MBA 614 – Final Project

Mutual Fund Analyzer

Everett Crane

11

Executive Summary

Description

Before returning to school for further education, I participated in my employers 401K plan, which offered several dozen different funds to choose from. I often found myself frustrated with the seemingly endless list of funds that I had to choose from. A common measure of overall mutual fund performance is to compare a particular mutual fund's return with the S&P 500 index. With several thousand mutual funds to choose from, it can be a daunting task to go through each fund individually and determine whether or not the fund is right for you. As an investor, it is important to balance cost, risk and performance when choosing a mutual fund.

System Overview

To help assist me in this decision, I chose to build a model that will pull in mutual fund information based off of the user's input of mutual fund ticker symbols. Once the data is pulled, the user can then customize his portfolio by selecting the funds that meet his criteria along with a desired investment amount. After the funds are selected, the user can then choose to maximize the portfolio's return or minimize the portfolio's risk simply by using the built in user forms.
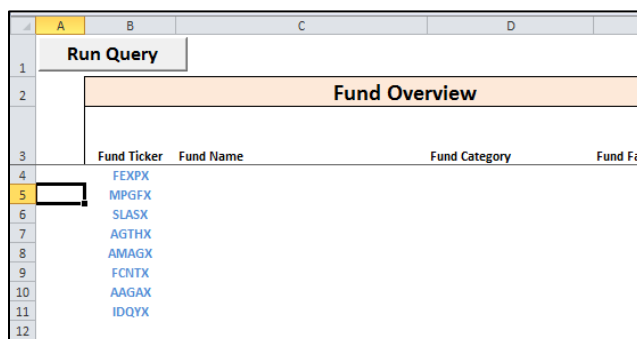
Implementation documentation

To implement the functionality of my mutual fund model, I divided the VBA programing task into three overall main parts:

1) Data Retrieval
2) User Form Functionality
3) Solver Capabilities

Data Retrieval

Once a user opens the Excel file, the user first begins by entering in mutual fund tickers below the "Fund Ticker" heading in column B (**See Figure 1**). The ticker symbols are displayed in blue font, identifying input information. Black fonts represent data that are not to be changed.

Figure 1 – Mutual Fund Ticker Input

Once the user has inputted the desired amount of tickers, the "Run Query" button is then pushed. The assigned VBA code is then initiated which automatically retrieves the data from Yahoo! Finance based off of the corresponding ticker symbol (**See Figure 2**).

Figure 2 – Data Querying

		Fund Overview				Investment Information				Fees			
							Min Initial	Min	Min	Annual Report	Prospectus	Prospectus	
		Fund				Min Initial	Investment,	Subsequent	Subsequent	Expense Ratio	Net Expense	Gross	Max 12b1
		Ticker	Fund Name	Fund Category	Fund Family	Investment:	IRA:	Investment:	Investment,	(net):	Ratio:	Expense	Fee:
	1	FEXPX	Fidelity Export & Multinational	Large Growth	Fidelity Investments	2,500	500	N/A	N/A	0.84%	0.86%	0.86%	N/A
	2	MPGFX	Mairs & Power Growth Inv	Large Blend	Mairs & Power	2,500	1,000	100	100	0.71%	0.71%	0.71%	N/A
	3	SLASX	Selected American Shares S	Large Blend	Selected Funds	1,000	1,000	25	25	0.93%	0.94%	0.94%	0.25%
	4	AGTHX	American Funds Growth Fund of Amer A	Large Growth	American Funds	250	250	50	50	0.69%	0.69%	0.69%	0.23%
	5	AMAGX	Amana Trust Growth	Large Growth	Amana	250	100	25	25	1.20%	1.21%	1.21%	0.25%
	6	FCNTX	Fidelity Contrafund	Large Growth	Fidelity Investments	2,500	500	N/A	N/A	0.91%	0.92%	0.92%	N/A
	7	AAGAX	American Beacon Lg Cap Value AMR	Large Value	American Beacon	0	0	N/A	0	0.34%	0.35%	0.35%	0.00%
	8	IDQYX	Columbia Diversified Equity Income R4	Large Value	Columbia	2,000	1,000	100	100	0.97%	0.92%	0.92%	0.00%
	9	UMBIX	Columbia Value & Restructuring Z	Large Value	Columbia	2,500	1,000	0	0	0.89%	0.91%	0.91%	0.00%
	10	CRMMX	CRM Mid Cap Value Inv	Mid-Cap Blend	CRM	2,500	2,000	100	100	1.03%	1.04%	1.04%	N/A
	11	TAVFX	Third Avenue Value Instl	World Stock	Third Avenue	100,000	100,000	1,000	200	1.15%	1.15%	1.19%	N/A
	12	WWNPX	Kinetics Paradigm No Load	World Stock	Kinetics	2,500	2,500	100	100	1.64%	1.64%	1.74%	N/A
	13	CVGRX	Calamos Growth A	Large Growth	Calamos	2,500	500	50	50	1.27%	1.28%	1.28%	0.25%
	14	FISGX	Nuveen Mid Cap Growth Opp I	Mid-Cap Growth	Nuveen	100,000	N/A	N/A	N/A	0.98%	1.04%	1.04%	0.00%
	15	MMELX	MassMutual Select Mid Cap Gr Eq II L	Mid-Cap Growth	MassMutual	0	N/A	0	N/A	1.10%	1.10%	1.20%	N/A
	16	FDVLX	Fidelity Value	Mid-Cap Value	Fidelity Investments	2,500	500	N/A	N/A	0.63%	0.64%	0.64%	N/A

