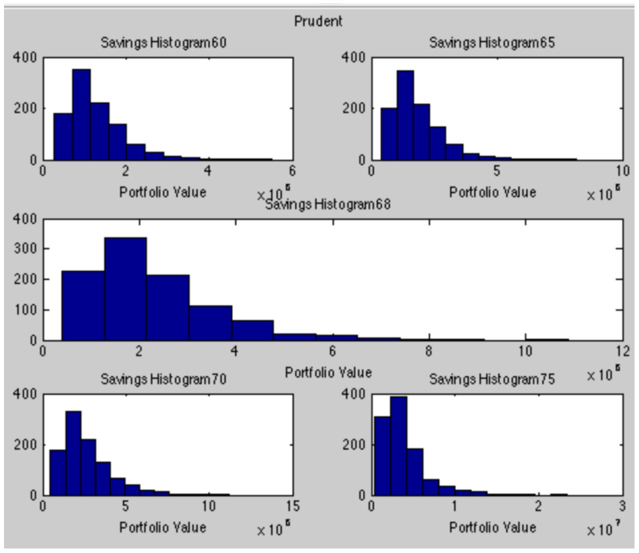


Figure 6: Savings Wealth Ages 60,65,68,70,75

As per our first objective, the primary goal is to maximize the probability that Jennifer has reached her target retirement wealth by her projected retirement age of 68. Figure 7 shows the distribution of Jennifer's wealth once she reaches 68 years of age compared to her target wealth. Rather than looking at Jennifer's distribution, we can instead look at the distribution of Jennifer's age (shown in figure8)where she has achieved a funding ratio of one.

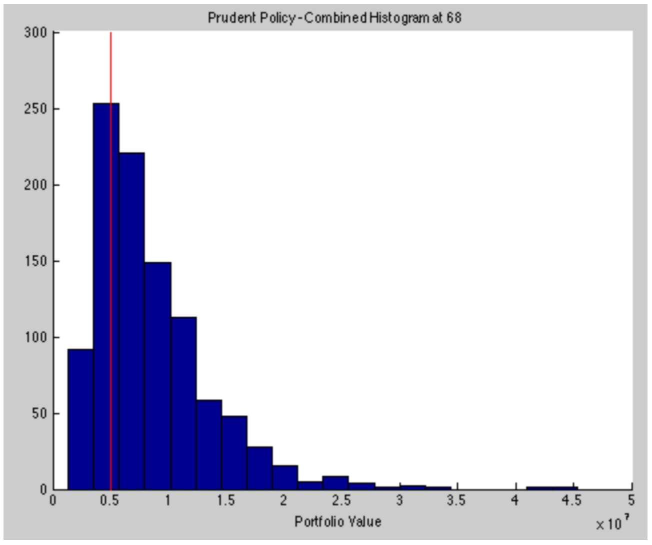


Figure 7: Wealth Distribution at 68

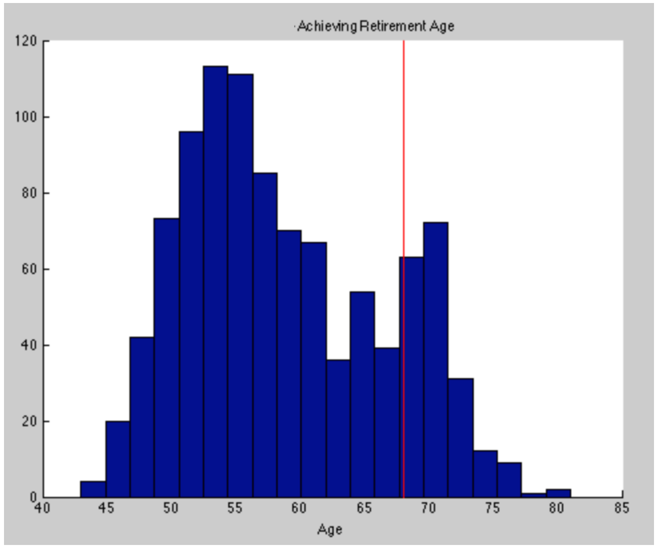


Figure 8: Age Distribution at Funding Ratio of 1

A retirement diamond describes an 80% confidence interval of both the expected age of retirement and the expected wealth at Jennifer's projected retirement age of 68. The boundaries of these two confidence intervals create the diamond shape and enclose a space where we expect Jennifer's retirement path to lie near retirement in a majority of the simulations. Figure 9 displays this retirement diamond for Jennifer's Fixed-Mix strategy.

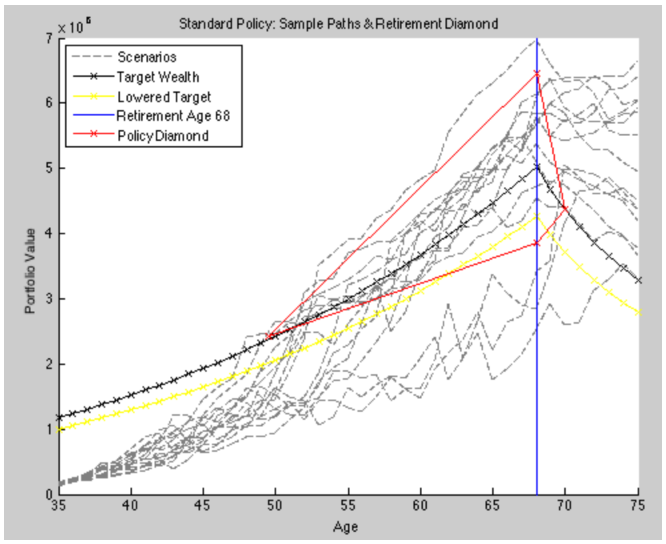


Figure 9: Retirement Diamond for Fixed-Mix Policy

Finally, figure 11 shows a plot of every simulated path through time to see how wealth develops under the specified policy.

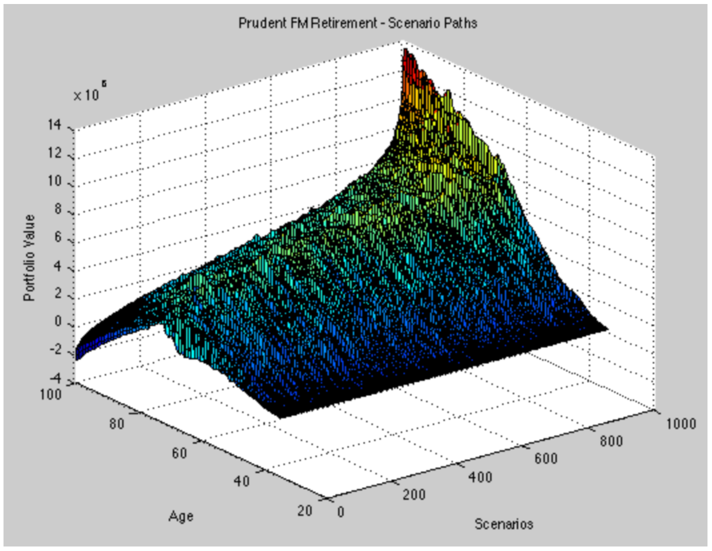


Figure 10: Fixed-Mix Scenarios

This sample calculation shows the major descriptive figures associated with each policy. We have computed these in addition to explanatory metrics for all ten potential policies (which have already been optimized with a given volatiltiy constraint) and compare the metrics of the policies in figure 11.

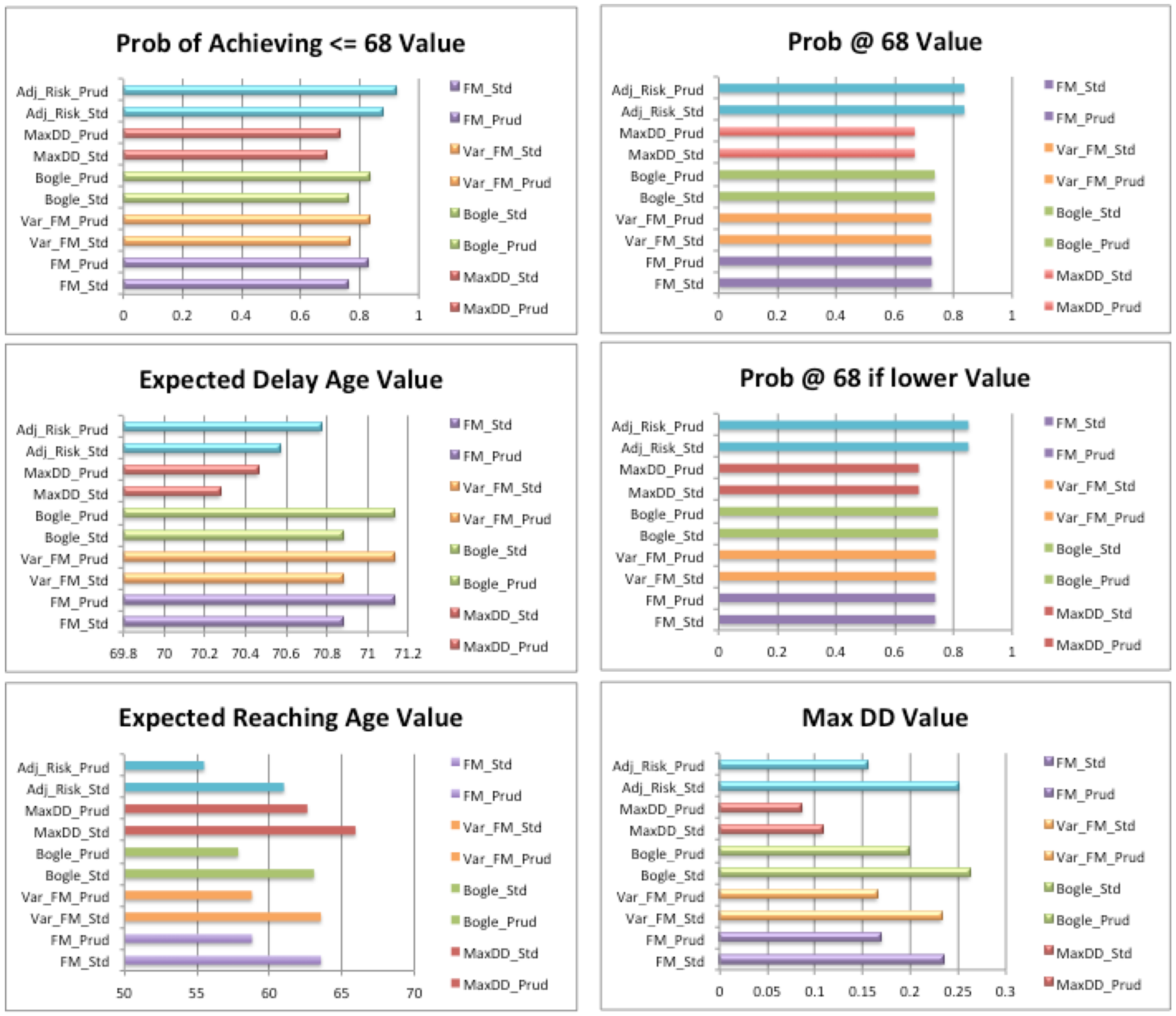


Figure 11: Statistic Comparison

From figure 11 we see immediately that there is a significant difference between the standard savings policy and the prudent savings rule. The prudent rule outperforms the standard rule across investment policies in terms of drawdowns, the probability of achieving the wealth target by age 68, and expected age when reaching the target wealth value. In terms of investment policies, the Fixed-Mix policy outlined above produces inferior results in every measured statistic but serves as a base policy upon which other policies build. The policy is inflexible and does not take into account Jennifer's age or drawdowns of the stock index in the optimal allocation. The variable Fixed-Mix policy is a natural extension of the Fixed-Mix policy. This policy is slightly more variable in that it allows allocations to change at a prespecified interval, but it still does not account for the drawdowns or Jennifer's age similar to the Fixed-Mix rule. The variable policy outperforms the Fixed-Mix rule as we would expect, but not by much as it adds minimal variability and does not improve on the major constraints.

