

Using Microsoft Excel to build Efficient Frontiers via the Mean Variance Optimization Method

Submitted by John Alexander McNair

ID #: 0061216

Date: April 14, 2003



The Optimal Portfolio Problem

- Consider the dilemma an investor faces when trying to decide what proportion of his/her wealth should be allocated to the different types of available investments on the market place (i.e. Canadian bonds, Canadian equity, U.S. equity, Real Estate etc.).
- Historical experience has shown that each of the above stated types of investments exhibit different risk/return characteristics under different economic conditions (see Appendix B).
- Adding to the complexity of this problem, the above investments also exhibit correlation to one another.
- An investor may ask the following question “How can I minimize portfolio risk for a level of acceptable return?”.
- Mean Variance Optimization can help answer this question.

Mean-Variance Optimization (MVO) Method

- MVO, as developed by Markowitz, aims to find the set of “optimal” portfolios (*efficient frontier*), which have the highest rate of investment return for a given level of risk (i.e. standard deviation of returns) or the lowest level of risk for a given rate of return. To accomplish this task, the MVO method requires three inputs which can be calculated from the historical returns of the assets being considered for optimization.
- These inputs are: returns on each investment (r_i), standard deviation of returns on each investment (σ_i) and covariance (σ_{ij}) between the investments.
- It is important to mention that standard deviation is not the only measure of investment risk studied by investment practitioners, but MVO requires this statistic as input. Other types of risk are tracking error (standard deviation of excess returns versus a benchmark), Value at Risk (VaR) etc.
- I will discuss some “pathologies” of MVO related to Parametric versus Numerical VaR.

Mean-Variance Optimization (MVO) Method

- The main assumption underlying MVO as developed by Markowitz is that returns are Normally distributed.
- If this is the case, then making use of σ_i as the risk measure in MVO makes sense because the occurrence (probability) of outperformance versus the mean investment performance is exactly offset by the occurrence of underperformance versus the mean performance.
- Historical experience has demonstrated however that investment returns are not symmetrically distributed (see Appendix C) and as such MVO has been the subject of debate amongst investment practitioners.

Mean-Variance Optimization (MVO) Method

- The following is the MVO problem:

- Minimize the portfolio variance i.e. $\min_{\omega} (\omega^t \Sigma \omega)$ -----(1.0)

- Subject to:

$$(1.1) \left\{ \begin{array}{l} \diamond \omega^t \mathbf{r} = \mathbf{r}_T \\ \diamond \sum_{i=1 \text{ to } n} \omega_i = 1 \\ \diamond \omega_i \geq 0 \text{ (not necessarily the only option....short selling!)} \\ \bullet \text{ And perhaps other constraints} \end{array} \right.$$

- Where ω is an nx1 column vector of portfolio weights, Σ is the nxn covariance matrix of investment returns, \mathbf{r} is an nx1 column vector of investment returns, \mathbf{r}_T is the total portfolio return and n is the number of different investments in the optimization algorithm.

MVO – Sample calculation

- Consider the following two asset MVO problem: An investor wants to build a portfolio of stocks and bonds that exhibits the lowest possible portfolio variance. To simplify calculations further, assume that all of the investor's wealth must be invested (without borrowing more!). Let ω_b be the percent of the investor's wealth allocated to bonds ($1 - \omega_b$ to stocks!). Let σ_b^2 and σ_s^2 be the variance of returns of bonds and stocks respectively and let the covariance of returns between these two investments be σ_{bs} .
- Making use of (1.0) on the previous page yields the following function to be minimized:
 - $f(\omega_b) = \omega_b^2(\sigma_b^2 - 2\sigma_{bs} + \sigma_s^2) + \omega_b(2\sigma_{bs} - 2\sigma_s^2) + \sigma_s^2$

MVO – Sample calculation

- Taking the derivative of $f(\omega_b)$ with respect to ω_b and equating to zero yields the following minimum variance portfolio (this would be the first portfolio on the efficient frontier):
 - $\omega_b = -(\sigma_{bs} - \sigma_s^2) / (\sigma_b^2 - 2\sigma_{bs} + \sigma_s^2)$
 - σ_{bs} , σ_s^2 and σ_b^2 are known inputs.
- We know that the above portfolio is the global minimum variance portfolio because MVO is a strictly convex quadratic programming problem due to the covariance matrix being positive definite.
- Now as you can see adding more investments to this algorithm and adding other constraints makes for a computationally rigorous problem. Luckily many mainstream software packages have built in quadratic programming solvers. My model makes use of the built in solver within Microsoft Excel, which makes use of the Simplex method to solve the MVO problem (see Appendix E).

