Two-Stage Stochastic Programming for Asset-Liability Matching

Matthew Reiter[1] and Daniel Kecman[2]

[1] matthew.reiter@mail.utoronto.ca
[2] daniel.kecman@mail.utoronto.ca

University of Toronto, Division of Engineering Science

Abstract. This project implemented a two-stage stochastic linear programming model to solve an asset-matching problem. Six assets from the S&P500 were considered and data was collected over a two-year timespan from January 2014 to December 2015. The data-calibration period ranged from January-December 2014 whereby the returns and covariances of the holding stocks were estimated. From these calibration estimate, the stock returns were modeled from a multivariate normal and the stock prices were simulated over the investment period lasting from January-December 2015. The simulation implemented a Monte-Carlo method based on a Geometric Brownian motion which led to the formulation of five discrete scenarios. These scenarios led to a stochastic program that improved the performance of the underlying deterministic model.

1 Methodology

Stochastic Programming provides a framework for optimized decision-making under uncertainty. Specifically, the goal of a two-stage stochastic program is to formulate an optimization model that will yield a robust solution to a problem that involves two sets of decisions. The first set of decisioning involves 'first-stage' variables which are made initially and the second set of decisions refer to 'recourse-variables' which are made at some time in the future.

Deterministic and Stochastic Model: Consider the following linear program, for which we will expand using a stochastic model:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^T\mathbf{x} \\ st. \quad & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned}$$

We will call this the deterministic model and we apply it when there is no uncertainty. The approach for setting up the stochastic problem allows for discrete

scenarios to be generated and the optimization model to incorporate these scenarios. To understand how stochastic programming builds on the deterministic model, consider figure 1 which outlines a simplified two-staged scenario tree.

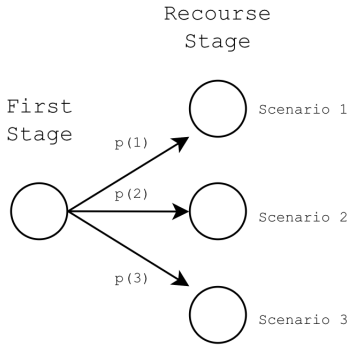

Fig. 1. A simplified scenario tree for a stochastic program.

For up to S scenarios, we modify the deterministic model in the following two ways

- determine the expected recourse function $\mathbb{E}Q(\mathbf{x}, \xi)$ and include this in the objective function of the optimization
- incorporate constraints for the recourse variables taking the form $W\mathbf{y} = \mathbf{h} - T\mathbf{x}$

With that, our stochastic model becomes

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^T\mathbf{x} + \mathbb{E}Q(\mathbf{x}, \xi) \\ st. \quad & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0 \\ & Q = \min_{\mathbf{y}}\{\mathbf{q}^T\mathbf{y} | W\mathbf{y} = \mathbf{h} - T\mathbf{x}, \mathbf{y} \geq 0\} \end{aligned}$$

This stochastic model can be simplified to a great expect by modeling the expected recourse function as

$$\mathbb{E}Q(\mathbf{x}, \xi) = \sum_{s=1}^{S} p(s)Q(\mathbf{x}, \xi_s) \tag{1}$$

Scenario Generation with Monte Carlo Simulations: Attention now is directed towards the generation of appropriate scenarios. This is an important part

of stochastic programming and efficient scenario generation makes the model more robust.

As will be developed in section 2, this project will solve an asset-liability matching problem. This means that the generated scenarios pertain to the returns of a portfolio of up to n assets and l liabilities. Assume that all assets and liabilities are random variables. In this section, we will address how to simulate scenarios for our stochastic program by capturing the uncertainty of these variables.