To create this functionality, I created several sub procedures to execute the necessary code. The first sub procedure creates 3 new sheets each with a web query querying a different web page within Yahoo! Finance (part of this code is shown in the figure to the left). I attempted to pull information straight from Morningstar.com since Yahoo! Finance gets its

```
Sub GetYahooData()

    Sheets("List").Range("B4").Select
    fundTicker = Selection.Value
    Sheets.Add.Name = "ProfileData"

    With ActiveSheet.QueryTables.Add(Connection:= _
        "URL;http://finance.yahoo.com/q/pr?s=" & fundTicker & "+Profile", _
        "ProfileData!$A$1"))
        .Name = "pr?s=" & fundTicker & "+Profile"
        .FieldNames = True
        .RowNumbers = False
        .FillAdjacentFormulas = False
        .PreserveFormatting = True
        .RefreshOnFileOpen = False
        .BackgroundQuery = True
```

information from Morningstar.com, but the actual formatting of the data was much easier with Yahoo! Finance. From this procedure, I created a loop that will continue through all of the tickers that are listed on the "List" tab in column B. The next sub procedure goes through all three tabs and copies the

```
Sub CopyFundData()
Dim percent As Double

On Error Resume Next

'Copy Fund Name

    Sheets("ProfileData").Range("A1").Copy
    Sheets("List").Select
    Cells(rowNum, 3).Select
    ActiveCell.PasteSpecial xlPasteValues, xlPasteSpecialOperationNone, False, False
    Application.CutCopyMode = False
    ActiveCell.Value = Trim(Left(Cells(rowNum, 3), (InStr(Cells(rowNum, 3), "(")) - 1))
```

relevant data. Several lines of code had to be created to search out each piece of information that is listed in the column headers.

After all information is inputted for one ticker, the code then goes into a loop and repeats until all information for all tickers is complete. Once the loo p finishes, the code then deletes the worksheets that contained the three web queries. This code is shown at the right.

```
Do Until ActiveCell.Value = ""

    DownloadData

Loop

' Delete the temporary worksheets

Application.DisplayAlerts = False

Worksheets("ProfileData").Select
Worksheets("ProfileData").Delete

Worksheets("PerfomanceData").Select
Worksheets("PerfomanceData").Delete

Worksheets("RiskData").Select
Worksheets("RiskData").Delete

Application.DisplayAlerts = True
```

To get the functionality that I desired, I included error handling in the code that would skip a line of code if an error was returned. This was particularly important when certain criteria were not found using Excel's find capabilities. It is important for the user to go through the data and identify missing pieces of information. I am suspicious that the cause of the missing information is because the web query is not refreshing fast enough for the find function to pick up the necessary data.