MVO – Microsoft Excel Optimizer

- The following quote encapsulates why I built an efficient frontier builder (which I called the Optimizer) in Microsoft Excel “Necessity, the mother of invention” (George Farquhar – (1678-1707)). Current providers of MVO do not calculate key statistics that investment practitioners want to examine and are very expensive.
- My Optimizer finds the maximum portfolio return then finds the minimum variance portfolio and the return associated with it. These two returns are then subtracted and then divided by 100. This calculation in essence is the different levels of return for which the optimizer will then find the associated minimum variance portfolio. The optimizer then stores these minimum variance portfolios in a spreadsheet in excel to which essential portfolio statistics are calculated and descriptive graphs are built.
- An easy to use interface was created so that even a novice user of Microsoft Excel can generate efficient frontiers.

MVO – Microsoft Excel Optimizer

- The Optimizer

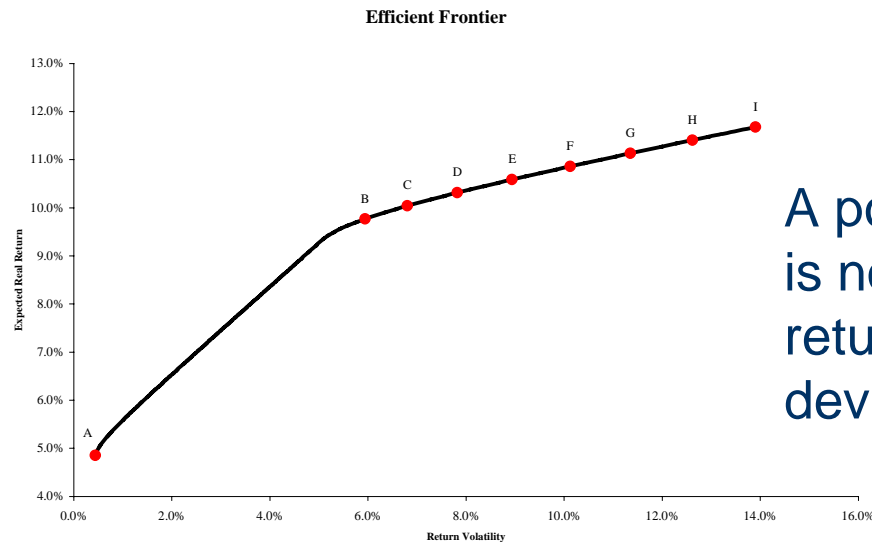
MVO – 10 year Annualized Statistics

Covariance Matrix

	Return	STDEV	SC91TBIL	SCUNOVER	S&P/TSX	SP500C	
SC91TBIL	4.85%	0.44%	0.00%	0.01%	0.01%	0.01%	0.00%
SCUNOVER	8.91%	5.08%	0.01%	0.26%	0.24%	0.10%	0.02%
S&P/TSX	9.07%	16.80%	0.01%	0.24%	2.82%	1.64%	1.49%
SP500C	11.68%	13.90%	0.01%	0.10%	1.64%	1.93%	1.30%
MSEAFEC	6.54%	14.35%	0.00%	0.02%	1.49%	1.30%	2.06%

Annualized Statistics (Jan 93 - Dec 02)

MVO – the Efficient Frontier



A portfolio is considered efficient if it is not possible to obtain a higher return without increasing standard deviation.

	A	B	C	D	E	F	G	H	I
SC91TBIL	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
SCUNOVER	0.0%	69.0%	59.1%	49.2%	39.4%	29.5%	19.7%	9.8%	0.0%
S&P/TSX	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
SP500C	0.0%	31.0%	40.9%	50.8%	60.6%	70.5%	80.3%	90.2%	100.0%
MSEAFEC	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Total Equity	0.0%	31.0%	40.9%	50.8%	60.6%	70.5%	80.3%	90.2%	100.0%
Total Foreign Equity	0.0%	31.0%	40.9%	50.8%	60.6%	70.5%	80.3%	90.2%	100.0%
Total Fixed Income	100.0%	69.0%	59.1%	49.2%	39.4%	29.5%	19.7%	9.8%	0.0%
Expected Real Return	4.85%	9.77%	10.04%	10.31%	10.59%	10.86%	11.13%	11.41%	11.68%
Return Volatility	0.44%	5.94%	6.81%	7.82%	8.94%	10.12%	11.35%	12.62%	13.90%
VaR (Absolute)	-\$56	-\$195	-\$212	-\$232	-\$253	-\$275	-\$298	-\$322	-\$346
VaR Confidence Min	-\$57	-\$210	-\$229	-\$251	-\$274	-\$299	-\$325	-\$352	-\$379
VaR Confidence Max	-\$55	-\$184	-\$200	-\$217	-\$236	-\$256	-\$277	-\$298	-\$320