We can account for Jennifer's age by instituting the Bogle policy rule. This rule varies the allocation to the stock index relative to Jennifer's age but does not take into account drawdowns in the stock index or offer any flexibility in allocation. The Bogle rule performs very similarly to the Fixed-Mix and variable Fixed-Mix rules but is still inferior to other strategies. The first rule to incorporate the negative performace of the stock index in the maximum drawdown policy rule. The drawdown rule finds the optimal Fixed-Mix for Jennifer but constrains the maximum drawdown of the portfolio to a specified target level. Though ideal for a risk averse investor because of the reduced maximum drawdown, this policy rule is too conservative and does not take enough risk to achieve Jennifer's goals. The final strategy is the adjustable risk rule. By increasing the allocation to the stock index when the index performed very poorly in the prior year, the policy attempts to recover the losses suffered in the prior period. The policy rule is the most flexible and takes into account drawdowns, though it is oblivious to Jennifer's age. The combination of flexibility and responding to large drawdowns provided the best performance by far. The adjustable risk rule is described in detail above as we suggest the policy as the optimal choice for Jennifer to realize her retirement goals.

Key Policy Diamonds Comparison:

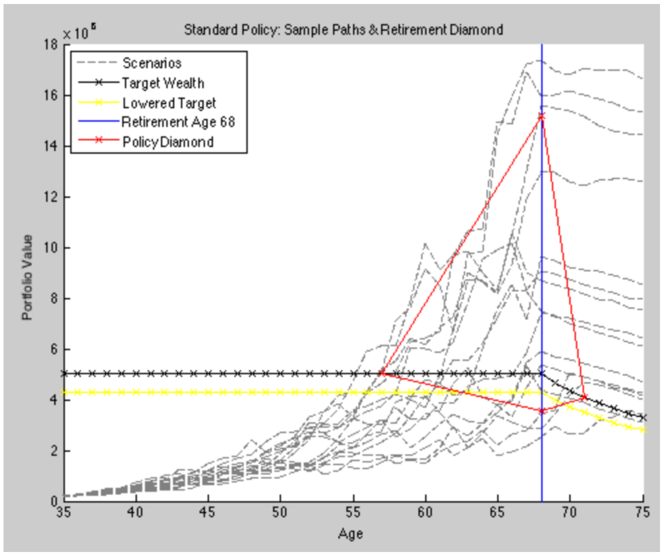


Figure 12: Fixed-Mix Standard

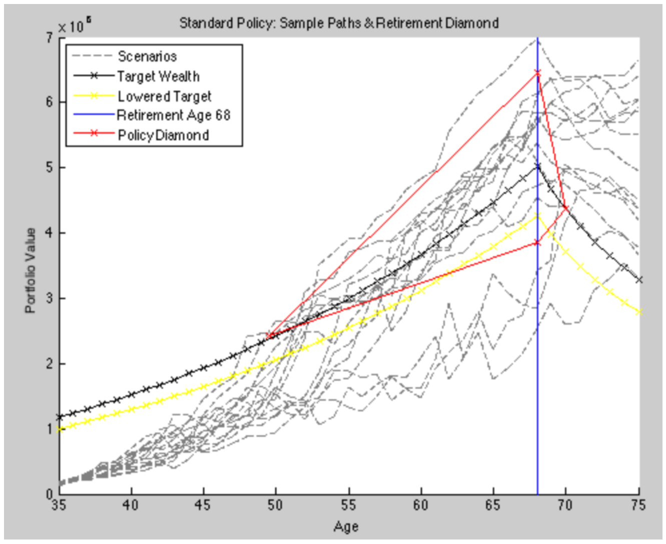


Figure 13: Fixed-Mix Prudent

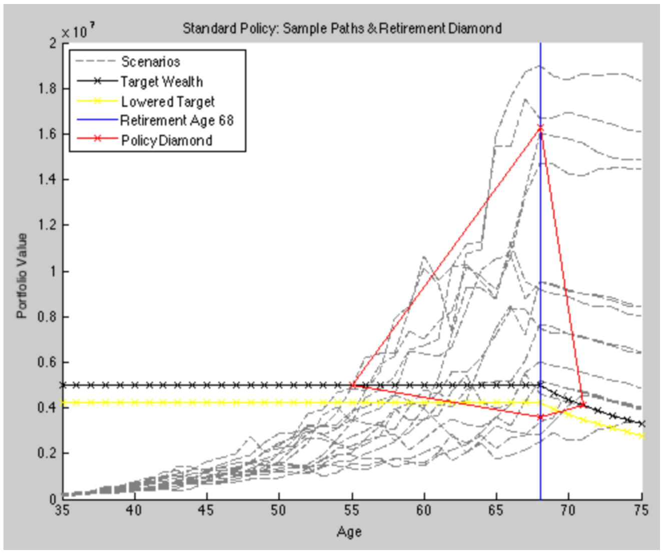


Figure 14: Bogle Standard

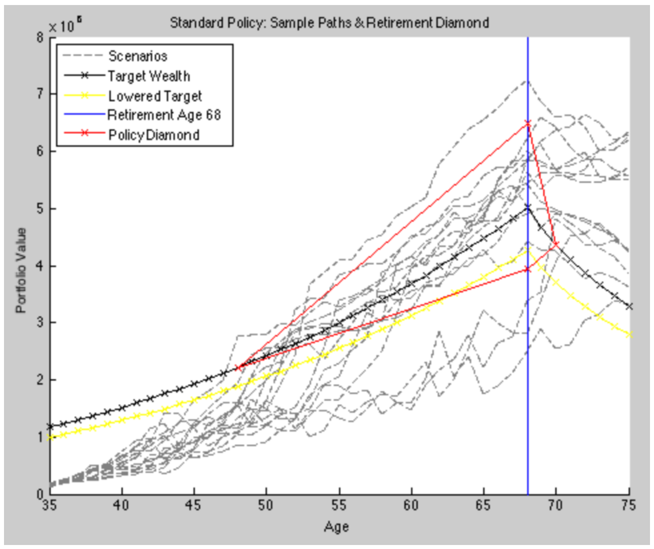


Figure 15: Bogle Prudent

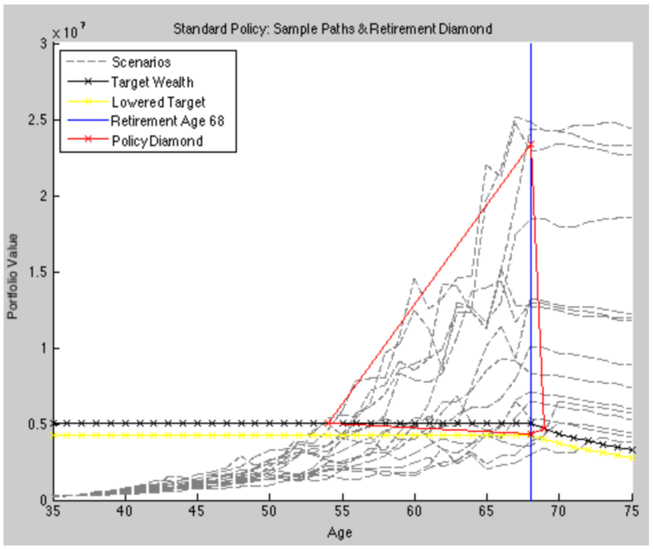


Figure 16: Adjustable-Risk Standard (20% savings)

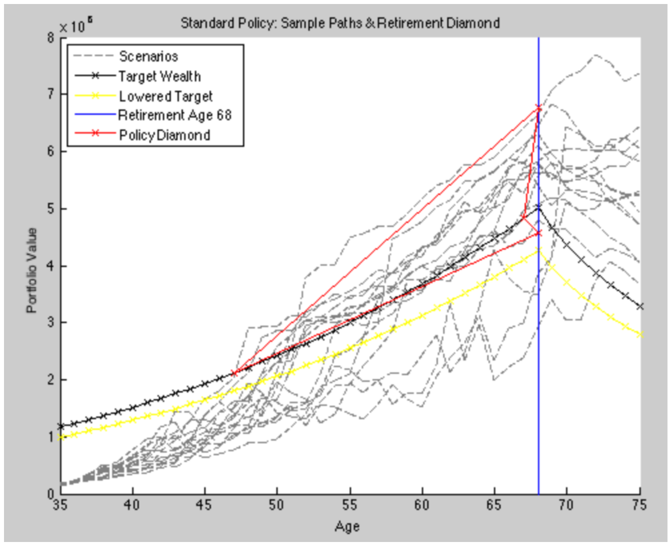


Figure 17: Adjustable-Risk Prudent (20% savings

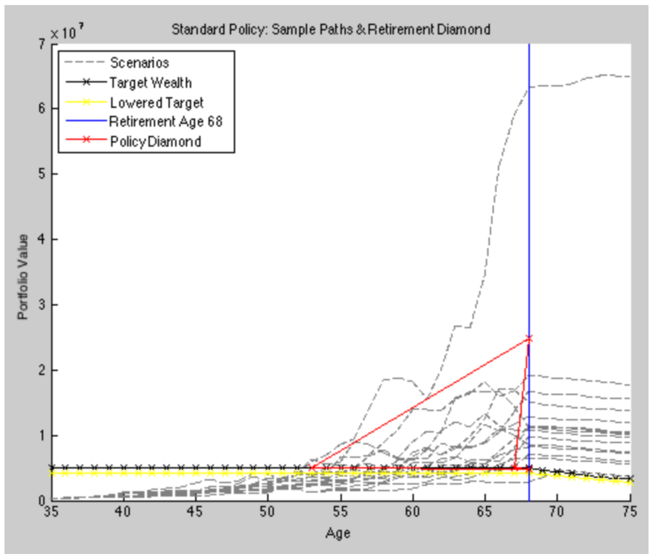


Figure 18: Adjustable-Risk Standard (24% savings)