The final sub procedure for data querying is designed to retrieve S&P 500 data which is used as a benchmark the user's portfolio. This sub procedure works in much the same way as the sub procedure that gathers the ticker information on the "List" tab. The only difference is that the S&P 500 query pulls only one ticker, IVV, which is an index fund created to match the holdings of the S&P 500 (**See Figure 3**).

Figure 3 – S&P 500 Data Retrieval

							Return				Risk			
	Create Your Portfolio		Pull S&P Data		Maximize Portfolio									
							1-Yr Return	3-Yr Return	5-Yr Return	10-Yr Return	3 Yr Beta	3 Yr Standard Deviation	5 Yr Beta	5 Yr Star Deviat
	S&P 500 Index													
	S&P 500 Index						22.57%	2.19%	2.87%	0.00%	1.00	21.84	1.00	17.8
	Total Investment Holding:		$	-										

User Form Functionality

The most complicated part of my project was providing user form functionality. Once the user has pulled all of the necessary information to create his or her portfolio, it is then time to create their portfolio. The "Create Your Portfolio" button fires the code that generates a user form which is based off of the data that was queried in the "List" tab (**See Figure 4)**. The user can then select the desired funds he or she wants added to their portfolio. The user form also allows a user to remove a fund from the portfolio list. The tricky part to this code was creating a sub procedure that would not allow the portfolio list to have any duplicate fund names, even if the user added a particular fund multiple times (**See Figure 5** for details).

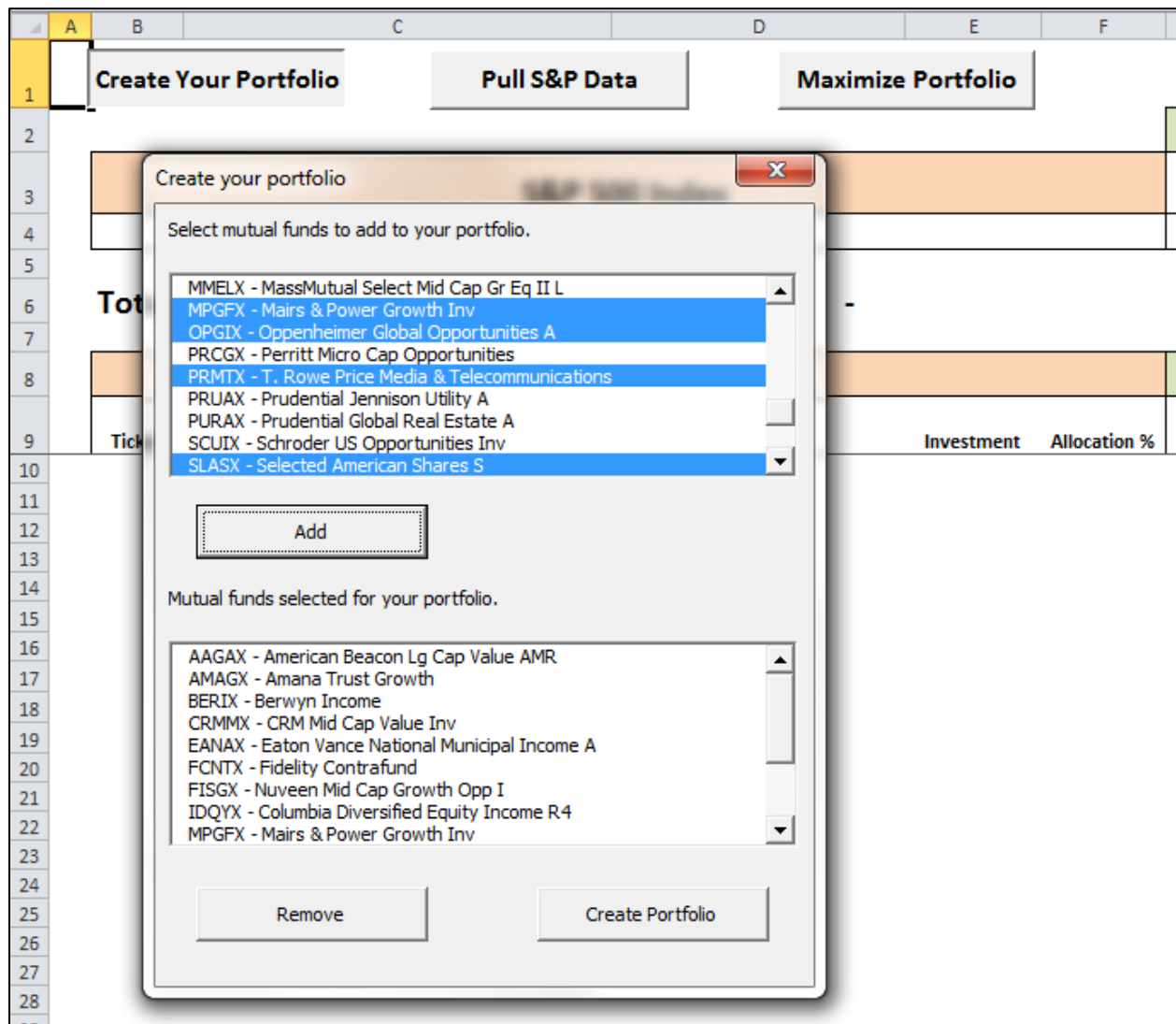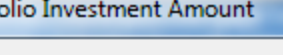**Figure 4 – User Form Functionality**