Pathologies: Parametric versus Numerical VaR Calculation

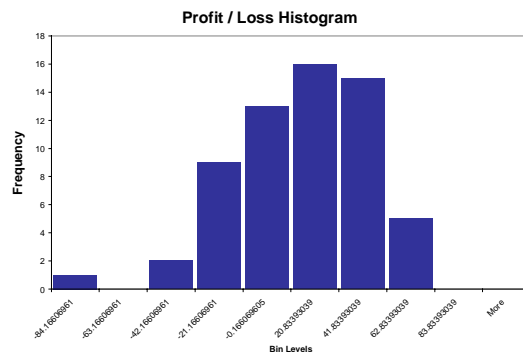
- Value at Risk (VaR) is a statistic that aims to quantify the maximum amount of wealth an investor is likely to lose over a given amount of time at a specific confidence level. There are two common ways in portfolio theory to calculate VaR. There is the Parametric VaR (typically making use of Standard Normal quantiles) and numerical VaR (making use of historical Profit/Loss histograms).
- Consider an investor who now wants to build the following portfolio:
 - 5% Cash (SC 91 Day T-Bills)
 - 45% Bonds (SC Universe Bonds)
 - 20% Canadian Equity (S&P/TSX Composite)
 - 15% U.S. Equity (S&P 500)
 - 15% International Equity (MSCI EAFE)

Pathologies: Parametric versus Numerical VaR Calculation

- The 95% Parametric annualized VaR (Absolute) of this continuously rebalanced portfolio (under Normal distribution assumption i.e. assumption in MVO) considering the last 5 years of historical monthly returns with \$1000 initial wealth (W) is then:
 - $\text{VaR (Absolute)} = -\mu W - \alpha \sigma W = -4.15\%(1000) - 1.645(8.11\%)(1000) = -\175 (i.e. 1 in 20 years we can expect to lose more than \$175)
 - The 95% confidence interval for this VaR is [-\$204,-\$155]
- Now let us examine the difference between Parametric and Numerical VaR considering the above portfolio's performance over the last 5 years ending December 31, 2002.

Pathologies: Parametric versus Numerical VaR Calculation

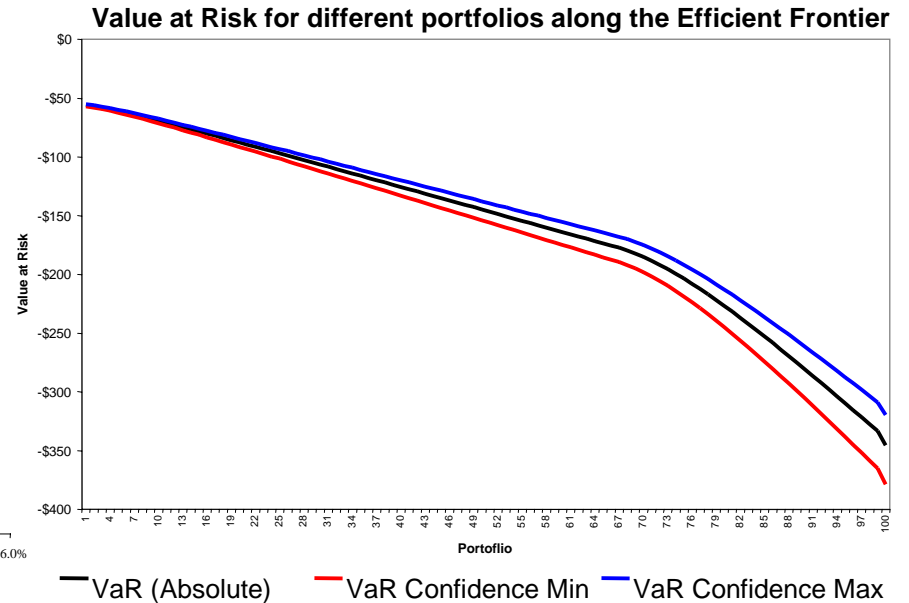
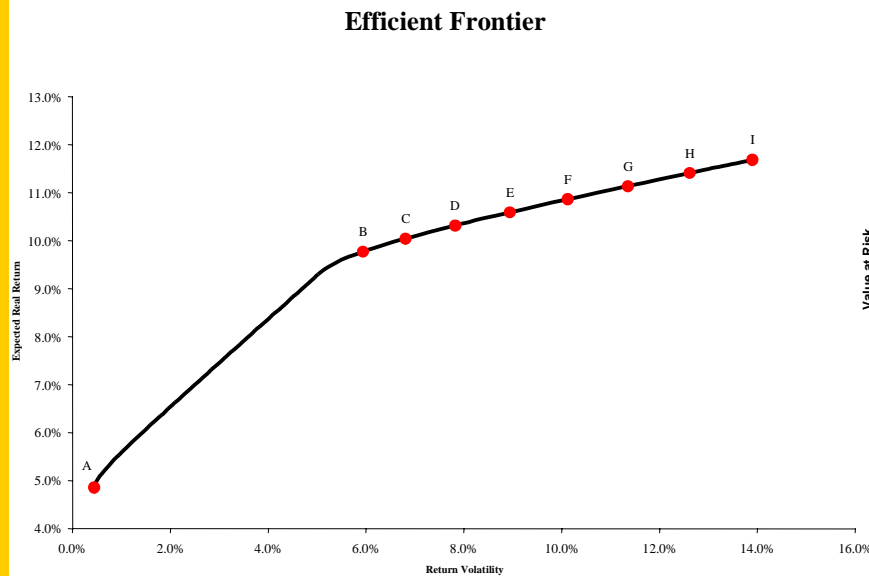
- Below is the Profit/Loss histogram generated using portfolio monthly returns for the last 5 years ending December 31, 2002.