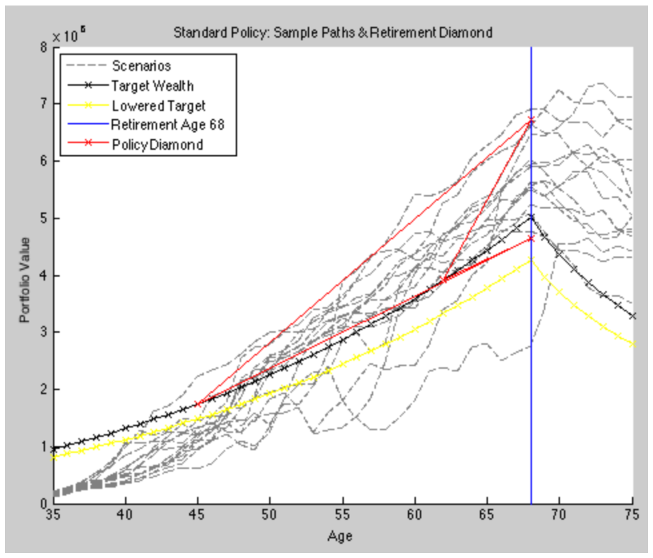


Figure 19: Adjustable-Risk Prudent (24% savings

7 Extensions

Some further extensions to the project include:

- Explore alternative policy rules on investment strategy such as Momentum, Risk Parity, and others.
- Allow for borrowing when optimising the retirement plan.
- Include more assets to choose from, allowing for small allocations to extremely risk assets such as financial derivatives. This could work well with the prudent policy rule for early funding ratio target hits.
- Examine possibilities of investing in cash flow matching method.
- Examine portfolio optimisation on any surplus wealth if funding ratio is large. This could improve the distribution of assets left to family members.
- Employ other variance reduction methods on simulations such as control variate.

8 Attached Code - Ed Stivers, Lu Xin

8.1 Saving Policies

Standard Pension Plan Simulator:

```
function [ standard_port ] = PensionPlanSimulator( NAME, RETIRE_AGE, ...
    MIN_DEATH_AGE, RETURNS, CASH_RETS, lower_target_vector)

SOCIAL_SEC     = 30000;
PERC_TARGET    = 0.04;

PLOT_NUM       = 15;
SCENARIOS      = size(RETURNS,2);
% Read in all data for named person
configs = getConfiguration(NAME);
[Age, Sex, Salary, SalaryIncr, MatchingContr,...
    AdditionContr, MaxSalarySaved, DCBal, SavingsBal] = GetValues(configs);

TOTAL_YEARS       = 100-Age;
years             = 0:TOTAL_YEARS;

% Calculate retirement goal
annual_salary = Salary*((1+SalaryIncr).^years);
[target_wealth, Moving_Target, yearly_spending ] = TargetWealth(annual_salary, Age, ...
    RETIRE_AGE, SOCIAL_SEC, PERC_TARGET, MIN_DEATH_AGE, MaxSalarySaved);
target_wealth
lower_target_wealth = target_wealth * 0.85;

RETIREMENT_INDEX = RETIRE_AGE - Age + 1;
Moving_Target(1:RETIREMENT_INDEX) = target_wealth;
Lower_Moving_Target = Moving_Target * 0.85;

% DO OPTIMISATION ON SAVINGS AND DC ALLOCATION
function [c, ceq] = DC_savings_confun(s_w)
    c = [s_w - (MaxSalarySaved - MatchingContr);
        (MaxSalarySaved - AdditionContr - MatchingContr)-s_w];
    ceq = [];
end

function [prob_achieve] = CalcProb(w)
    pw           = MaxSalarySaved - w + MatchingContr;
    DC_port      = PensionPort(DCBal, annual_salary, pw, RETURNS, Age);
    Savings_port = NonPensionPort(SavingsBal, annual_salary, w, RETURNS, Age);
    PENSION_PLAN = DC_port + Savings_port;
    prob_achieve = sum(PENSION_PLAN(RETIREMENT_INDEX,:)>target_wealth)/SCENARIOS;
end

op = optimset('MaxFunEvals',100000, 'MaxIter', 100000, ...
        'TolX', 0.001,'Display', 'off');

[opt_sw, ~]=fmincon(@(w) (-CalcProb(w)),...
    0.06,[],[],[],[],[],[],@(w) (DC_savings_confun(w)),op);

opt_pw       = (MaxSalarySaved + MatchingContr) - opt_sw;
DC_port      = PensionPort(DCBal, annual_salary, opt_pw, RETURNS, Age);
Savings_port = NonPensionPort(SavingsBal, annual_salary, opt_sw, RETURNS, Age);
PENSION_PLAN = DC_port + Savings_port;

% Adjust Spendings after

% HISTOGRAMS
PlotPortHist(DC_port, Age, 'DC ', 'Standard');
PlotPortHist(Savings_port, Age, 'Savings ', 'Standard');

% SHE CAN LOWER BY 15% IF GOTTEN BIG HITS IN THE STOCK MARKET
adjusted_target_wealth = lower_target_vector * lower_target_wealth + ...
    (1-lower_target_vector) * target_wealth;

prob_achieve = sum(PENSION_PLAN(RETIREMENT_INDEX,:)>target_wealth)/SCENARIOS
prob_achieve_lower = ...
    sum(PENSION_PLAN(RETIREMENT_INDEX,:)>adjusted_target_wealth)/SCENARIOS
% SEE WHEN TARGET
TARGET_REACHED = PENSION_PLAN'>repmat(Moving_Target,SCENARIOS,1);

ages_reached   = zeros(1,SCENARIOS);

for i = 1:SCENARIOS
    reach_index     = find(TARGET_REACHED(i,:) > 0.5,1);
    ages_reached(i) = reach_index+ Age - 1;
    TARGET_REACHED(i,reach_index:end) = 1;
end

EXPECTED_DELAYED_AGE         = mean(ages_reached(ages_reached>RETIRE_AGE))
MEAN_REACH_TARGET_AGE        = mean(ages_reached)
PROB_REACH_TARGET_BEFORE_RET = sum(ages_reached<=RETIRE_AGE)/SCENARIOS

figure;
hold on;
hist(PENSION_PLAN(RETIREMENT_INDEX,:),20);
plot([target_wealth target_wealth],get(gca,'ylim'),'r');
title('Standard Policy - Combined Histogram at 68');
xlabel('Portfolio Value');
hold off;

figure;
hold on;
```

```
hist(ages_reached,20);
plot([RETIRE_AGE RETIRE_AGE],get(gca,'ylim'),'r');
title('Standard Policy - Achieving Retirement Age');
xlabel('Age');
hold off;

% PUT IN CASH
STD_PENSION_PLAN = StandardPolicy(PENSION_PLAN, TARGET_REACHED',...
    CASH_RETS, opt_pw, opt_sw, annual_salary, RETIREMENT_INDEX, yearly_spending);

% CALCULATE 90% CONF INTERVAL
QUANTILE    = 0.1;
Age_lower_Q = quantile(ages_reached,QUANTILE);
Age_upper_Q = quantile(ages_reached,1-QUANTILE);
Age_l_index = ceil(Age_lower_Q-Age+1);
Age_u_index = floor(Age_upper_Q-Age+1);

Plan_lower_Q = quantile(STD_PENSION_PLAN(RETIREMENT_INDEX,:),QUANTILE);
Plan_upper_Q = quantile(STD_PENSION_PLAN(RETIREMENT_INDEX,:),1-QUANTILE);

figure
hold on;
hline0 = plot(repmat(Age:MIN_DEATH_AGE,1,1)', ...
    STD_PENSION_PLAN(1:(MIN_DEATH_AGE-Age+1),1),'k--');
set(hline0, 'color', [0.5 0.5 0.5])
hl = plot(repmat(Age:MIN_DEATH_AGE,PLOT_NUM-1,1)', ...
    STD_PENSION_PLAN(1:(MIN_DEATH_AGE-Age+1),2:PLOT_NUM),'k--');
set(hl, 'color', [0.5 0.5 0.5])
hline1 = plot(Age:MIN_DEATH_AGE,Moving_Target(1:length(Age:MIN_DEATH_AGE)),'k-x');
hline2 = plot(Age:MIN_DEATH_AGE,Lower_Moving_Target(1:length(Age:MIN_DEATH_AGE)),'y-x');
hline3 = plot([RETIRE_AGE RETIRE_AGE],get(gca,'ylim'),'b-');
hline4 = plot([Age_lower_Q RETIRE_AGE],[Moving_Target(Age_l_index) Plan_lower_Q],'r-x');
plot([Age_lower_Q RETIRE_AGE],[Moving_Target(Age_l_index) Plan_upper_Q],'r-x');
plot([Age_upper_Q RETIRE_AGE],[Moving_Target(Age_u_index) Plan_lower_Q],'r-x');
plot([Age_upper_Q RETIRE_AGE],[Moving_Target(Age_u_index) Plan_upper_Q],'r-x');
title('Standard Policy: Sample Paths & Retirement Diamond');
xlabel('Age');
ylabel('Portfolio Value');
legend([hline0, hline1,hline2,hline3,hline4],...
    'Scenarios','Target Wealth','Lowered Target','Retirement Age 68',...
    'Policy Diamond','Location','northwest')

hold off;

standard_port = STD_PENSION_PLAN;
end
```

```
function [ std_pension ] = StandardPolicy( pension_plan, target_reached,...
    cash_rets, dc_weight, savings_weight, salaries, retirement_id, spending )
cash_rets = 1+cash_rets;

CAP_GAINS_TAX = 0.1;
REAL_GAINS    = 1 - CAP_GAINS_TAX;

for i = 1:size(target_reached,2)
first = find(target_reached(:,i) > 0.5,1);
target_reached(first,i) = 0;
end

bool_multi = 1-target_reached;
bool_multi(1:retirement_id,:) = 1;
% PUT ALL IN CASH
std_pension = pension_plan .* bool_multi;

for s_id = 1:size(std_pension,2)
    start_index = find(std_pension(:,s_id) < 0.01, 1);

    for age_id = start_index:max(start_index,size(std_pension,1))
        std_pension(age_id, s_id) = std_pension(age_id-1, s_id) * ...
            cash_rets(age_id,s_id);
        std_pension(age_id, s_id) = std_pension(age_id, s_id) - spending;
    end
end
end
```