Figure 5 – Removing Duplicates From a List

```
Private Sub DeDupeList()

' This makes it so the user will not enter in duplicate values in the list

    On Error Resume Next
    Dim lstCollection As New Collection
    Dim i As Long
        For i = 0 To LstPortfolio.ListCount - 1
            lstCollection.Add LstPortfolio.list(i), LstPortfolio.list(i)
        Next i
        LstPortfolio.Clear
        For i = 1 To lstCollection.Count
            LstPortfolio.AddItem lstCollection.Item(i)
        Next i
End Sub
```

Once the user selects the desired funds and hits the "Create Portfolio" button, an input box is generated that prompts the user for an investment amount. This input box will only accept a numeric amount.

After the investment amount is established, the code then pulls the appropriate data from the "List" tab. This part of the code was particularly complicated because I wanted the model to be dynamic and automatically adjust to different portfolio sizes (**See Figure 6**).

Figure 6 – Dynamic Portfolio

Ticker	Fund Name	Category	Investment	Allocation %	Load Adj 1-Yr Return	Load Adj 3-Yr Return	Load Adj 5-Yr Return	Load Adj 10-Yr Return	3 Yr Beta	3 Yr Standard Deviation	5 Yr Beta	5 Yr St Devi
	S&P 500 Index				22.57%	2.19%	2.87%	0.00%	1.00	21.84	1.00	17
AGTHX	American Funds Growth Fund of Amer A	Large Growth	$ 4,000.00	4.00%	6.96%	-0.30%	1.49%	4.28%	0.96	21.58	0.96	17
AMAGX	Amana Trust Growth	Large Growth	$ 4,000.00	4.00%	20.70%	6.17%	6.78%	7.55%	0.77	17.96	0.79	15
AMSFX	Forward Mortgage Securities Instl	Intermediate Government	$ 4,000.00	4.00%	6.42%	5.69%	5.98%	5.18%	0.62	3.41	0.68	3.
BERIX	Berwyn Income	Conservative Allocation	$ 4,000.00	4.00%	10.84%	10.10%	8.35%	8.93%	0.57	9.04	0.56	7.
CGMRX	CGM Realty	Real Estate	$ 4,000.00	4.00%	37.83%	0.81%	9.50%	20.29%	1.55	40.24	1.53	33
MALTX	BlackRock Latin America Instl	Latin America Stock	$ 4,000.00	4.00%	20.04%	3.44%	15.37%	20.89%	1.33	38.81	1.36	33
	Weighted Portfolio		$ 100,000.00	100.00%	15.79%	2.38%	3.23%	8.27%	1.17	23.57	1.16	19