Simulating Asset Returns: We begin with the portfolio of correlated assets. Assume that their returns $\mathbf{r_t} \in \mathbb{R}^n$ at time t follow a multivariate normal distribution with mean $\mu \in \mathbb{R}^n$ and covariance $Q \in \mathbb{R}^{n \times n}$. This is to say that

$$\mathbf{r_t} \sim \mathcal{N}(\mu, Q)$$

By this assumption, it follows that our asset prices follow a Geometric Brownian Motion. From this, we can readily derive a stochastic equation governing the asset prices, starting first from the asset returns. For asset i with price P at time t, we have

$$r_t = \frac{S_t^{(i)} - S_{t-1}^{(i)}}{S_{t-1}^{(i)}} = \mu_i + \xi_t^{(i)} \tag{2}$$

Where ξ_i is a normal random variable which accounts for correlation between your holding assets. We will address shortly how this is determined. Under a continuous pricing model we have the following stochastic differential equation (SDE)

$$dS_t^{(i)} = \mu_i S_t^{(i)} dt + \sigma_i \xi_t^{(i)} \sqrt{dt} \tag{3}$$

Solving this SDE leads to the following equation

$$S_{t+1}^{(i)} = S_t^{(i)} e^{(\mu_i - \frac{1}{2}\sigma_i^2)dt + \sigma_i \xi_t^{(i)} \sqrt{dt}} \tag{4}$$

Now we will address where ξ_i comes from. Given our covariance matrix Q we can readily define our correlation matrix $\rho \in \mathbb{R}^{n \times n}$ such that

$$\rho = \begin{pmatrix} 1 & \rho_{12} & \dots & \rho_{1n} \\ \rho_{21} & 1 & \dots & \rho_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{n1} & \rho_{n2} & \dots & 1 \end{pmatrix} \tag{5}$$

where $\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}$. With that, we define $L \in \mathbb{R}^{n \times n}$ to be the lower Cholesky factorization of ρ and take $\xi \in \mathbb{R}^{n \times 1}$ to be defined as follows

$$\xi := L\varepsilon \tag{6}$$

Where $\varepsilon \in \mathbb{R}^{n\times 1} \sim \mathcal{N}(\mathbf{0}, I)$. It readily follows then that $\xi \sim \mathcal{N}(\mathbf{0}, LL^T)$.

Given the above framework for determining the price of an asset, a Monte Carlo simulation can be run by repeatably calculating the price of an asset as it follows different paths. Upon each iteration, the asset's price will be simulated from a different random path and given a sufficient number of simulations, parameters can be estimated as done with any other Monte Carlo method. The results for this simulation is shown in figure 2. Note that the prices estimated for scenarios were done over a yearly basis with daily time-steps.

Fig. 2. A mesh-plot showing six simulated prices of the holding assets.

Simulating Liabilities: For the sake of simplicity, the liabilities will be modeled from a normal distribution. This means to generate our scenarios, we sample up to S times from the following random variable

$$L_s \sim \mathcal{N}(\mathbb{E}L, \sigma_L)$$

The specification for these parameters will be outlined in section 2.

Value of Stochastic Solution: The value of the stochastic solution (VSS) will be the metric that tells us how good our stochastic model was. VSS is calculated by taking the difference of the optimal solution under the deterministic model and the optimal solution under the stochastic model. For a meaningful comparison, the deterministic model is evaluated under the most-likely scenario. This is calculated as

$$\text{VSS} := \mathbb{E}f(\mathbf{x}(\bar{\xi}), \xi) - \mathbb{E}f(\mathbf{x}, \xi) \tag{7}$$

What the VSS amounts the difference between the deterministic model under an expected case and the stochastic model which incorporates all scenarios. Therefore, a larger VSS indicates that the the stochastic model is more robust to uncertainty and a smaller VSS means that the uncertainty of the scenarios do not contribute greatly to the optimal solution.

With the sufficient background, we now introduce the asset-liability matching problem which the two-stage stochastic model is geared towards.

2 Problem Formulation

Suppose you are a student facing tuition payments. For the previous year, tuition was. For the previous school year, tuition was $17,000 and you expect that your tuition will increase by $500 $\pm$200. Therefore, are looking to invest $17,500 to invest in up to six stocks from the S&P500. Your goal is to invest in order to cover the cost of your tuition. The stocks you can select are from different asset classes and are presented in table 1.