Prudent Pension Plan Simulator:

```
function [ prudent_returns ] = PrudentPensionPlanSimulator( NAME, RETIRE_AGE, ...
    MIN_DEATH_AGE, RETURNS, BOND_RETS, lower_target_vector )

SOCIAL_SEC      = 30000;
PERC_TARGET     = 0.04;
FUND_RATIO_CAP  = 1.01;

PLOT_NUM        = 15;
SCENARIOS       = size(RETURNS,2);
% Read in all data for named person
configs = getConfiguration(NAME);
[Age, Sex, Salary, SalaryIncr, MatchingContr,...
    AdditionContr, MaxSalarySaved, DCBal, SavingsBal] = GetValues(configs);

TOTAL_YEARS        = 100-Age;
years              = 0:TOTAL_YEARS;

% Calculate retirement goal
annual_salary = Salary*((1+SalaryIncr).^years);
[target_wealth, Moving_Target, yearly_spending ] = TargetWealth(annual_salary, Age, ...
    RETIRE_AGE, SOCIAL_SEC, PERC_TARGET, MIN_DEATH_AGE, MaxSalarySaved);
target_wealth
lower_target_wealth = target_wealth * 0.85;


RETIREMENT_INDEX = RETIRE_AGE - Age + 1;
Lower_Moving_Target = Moving_Target * 0.85;

% DO OPTIMISATION ON SAVINGS AND DC ALLOCATION
function [c, ceq] = DC_savings_confun(s_w)
    c = [s_w - (MaxSalarySaved - MatchingContr);
        (MaxSalarySaved - AdditionContr - MatchingContr)-s_w];
    ceq = [];
end

function [prob_achieve] = CalcProb(w)
    pw           = MaxSalarySaved - w + MatchingContr;
    DC_port      = PensionPort(DCBal, annual_salary, pw, RETURNS, Age);
    Savings_port = NonPensionPort(SavingsBal, annual_salary, w, RETURNS, Age);
    PENSION_PLAN = DC_port + Savings_port;
    prob_achieve = sum(PENSION_PLAN(RETIREMENT_INDEX,:)>target_wealth)/SCENARIOS;
end

op = optimset('MaxFunEvals',100000, 'MaxIter', 100000, ...
        'TolX', 0.001,'Display', 'off');

[opt_sw, ~]=fmincon(@(w) (-CalcProb(w)),...
    0.06,[],[],[],[],[],[],@(w) (DC_savings_confun(w)),op);

opt_pw       = (MaxSalarySaved + MatchingContr) - opt_sw;
DC_port      = PensionPort(DCBal, annual_salary, opt_pw, RETURNS, Age);
Savings_port = NonPensionPort(SavingsBal, annual_salary, opt_sw, RETURNS, Age);
PENSION_PLAN = DC_port + Savings_port;

% HISTOGRAMS
PlotPortHist(DC_port, Age, 'DC ', 'Prudent');
PlotPortHist(Savings_port, Age, 'Savings ', 'Prudent');

figure;
hold on;
hist(PENSION_PLAN(RETIREMENT_INDEX,:),20);
plot([target_wealth target_wealth],get(gca,'ylim'),'r');
title('Prudent Policy - Combined Histogram at 68');
xlabel('Portfolio Value');
hold off;

% SHE CAN LOWER BY 15% IF GOTTEN BIG HITS IN THE STOCK MARKET
adjusted_target_wealth = lower_target_vector * lower_target_wealth + ...
```

```
    (1-lower_target_vector) * target_wealth;
prob_achieve = sum(PENSION_PLAN(RETIREMENT_INDEX,:)>target_wealth)/SCENARIOS
prob_achieve_lower = ...
    sum(PENSION_PLAN(RETIREMENT_INDEX,:)>adjusted_target_wealth)/SCENARIOS

% SEE WHEN FUNDING RATIO IS ABOVE 1
TARGET_REACHED = PENSION_PLAN'>repmat(Moving_Target*FUND_RATIO_CAP,SCENARIOS,1);
ages_reached   = zeros(1,SCENARIOS);

for i = 1:SCENARIOS
    reach_index     = find(TARGET_REACHED(i,:) > 0.5,1);
    ages_reached(i) = reach_index+ Age - 1;
    TARGET_REACHED(i,reach_index:end) = 1;
end

EXPECTED_DELAYED_AGE         = mean(ages_reached(ages_reached>RETIRE_AGE))
MEAN_REACH_TARGET_AGE        = mean(ages_reached)
PROB_REACH_TARGET_BEFORE_RET = sum(ages_reached<=RETIRE_AGE)/SCENARIOS

figure;
hold on;
hist(ages_reached,20);
plot([RETIRE_AGE RETIRE_AGE],get(gca,'ylim'),'r');
title('Prudent Policy - Achieving Retirement Age');
xlabel('Age');
hold off;

% IMMUNISED
immunised_port = ImmunisePolicy(PENSION_PLAN, TARGET_REACHED',...
    BOND_RETS, opt_pw, opt_sw, annual_salary, RETIREMENT_INDEX, yearly_spending);
% CALCULATE 80% CONF INTERVAL
QUANTILE    = 0.1;
Age_lower_Q = quantile(ages_reached,QUANTILE);
Age_upper_Q = quantile(ages_reached,1-QUANTILE);
Age_l_index = ceil(Age_lower_Q-Age+1);
Age_u_index = floor(Age_upper_Q-Age+1);

Plan_lower_Q = quantile(immunised_port(RETIREMENT_INDEX,:),QUANTILE);
Plan_upper_Q = quantile(immunised_port(RETIREMENT_INDEX,:),1-QUANTILE);
figure
hold on;
hline0 = plot(repmat(Age:MIN_DEATH_AGE,1,1)', ...
    immunised_port(1:(MIN_DEATH_AGE-Age+1),1),'k--');
set(hline0, 'color', [0.5 0.5 0.5])
hl = plot(repmat(Age:MIN_DEATH_AGE,PLOT_NUM-1,1)', ...
    immunised_port(1:(MIN_DEATH_AGE-Age+1),2:PLOT_NUM),'k--');
set(hl, 'color', [0.5 0.5 0.5])
hline1 = plot(Age:MIN_DEATH_AGE,Moving_Target(1:length(Age:MIN_DEATH_AGE)),'k-x');
```

```
hline2 = plot(Age:MIN_DEATH_AGE,Lower_Moving_Target(1:length(Age:MIN_DEATH_AGE)),'y-x');
hline3 = plot([RETIRE_AGE RETIRE_AGE],get(gca,'ylim'),'b-');
hline4 = plot([Age_lower_Q RETIRE_AGE],[Moving_Target(Age_l_index) Plan_lower_Q],'r-x');
plot([Age_lower_Q RETIRE_AGE],[Moving_Target(Age_l_index) Plan_upper_Q],'r-x');
plot([Age_upper_Q RETIRE_AGE],[Moving_Target(Age_u_index) Plan_lower_Q],'r-x');
plot([Age_upper_Q RETIRE_AGE],[Moving_Target(Age_u_index) Plan_upper_Q],'r-x');
title('Standard Policy: Sample Paths & Retirement Diamond');
xlabel('Age');
ylabel('Portfolio Value');
legend([hline0, hline1,hline2,hline3,hline4],...
    'Scenarios','Target Wealth','Lowered Target','Retirement Age 68',...
    'Policy Diamond','Location','northwest')
hold off;

prudent_returns = immunised_port;
end
```

```
function [ imm_pension_plan ] = ImmunisePolicy( pension_plan, target_reached,...
    bond_rets, dc_weight, savings_weight, salaries, retirement_id, spending )
bond_rets = 1+bond_rets;

CAP_GAINS_TAX = 0.1;
REAL_GAINS    = 1 - CAP_GAINS_TAX;

for i = 1:size(target_reached,2)
first = find(target_reached(:,i) > 0.5,1);
target_reached(first,i) = 0;
end

% PUT ALL IN BONDS ONCE ABOVE LEVEL
imm_pension_plan = pension_plan .* (1-target_reached);

for s_id = 1:size(imm_pension_plan,2)
start_index = find(imm_pension_plan(:,s_id) < 0.01, 1);

for age_id = start_index:max(start_index,size(imm_pension_plan,1))
imm_pension_plan(age_id, s_id) = imm_pension_plan(age_id-1, s_id) * ...
    bond_rets(age_id,s_id);

% STILL CONTRIBUTE
if age_id <= retirement_id
imm_pension_plan(age_id, s_id) = imm_pension_plan(age_id, s_id) +...
        dc_weight * salaries(age_id) + ...
        (1 + (bond_rets(age_id,s_id)-1) * REAL_GAINS) ...
        * savings_weight * salaries(age_id);
else
imm_pension_plan(age_id, s_id) = imm_pension_plan(age_id, s_id) - spending;
end
```

```

end
end
end
```

Additional:

```
function [ Big_Hits ] = BigHitsVector(rets, QUANTILE)
SCENARIOS   = size(rets,1);
TOTAL_YEARS = size(rets,3);

new_rets    = reshape(rets(:,1,:),SCENARIOS,TOTAL_YEARS);
Big_Hits    = zeros(1,SCENARIOS);
%QUANTILE    = .25;

for i = 1:SCENARIOS
    sorted_rets = sort(new_rets(i,:));

    index       = ceil(length(sorted_rets)*QUANTILE);

    Big_Hits(i) = sorted_rets(index);
end
end
```

```
function [ avg_return, avg_std, yearly_returns ] = FixedMix( rets, w)
SCENARIOS   = size(rets,1);
NUM_ASSETS  = size(rets,2);
TOTAL_YEARS = size(rets,3);

avg_scenario_ret = zeros(NUM_ASSETS,1);
avg_scenario_var = zeros(NUM_ASSETS,1);
total_returns    = zeros(TOTAL_YEARS,SCENARIOS);
yearly_returns   = zeros(TOTAL_YEARS,SCENARIOS);

for i = 1:SCENARIOS

    scenario_rets    = reshape(rets(i,:,:),NUM_ASSETS,TOTAL_YEARS);
    weighted_returns = scenario_rets' .* repmat(w,TOTAL_YEARS,1);

    yearly_returns(:,i) = sum(weighted_returns,2) + 1;
    total_returns(:,i)  = cumprod(yearly_returns(:,i));

    avg_scenario_ret(i) = geomean(yearly_returns(:,i))-1;
    avg_scenario_var(i) = var(yearly_returns(:,i));

end

avg_return = (mean(avg_scenario_ret));
avg_std    = sqrt(mean(avg_scenario_var));
```

```

end
```