Total Investment Holding: $ 100,000.00

Solver Capabilities

The last feature that I added to my model was to include the solver capability. Once the user presses the "Maximize Portfolio" button, additional user forms are initiated to collect necessary information for the solver. The fist user form is a form that has two options. The user must select one of the options, which is a risk and return trade-off option.
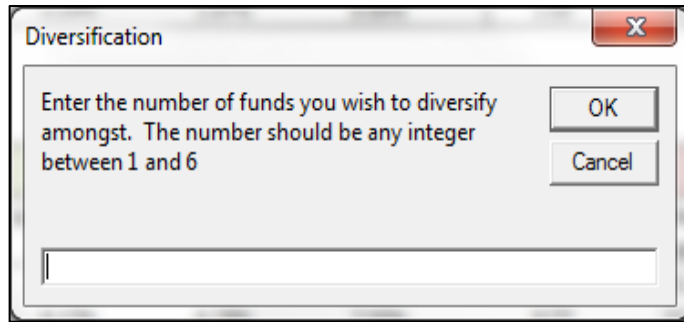
Once an option is selected, another user form then prompts the user to enter in an integer that will be used to identify the number of funds he or she wishes to diversify among. Creating the code that will only accept an integer between one and the number of funds in the portfolio proved to be quite difficult, so I have also included this code in the write-up (**See Figure 7**).

Figure 7 – Dynamic Integer Check Code

```vb
        Do
        Do
        Do
        Do
        Do

        check1 = False
        check2 = False
        check3 = False
        check4 = False
        check5 = False


On Error Resume Next

        If IsNumeric(fundDiversification) = True Then
            check1 = True
        End If
        If fundDiversification <> "" Then
            check2 = True
        End If
        If Round((fundDiversification), 0) / 1 = fundDiversification Then
            check3 = True
        End If
        If fundDiversification <= Range("PortfolioTickerCount").Count Then
            check4 = True
        End If
        If fundDiversification >= 1 And fundDiversification <= Range("PortfolioTickerCount").Count Then
            check5 = True
        Else
            check5 = False
        End If
        If check1 = False Or check2 = False Or check3 = False Or check4 = False Or check5 = False Then
            MsgBox "Please enter in valid number."
            fundDiversification = InputBox("Enter the number of funds you wish to diversify amongst.  The
        End If

    Loop While check1 = False
    Loop While check2 = False
    Loop While check3 = False
    Loop While check4 = False
    Loop While check5 = False
```

After the user inputs an integer, depending on what option was chosen, either the risk or return option, one final user form is presented. This user form specifies what cell will be the objective cell for the solver to alter to achieve for the best results (**See Figure 8**).

Figure 8 – Setting the Risk/Return Objective Cells for Solver
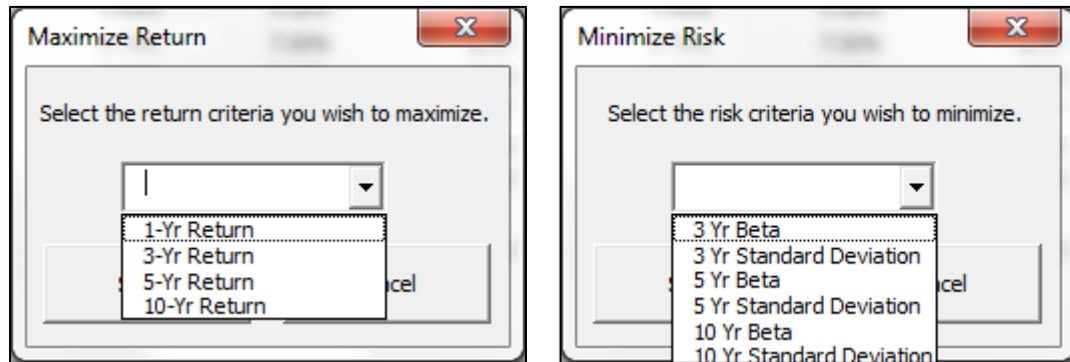
When the user pushes the "Solve" button, the solver initiates and determines an optimal level of allocation of investment (**See Figure 9**).

Figure 9 – Resulting Portfolio Allocation After Solver

Learning and Conceptual Difficulties

The whole project itself was extremely challenging and took much more time than I originally anticipated. Prior to the class, I had very little programing knowledge. Probably the most frustrating part was trying to get the code to do something relatively simple, yet not knowing exactly how to compose the code. Often times I felt like I asked very simple questions to others who had better knowledge than I did.

Overall, I struggled with the ## main areas while creating my code.