You are confident that the random fluctuation in the tuition is captured by a normally distributed variable, so you model your tuition under scenario s as $T_s = 17000 + \varepsilon_s$ where $\varepsilon_s \sim \mathcal{N}(500, 200)$.

Table 1. Six Stocks from the S&P500 tracked

S&P500 Sector	**Company Tickers**		
Consumer Discretionary	F		
Financials	C	JPM	WFC
Information Technology	AAPL	IBM	

We will be considering $S = 5$ scenarios, all which are equally likely at $p = 1/5$. To simulate the asset returns, we will follow the methodology from section 1 and model the stock prices as a Geometric Brownian Motion based on normally distributed rates of returns. Please refer to section 5 for the code that generated

the scenario values from the Monte Carlo simulation.

Given the number of scenarios, we develop a two-stage stochastic program which optimizes our investment strategy. We summarize our decision variables in the following section.

Decision Variables: We will have 16 decision variables in total, which are specifically

- x_i for $i = 1 \dots 6$ which will represent the dollar amount of wealth to invest in asset i at the beginning. These are our first-stage decision-variables.
- $x_s^{(+)}$ for $s = 1 \dots 5$ which represents the surplus wealth under scenario s. This is beneficial and therefore comes with a reward of \$1 per amount in surplus, which incentives this outcome.
- $x_s^{(-)}$ for $s = 1 \dots 5$ which represents the shortfall wealth under scenario s. This is not desired as it threatens your ability to make a tuition payment, so this outcome comes with a penalty of \$2 per amount in shortfall. This discourages running a shortfall.

Optimization Parameters: Our constraints will come directly from our budgetary requirements as well as our ability to meet our fluctuating tuition requirements under each scenario. See table 2 for all of the scenarios and expected results. With that, we have the following:

- $\mu_{i,s}$ is the simulated rate of return for asset i under scenario s.
- T_s is the randomly generated tuition amount under scenario s.

Table 2. Simulated Scenario Results

	s = 1	**s = 2**	**s = 3**	**s = 4**	**s = 5**	**expected**
μ_F	-0.009	0.002	-0.003	0.008	-0.018	0.022
μ_C	0.010	-0.001	0.001	0.025	-0.012	0.017
μ_{WFC}	-0.001	0.004	-0.012	0.022	0.003	0.215
μ_{JPM}	-0.003	-0.004	-0.017	0.014	-0.001	0.09
μ_{AAPL}	-0.011	0.014	0.008	0.003	0.005	0.369
μ_{IBM}	-0.008	0.018	0.017	-0.004	-0.011	-0.117
Tuition (\$)	17,388	17,675	17,445	17,780	17,658	17,500

Two-Stage Stochastic Model: Given the above information, we present the following two-stage stochastic linear program to solve the asset-liability matching problem:

$$\begin{aligned}
\min_{\mathbf{x},\mathbf{x}^{(\pm)}} \quad & \sum_{i=1}^{6} x_i + \frac{1}{5}\sum_{s=1}^{5}\left(2x_s^{(-)} - x_s^{(+)}\right) \\
st. \quad & \sum_{i=1}^{6} x_i \leq 17500 \\
& -x_s^{(+)} + x_s^{(-)} + \sum_{i=1}^{6}(1+\mu_{i,s})x_i = T_s \quad \forall s = 1\ldots 5 \\
& \mathbf{x} \geq 0, \mathbf{x}^{(\pm)} \geq 0
\end{aligned}$$

We are interested in comparing in calculating the VSS, so for this we need to use the deterministic model under the mostly likely scenario in order to determine our optimal recourse variables. For this purpose, we treat the most likely tuition liability to be \$17,000 which is its expected value and take the expected returns to be the geometric average of the yearly historical prices.

Which leads us to the deterministic and recourse optimization models.