8.2 Investment Policies

```
function [ avg_return, avg_std, yearly_returns ] = FixedMix( rets, w)
SCENARIOS   = size(rets,1);
NUM_ASSETS  = size(rets,2);
TOTAL_YEARS = size(rets,3);

avg_scenario_ret = zeros(NUM_ASSETS,1);
avg_scenario_var = zeros(NUM_ASSETS,1);
total_returns    = zeros(TOTAL_YEARS,SCENARIOS);
yearly_returns   = zeros(TOTAL_YEARS,SCENARIOS);

for i = 1:SCENARIOS

    scenario_rets    = reshape(rets(i,:,:),NUM_ASSETS,TOTAL_YEARS);
    weighted_returns = scenario_rets' .* repmat(w,TOTAL_YEARS,1);

    yearly_returns(:,i) = sum(weighted_returns,2) + 1;
    total_returns(:,i)  = cumprod(yearly_returns(:,i));

    avg_scenario_ret(i) = geomean(yearly_returns(:,i))-1;
    avg_scenario_var(i) = var(yearly_returns(:,i));

end

avg_return = (mean(avg_scenario_ret));
avg_std    = sqrt(mean(avg_scenario_var));

end
```

```
function [ opt_w, opt_ret ] = OptimalFixedMixWts( rets, cov_ret, target_risk )

SCENARIOS   = size(rets,1);
TOTAL_YEARS = size(rets,3);
INIT_W      = [.5 .5 .5 0];

function [ r ] = FM_Returns( w )
   [r, s , ~] = FixedMix(rets,w);
   %returns = r/s;
end

function [ c, ceq ] = ConfunFixedMixWrap( w )
   [c, ceq] = ConfunFixedMix(w, cov_ret, target_risk);
end
```

```

 op = optimset('MaxFunEvals',100000, 'MaxIter', 100000, ...
        'TolX', 0.001,'Display', 'off');
[opt_w, opt_ret] = fmincon(@(w) (-FM_Returns(w)),...
    INIT_W,[],[],[],[],[],[],@(w) (ConfunFixedMixWrap(w)),op);
end
```

```
function [ avg_return, avg_std, yearly_returns ] = VarFixMix(rets, CURRENT_AGE, new_age, VAR_TIME,
  w)
% Fixed Mix over different periods of time
%  eg change fixed mix every n years
startt   = (new_age - CURRENT_AGE +1);
endt     = min((new_age - CURRENT_AGE + VAR_TIME),65);
sub_rets = rets(:,:,(startt:endt));

SCENARIOS   = size(sub_rets,1);
NUM_ASSETS  = size(sub_rets,2);
TOTAL_YEARS = size(sub_rets,3);

avg_scenario_ret = zeros(NUM_ASSETS,1);
avg_scenario_var = zeros(NUM_ASSETS,1);
total_returns    = zeros(TOTAL_YEARS,SCENARIOS);
yearly_returns   = zeros(TOTAL_YEARS,SCENARIOS);

for i = 1:SCENARIOS

    scenario_rets    = reshape(sub_rets(i,:,:),NUM_ASSETS,TOTAL_YEARS);
    weighted_returns = scenario_rets' .* repmat(w,TOTAL_YEARS,1);

    yearly_returns(:,i) = sum(weighted_returns,2) + 1;
    total_returns(:,i)  = cumprod(yearly_returns(:,i));

    avg_scenario_ret(i) = geomean(yearly_returns(:,i))-1;
    avg_scenario_var(i) = var(yearly_returns(:,i));

end

avg_return = (mean(avg_scenario_ret));
avg_std    = sqrt(mean(avg_scenario_var));

end
```

```
function [ avg_return, avg_std, annual_rets ] = VarFixMixOutput(rets, CURRENT_AGE, VAR_TIME, wts,
  RETIRE_AGE)
new_age     = CURRENT_AGE;
SCENARIOS   = size(rets,1);
NUM_ASSETS  = size(rets,2);
TOTAL_YEARS = size(rets,3);
```

```matlab
RETIRE_AGE  = 100;

opt_ret           = zeros(TOTAL_YEARS,1);
opt_std           = zeros(TOTAL_YEARS,1);
annual_rets       = zeros(TOTAL_YEARS,SCENARIOS);
avg_scenario_ret = zeros(NUM_ASSETS,1);
avg_scenario_var = zeros(NUM_ASSETS,1);

i = 1;
j = 1;
while new_age < RETIRE_AGE


  [avg_return, avg_std, yearly_returns] = VarFixMix(rets, CURRENT_AGE, new_age, VAR_TIME, wts(:,j)
    ');

  endtt                           = min(65,(i+VAR_TIME)-1);
  opt_ret(i:(i+VAR_TIME))          = avg_return;
  opt_std(i:(i+VAR_TIME))          = avg_std;
  annual_rets(i:endtt,:)           = yearly_returns;


  new_age   = new_age + VAR_TIME;
  i         = i + VAR_TIME;
  j         = j + 1;
end

for i = 1:SCENARIOS

    avg_scenario_ret(i) = geomean(yearly_returns(:,i))-1;
    avg_scenario_var(i) = var(yearly_returns(:,i));

end

avg_return = (mean(avg_scenario_ret));
avg_std    = sqrt(mean(avg_scenario_var));

end
```

```matlab
function [ var_opt_wts, var_opt_ret ] = OptimalVarFixedMixWts(rets, CURRENT_AGE, RETIRE_AGE,
  VAR_TIME, cov_ret, target_risk)
new_age     = CURRENT_AGE;
SCENARIOS   = size(rets,1);
NUM_ASSETS  = size(rets,2);
TOTAL_YEARS = size(rets,3);
INIT_W      = [0.8 0 0.2 0];
RETIRE_AGE = 100;
var_opt_wts  =  zeros(NUM_ASSETS,ceil((RETIRE_AGE-CURRENT_AGE)/10));
```

```
var_opt_ret  =  zeros(1,ceil((RETIRE_AGE-CURRENT_AGE)/10));

function [ r ] = FM_Returns( w )
   [r, s , ~] = VarFixMix(rets, CURRENT_AGE, new_age, VAR_TIME, w);

end

function [ c, ceq ] = ConfunFixedMixWrap( w )
   [c, ceq] = ConfunFixedMix(w, cov_ret, target_risk);
end

i = 1;

while new_age < RETIRE_AGE
    op = optimset('MaxFunEvals',100000, 'MaxIter', 100000, ...
        'TolX', 0.001,'Display', 'off');

    [opt_wts, opt_ret] = fmincon(@(w) (-FM_Returns(w)),...
        INIT_W,[],[],[],[],[],[],@(w) (ConfunFixedMixWrap(w)),op);
    INIT_W = opt_wts;
    var_opt_wts(:,i) = opt_wts;
    var_opt_ret(i)   = opt_ret;

    new_age  = new_age + VAR_TIME;
    i       = i + 1;
end

end
```

```
function [ opt_w, opt_ret ] = OptimalDDWts( rets, cov, targ, targetDD )

SCENARIOS   = size(rets,1);
TOTAL_YEARS = size(rets,3);
INIT_W      = [.5 .5 .5 0];

function [ r ] = FM_Returns( w )
   [r, s , ~] = FixedMix(rets,w);
   %returns = r/s;
end

function [ c, ceq ] = ConfunFixedMixWrap( w )
   [c, ceq] = DDConFun( w , cov, targ, rets, targetDD );
end

 op = optimset('MaxFunEvals',100000, 'MaxIter', 100000, ...
        'TolX', 0.001,'Display', 'off');
[opt_w, opt_ret] = fmincon(@(w) (-FM_Returns(w)),...
    INIT_W,[],[],[],[],[],[],@(w) (ConfunFixedMixWrap(w)),op);
```

```


maxDD    = mean(max(DrawDown(opt_w,rets)))
end
```

```
function [ max_dds ] = maxDD( returns )
% Get cumulative returns
max_dds  = zeros(1,size(returns,2));

for i = 1:length(max_dds)
port_ret = returns(:,i);
cum_ret  = cumprod(port_ret);

% Calculate difference with the max point
dd = @(index) ((max(cum_ret(1:index)) - cum_ret(index))...
    /max(cum_ret(1:index)));

drawdowns = arrayfun (dd, 1:length(cum_ret));
max_dds(i) = max(drawdowns);
end
end
```

```
function [c,ceq] = DDConFun( weights , cov, targ, rets, targetDD )
PortfolioRisk     = @(weights,cov) (weights*cov*weights');
dd                = DrawDown(weights, rets);
maxDD             = mean(max(dd));

ceq = weights(1)+weights(2)+weights(4)-1;

c   = [weights(3)-.5;
    -weights(1);
    -weights(2);
    -weights(3);
    -weights(4);
    sqrt(weights*cov*weights')-targ;
    (maxDD - targetDD)];


end
```

```
function [ drawdowns ] = DrawDown( w, rets )
SCENARIOS   = size(rets,1);
NUM_ASSETS  = size(rets,2);
TOTAL_YEARS = size(rets,3);

drawdowns   = zeros(TOTAL_YEARS,SCENARIOS);

for i = 1:SCENARIOS
```

```
port_ret = w * reshape(rets(1,:,:),NUM_ASSETS,TOTAL_YEARS) + 1;
cum_ret   = cumprod(port_ret);

% Calculate difference with the max point
dd = @(index) ((max(cum_ret(1:index)) - cum_ret(index))...
    /max(cum_ret(1:index)));

drawdowns(:,i) = arrayfun (dd, 1:length(cum_ret));

end

end
```

```
function [ avg_return, avg_std, yearly_returns ] = Bogle( rets , WORKING_AGE_INDEX)
SCENARIOS    = size(rets,1);
NUM_ASSETS   = size(rets,2);
TOTAL_YEARS = size(rets,3);

avg_scenario_ret = zeros(NUM_ASSETS,1);
avg_scenario_var = zeros(NUM_ASSETS,1);
total_returns     = zeros(TOTAL_YEARS,SCENARIOS);
yearly_returns    = zeros(TOTAL_YEARS,SCENARIOS);

WEIGHTS       = zeros(NUM_ASSETS,TOTAL_YEARS);
AGE_INDEX     = (1:1:TOTAL_YEARS)/100;

for i = 1:TOTAL_YEARS
WEIGHTS(1,i)       = 1-AGE_INDEX(i);
WEIGHTS(2,i)       = 0+AGE_INDEX(i);
WEIGHTS(3,i)       = .5;
WEIGHTS(4,i)       = 0;
end


for i = 1:SCENARIOS
scenario_rets      = reshape(rets(i,:,:),NUM_ASSETS,TOTAL_YEARS);
weighted_returns   = scenario_rets' .* WEIGHTS';

    yearly_returns(:,i) = sum(weighted_returns,2) + 1;
    total_returns(:,i)  = cumprod(yearly_returns(:,i));

    avg_scenario_ret(i) = geomean(yearly_returns(:,i))-1;
    avg_scenario_var(i) = var(yearly_returns(:,i));

end

avg_return = (mean(avg_scenario_ret));
```

```
avg_std     = sqrt(mean(avg_scenario_var));

end
```

```
function [ avg_return, avg_std, yearly_returns ] = VolPump( rets , PUMP_AMT, w , THRESHOLD )
SCENARIOS    = size(rets,1);
NUM_ASSETS   = size(rets,2);
TOTAL_YEARS  = size(rets,3);
%THRESHOLD    = -.3;
%PUMP_AMT     = .05;
%w            = [.7, .3, .5, 0];
BigHit       = zeros(1, SCENARIOS);
HitCounter   = zeros(1, SCENARIOS);
lookup       = .4:(PUMP_AMT):1;
lookup_vect  = zeros(length(lookup), 2);
wts_array    = zeros(SCENARIOS,NUM_ASSETS,TOTAL_YEARS);

lookup_vect(:,1) = 1:(-PUMP_AMT):.4;
lookup_vect(:,2) = 1-lookup_vect(:,1);
startt           = (length(lookup_vect)+1)/2;
HitCounter(:)       = startt;
wts_array(1:SCENARIOS,1,1) = w(1);
wts_array(1:SCENARIOS,2,1) = w(2);
wts_array(1:SCENARIOS,3,1) = w(3);
wts_array(1:SCENARIOS,4,1) = w(4);

avg_scenario_ret = zeros(NUM_ASSETS,1);
avg_scenario_var = zeros(NUM_ASSETS,1);

yearly_returns = zeros(TOTAL_YEARS,SCENARIOS);

for i = 1:SCENARIOS

    scenario_rets    = reshape(rets(i,:,:),NUM_ASSETS,TOTAL_YEARS);
    for j = 2: TOTAL_YEARS

        if scenario_rets(1,j) < THRESHOLD

            BigHit(i)     = BigHit(i) + 1;
            HitCounter(i) = min(length(lookup),HitCounter(i) + 1);

        elseif scenario_rets(1,j) > -THRESHOLD

            HitCounter(i) = max(1,HitCounter(i) - 1);

        end

        wts_array(i,1,j) = lookup_vect(HitCounter(i),1);
        wts_array(i,2,j) = lookup_vect(HitCounter(i),2);
        wts_array(i,3,j) = w(3);
        wts_array(i,4,j) = w(4);
    end

    wtd_ret_array = reshape(wts_array(i,:,:),NUM_ASSETS,TOTAL_YEARS).*scenario_rets;
    yearly_returns(:,i) = sum(wtd_ret_array,1)+1;

    avg_scenario_ret(i) = geomean(yearly_returns(:,i))-1;
    avg_scenario_var(i) = var(yearly_returns(:,i));

end

avg_return = (mean(avg_scenario_ret));
avg_std    = sqrt(mean(avg_scenario_var));

end
```

```
function [ opt_w, opt_ret ] = OptimalDDWts( rets, cov, targ, targetDD )

SCENARIOS   = size(rets,1);
TOTAL_YEARS = size(rets,3);
INIT_W      = [.5 .5 .5 0];

function [ r ] = FM_Returns( w )
   [r, s , ~] = FixedMix(rets,w);
   %returns = r/s;
end

function [ c, ceq ] = ConfunFixedMixWrap( w )
   [c, ceq] = DDConFun( w , cov, targ, rets, targetDD );
end

 op = optimset('MaxFunEvals',100000, 'MaxIter', 100000, ...
        'TolX', 0.001,'Display', 'off');
[opt_w, opt_ret] = fmincon(@(w) (-FM_Returns(w)),...
    INIT_W,[],[],[],[],[],[],@(w) (ConfunFixedMixWrap(w)),op);


maxDD   = mean(max(DrawDown(opt_w,rets)))
end
```

