Robert Pixton
9 December 2010
ISYS 520

Final Project: Mutual Fund Manager

Executive Summary

My project is based on an assignment given to students in the Financial Planning class (BUSM 418) at Brigham Young University. In this class we were given the task of determining our desired individual portfolio. This task was extremely time-consuming and so I decided to create a tool that will make this process easier and faster. My tool accomplishes three major objectives:

1. It automatically checks to see if a mutual fund's asset allocation is weighted toward stocks or bonds.
2. It performs an assessment to determine the user's risk tolerance.
3. It uses solver to create an optimized mutual fund portfolio based on asset allocation constraints entered by the user.

Introduction

Brigham Young University offers a class on financial planning (BUSM 418) taught by Professor Bryan Sudweeks. This course covers many areas of personal finance including investments. One assignment we had in this class was to create a personal investment plan that among other things outlined the mutual fund portfolio we will create when we have the financial means to do so. I found this assignment extremely worthwhile, but very time consuming. Researching and choosing the mutual funds was easy enough, but then planning out the desired weights and asset allocation of the funds proved to be more difficult.

For my final project I developed a tool called the Mutual Fund Manager. I wanted this tool to be able to accomplish three objectives:

1. I wanted it to sort between equity and fixed income based mutual funds. In BUSM 418, our teacher outlined a framework to be used as a general allocation "rule-of-thumb." He told us that as a general rule, a person should have an allocation of bonds equal to their age with the remainder in stocks. In order to use this framework, I decided it would be nice if my tool would automatically determine if a mutual fund was allocated more heavily toward stocks or bonds.

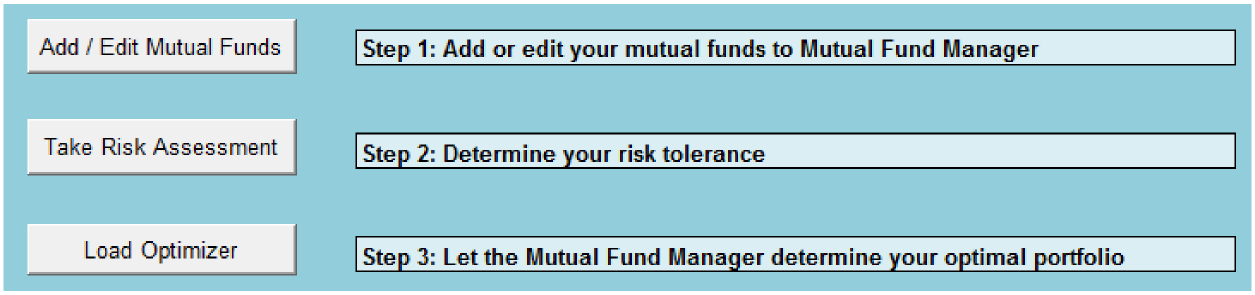
2. I wanted my tool to perform a risk assessment quiz to determine the user's risk tolerance in order to make adjustments to the "rule-of-thumb" outlined above. The results of this quiz will tell the user what his or her allocation should be between stock and bond based mutual funds.

3. I wanted my tool to allow the user to input minimum and maximum constraints regarding allocation between different asset classes and then run solver to find the optimal weights for each mutual fund in order to satisfy the objective of the user. This objective would be to either maximize portfolio return or minimize portfolio risk.

The tool I built accomplishes my objectives. The Mutual Fund Manger tool was a lot of work that gave me a useful tool that I can use to develop my portfolio. Furthermore, it also gave me the opportunity to practice using VBA and apply many of the things I have learned in ISYS 520.

Project Implementation

When beginning my project, I spent some time physically drawing out what I wanted the tool to look like. This helped me visualize how each component of my tool would help solve my objectives. I wanted a layout that would be easy for anyone to use and understand. The tool opens to a worksheet that displays the following buttons:

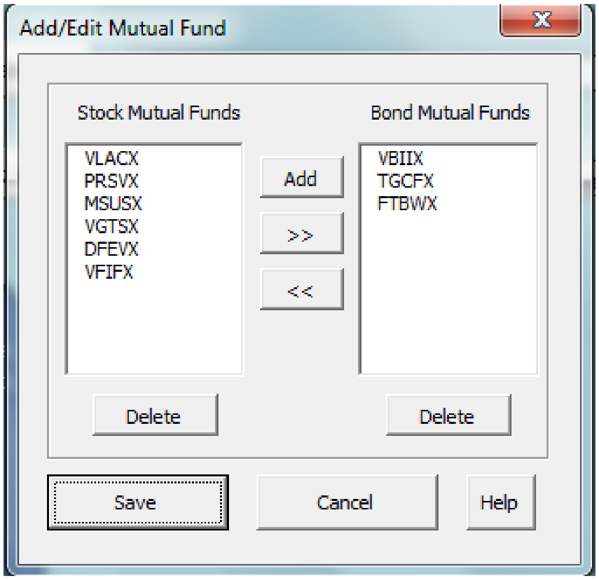


Step 1: Add or edit your mutual funds to Mutual Fund Manager

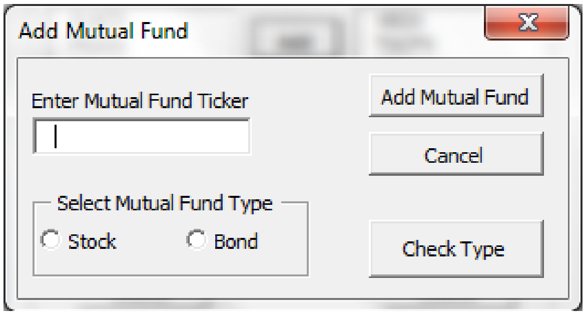
Step 2: Determine your risk tolerance

Step 3: Let the Mutual Fund Manager determine your optimal portfolio

The user can easily see the order in which to interact with the tool. First they are able to add new mutual funds to the portfolio or edit existing mutual funds. When the user presses the “Add / Edit Mutual Funds” button, it opens the Add / Edit form (pictured below).



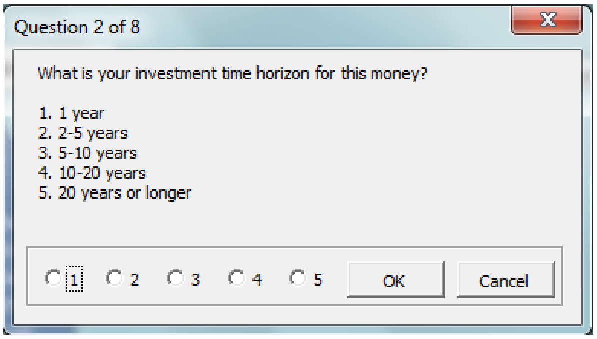
This form shows the user the current mutual funds added to the portfolio. The form provides a list of mutual funds that are primarily allocated toward stocks and another list that shows funds primarily allocated toward bonds. The user can select one or multiple funds from a list and either move them to the other list or delete them. If a mutual fund is listed under the incorrect list, the user can move them by selecting the fund(s) and hitting the arrow button pointing toward the correct list. The ">>" button moves the selected fund(s) from the stock list to the bond list and the "<<" button does the opposite. If the user needs help, the "Help" button offers some basic guidance. The user can add a new fund by clicking on the "Add" button. This will open the Add form (pictured below).



The user can type in the ticker of the mutual fund he or she would like to add and then they can select the mutual fund type which lets the tool know which list to add the input to. If the user doesn't know the type, he or she can hit the "Check Type" button. This launches a subprocedure in VBA that performs a web query. This query finds the fund on Google Finance and then performs a calculation to determine whether the majority of the fund is allocated toward stocks or bonds. It then selects the correct option on the form. This form also has some basic checks to make sure the user enters a valid ticker. Invalid tickers will result in an error message when the "Check Type" and/or "Add Mutual Fund" buttons are pressed. When the user has the information they want, they can select the "Add Mutual Fund" button to add the fund to the previous form. When the user has added and/or edited his or her funds, they can select the "Save" button to save the funds to the Mutual Fund worksheet. This concludes the first step.