Bin	Frequency	Cumulative %
-84.17	1	1.64%
-63.17	0	1.64%
-42.17	2	4.92%
-21.17	9	19.67%
-0.17	13	40.98%
20.83	16	67.21%
41.83	15	91.80%
62.83	5	100.00%
83.83	0	100.00%
More	0	100.00%

- As you can see the above returns are not Symmetrically distributed. The loss level that delineates the lowest 5% of losses from the other profit/loss levels is approximately -\$42.2, therefore the numerical annualized VaR is:
 - Numerical VaR = $-\$42.2(12)^{0.5} = -\146 (i.e. 1 in 20 years we can expect to lose more than \$146)
 - This VaR lies outside the Parametric VaR's confidence interval and hence evidence of one of the pathologies of MVO.

MVO - Optimizer versus other software providers



	A	B	C	D	E	F	G	H	I
SC91TBIL	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
SCUNOVER	0.0%	69.0%	59.1%	49.2%	39.4%	29.5%	19.7%	9.8%	0.0%
S&P/TSX	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
SP500C	0.0%	31.0%	40.9%	50.8%	60.6%	70.5%	80.3%	90.2%	100.0%
MSEAFEC	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Total Equity	0.0%	31.0%	40.9%	50.8%	60.6%	70.5%	80.3%	90.2%	100.0%
Total Foreign Equity	0.0%	31.0%	40.9%	50.8%	60.6%	70.5%	80.3%	90.2%	100.0%
Total Fixed Income	100.0%	69.0%	59.1%	49.2%	39.4%	29.5%	19.7%	9.8%	0.0%
Expected Real Return	4.85%	9.77%	10.04%	10.31%	10.59%	10.86%	11.13%	11.41%	11.68%
Return Volatility	0.44%	5.94%	6.81%	7.82%	8.94%	10.12%	11.35%	12.62%	13.90%
VaR (Absolute)	-\$56	-\$195	-\$212	-\$232	-\$253	-\$275	-\$298	-\$322	-\$346
VaR Confidence Min	-\$57	-\$210	-\$229	-\$251	-\$274	-\$299	-\$325	-\$352	-\$379
VaR Confidence Max	-\$55	-\$184	-\$200	-\$217	-\$236	-\$256	-\$277	-\$298	-\$320

$$\text{VaR 95\% Confidence Min} = -\mu W - 1.645 W s \left(\frac{(n-1)}{\chi^2_{(0.975)}} \right)^{1/2}$$

$$\text{VaR 95\% Confidence Max} = -\mu W - 1.645 W s \left(\frac{(n-1)}{\chi^2_{(0.025)}} \right)^{1/2}$$