8.3 General Files

Execution:

```
%% Financial Risk Management - Final Project
% Lu Xin, Edward Stivers     - Jan 19, 2015
```

```matlab
clc; clear all;
load('ftp_scenario');

%% Vol Pump Strategy
% Get returns of years from present until retirement
% Pump values must divide cleanly into 60

% USE WEIGHTS FROM BEST FM
[w, ~] =OptimalFixedMixWts( rets(:,:,WORKING_AGE_INDEX),...
    cov_ret,TARGET_RISK);

PUMP   = [.01, .02, .05, .10,.15];
THRESH = .10:.05:.30;
Record = zeros(length(PUMP)*length(THRESH),5);
for i = 1:length(PUMP)
    for j = 1:length(THRESH)
        PUMP_AMT  = PUMP(i);
        THRESHOLD = -THRESH(j);
        [Vol_ret, Vol_std, ~] = VolPump(rets , PUMP_AMT, w , THRESHOLD);
        y = 5*i + j - 5;
        Record(y,1) = PUMP_AMT;
        Record(y,2) = THRESHOLD;
        Record(y,3) = Vol_ret;
        Record(y,4) = Vol_std;
        Record(y,5) = Vol_ret/Vol_std;
    end
end
Record;
[row, col] = find(ismember(Record, max(Record(:,5))));

[Vol_ret, Vol_std, Vol_returns] = VolPump(rets , Record(row,1), w , Record(row,2));
Vol_ret
Vol_std
Vol_SR = Vol_ret/Vol_std
Vol_DD = mean(maxDD(Vol_returns))

%% Simulate Standard Pension Plan - Invest in CASH after Retirement
[PensionPortfolio ]= ...
PensionPlanSimulator(NAME, RETIRE_AGE, MIN_DEATH_AGE, Vol_returns, ...
CASH_RETS, lower_target_vector);

PortSaving = PensionPortfolio(1:RETIRE_AGE-Age+1,:);
Returns    = diff(PortSaving)./PortSaving(1:end-1,:);
MaxDD      = mean(maxDD(Returns+1))

%% Surf Standard Pension Plan Paths
[~, sort_ind] = sort(PensionPortfolio(end,:));
figure
```

```matlab
X_AXIS = 1:size(PensionPortfolio,2);
Y_AXIS = (1:size(PensionPortfolio,1)) + Age - 1;
[X Y] = meshgrid(X_AXIS, Y_AXIS);
surf(X,Y,PensionPortfolio(:,sort_ind));
title('Standard Bogle Retirement - Scenario Paths');
xlabel('Scenarios');
ylabel('Age');
zlabel('Portfolio Value');


%% NEXT POLICY:


%% Simulate Prudent Pension Plan - Invest in Bonds After FR = 1.05
[PensionPortfolioPrudent ]= ...
PrudentPensionPlanSimulator(NAME, RETIRE_AGE, MIN_DEATH_AGE, Vol_returns, ...
BOND_RETS,lower_target_vector);

PrudentPortSaving = PensionPortfolioPrudent(1:RETIRE_AGE-Age+1,:);
PrudentReturns    = diff(PrudentPortSaving)./PrudentPortSaving(1:end-1,:);
PrudentMaxDD      = mean(maxDD(PrudentReturns+1))

%% Surf Prudent Pension Plan Paths
[~, sort_ind] = sort(PensionPortfolioPrudent(end,:));
figure
X_AXIS = 1:size(PensionPortfolioPrudent,2);
Y_AXIS = (1:size(PensionPortfolioPrudent,1)) + Age - 1;
[X, Y] = meshgrid(X_AXIS, Y_AXIS);
surf(X,Y,PensionPortfolioPrudent(:,sort_ind));
title('Prudent Bogle Retirement - Scenario Paths');
xlabel('Scenarios');
ylabel('Age');
zlabel('Portfolio Value');

%%
term_wealth = LongevityFinder(PensionPortfolioPrudent',Sex);
```

Longevity:

```matlab
function [ MATRIX ] = MortalityEngine( sex, scenarios, age)
data = csvread('../data/MortTable.csv',1,0);
AGES = data(:,1);
M_PX = data(:,2);
M_LE = data(:,3);
F_PX = data(:,4);
F_LE = data(:,5);

% Sex:1 = Male % Sex:0 = Female
if sex == 1
```

```matlab
    PROBS = M_PX;
else
    PROBS = F_PX;
end

years = 100 - age;
index = find(AGES == age);

transition_prob = PROBS(index:(index+years-1));
MATRIX = zeros(years, scenarios);

MATRIX(1,:) = rand(1,scenarios) < (1-transition_prob(1));
for i = 2:years
    MATRIX(i,:) = MATRIX(i-1,:) .* (rand(1,scenarios) < (1-transition_prob(i)));
end
end
```

```matlab
function [ Term_Wealth ] = LongevityFinder( rets, Sex )
Age               = 75;
Sex               = 0;
SCENARIOS         = size(rets,1);
Term_Age          = zeros(1,SCENARIOS);
Term_Age(:)       = Term_Age(:) + 75;
Prob_Age          = zeros(100-Age,1);
Mortality_matrix  = MortalityEngine( Sex ,SCENARIOS, Age);
Term_Wealth       = zeros(1,SCENARIOS);
BINS              = 15;

for i = 1:SCENARIOS
    if sum(Mortality_matrix(:,i))>0
    Term_Age(i)   = Term_Age(i) + find(Mortality_matrix(:,i),1,'last');
    else
    end
end

for j = 1:(100-Age)
    Prob_Age(j) = sum(Mortality_matrix(j,:));
end
    Prob_Age = Prob_Age/SCENARIOS;
    figure;
    bar(Prob_Age,1);
    title('Longevity Histogram');
    xlabel('Age');
    labels = 76:1:100;
    set(gca, 'XTick', 1:length(labels));
    set(gca, 'XTickLabel', labels); % Change x-axis ticks labels.

    %for i = 1:65
    %rets(:,i)  = i;
    %end
    Find_Term_Age = Term_Age - Age + 41;

    for i = 1:SCENARIOS
    Term_Wealth(i) = rets(i,Find_Term_Age(i));
    end

    figure; hold on;
    [f,x]=hist(Term_Wealth,BINS);
    bar(x,f/sum(f));
    title('Terminal Wealth');
    xlabel('Wealth');
    ylabel('Probability');
    plot([0 0],get(gca,'ylim'),'r-');
    hold off;
end
```