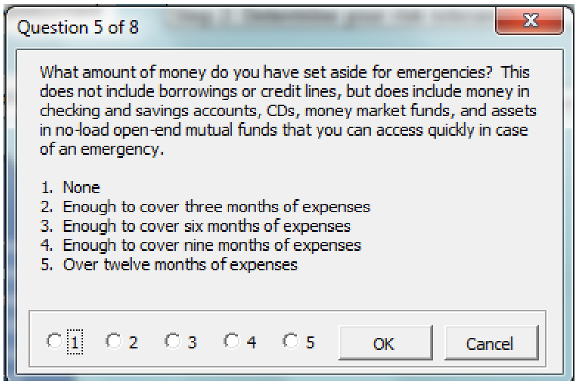
The reason I decided to split the mutual funds up by stock and bond allocation is because it coincides with a handy framework taught in BUSM 418. Professor Sudweeks taught us that your portfolio asset allocation should consist of a percentage of bonds equal to your age and the remainder should be allocated into stocks. Sorting the funds by type makes it easy to see and visualize your portfolio to make sure you are in sync with this rule. For example, it will be obvious if a user is out of balance if they have a long list of bond mutual funds when they should be weighting more toward stock mutual funds.

Professor Sudweeks rule is pretty handy as a rule of thumb, but he also taught us that this general rule should be modified based on the individual's risk tolerance. Therefore, for my second objective, I wanted my tool to determine risk tolerance. I received permission from Professor Sudweeks to use a risk assessment quiz that he wrote to help me determine the user's risk tolerance. When the user begins step two by clicking the "Take Risk Assessment" button, it launches the quiz (pictured below).



What is your investment time horizon for this money?

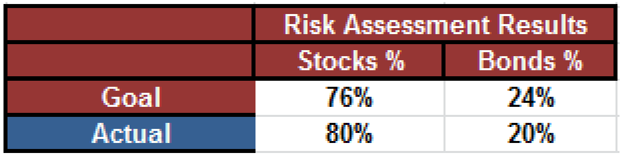
1. 1 year
2. 2-5 years
3. 5-10 years
4. 10-20 years
5. 20 years or longer



What amount of money do you have set aside for emergencies? This does not include borrowings or credit lines, but does include money in checking and savings accounts, CDs, money market funds, and assets in no-load open-end mutual funds that you can access quickly in case of an emergency.

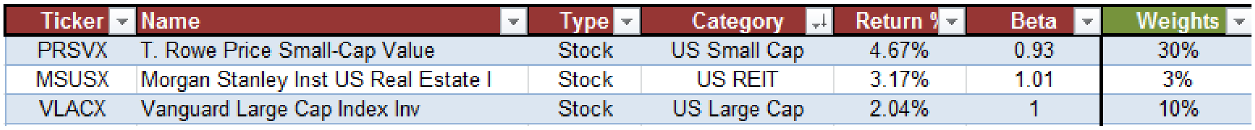
1. None
2. Enough to cover three months of expenses
3. Enough to cover six months of expenses
4. Enough to cover nine months of expenses
5. Over twelve months of expenses

The quiz presents eight questions and asks the user to pick a numbered answer one through five. If the user answers by selecting answer one, they get one point and if they select answer two they get two points, etc. Their scores are added up at the end of the quiz and are used to determine whether the user is a very conservative, conservative, moderate, aggressive, or very aggressive investor. Their optimum allocation is then determined by taking the general "rule-of-thumb" allocation, and then modifying it. If the user is a very conservative investor, it shifts their bond allocation to their age plus 20 with the remainder being stock. So if a 30 year old person was labeled as being very conservative, they would allocate their portfolio 50% (their age of 30 + 20 = 50%) into bonds and the rest in stock. If they are a conservative investor, it adds 10 to their age. Moderate investors add 0 to their age, aggressive investors subtract 10 from their age and very aggressive investors subtract 20 from their age. The results of this test are shown as the Risk Assessment Results Goal on the Analysis worksheet (pictured left). This concludes the second step.



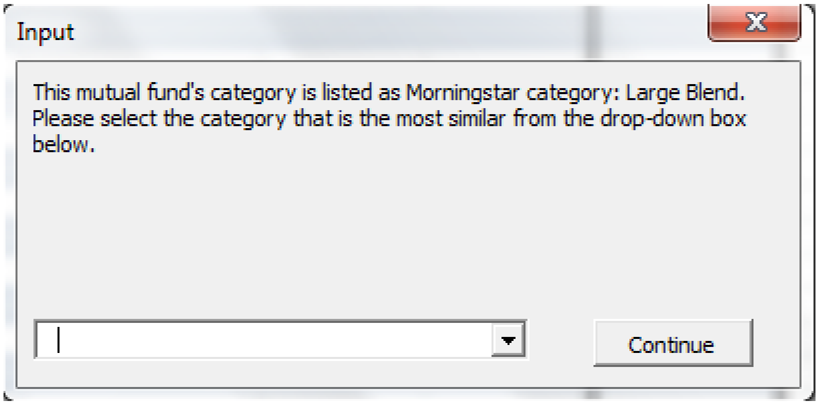
	Risk Assessment Results	
	Stocks %	Bonds %
Goal	76%	24%
Actual	80%	20%