Deterministic Model: Given the above expected returns, which we shall refer to below as $\mathbf{r} \in \mathbb{R}^{n\times n}$, we have

$$\begin{aligned}
\min_{\mathbf{x},\mathbf{x}^{(\pm)}} \quad & \sum_{i=1}^{6} x_i + 2x^{(-)} - x^{(+)} \\
st. \quad & \sum_{i=1}^{6} x_i \leq 17500 \\
& -x^{(+)} + x^{(-)} + \sum_{i=1}^{6}(1+r_i)x_i = 17500 \\
& \mathbf{x} \geq 0, \mathbf{x}^{(\pm)} \geq 0
\end{aligned}$$

Recourse Model: Given the first-stage values from the optimal solution of the deterministic model, we solve the for the optimal recourse values. We shall call the optimal first-stage variables from the deterministic model $\mathbf{x}^{(\text{det})}$.

$$\begin{aligned}
\min_{\mathbf{x}^{(\pm)}} \quad & \frac{1}{5}\sum_{s=1}^{5}\left(2x_s^{(-)} - x_s^{(+)}\right) \\
st. \quad & -x_s^{(+)} + x_s^{(-)} = T_s - \sum_{i=1}^{6}(1+\mu_{i,s})x_i^{(\text{det})} \quad \forall s = 1\ldots 5 \\
& \mathbf{x}^{(\pm)} \geq 0
\end{aligned}$$

VSS: To find the VSS, we take the difference between the optimal value of the recourse and deterministic model with the stochastic model. Note that our deterministic model solves for the optimal wealth investment in the stocks and the recourse model solves for the optimal spread of surplus and shortfall wealth.

3 Results

We present our results in table 3. Note that in every case, our optimization models look to maximize the wealth allocation in the assets with the maximum return. This is to be expected since there were no cardinality constraints imposed to enforce diversification.

Under the stochastic model, in scenario's two, four and five the asset returns failed to provide an adequate return and a small shortfall was required in order to meet the tuition payment. Under the recourse model with the deterministic first-stage decisions, a shortfall was forced on the strategy in scenarios one, four and five.

The deterministic model outperformed the stochastic model considerably. This was due to the fact that the stock 'AAPL' has a massive yearly expected return observed in the calibration period. It is to be expected then that the deterministic model would run a large surplus. However it is interesting to note that despite the strength in the deterministic model, the generated scenarios predicting the stock prices over the investment horizon meant that the recourse model underperformed the stochastic model.

The VSS was calculated to be \$45.21 which indicates that the two-stage stochastic program was marginally more robust than the deterministic model. Under perfect information and assuming that the expected returns will be realized, the far superior investment strategy is the pure deterministic model (without then optimizing the recourse with scenarios).

As an interesting exercise, we can track the value of the portfolio recommended by the stochastic model. The results from this tracking is shown in figure 3. Recall that the investment period for this project was from January to December 2015, and so the weekly portfolio values was tracked. Despite a strong performance throughout the secnd and third quarters of 2015, the value of the portfolio by the end of the investment period dropped down to approximately where it started at the beginning. This means that the value of the stochastic program predicted a result that was very close to what was realized.

Table 3. Optimization Results

	stochastic	**deterministic**	**recourse**
F (x_1)	0	0	0
C (x_2)	0	0	0
WFC (x_3)	9,826	0	0
JPM (x_4)	0	0	0
AAPL (x_5)	7,674	17,500	17,500
IBM (x_6)	0	0	0
Surplus $x_1^{(+)}$	17.33	-	0
Shortfall $x_1^{(-)}$	0	-	87.28
Surplus $x_2^{(+)}$	0	-	70.00
Shortfall $x_2^{(-)}$	25.77	-	0
Surplus $x_3^{(+)}$	0	-	193.60
Shortfall $x_3^{(-)}$	0	-	0
Surplus $x_4^{(+)}$	0	-	0
Shortfall $x_4^{(-)}$	35.85	-	228.24
Surplus $x_5^{(+)}$	0	-	0
Shortfall $x_5^{(-)}$	86.98	-	68.75
Expected Surplus $\bar{x}^{(+)}$	-	6,457	-
Expected Shortfall $\bar{x}^{(-)}$	-	0	-
Optimal Value	17,560	10,426	17,601