Appendix A – Visual Basic code

```
• Sub Frontier_Builder()
• SolverReset
• Application.ScreenUpdating = False
• 'Finding Maximum Surplus
• SolverOk SetCell:="$K$38", MaxMinVal:=1, ValueOf:="0", ByChange:="$B$18:$B$27"
• SolverAdd CellRef:="$B$28", Relation:=2, FormulaText:="1"
• SolverAdd CellRef:="$B$18:$B$27", Relation:=1, FormulaText:="Optimal_Mix_Max"
• SolverAdd CellRef:="$B$18:$B$27", Relation:=3, FormulaText:="Optimal_Mix_Min"
• SolverAdd CellRef:="$H$31:$H$33", Relation:=1, FormulaText:="$I$31:$I$33"
• SolverOptions MaxTime:=1000, Iterations:=1000, Precision:=0.00000001, _
• AssumeLinear:=False, StepThru:=False, Estimates:=2, Derivatives:=2, _
• SearchOption:=1, IntTolerance:=0.000005, Scaling:=False, Convergence:=0, _
• AssumeNonNeg:=False
• Solution = SolverSolve(True)
• MaxSurplus = Range("Surplus").Value
• SolverReset
• 'Finding Minimum Surplus Volatility and its Surplus
• SolverOk SetCell:="$K$36", MaxMinVal:=2, ValueOf:="0", ByChange:="$B$18:$B$27"
• SolverAdd CellRef:="$B$28", Relation:=2, FormulaText:="1"
• SolverAdd CellRef:="$B$18:$B$27", Relation:=1, FormulaText:="Optimal_Mix_Max"
• SolverAdd CellRef:="$B$18:$B$27", Relation:=3, FormulaText:="Optimal_Mix_Min"
• SolverAdd CellRef:="$H$31:$H$33", Relation:=1, FormulaText:="$I$31:$I$33"
• SolverOptions MaxTime:=1000, Iterations:=1000, Precision:=0.00000001, _
• AssumeLinear:=False, StepThru:=False, Estimates:=2, Derivatives:=2, _
• SearchOption:=1, IntTolerance:=0.000005, Scaling:=False, Convergence:=0, _
• AssumeNonNeg:=False
• Solution = SolverSolve(True)
• MinSurplus = Range("Surplus").Value
• SolverReset
• ReDim Asset_Mix_Array(1 To 10, 1 To 100) As Double
• ReDim Frontier_Check_Array(1, 1 To 100) As Double
```