At this point the user has entered their mutual funds and determined their risk tolerance which then recommended an allocation. The third step is to optimize their portfolio. This step begins when the user selects the "Load Optimizer" button. This begins a subprocedure that copies the mutual funds to a table on the Analysis worksheet. It then runs a web query and gathers the name, return %, and beta. It populates this data into the table for each mutual fund. It also lists the mutual fund type (stock or bond) and category (which is a broader list of asset classes).



Ticker	Name	Type	Category	Return %	Beta	Weights
PRSVX	T. Rowe Price Small-Cap Value	Stock	US Small Cap	4.67%	0.93	30%
MSUSX	Morgan Stanley Inst US Real Estate I	Stock	US REIT	3.17%	1.01	3%
VLACX	Vanguard Large Cap Index Inv	Stock	US Large Cap	2.04%	1	10%

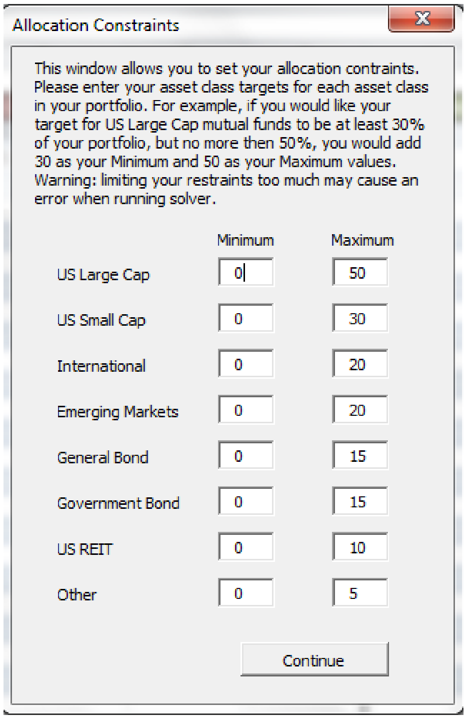
The category column is one item that requires input from the user. My goal was to allow the user to set asset allocation constraints for each separate asset class, but there are too many different types to do this within reason. Therefore, my solution was to have a form appear for each mutual fund that shows what the Morningstar category is and it then asks the user to select the closest asset class from a drop down box (pictured below). This allowed me to reduce the asset classes to eight which I felt hit all the major asset classes. The user can choose between Large Cap stock, Small Cap stock, International stock, emerging markets stock, US REIT, General Bonds (includes corporate bonds), Government bonds (includes all government bonds and T-bills), and an Other category for funds that may not quite fit in any of these categories.



Input

This mutual fund's category is listed as Morningstar category: Large Blend. Please select the category that is the most similar from the drop-down box below.

The subprocedure continues until the table is completely filled out for all mutual funds in the portfolio. Finally, it pulls up a form that asks the user to set their allocation constraints for each of the eight asset classes. The user can set a minimum and maximum limit that will help solver later create a portfolio that contains the allocation the user desires (form pictured below).



Allocation Constraints

This window allows you to set your allocation contraints. Please enter your asset class targets for each asset class in your portfolio. For example, if you would like your target for US Large Cap mutual funds to be at least 30% of your portfolio, but no more then 50%, you would add 30 as your Minimum and 50 as your Maximum values. Warning: limiting your restraints too much may cause an error when running solver.

	Minimum	Maximum
US Large Cap	0	50
US Small Cap	0	30
International	0	20
Emerging Markets	0	20
General Bond	0	15
Government Bond	0	15
US REIT	0	10
Other	0	5

When they are finished setting their allocation constraints, the tool then lets the user check over the information to make sure it looks correct. When they are finished checking, they can hit the “Optimize” button to run solver. When they hit “Optimize” a form appears that asks the user to select their primary objective in determining their optimum portfolio. They can maximize returns which causes solver to weight each mutual fund in a way that maximizes the weighted average return of the portfolio while obeying the asset class allocation constraints set by the user. Alternatively, the user can choose to minimize risk which then asks solver to minimize the beta of the portfolio by adjusting the weights of the mutual funds in the portfolio. If for any reason the user has a question, they can select the “Help” button to get some basic guidance. One noteworthy help item is that the solver add-in must be installed and the solver reference must be enabled in the VBA editor.

