Computing Efficient Portfolios in R

Eric Zivot

November 11, 2008

Abstract

This note describes the computation of mean-variance efficient portfolios using R.

1 Portfolio Analysis Functions

I have written a few R functions for computing Markowitz mean-variance efficient portfolios allowing for short sales. These functions are meant to be used for learning the basics of portfolio theory. They are summarized in Table 1.

1.1 Examples

The following examples illustrate the use of the functions in Table 1. These examples are also available in the R script file `testport.r` on the class R hints web page. To begin, open the script file `portfolio.r` and select Edit/Run all to source the portfolio functions into memory.

Next, consider the input data ($\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ and r_f) used for the three firm example used in the class lectures (see `3firmExample.xls` on the class syllabus web page).

```
> asset.names <- c("MSFT", "NORD", "SBUX")
> er <- c(0.0427, 0.0015, 0.0285)
> names(er) <- asset.names
```

Function	Description
`getPortfolio`	create portfolio object
`globalMin.portfolio`	compute global minimum variance portfolio
`efficient.portfolio`	compute minimum variance portfolio subject to target return
`tangency.portfolio`	compute tangency portfolio
`efficient.frontier`	compute efficient frontier of risky assets

Table 1: R functions for computing mean-variance efficient portfolios

```
> covmat <- matrix(c(0.0100, 0.0018, 0.0011,
+    0.0018, 0.0109, 0.0026,
+    0.0011, 0.0026, 0.0199),
+  nrow=3, ncol=3)
> rk.free <- 0.005
> dimnames(covmat) <- list(asset.names, asset.names)
```

To specify a portfolio, you need an expected return vector and covariance matrix for the assets under consideration as well as a vector of portfolio weights. To create an equally weighted portfolio use

```
> ew = rep(1,3)/3
> equalWeight.portfolio = getPortfolio(er=er,cov.mat=covmat,weights=ew)
> class(equalWeight.portfolio)
[1] "portfolio"
```

Portfolio objects have the following components

```
> names(equalWeight.portfolio)
[1] "call"    "er"      "sd"      "weights"
```

There are `print()`, `summary()` and `plot()` methods for portfolio objects. The `print()` method gives

```
> equalWeight.portfolio
Call:
getPortfolio(er = er, cov.mat = covmat, weights = ew)

Portfolio expected return:     0.02423333
Portfolio standard deviation:  0.07586538
Portfolio weights:
  MSFT   NORD   SBUX
0.3333 0.3333 0.3333
```

The `plot()` method shows a bar chart of the portfolio weights

```
> plot(equalWeight.portfolio)
```

The global minimum variance portfolio (allowing for short sales) $\mathbf{m}$ solves the optimization problem

$$\min_{\mathbf{m}} \mathbf{m}'\mathbf{\Sigma}\mathbf{m} \text{ s.t. } \mathbf{m}'\mathbf{1} = 1.$$

To compute this portfolio use the function `globalMin.portfolio()`

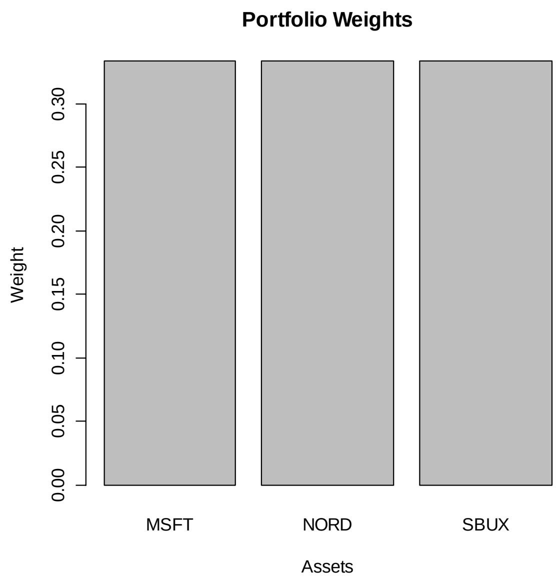

Figure 1:

```
> gmin.port <- globalMin.portfolio(er, covmat)
> attributes(gmin.port)
$names
[1] "call"    "er"      "sd"      "weights"

$class
[1] "portfolio"
> gmin.port
Call:
globalMin.portfolio(er = er, cov.mat = covmat)

Portfolio expected return:     0.02489184
Portfolio standard deviation:  0.07267607
Portfolio weights:
  MSFT   NORD   SBUX
0.4411 0.3656 0.1933
```

A mean-variance efficient portfolio $\mathbf{x}$ that achieves the target expected return μ_0 solves the optimization problem

$$\min_{x} \ \mathbf{x}'\mathbf{\Sigma}\mathbf{x} \text{ s.t. } \mathbf{x}'\mathbf{1} = 1 \text{ and } \mathbf{x}'\boldsymbol{\mu} = \mu_0.$$

To compute this portfolio for the target expected return $\mu_0 = E[R_{msft}] = 0.04275$ use the `efficient.portfolio()` function

```
> target.return <- er[1]
> e.port.msft <- efficient.portfolio(er, covmat, target.return)
> e.port.msft
Call:
efficient.portfolio(er = er, cov.mat = covmat, target.return = target.return)

Portfolio expected return:     0.0427
Portfolio standard deviation:  0.091656
Portfolio weights:
   MSFT    NORD    SBUX
 0.8275 -0.0907  0.2633
```