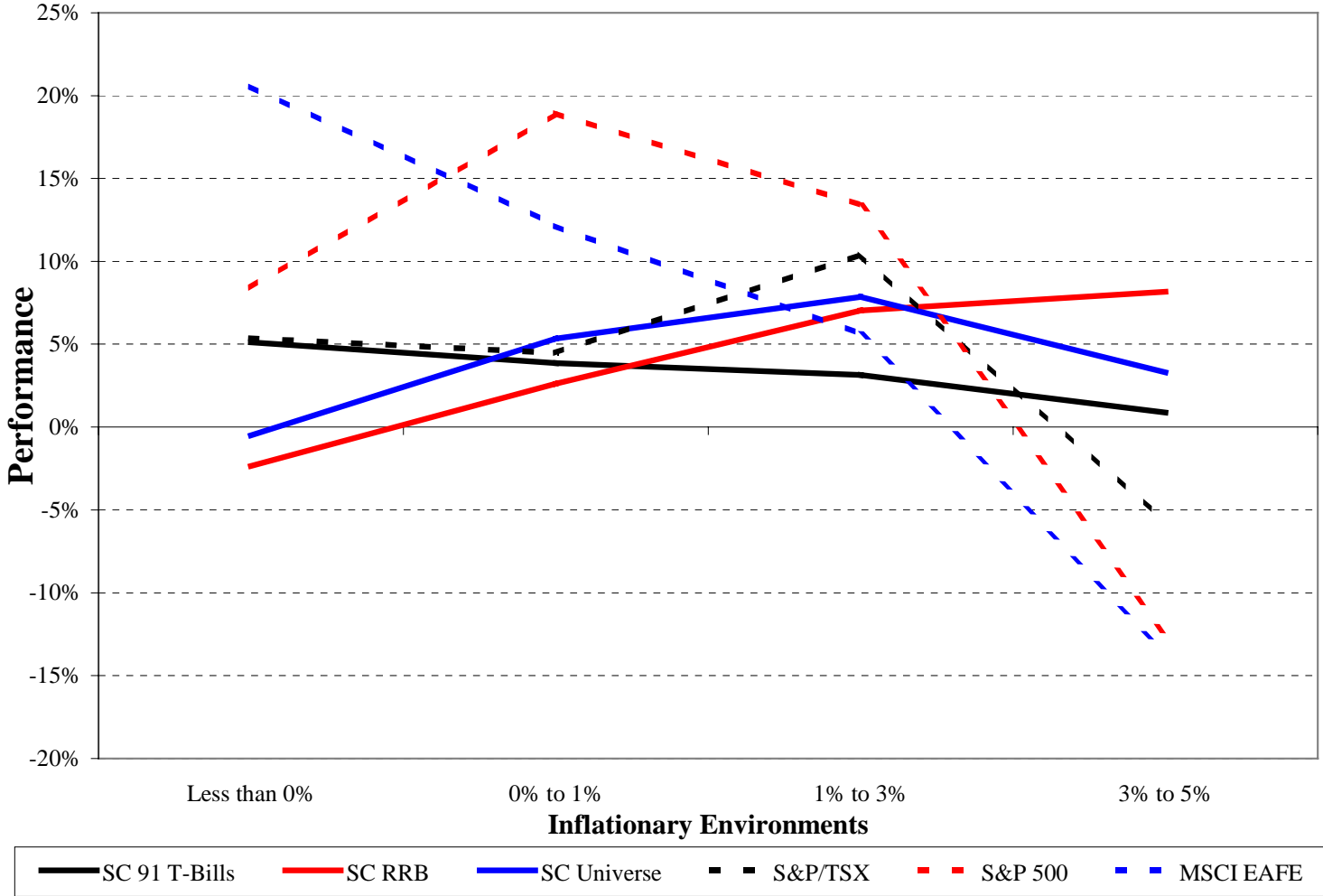

Appendix A – Visual Basic code continued

- NextRow = 1
- Increment = (MaxSurplus - MinSurplus) / 100
- SolverOk SetCell:="\$K\$36", MaxMinVal:=2, ValueOf:="0", ByChange:="\$B\$18:\$B\$27"
- SolverAdd CellRef:="\$B\$28", Relation:=2, FormulaText:="1"
- SolverAdd CellRef:="\$B\$18:\$B\$27", Relation:=1, FormulaText:="Optimal_Mix_Max"
- SolverAdd CellRef:="\$B\$18:\$B\$27", Relation:=3, FormulaText:="Optimal_Mix_Min"
- SolverAdd CellRef:="\$H\$31:\$H\$33", Relation:=1, FormulaText:="\$I\$31:\$I\$33"
- SolverAdd CellRef:="\$K\$38", Relation:=2, FormulaText:="Desired_Surplus"
- SolverOptions MaxTime:=1000, Iterations:=1000, Precision:=0.00000001, _
- AssumeLinear:=False, StepThru:=False, Estimates:=2, Derivatives:=2, _
- SearchOption:=1, IntTolerance:=0.000005, Scaling:=False, Convergence:=0, _
- AssumeNonNeg:=False
- For i = MinSurplus To MaxSurplus - Increment Step Increment
- Range("Desired_Surplus").Value = i
- Solution = SolverSolve(True)
- If Solution = 0 Then
- Frontier_Check_Array(1, NextRow) = Range("Optimal_Mix_Total").Value
- Asset_Mix_Array(1, NextRow) = Range("Class1").Value
- Asset_Mix_Array(2, NextRow) = Range("Class2").Value
- Asset_Mix_Array(3, NextRow) = Range("Class3").Value
- Asset_Mix_Array(4, NextRow) = Range("Class4").Value
- Asset_Mix_Array(5, NextRow) = Range("Class5").Value
- Asset_Mix_Array(6, NextRow) = Range("Class6").Value
- Asset_Mix_Array(7, NextRow) = Range("Class7").Value
- Asset_Mix_Array(8, NextRow) = Range("Class8").Value
- Asset_Mix_Array(9, NextRow) = Range("Class9").Value
- Asset_Mix_Array(10, NextRow) = Range("Class10").Value
- NextRow = NextRow + 1
- End If
- Next