Additional files:

```
function [ SCENARIO_RETS , cov_ret] = ScenarioGen (data, GivenRets, SAMPLE_SIZE, ...
    SCENARIOS, TOTAL_YEARS)
daily_ret   = data(2:end,:) ./ data(1:end-1,:)-1;

% Generate random date points to calculate covariance
days           = size(data,1);
trading_days   = 252;
starting_days  = zeros(1,SAMPLE_SIZE)-1;
MAX_SAMPLE_DAY = days - 1 - trading_days;

for i = 1:SAMPLE_SIZE
    day = floor(rand * MAX_SAMPLE_DAY) + 1;
    while ismember(day, starting_days)
        day = floor(rand * MAX_SAMPLE_DAY) + 1;
    end
    starting_days(i) = day;
end


getGeoRet    = @(start) prod(daily_ret(start:(start+trading_days),:)+1)-1;
data_rets    = arrayfun (@(x) getGeoRet(x), starting_days,'UniformOutput',false);
mean_rets    = cell2mat(data_rets');

cor_ret      = corr(mean_rets);
cor_ret(4,4) = 1;

cov_ret      = cov(mean_rets);
cov_ret(4,4) = 0.01^2;

```

```
% Cholesky Factorisation - F'*F = cov_ret
F = chol(cov_ret);

%% Generate 1000 Scenarios
% Col 1 - Normal. Col 2 - Crash
antithetic_scenarios = SCENARIOS/2;

r_bar          = GivenRets;
SCENARIO_RETS  = zeros(SCENARIOS,4,TOTAL_YEARS);

for i = 1:antithetic_scenarios
    WT = randn(4,TOTAL_YEARS);
    SCENARIO_RETS(2*i-1,:,:) = repmat(r_bar',1,TOTAL_YEARS) + F * WT;
    SCENARIO_RETS(2*i,:,:)   = repmat(r_bar',1,TOTAL_YEARS) + F * (-WT);
end

end
```

```
%% Financial Risk Management - Final Project
% Lu Xin, Edward Stivers      - Jan 19, 2015
clc; clear all;
% SETTINGS
SCENARIOS     = 1000;
RETIRE_AGE    = 68;
MIN_DEATH_AGE = 75;
SAMPLE_SIZE   = 1000;
TARGET_RISK   = 0.15;

BIG_HITS_QUANTILE = 0.20;

% INPUT NAME HERE
NAME = 'Jennifer';
% Read in all data for named person
configs = getConfiguration(NAME);
[Age, Sex, ~, ~, ~, ~, ~, ~, ~] = GetValues(configs);

%% Get Data
% Returns
STOCKS_RET = 0.06;
BONDS_RET  = 0.025;
FTSE_RET   = 0.05;
CASH_RET   = 0.005;
GivenRets  = [STOCKS_RET BONDS_RET FTSE_RET CASH_RET];

% Historical Data
data = csvread('../data/Data.csv',1,1);

sp = 1; bonds = 2; ftse = 3; cash = 4;
```

```

%% Generation of Scenarios
% Calculate the Annual Salaries
TOTAL_YEARS       = 100-Age;
years             = 0:TOTAL_YEARS;

WORKING_AGE_INDEX = 1:(RETIRE_AGE-Age+1);

have_generated = exist('rets');
if have_generated == 0
    [rets, cov_ret] = ScenarioGen (data, GivenRets, SAMPLE_SIZE, ...
        SCENARIOS, TOTAL_YEARS+1);
end
BOND_RETS   = reshape(rets(:,bonds,:),SCENARIOS,TOTAL_YEARS+1)';
CASH_RETS   = reshape(rets(:,cash,:),SCENARIOS,TOTAL_YEARS+1)';

% 20% of time she experiences a hit at least this big
LARGE_STOCK_HITS = BigHitsVector(rets, BIG_HITS_QUANTILE);

% Allow her to lower her target wealth in worst 10% of cases
lower_target_vector = LARGE_STOCK_HITS<=quantile(LARGE_STOCK_HITS,0.1);


save('ftp_scenario')
```

```
function [ target, Moving_Target, yearly_spending ] = TargetWealth( salaries, age, retirement_age,
  ...
    social, percentage_target, min_death_age, max_save)
% Calculate spending per year
three_years_before_ret = (1:3) + retirement_age - 3 - age;

yearly_spending = (0.7 * mean(salaries(three_years_before_ret)) - social);

target = yearly_spending/percentage_target;


% Delay retirement she sacrifices 0.3% per year
delayed_target = zeros(1,min_death_age - retirement_age);

for i = 1:(100 - retirement_age)
    delayed_target(i) = yearly_spending/(percentage_target + 0.003*i);
end

% Early retirement account for 2.5% bond
early_target     = zeros(1,length(age:retirement_age));
early_target(1)  = target;
BOND_RET         = 0.025;
effective_disc   = (BOND_RET*0.9)/4 + BOND_RET*3/4;
```

```

for i = 2:length(early_target)
    early_target(i) = early_target(i-1) * exp(-effective_disc);

    year_index = retirement_age - age + 2 - i;

    early_target(i) = early_target(i) - salaries(year_index) * max_save;
end

% Combine Targets
Moving_Target   =  [ early_target(end:-1:1) delayed_target];

end
```

```
function [ Big_Hits ] = BigHitsVector(rets, QUANTILE)
SCENARIOS   = size(rets,1);
TOTAL_YEARS = size(rets,3);

new_rets    = reshape(rets(:,1,:),SCENARIOS,TOTAL_YEARS);
Big_Hits    = zeros(1,SCENARIOS);
%QUANTILE    = .25;

for i = 1:SCENARIOS
    sorted_rets = sort(new_rets(i,:));

    index       = ceil(length(sorted_rets)*QUANTILE);

    Big_Hits(i) = sorted_rets(index);
end
end
```

```
%% Financial Risk Management - Final Project
% Lu Xin, Edward Stivers     - Jan 19, 2015
clc; clear all;
% SETTINGS
SCENARIOS     = 1000;

RETIRE_AGE    = 68;
MIN_DEATH_AGE = 75;
SAMPLE_SIZE   = 1000;

% INPUT NAME HERE
NAME = 'Jennifer';
% Read in all data for named person
configs = getConfiguration(NAME);
[Age, ~, ~, ~, ~, ~, ~, ~, ~] = GetValues(configs);

%% Get Data
```

```matlab
% Returns
STOCKS_RET = 0.06;
BONDS_RET  = 0.025;
FTSE_RET   = 0.05;
CASH_RET   = 0.005;
GivenRets  = [STOCKS_RET BONDS_RET FTSE_RET CASH_RET];

% Historical Data
data = csvread('../data/Data.csv',1,1);

sp = 1; bonds = 2; ftse = 3; cash = 4;

%% Generation of Scenarios
% Calculate the Annual Salaries
TOTAL_YEARS       = 100-Age;
years             = 0:TOTAL_YEARS;

WORKING_AGE_INDEX = 1:(RETIRE_AGE-Age+1);

have_generated = exist('rets');
if have_generated == 0
    [rets, cov_ret] = ScenarioGen (data, GivenRets, SAMPLE_SIZE, ...
        SCENARIOS, TOTAL_YEARS+1);
end
BOND_RETS   = reshape(rets(:,bonds,:),SCENARIOS,TOTAL_YEARS+1)';

%% Get fixed mix - EFF
% Get returns of years from present until retirement
TARGET_RISK = 0.08:0.01:0.18;
EFF         = zeros(length(TARGET_RISK),3);

for i = 1:length(TARGET_RISK)

[w, ~] =OptimalFixedMixWts( rets(:,:,WORKING_AGE_INDEX),...
    cov_ret,TARGET_RISK(i));
[FM_rbar, FM_vol, ~] = FixedMix(rets,w);
EFF(i,1:2) = [FM_rbar FM_vol];
EFF(i,3)   = FM_rbar/FM_vol;
end

plot(EFF(:,2),EFF(:,1))
title('Fixed Mix Efficient Frontier');
xlabel('Volatility');
ylabel('Return');

%% Get variable fixed mix - EFF
VAR_TIME    = 10;
TARGET_RISK = 0.08:0.01:0.18;
```

```
EFF          = zeros(length(TARGET_RISK),3);

for i = 1:length(TARGET_RISK)

[w, ~] = OptimalVarFixedMixWts(rets, Age, RETIRE_AGE, ...
    VAR_TIME, cov_ret, TARGET_RISK(i));
[FM_rbar, FM_vol, ~] = VarFixMixOutput(rets, Age, VAR_TIME, w, RETIRE_AGE);
EFF(i,1:2) = [FM_rbar FM_vol];
EFF(i,3)   = FM_rbar/FM_vol;
end

plot(EFF(:,2),EFF(:,1))
title('10 Year Variable Fixed Mix Efficient Frontier');
xlabel('Volatility');
ylabel('Return');

EFF
```

```
function [] = PlotPortHist( port, age, portname, policyname)
BINS = 12;
PLOT_YEARS = [ 60 65 68 70 75 ];
PLOT_INDEX = PLOT_YEARS - age + 1;

figure;
annotation('textbox', [0 0.9 1 0.1], ...
    'String', policyname, ...
    'EdgeColor', 'none', ...
    'HorizontalAlignment', 'center')

subplot(3,2,1);
hist(port(PLOT_INDEX(1),:),BINS);
title(strcat(portname,' Histogram  ', num2str(PLOT_YEARS(1))));
xlabel('Portfolio Value');
subplot(3,2,2);
hist(port(PLOT_INDEX(2),:),BINS);
title(strcat(portname,' Histogram  ', num2str(PLOT_YEARS(2))));
xlabel('Portfolio Value');
subplot(3,2,[3 4]);
hist(port(PLOT_INDEX(3),:),BINS);
title(strcat(portname,' Histogram  ', num2str(PLOT_YEARS(3))));
xlabel('Portfolio Value');
subplot(3,2,5);
hist(port(PLOT_INDEX(4),:),BINS);
title(strcat(portname,' Histogram  ', num2str(PLOT_YEARS(4))));
xlabel('Portfolio Value');
subplot(3,2,6);
hist(port(PLOT_INDEX(5),:),BINS);
title(strcat(portname,' Histogram  ', num2str(PLOT_YEARS(5))));
```

```
xlabel('Portfolio Value');
end
```

```
%% Financial Risk Management - Final Project
% Lu Xin, Edward Stivers     - Jan 19, 2015
clc; clear all;
load('ftp_scenario');

%% Vol Pump Strategy
% Get returns of years from present until retirement
% Pump values must divide cleanly into 60

% USE WEIGHTS FROM BEST FM
[w, ~] =OptimalFixedMixWts( rets(:,:,WORKING_AGE_INDEX),...
    cov_ret,TARGET_RISK);

PUMP   = [.01, .02, .05, .10,.15];
THRESH = .10:.05:.30;
Record = zeros(length(PUMP)*length(THRESH),5);
for i = 1:length(PUMP)
    for j = 1:length(THRESH)
        PUMP_AMT  = PUMP(i);
        THRESHOLD = -THRESH(j);
        [Vol_ret, Vol_std, ~] = VolPump(rets , PUMP_AMT, w , THRESHOLD);
        y = 5*i + j - 5;
        Record(y,1) = PUMP_AMT;
        Record(y,2) = THRESHOLD;
        Record(y,3) = Vol_ret;
        Record(y,4) = Vol_std;
        Record(y,5) = Vol_ret/Vol_std;
    end
end
Record;
[row, col] = find(ismember(Record, max(Record(:,5))));

[Vol_ret, Vol_std, Vol_returns] = VolPump(rets , Record(row,1), w , Record(row,2));
Vol_ret
Vol_std
Vol_SR = Vol_ret/Vol_std
Vol_DD = mean(maxDD(Vol_returns))

%% Simulate Standard Pension Plan - Invest in CASH after Retirement
[PensionPortfolio ]= ...
PensionPlanSimulator(NAME, RETIRE_AGE, MIN_DEATH_AGE, Vol_returns, ...
CASH_RETS, lower_target_vector);

PortSaving = PensionPortfolio(1:RETIRE_AGE-Age+1,:);
Returns    = diff(PortSaving)./PortSaving(1:end-1,:);
```

```
MaxDD       = mean(maxDD(Returns+1))

%% Surf Standard Pension Plan Paths
[~, sort_ind] = sort(PensionPortfolio(end,:));
figure
X_AXIS = 1:size(PensionPortfolio,2);
Y_AXIS = (1:size(PensionPortfolio,1)) + Age - 1;
[X Y] = meshgrid(X_AXIS, Y_AXIS);
surf(X,Y,PensionPortfolio(:,sort_ind));
title('Standard Bogle Retirement - Scenario Paths');
xlabel('Scenarios');
ylabel('Age');
zlabel('Portfolio Value');


%% NEXT POLICY:


%% Simulate Prudent Pension Plan - Invest in Bonds After FR = 1.05
[PensionPortfolioPrudent ]= ...
PrudentPensionPlanSimulator(NAME, RETIRE_AGE, MIN_DEATH_AGE, Vol_returns, ...
BOND_RETS,lower_target_vector);

PrudentPortSaving  = PensionPortfolioPrudent(1:RETIRE_AGE-Age+1,:);
PrudentReturns     = diff(PrudentPortSaving)./PrudentPortSaving(1:end-1,:);
PrudentMaxDD       = mean(maxDD(PrudentReturns+1))

%% Surf Prudent Pension Plan Paths
[~, sort_ind] = sort(PensionPortfolioPrudent(end,:));
figure
X_AXIS = 1:size(PensionPortfolioPrudent,2);
Y_AXIS = (1:size(PensionPortfolioPrudent,1)) + Age - 1;
[X, Y] = meshgrid(X_AXIS, Y_AXIS);
surf(X,Y,PensionPortfolioPrudent(:,sort_ind));
title('Prudent Bogle Retirement - Scenario Paths');
xlabel('Scenarios');
ylabel('Age');
zlabel('Portfolio Value');

%%
term_wealth = LongevityFinder(PensionPortfolioPrudent',Sex);
```