1) Trying to figure out a better way to pull data from the web. The Internet connection at BYU has been extremely slow, which takes quite a bit of time to pull relatively little information. I asked Professor Allen if there was a better and quicker way to pull the data from off of the Internet, but he informed me that it was more of a network issue.

2) Creating a dynamic portfolio summary (located on the "Dashboard" tab). Conceptually the ideas was simple, but creating the code proved to be quite challenging for me. I wanted to make the Weighted Portfolio line to be dynamic rather than static, but accomplishing this took many lines of code. I'm sure there are better ways to be more efficient with the code, but I just did not have the time to really go for optimizing my code.

Create Your Portfolio	Pull S&P Data	Maximize Portfolio					
							Retu
		S&P 500 Index			1-Yr Return	3-Yr Return	
		S&P 500 Index			22.57%	2.19%	

Total Investment Holding: $ 100,000.00

Portfolio Information					Retu		
					Load Adj 1-Yr Return	Load Adj 3-Yr Return	L
Ticker	Fund Name	Category	Investment	Allocation %			
AAGAX	American Beacon Lg Cap Value AMR	Large Value	$ -	0.00%	20.59%	0.84%	
ACRNX	Columbia Acorn Z	Mid-Cap Growth	$ 20,000.00	20.00%	30.87%	7.15%	
CGMRX	CGM Realty	Real Estate	$ -	0.00%	37.83%	0.81%	
CRMMX	CRM Mid Cap Value Inv	Mid-Cap Blend	$ 20,000.00	20.00%	26.86%	4.28%	
FCNTX	Fidelity Contrafund	Large Growth	$ 20,000.00	20.00%	24.79%	3.31%	
FSHOX	Fidelity Select Construction & Housing	Miscellaneous Sector	$ 20,000.00	20.00%	26.24%	5.99%	
IDQYX	Columbia Diversified Equity Income R4	Large Value	$ -	0.00%	17.03%	1.56%	
TAVFX	Third Avenue Value Instl	World Stock	$ -	0.00%	16.27%	1.91%	
UMBIX	Columbia Value & Restructuring Z	Large Value	$ -	0.00%	25.26%	0.18%	
VSEAX	JPMorgan Small Cap Equity A	Small Growth	$ 20,000.00	20.00%	23.65%	9.26%	
WWNPX	Kinetics Paradigm No Load	World Stock	$ -	0.00%	26.32%	-2.71%	
	Weighted Portfolio		$ 100,000.00	100.00%	26.48%	6.00%	

3) Creating a user input box that would only allow an integer between one and what ever the number of funds were in the portfolio proved to be very difficult for me to create. This probably took me a couple of hours to finally figure out. I came up with the idea to create 5 check flags that would be flipped if certain conditions were met. If one of the flags were not flipped, then the whole sequence of code would be looped back again and the flags would all be reset.

```vba
        Do
        Do
        Do
        Do
        Do

        check1 = False
        check2 = False
        check3 = False
        check4 = False
        check5 = False


On Error Resume Next

            If IsNumeric(fundDiversification) = True Then
                check1 = True
            End If
            If fundDiversification <> "" Then
                check2 = True
            End If
            If Round((fundDiversification), 0) / 1 = fundDiversification Then
                check3 = True
            End If
            If fundDiversification <= Range("PortfolioTickerCount").Count Then
                check4 = True
            End If
            If fundDiversification >= 1 And fundDiversification <= Range("PortfolioTickerCount").Cou
                check5 = True
            Else
                check5 = False
            End If
            If check1 = False Or check2 = False Or check3 = False Or check4 = False Or check5 = Fals
                MsgBox "Please enter in valid number."
                fundDiversification = InputBox("Enter the number of funds you wish to diversify amon
            End If

        Loop While check1 = False
        Loop While check2 = False
        Loop While check3 = False
        Loop While check4 = False
        Loop While check5 = False
```

4) The final part that I struggled with was configuring the solver through user forms. I learned that whenever you use user forms, your project becomes significantly more complicated. I should have learned my lesson from our user forms assignment and not have implemented them into my project, but all well.

Overall, I learned a ton during this project, even though it was a bear. This assignment really did give me confidence that I can handle my self in a small way when it comes to VBA.