

# Portfolio Optimization with the Omega Ratio

## Introduction

Traditional investment performance benchmarks, like the Sharpe Ratio, approximate the returns distribution with mean and standard deviation. This, however, assumes the distribution is normal. Many modern investments vehicles, like hedge funds, display fat tails, and skew and kurtosis in the returns distribution. Hence, they cannot be adequately benchmarked with traditional approaches.

One solution, proposed by Shadwick and Keating in 2002 is the Omega Ratio. This divides the returns distribution into two halves – the area below a target return, and the above a target return. The Omega Ratio is simply the former divided by the latter. A higher value is better.

For a set of discrete returns, the Omega Ratio is given by

$$\Omega(L) = \frac{E[\max(R - L, 0)]}{E[\max(L - R, 0)]}$$

where  $L$  is a target return and  $R$  is a vector of returns.

This application finds the asset weights that maximize the Omega Ratio of a portfolio of ten investments, given their simulated monthly returns and a target return.

This is a non-convex problem, and requires global optimizers for a rigorous solution. However, a transformation of the variables (only valid for Omega Ratios of over 1) converts the optimization into a linear program.

This application implements both approaches, the former using Maple's Global Optimization Toolbox, and the latter using Maple's linear programming features. For the data set provided in this application, both approaches give comparable results.

## Returns Data and Minimum Acceptable Return

> restart :

Monthly hedge fund returns

	1	2	3	4	5	6	7	8
1	1.283	0.865	0.654	0.709	2.023	0.813	2.234	-0.311
2	0.429	0.433	0.601	-0.549	0.669	0.22	-0.291	0.367
3	2.657	2.238	1.592	1.26	3.115	0.328	2.881	3.684
4	1.731	1.4	1.184	1.717	2.112	0.508	1.438	2.818
5	1.899	1.306	-0.322	-0.015	2.758	1.304	1.732	3.393

Number of funds

> N := LinearAlgebra[ColumnDimension](data)  
N := 10

Number of returns for each fund

> S := LinearAlgebra[RowDimension](data)  
S := 36

Target Return

> L := 0.1 :

## Omega Ratio

> OmegaRatio := **proc** (L, returns, weights)  
**local** weightedReturns, above, below, N, S :

N := LinearAlgebra[ColumnDimension](returns) :

S := LinearAlgebra[RowDimension](returns) :

weightedReturns := [seq(add(returns[i,j]·weights[j], j = 1..N), i = 1..S)] :

below := select(x → evalb(x < L), weightedReturns) :

above := select(x → evalb(x > L), weightedReturns) :

**return**  $\frac{\text{add}(a - L, a \text{ in above})}{\text{add}(L - a, a \text{ in below})}$

**end proc**:

"Strawman" portfolio of equal weights at the target return

> OmegaRatio(L, data, [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1])  
4.00319251000761

## Global Optimization

> resultsGO := GlobalOptimization[GlobalSolve]( 'OmegaRatio'(L, data, [seq(w<sub>i</sub>, i = 1..10)]), {add(w<sub>i</sub>, i = 1..N) = 1}, seq(w<sub>i</sub> = 0..1, i = 1..N), maximize) :

Optimized Omega Ratio

> resultsGO<sub>1</sub>

6.97971552601530920

Optimized investment weights

> weightsGO := resultsGO<sub>2</sub>

weightsGO := [w<sub>1</sub> = 0., w<sub>2</sub> = 0., w<sub>3</sub> = 0., w<sub>4</sub> = 0.409773987791656, w<sub>5</sub> = 0., w<sub>6</sub> = 0.219620283939146, w<sub>7</sub> = 0.333220772149511, w<sub>8</sub> = 0., w<sub>9</sub> = 0., w<sub>10</sub> = 0.0373849561196863]

## Linear Program

The transformation of the optimization problem into a linear program is described [here](#)

> eq1 := seq(add(data<sub>i,j</sub> w<sub>j</sub>, j = 1..N) - u<sub>i</sub> + d<sub>i</sub> - L t = 0, i = 1..S) :

> eq2 := add(w<sub>j</sub>, j = 1..N) = t :

> eq3 := add( $\frac{d_i}{S}$ , i = 1..S) = 1 :

> obj := add( $\frac{u_i}{S}$ , i = 1..S) :

> cons := seq([u<sub>i</sub> ≥ 0, d<sub>i</sub> ≥ 0]<sub>[i]</sub>, i = 1..S), seq([w<sub>j</sub> ≥ 0]<sub>[j]</sub>, j = 1..N) :

> resultsLP := Optimization[LPSolve](obj, {eq1, eq2, eq3, cons}, maximize, assume = nonnegative) :

Optimized Omega Ratio

> resultsLP<sub>1</sub>

6.97971754550025

Optimized investment weights

> assign(select(has, resultsLP<sub>2</sub>, t))

> weightsLP := map(i → lhs(i) =  $\frac{\text{rhs}(i)}{t}$ , select(has, resultsLP<sub>2</sub>, w))

weightsLP := [w<sub>1</sub> = 0., w<sub>2</sub> = 0., w<sub>3</sub> = 0., w<sub>4</sub> = 0.409790222161672, w<sub>5</sub> = 0., w<sub>6</sub> = 0.219617381328148, w<sub>7</sub> = 0.333211183439266, w<sub>8</sub> = 0., w<sub>9</sub> = 0., w<sub>10</sub> = 0.0373812133486924]