XML Parser:

```
function [ s ] = xml2struct( file )
%Convert xml file into a MATLAB structure
    if (nargin < 1)
        clc;
        help xml2struct
        return
    end

    if isa(file, 'org.apache.xerces.dom.DeferredDocumentImpl') || isa(file, 'org.apache.xerces.dom.
      DeferredElementImpl')
        % input is a java xml object
        xDoc = file;
    else
        %check for existance
        if (exist(file,'file') == 0)
            %Perhaps the xml extension was omitted from the file name. Add the
            %extension and try again.
            if (isempty(strfind(file,'.xml')))
                file = [file '.xml'];
            end

            if (exist(file,'file') == 0)
                error(['The file ' file ' could not be found']);
            end
        end
        %read the xml file
        xDoc = xmlread(file);
    end

    %parse xDoc into a MATLAB structure
    s = parseChildNodes(xDoc);

end

% ----- Subfunction parseChildNodes -----
function [children,ptext,textflag] = parseChildNodes(theNode)
    % Recurse over node children.
    children = struct;
    ptext = struct; textflag = 'Text';
    if hasChildNodes(theNode)
        childNodes = getChildNodes(theNode);
        numChildNodes = getLength(childNodes);

        for count = 1:numChildNodes
            theChild = item(childNodes,count-1);
            [text,name,attr,childs,textflag] = getNodeData(theChild);

            if (~strcmp(name,'#text') && ~strcmp(name,'#comment') && ~strcmp(name,'#
              cdata_dash_section'))
                %XML allows the same elements to be defined multiple times,
                %put each in a different cell
                if (isfield(children,name))
                    if (~iscell(children.(name)))
```

```matlab
                        %put existsing element into cell format
                        children.(name) = {children.(name)};
                    end
                    index = length(children.(name))+1;
                    %add new element
                    children.(name){index} = childs;
                    if(~isempty(fieldnames(text)))
                        children.(name){index} = text;
                    end
                    if(~isempty(attr))
                        children.(name){index}.('Attributes') = attr;
                    end
                else
                    %add previously unknown (new) element to the structure
                    children.(name) = childs;
                    if(~isempty(text) && ~isempty(fieldnames(text)))
                        children.(name) = text;
                    end
                    if(~isempty(attr))
                        children.(name).('Attributes') = attr;
                    end
                end
            else
                ptextflag = 'Text';
                if (strcmp(name, '#cdata_dash_section'))
                    ptextflag = 'CDATA';
                elseif (strcmp(name, '#comment'))
                    ptextflag = 'Comment';
                end

                %this is the text in an element (i.e., the parentNode)
                if (~isempty(regexprep(text.(textflag),'[\s]*','')))
                    if (~isfield(ptext,ptextflag) || isempty(ptext.(ptextflag)))
                        ptext.(ptextflag) = text.(textflag);
                    else

                        ptext.(ptextflag) = [ptext.(ptextflag) text.(textflag)];
                    end
                end
            end

        end
    end
end

% ----- Subfunction getNodeData -----
function [text,name,attr,childs,textflag] = getNodeData(theNode)
    % Create structure of node info.

    %make sure name is allowed as structure name
    name = toCharArray(getNodeName(theNode))';
    name = strrep(name, '-', '_dash_');
    name = strrep(name, ':', '_colon_');
    name = strrep(name, '.', '_dot_');

    attr = parseAttributes(theNode);
    if (isempty(fieldnames(attr)))
        attr = [];
    end

    %parse child nodes
    [childs,text,textflag] = parseChildNodes(theNode);

    if (isempty(fieldnames(childs)) && isempty(fieldnames(text)))
        text.(textflag) = toCharArray(getTextContent(theNode))';
    end

end

% ----- Subfunction parseAttributes -----
function attributes = parseAttributes(theNode)
    % Create attributes structure.

    attributes = struct;
    if hasAttributes(theNode)
       theAttributes = getAttributes(theNode);
       numAttributes = getLength(theAttributes);

       for count = 1:numAttributes
            str = toCharArray(toString(item(theAttributes,count-1)))';
            k = strfind(str,'=');
            attr_name = str(1:(k(1)-1));
            attr_name = strrep(attr_name, '-', '_dash_');
            attr_name = strrep(attr_name, ':', '_colon_');
            attr_name = strrep(attr_name, '.', '_dot_');
            attributes.(attr_name) = str((k(1)+2):(end-1));
       end
    end
end
```

```
function [ configs ] = getConfiguration( name )
all_configs     = xml2struct('configurations.xml');
no_of_entries   = size(all_configs.Persons.Entry,2);

configs = NaN;

```

```
for i = 1:no_of_entries
    if strcmp(all_configs.Persons.Entry{i}.Name.Text, name)
       configs =  all_configs.Persons.Entry{i};
    end
end
end
```

```
function [ Age Sex Salary SalaryIncr ...
   MatchingContr AdditionContr MaxSalarySaved DCBal SavingsBal] = GetValues( configs )

% GET AGE
if isfield(configs, 'Age') == 1
    Age           = str2double(configs.Age.Text);
else
    Age           = 25;
end

% GET SEX: 1 = MALE ; 0 = FEMALE
if isfield(configs, 'Sex') == 1
    Sex           = strcmp(configs.Age.Text,'M');
else
    Sex           = 'M';
end

% GET SALARY
if isfield(configs, 'Salary') == 1
    Salary        = str2double(configs.Salary.Text);
else
    Salary        = 100000;
end

% GET EXPECTED SALARY INCREASE
if isfield(configs, 'SalaryIncr') == 1
    SalaryIncr    = str2double(configs.SalaryIncr.Text);
else
    SalaryIncr    = 0.01;
end

% GET MATRCHING CONTRIBUTION
if isfield(configs, 'MatchingContr') == 1
    MatchingContr = str2double(configs.MatchingContr.Text);
else
    MatchingContr = 0.04;
end

% GET ADDITIONAL CONTRIBUTION
if isfield(configs, 'Age') == 1
    AdditionContr = str2double(configs.AdditionContr.Text);
```

```
else
    AdditionContr = 0.06;
end

% GET MAX SALARY CONTRIBUTION
if isfield(configs, 'MaxSalarySaved') == 1
    MaxSalarySaved = str2double(configs.MaxSalarySaved.Text);
else
    MaxSalarySaved = 0.06;
end


% GET DC PLAN BALANCE
if isfield(configs, 'DCBal') == 1
    DCBal          = str2double(configs.DCBal.Text);
else
    DCBal          = 0;
end

% GET NON-PENSION SAVINGS
if isfield(configs, 'SavingsBal') == 1
    SavingsBal     = str2double(configs.SavingsBal.Text);
else
    SavingsBal     = 0;
end
end
```