Appendix A – Visual Basic code continued

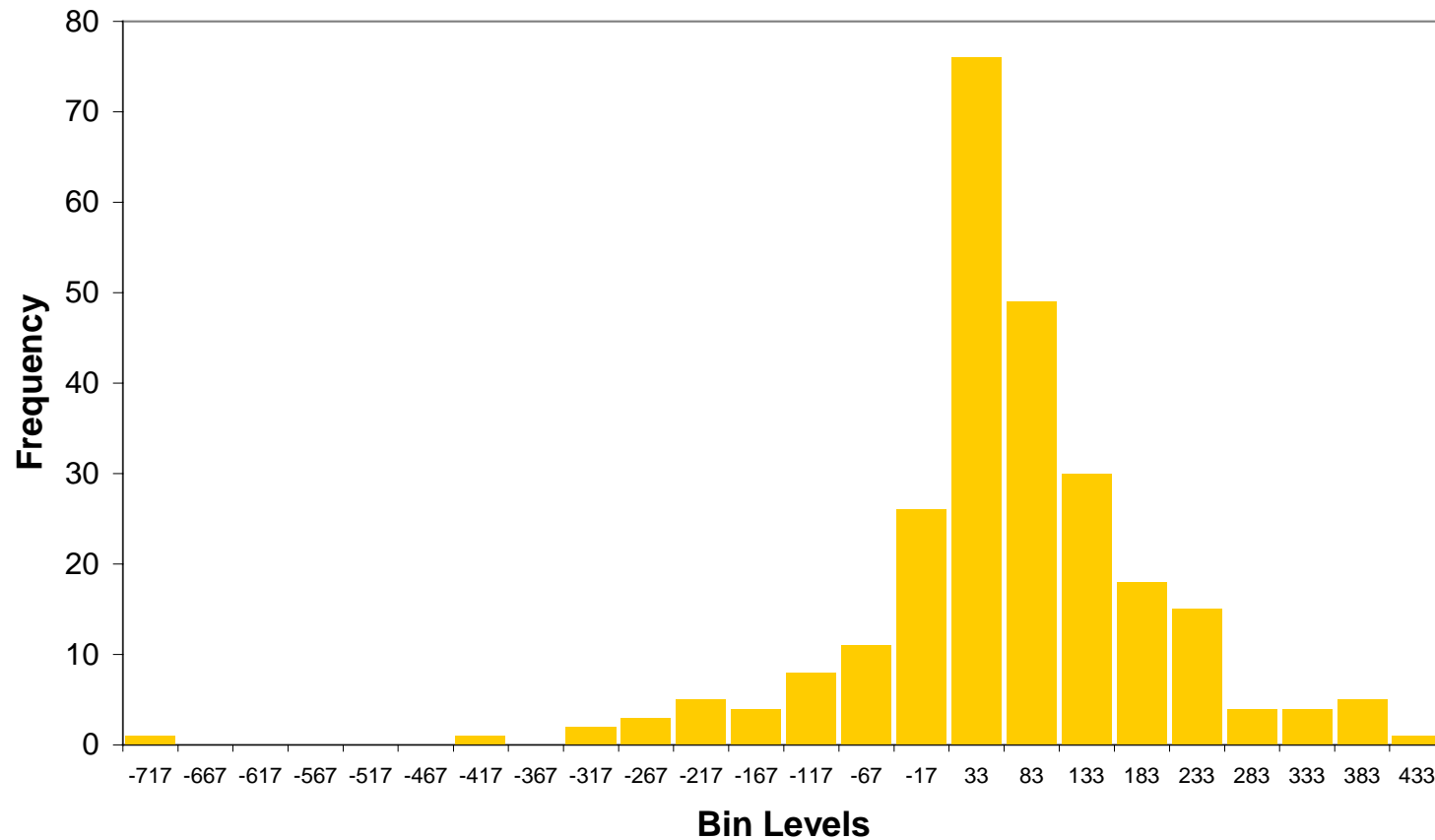
- Range("Desired_Surplus").Value = MaxSurplus
- Solution = SolverSolve(True)
- Asset_Mix_Array(1, 100) = Range("Class1").Value
- Asset_Mix_Array(2, 100) = Range("Class2").Value
- Asset_Mix_Array(3, 100) = Range("Class3").Value
- Asset_Mix_Array(4, 100) = Range("Class4").Value
- Asset_Mix_Array(5, 100) = Range("Class5").Value
- Asset_Mix_Array(6, 100) = Range("Class6").Value
- Asset_Mix_Array(7, 100) = Range("Class7").Value
- Asset_Mix_Array(8, 100) = Range("Class8").Value
- Asset_Mix_Array(9, 100) = Range("Class9").Value
- Asset_Mix_Array(10, 100) = Range("Class10").Value
- Worksheets("Frontier Points").Range("B1:CW10").Value = Asset_Mix_Array
- Application.ScreenUpdating = True
- SolverReset
- End Sub