After solver has run, the tool displays the weights required to achieve the selected objective (maximize return or minimize risk) for each of the mutual funds which is what I wanted. This shows what proportion each mutual fund in the user’s portfolio should be. This concludes the third and final step.

In conclusion, my tool successfully completes the three objectives I set out to do. In the end, the user has a list of mutual funds shown in a nice table with the weight of each fund that they need to optimize their portfolio. They can see their recommended allocation between stock and bond based funds taken from the risk tolerance quiz. They can also see how close they are to this recommendation. If they are not happy with their current allocation, they can change their allocation constraints and run solver again to update the portfolio until their actual allocation is in line with their goal. I built this tool for myself to help me optimize my personal portfolio, but I believe that this tool can help anyone who wants to manager their mutual fund portfolio as well.

Learning and Conceptual Difficulties

Building the Mutual Fund Manager tool gave me a great opportunity to learn and practice the VBA language. There are many things that I learned in class that I applied to this project. I was able to learn and relearn how to build and use forms, tables, web queries, if statements, loops, error handling techniques, “bulletproofing” techniques, and how to manipulate solver through VBA. I feel like my biggest accomplishment was actually building something that didn’t exist before. Prior to taking ISYS 520, I had never really used VBA before. It was great to be able to take a concept that I had and make an actual product out of it. Perhaps no one else will ever use this tool, but I will and building this tool gave me valuable experience that I can use the next time I want to make something using VBA.

The process of building this tool was not easy and took a substantial amount of time. However, the time spent was certainly worth it. One of the biggest challenges I faced was getting Excel to do what I wanted it to do. VBA is a language that I am nowhere near fluent in. I knew what I wanted to say with my code, but a lot of times I didn't know how to say it. It took a lot of searching the internet and reading my textbook to figure out the commands I needed to enter to get things to work correctly. Every time I learned something new, I tried to practice it a few times to reinforce it into my memory. Hopeful someday, with enough practice, I will be more proficient at this. Regardless, all the searching helped me learn that the answers to most VBA questions are out there if you get stuck; you just have to know where to look.

I spent a lot of time trying to "bulletproof" my program to reduce the amount of errors that could possibly occur. For example, the Add Mutual Fund form only accepts a valid mutual fund ticker that is 5 characters long, ends in "X", and has all letter characters. Furthermore, the web query attempts to find the 5 year return for a mutual fund. However, if the fund is newer and doesn't have five years of history, it will then return the 3 year return instead and finally the one year return if necessary.

One of the most time consuming and difficult parts of my project was error trapping anticipated run-time errors. I set up error handling tags that allows the program to continue running in the event that certain run-time errors occur. For example, if the web query gets the information for a mutual fund that doesn't list the fund's return, the code will have an error for run-time error 91 because the search doesn't find the text it is looking for on the page. This interrupts the tool and is very annoying. To avoid this from happing, I set up an "On Error" command that tells the code to instead ask the user to manually enter the funds return and then it continues running. The problem I had was if there were two or more run-time errors in a row, the error handling procedures stopped working the way I expected them to. After hours of searching for a solution, I decided to stop by Professor Allen's office hours and he showed me a clever way of getting around this by using if statements. In the end this solved the problem and taught me a new way to go about error handling.

In the end I gained valuable experience and built a tool that does what I set out to do. Some of the solutions I came up with may not be the most eloquent but they work. In the future as I learn more and become more familiar with the VBA language, I hope to write code that is more advanced and efficient. An example of my programming shortcomings is that I had to use a hidden worksheet to store the score values from the risk assessment quiz. There is probably a better way to do this and in the future I will strive to learn these better ways.

In conclusion, I hope that this tool will not only prove valuable to others, but that it will demonstrate some of the things that I have learned in ISYS 520. Taking a class on VBA has been one of the best decisions I have made in my academic career and I now know that I can create tools to help me both in my personal life, but also in my professional life going forward. These skills will certainly prove valuable in whatever career I choose.