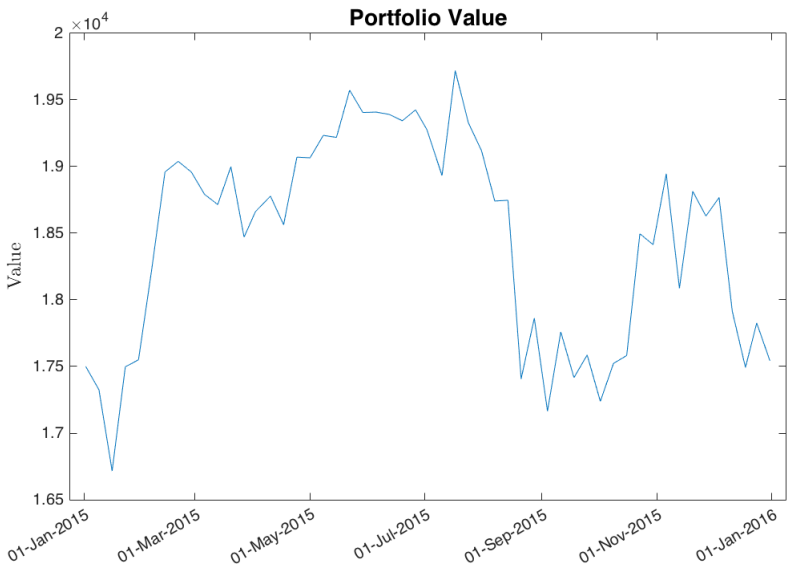

Fig. 3. The value of the portfolio over the investment period.

4 Next Steps and Improvements

More Assets and Scenarios: For the sake of simplicity, this project focused on a small basket of assets and a small number of scenarios. Both the number of considered assets and the scenarios generated can be ramped up in order to provide the optimization method with a robust option. Increasing the number of assets provides the investor with a greater degree of choice to select an optimal investment portfolio over a certain investment period. Increasing the number of scenarios improves the quality of decision-making under uncertainty by effectively increasing the sample size of our Monte Carlo simulation. Ideally, such simulation methods should be run *thousands* of times in order to best capture the true variability of the underlying probability distribution. This is a recognized shortcoming of the results presented in this project.

Probabilities for Scenarios:

This project considered all scenarios generated to be equally likely. This is a naive approach and can be expanded upon by empirically determining the corresponding likelihood of each scenario. The direct benefit that this would have is improving the performance of expected value taken in the objective of the stochastic model.

5 MATLAB Code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% the purpose of this project is to implement a two-stage stochastic linear
% program which solves a tuition investment-matching strategy.

% authors: Matthew Reiter and Daniel Kecman
% date: april 11, 2018

clc
clear all
format long

% note that this project is compatible with MATLAB 2015.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

1. read input files

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% load the stock weekly prices and factors weekly returns
data = readtable('price_data.csv');
data.Properties.RowNames = cellstr(datetime(data.Date));
data = data(:,2:size(data,2));

% n represents the number of stocks that we have
n = size(data,2);

% identify the tickers and the dates
tickers = data.Properties.VariableNames';
dates = datetime(data.Properties.RowNames);

% calculate the stocks' yearly returns
prices  = table2array(data);
returns = (prices(2:end,:) - prices(1:end-1,:)) ./ prices(1:end-1,:);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