Appendix B - Asset Class Performance over varying inflation

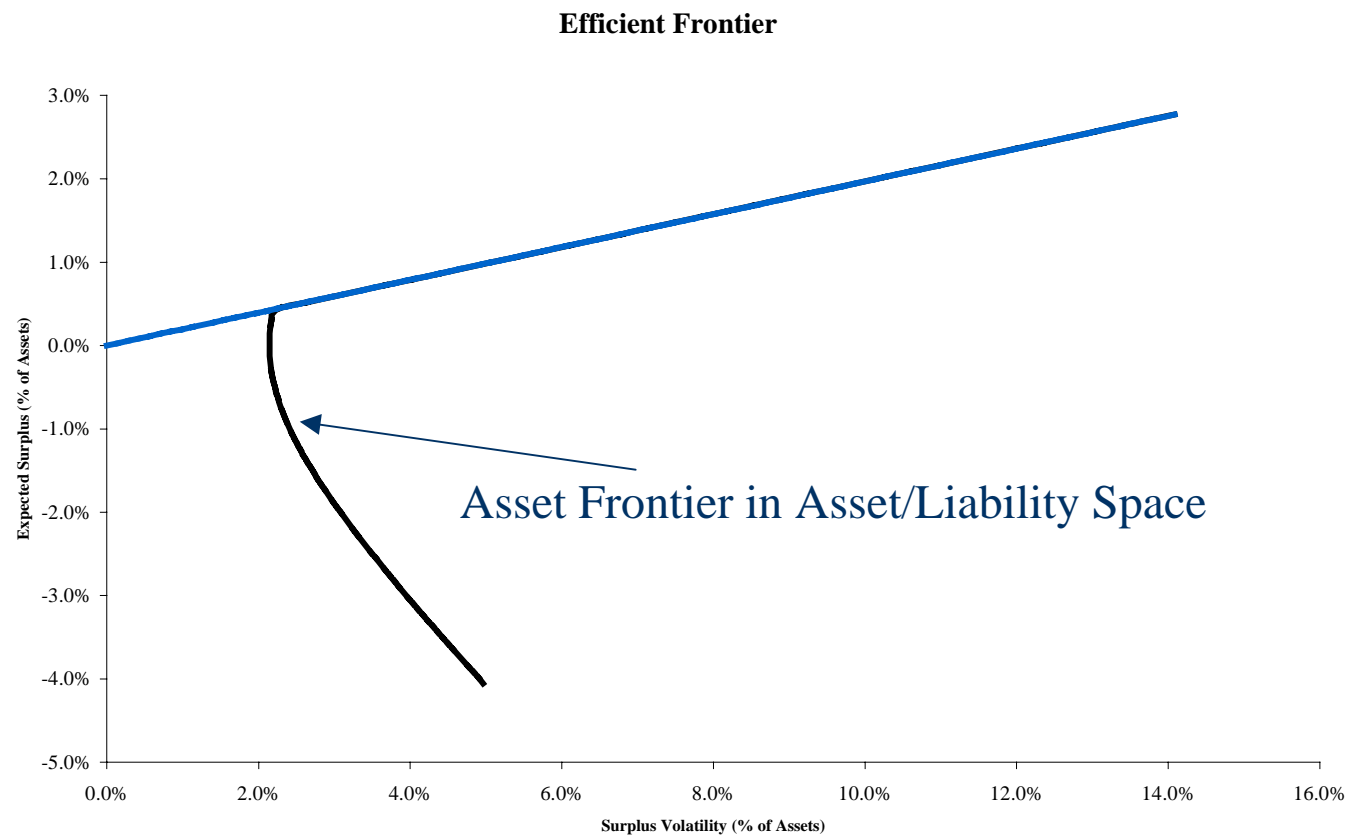


Appendix C - Portfolio Return Histograms

Monthly Profit/Loss Histogram over the last 20 years



Appendix D - Asset frontier in surplus space



Appendix E - Simplex Method for the MVO problem

- Before solving the MVO problem with the simplex method (1.1) has to be rewritten in equation form by introducing slack variables (suitable positive numbers) to transform the inequality constraints to equalities.
- The new minimization problem is then:
- $\min_{\omega} (\omega^t \Sigma \omega - \omega^t r + m^t (-\omega + s^2) + n^t (-\omega + \mathbf{1}))$ ----- (1.2) where:
 - s is a vector of slack variables
 - $m \geq 0$ & n are vectors of Lagrange multipliers for the inequality and equality constraints respectively.
 - $\mathbf{1}$ is a column vector of ones
- Now note that (1.2) is unconstrained and therefore the sufficient condition for its minimum is that all of its partial derivatives must be equal to zero. Performing these partial derivatives leads to a new set of equations to which is applied the simplex algorithm.

Appendix F – Accuracy and Computing Time

- Computing the accuracy of the optimal portfolio over different levels of portfolio return manually with the simplex method to ascertain accuracy would be quite rigorous. Instead when building the Optimizer I compared the optimal portfolios to those obtained with Ibbotson Encorr Optimizer, a widely used efficient frontier software. For identical levels of return, both models posted identical optimal portfolios (at least up to the third decimal place).
- Computing time for the Excel Optimizer is approximately 7 seconds on a Pentium IV 1.8 GHz with 256 MB or RAM. Ibbotson's product will generate an efficient frontier in less than 1 second. However from the input phase to efficient frontier generation my Excel based Optimizer is much simpler and faster to use.