The tangency portfolio $\mathbf{t}$ is the portfolio of risky assets with the highest Sharpe's slope and solves the optimization problem

$$\max_{\mathbf{t}} \frac{\mathbf{t}'\boldsymbol{\mu} - r_f}{(\mathbf{t}'\boldsymbol{\Sigma}\mathbf{t})^{1/2}} \text{ s.t. } \mathbf{t}'\mathbf{1} = 1,$$

where r_f denotes the risk-free rate. To compute this portfolio with $r_f = 0.005$ use the `tangency.portfolio()` function

```
> tan.port <- tangency.portfolio(er, covmat, rk.free)
> tan.port
Call:
tangency.portfolio(er = er, cov.mat = covmat, risk.free = rk.free)

Portfolio expected return:     0.05188967
Portfolio standard deviation:  0.1115816
Portfolio weights:
   MSFT    NORD    SBUX
 1.0268 -0.3263  0.2994
```

The the set of efficient portfolios of risky assets can be computed as a convex combination of any two efficient portfolios. It is convenient to use the global minimum variance portfolio as one portfolio and an efficient portfolio with target expected return equal to the maximum expected return of the assets under consideration as the other portfolio. Call these portfolios $\mathbf{m}$ and $\mathbf{x}$, respectively. For any number α, another efficient portfolio can be computed as

$$\mathbf{z} = \alpha\mathbf{m} + (1 - \alpha)\mathbf{x}$$

The function `efficient.frontier()` constructs the set of efficient portfolios using this method for a collection of α values. For example, to compute 20 efficient portfolios for values of α between -2 and 1.5 use

```
> ef <- efficient.frontier(er, covmat, alpha.min=-2,
+                          alpha.max=1.5, nport=20)
> attributes(ef)
$names
[1] "call"    "er"      "sd"      "weights"

$class
[1] "Markowitz"

> ef
Call:
efficient.frontier(er = er, cov.mat = covmat, nport = 20, alpha.min = -2,
    alpha.max = 1.5)

Frontier portfolios' expected returns and standard deviations
   port 1 port 2 port 3 port 4 port 5 port 6 port 7
ER 0.0783 0.0750 0.0718 0.0685 0.0652 0.0619 0.0586
SD 0.1826 0.1732 0.1640 0.1548 0.1458 0.1370 0.1284
   port 8 port 9 port 10 port 11 port 12 port 13 port 14
ER 0.0554 0.0521  0.0488  0.0455  0.0422   0.039  0.0357
SD 0.1200 0.1120  0.1044  0.0973  0.0908   0.085  0.0802
   port 15 port 16 port 17 port 18 port 19 port 20
ER  0.0324  0.0291  0.0258  0.0225  0.0193  0.0160
SD  0.0764  0.0739  0.0727  0.0730  0.0748  0.0779
```

Use the `summary()` method to show the weights of these portfolios. Use the `plot()` method to plot the efficient frontier

```
> plot(ef)
```

The resulting plot is shown in Figure 2.

To create a plot of the efficient frontier showing the original assets and the tangency portfolio use

```
> plot(ef, plot.assets=T)
> points(gmin.port$sd, gmin.port$er, col="blue")
> points(tan.port$sd, tan.port$er, col="red")
> sr.tan = (tan.port$er - rk.free)/tan.port$sd
> abline(a=rk.free, b=sr.tan)
```

The resulting plot is shown in Figure 3.

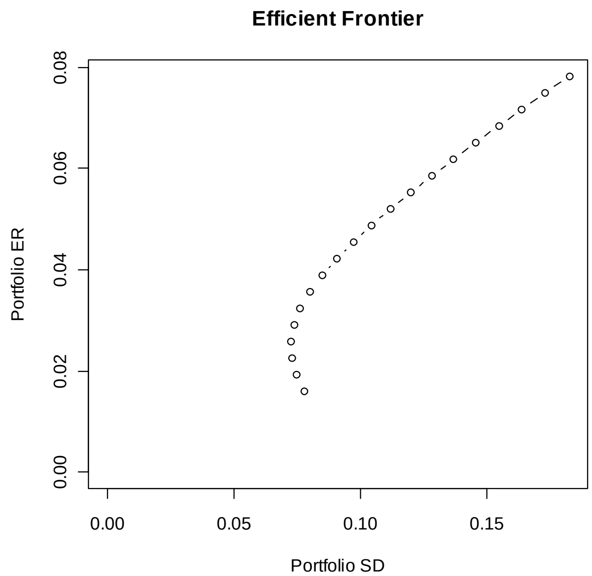

Figure 2: Plot method for Markowitz object.

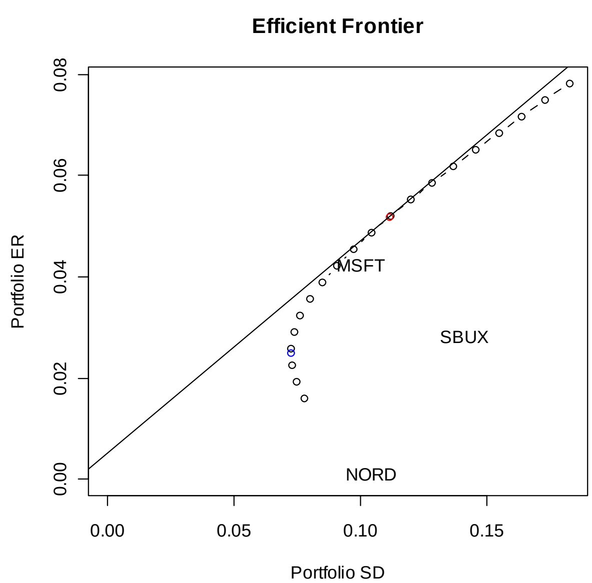

Figure 3: Efficient frontier for three firm example.