2. estimation of mean and variance

```
- we use historial data spanning a year from 2014-01-03 to 2014-12-26
- we use the geometric mean for stock returns and from this we formulate
the covariance matrix
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% calibration start data and end data
cal_start = datetime('2014-01-03');
cal_end = cal_start + calmonths(12) - days(2);

cal_returns = returns(cal_start <= dates & dates <= cal_end,:);
current_prices = table2array(data((cal_end - days(7)) <= dates & dates <= cal_end,:))';

% calculate the geometric mean of the returns of all assets
mu = 52*(geomean(cal_returns+1)-1)'
cov = 52*cov(cal_returns);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
mu =

   0.022234543556377
   0.016788212229012
   0.215424745393144
   0.089668871728315
   0.368991665804218
  -0.117177375020920
```

3. scenario generation

```
- we use two methods to generate scenarios, seperate for the scenario's
governing the asset returns and for the matched liabilities
- for asset returns, we model the stock price as a Geometric Brownian
Motion and do a Monte Carlo simulation to estimate stock returns for
our investment period spanning a year from 2015-01-02 to 2015-12-31
- for our liabilties, we sample from a normal distribution

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% we need the correlation matrix to simulate the correlated prices of
% portfolio
rho = corrcov(cov);

% we take the cholesky factorization of the correlation matrix
L = chol(rho, 'lower');

% define the number of randomized scenarios to sample for
S = 5;
```

```
% our simulated asset prices and returns
sim_price = zeros(n,S);
sim_returns = zeros(n,S);

% our scenario liabilities
sim_liabilities = zeros(S,1);

% we have yearly estimates for returns and we wish to simulate the
% price path after six months using monthly time-steps
dt = 1/252;

% setting the random seed so that our random draws are consistent across testing
rng(1);

tuition = 17000;

for i=1:S

    % our random correlated pertubations
    epsilon = L * normrnd(0,1,[n,1]);

    % randomize our liabilities
    sim_liabilities(i) = tuition + normrnd(500,200);

    % calculate our simulated prices
    sim_price(:,i) = current_prices .* exp((mu - 0.5 * diag(cov))*dt + sqrt(dt)*sqrt(diag(co

    % calculate our simulated returns
    sim_returns(:,i) = (sim_price(:,i) - current_prices) ./ current_prices;
end

sim_returns
sim_liabilities
mu

X = 1:n;
Y = 1:S;
mesh(sim_price);
title('Simulated Prices of Holding Assets', 'FontSize', 14)
ylabel('Asset','interpreter','latex','FontSize',12);
xlabel('Scenario','interpreter','latex','FontSize',12);
zlabel('Asset Price','interpreter','latex','FontSize',12);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

sim_returns =
```

```
  Columns 1 through 3

  -0.008810722158627   0.002429376891445  -0.003308027179416
   0.010314390970307  -0.001314447845343   0.000519451471810
  -0.000725330053245   0.004263153947304  -0.011740993590965
  -0.003484681445156  -0.003203387822546  -0.016884886520004
  -0.011372618581591   0.014010167086939   0.007861478445295
  -0.008396819040902  -0.017296638107479   0.016879572650751

  Columns 4 through 5

   0.008256261133632  -0.018472327969247
   0.025325940962147  -0.011701120780678
   0.022563041295995   0.003226072673106
   0.014146459532548  -0.001156991816241
   0.002982246006717   0.005081971009582
  -0.004487692711432  -0.011693004137588


sim_liabilities =

   1.0e+04 *

   1.738826384710521
   1.767517482956691
   1.744496855186486
   1.778043245726756
   1.765768184324549


mu =

   0.022234543556377
   0.016788212229012
   0.215424745393144
   0.089668871728315
   0.368991665804218
  -0.117177375020920
```

4. stochastic optimization

```
- implementation of the above scenarios, incorporating second stage and
first stage constraints
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% take uniform probability for each scenario
p = 1/S;

% we have an initial budget of 17,500
B = 17500;

% the benefit of running a surplus will be a 1 and the cost of running a
% shortfall will be -2
surplus = -1;
shortfall = 2;

% formulate our objective function
f = [ones(1,n) repmat(p*[surplus shortfall], 1, S)]';

% dealing with our first-stage constraints
A = [ones(1,n) zeros(1,2*S)];
b = B;

% handling our second stage constraints
temp = [-1 1];
temp_r = repmat(temp, 1, S);
temp_c = mat2cell(temp_r, size(temp,1), repmat(size(temp,2),1,S));
con = blkdiag(temp_c{:});

Aeq = [(sim_returns+1)' con];
beq = sim_liabilities;

lb = zeros(n+2*S,1);
ub = [];

[stochastic, sto_value] = linprog(f, A, b, Aeq, beq, lb, ub)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Optimization terminated.

stochastic =

   1.0e+03 *

   0.000000000003458
   0.000000000162112
   9.825664873736784
   0.000000000004996
   7.674335126090030
```

```
   0.000000000002647
   0.017332016614268
   0.000000000000315
   0.000000000000260
   0.025767790178600
   0.000000000001534
   0.000000000000410
   0.000000000000255
   0.035848819672729
   0.000000000000254
   0.086982785674876


sto_value =

     1.755597335488748e+04
```

5. value of the stochastic solution

```
- based on the expected values of the returns and the liability

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% first we determine the value of our deterministic problem considering
% only the expected returns and liabilities

% formulate our objective function
f = [ones(1,n) surplus shortfall]';

% dealing with our first-stage constraints
A = [ones(1,n) 0 0];
b = B;

Aeq = [(mu+1)' -1 1];
beq = tuition+500;

lb = zeros(n+2,1);
ub = [];

[deterministic, det_value] = linprog(f, A, b, Aeq, beq, lb, ub)

% now we solve the stochastic problem with the first stage variables as
% determined from our deterministic model

% formulate our objective function
```

```
f = repmat(p*[surplus shortfall], 1, S)';

% no first stage constraints need to be optimized since we are using the
% first stage constraints from the deterministics model
A = [];
b = [];

% constraints tied only to our recourse variables
temp = [-1 1];
temp_r = repmat(temp, 1, S);
temp_c = mat2cell(temp_r, size(temp,1), repmat(size(temp,2),1,S));
Aeq = blkdiag(temp_c{:});

% we update what the required liabilties must be with the given first stage
% variables set
beq = sim_liabilities - (sim_returns+1)'*deterministic(1:n);

lb = zeros(2*S,1);
ub = [];

[recourse, recourse_value] = linprog(f, A, b, Aeq, beq, lb, ub)
vss = recourse_value + sum(deterministic(1:n)) - sto_value

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Optimization terminated.

deterministic =

   1.0e+04 *

   0.000000000375794
   0.000000000369828
   0.000000000948572
   0.000000000471212
   1.749999997491551
   0.000000000267190
   0.645735414552603
   0.000000000166827


det_value =

     1.104264585705196e+04

Optimization terminated.
```

```
recourse =

   1.0e+02 *

   0.000000004489412
   0.872846732459476
   0.700030935865708
   0.000000002586185
   1.926073200688687
   0.000000002531623
   0.000000001955387
   2.282431527855161
   0.000000005460195
   0.687473521286146


recourse_value =

     1.011879884995559e+02


vss =

  45.214632853530929
```

6. tracking performance of stochastic model

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% testing period start date and end date
test_start = cal_end + days(1);
test_end = test_start + calmonths(12) - days(2);

% subset the prices corresponding to the current out-of-sample test period.
period_prices = table2array(data(test_start <= dates & dates <= test_end,:));

% the current value of our stock portfolio is the sum of all the wealth we
% allocate to each stock
current_value = sum(stochastic(1:n));

% get the prices of our assets at the beginning of our tracked period
current_prices = table2array(data((test_start) <= dates & dates <= test_start,:))';

% calculate the number of shares of each stock to purchase
```

```
shares = stochastic(1:n) ./ current_prices;

% weekly portfolio value during the out-of-sample window
portfolio_value = period_prices * shares

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```