Portfolio Managment: Homework 4

Gal Skarishevsky, Nathan Matare, Brian Thompson, Lior Sahaf

April 19, 2017

B.1. Estimating E and V by the sample estimates.

We write the function '`getPortfolio`' that constructs the appropriate portfolio based upon the homework prompt. The default setting constructs a portfolio based upon sample estimates: $\hat{E}$ and $\hat{V}$. Otherwise, several arguments: '`round returns, identity, CAPM, BAYES`' override the default settings and control the type of estimation technique. Finally, the argument '`portfolio`' controls whether to calculate either the minimum variance portfolio or the tangency portfolio.

```
getPortfolio <- function(data, portfolio, round_returns = FALSE,
                         identity = FALSE, CAPM = FALSE, BAYES = FALSE){

    rfree  = data$tbills # risk free returns
    rtrns  = data[ ,-grep("tbills", colnames(data))] # expected returns
    ertrns = apply(rtrns, 2, function(x) as.vector(x) - as.vector(rfree)) # rtrns - rfree

    N = NCOL(ertrns) # number of assets
    i = rep(1, N) # necessary column of ones for MVP
    T = NROW(ertrns) # number or periods
    Vhat = cov(rtrns) # covariance of expected returns
    Ehat = apply(ertrns, 2, mean) # expectation of returns

    if(round_returns)
        Ehat = round(Ehat, 2) # round to 2 decimal places

    if(CAPM)
        Ehat = c(0.6, 0.7, 1.2, 0.9, 1.2) * 0.005 # estimates based upon CAPM

    if(portfolio == "tangency"){ # tangency portfolio weights
        weight = solve(Vhat, Ehat) / sum(solve(Vhat, Ehat)) # no identity matrix
        if(identity) # with identity matrix
            weight = solve(Vhat, Ehat) / (Ehat %*% solve(Vhat, Ehat) * 100)
    }

    if(portfolio == "mvp"){ # min var portfolio weights
        weight = solve(Vhat, i) / sum(solve(Vhat, i)) # lay calculation
        if(identity)
            weight = solve(Vhat, i) / (i %*% solve(Vhat, i)) # w/ identity matrix
    }

    if(BAYES){ # layman bayesian estimate
        Ehat = (0.5 * Ehat) + (0.5 * c(0.6, 0.7, 1.2, 0.9, 1.2 * 0.005))
        D = mean(diag(Vhat)) * diag(NCOL(Vhat)) # average of diagnonal * identity matrix
        Vb = 0.5 * Vhat + 0.5 * D # average of two matrices
        weight = solve(Vb, Ehat) / sum(solve(Vb, Ehat)) # no longer have to invert matrix
    }
```

```
    portfolio_rtrn = Ehat %*% weight # E * w'
    portfolio_var  = weight %*% Vhat %*% weight # w' * V * w

    out = data.frame(E_rtrn = portfolio_rtrn, E_var = portfolio_var, t(weight))
    return(list(out, Ehat, Vhat))
}
```

a)

We compute the sample estimates of E and V, $\hat{E}$ and $\hat{V}$.

Table 1: Ehat

XOM	PG	PFE	INTC	WMT
0.00764	0.0061	0.00785	0.0168	0.014

Table 2: Vhat

	XOM	PG	PFE	INTC	WMT
XOM	0.002493	0.000835	0.00114	0.00163	0.000669
PG	0.000835	0.003207	0.00146	0.00156	0.001710
PFE	0.001137	0.001460	0.00504	0.00245	0.001358
INTC	0.001629	0.001563	0.00245	0.01428	0.003140
WMT	0.000669	0.001710	0.00136	0.00314	0.008069

b)

We set the '`portfolio`' flag to '`tangency`', and compute the expected return, variance, and corresponding weights for the tangency portfolio:

Table 3: Tangency

E_rtrn	E_var	XOM	PG	PFE	INTC	WMT
0.0104	0.0023	0.462	0.0534	0.09	0.126	0.268

The above weights are based upon sample estimates. While this procedure may provide unbiased estimators of both E and V, it lacks enough observations to be robust. That is, we only have 528 observations for five assets. As we will confirm later, this is a relatively large number of N(assets) to T(observations), and as such our estimates become unstable. More often than not, we will significantly deviate from the true value of E and V.

c)

We set the '`portfolio`' flag to '`mvp`', and compute the expected return, variance, and corresponding weights for the minimum variance portfolio:

Table 4: Minimum Variance Portfolio

E_rtrn	E_var	XOM	PG	PFE	INTC	WMT
0.0077	0.0017	0.508	0.3	0.115	-0.0118	0.0887

Naturally, because we were only concerned with minimizing the variance and expected return in our optimization routine, our total portfolio variance is much lower. Equally, our expected return is lower to account for the lower amount of variance. We also observed differing weights than the previous tangency portfolio.

d)

Next, we set the '`round returns`' flag TRUE and round returns to two decimal points.

Table 5: Tangency (Rounded) Portfolio

E_rtrn	E_var	XOM	PG	PFE	INTC	WMT
0.0113	0.0019	0.454	0.29	0.0752	0.132	0.0485

For such a seemingly small change in our estimated returns, our total variance, return, and corresponding weights are dramatically different. Given our previous discussion on the pitfalls of using $\hat{E}$ and $\hat{V}$ as estimators of E and V, it's unsurprising that we realize such large deviations when only rounding by two decimal points. We would, however, expect that as our sample size (T) increases to infinity, such rounding errors would cancel out and our estimators would be more robust.

B.2. Estimating V by the identity matrix.

a)

We set the '`identity`' flag to TRUE and use the identity matrix in place of V.

Table 6: Tangency (Identity Matrix) Portfolio

E_rtrn	E_var	XOM	PG	PFE	INTC	WMT
0.01	0.0021	0.443	0.0512	0.0862	0.121	0.257

b)

Next, we set both the '`identity`' and '`round returns`' flag to TRUE and compare differences:

Table 7: Tangency (Identity Matrix Rounded) Portfolio

E_rtrn	E_var	XOM	PG	PFE	INTC	WMT
0.01	0.0015	0.401	0.256	0.0664	0.117	0.0429

As before we notice a slight difference between the estimated mean return, variance, and corresponding weights– although the effect is much less pronounced; each metric is notably more stable than before. This, again, is due to our previously listed problems. That is, when N is large relative to T, our sample covariance matrix, $\hat{V}$, cannot be robustly inverted. And although we may understate correlations by using this method, it is more 'general' in that whatever estimation errors were uncovered from the naive $\hat{V}$ sample estimate, we are able to average away with the off diagonal identity matrix.

B.3. Estimating E using the CAPM

Now we set the '`CAPM`' flag to TRUE and estimate our portfolio metrics:

Table 8: Tangency (CAPM) Portfolio

E_rtrn	E_var	XOM	PG	PFE	INTC	WMT
0.0049	0.0022	0.274	0.134	0.385	-0.0197	0.227

Our portfolio expected return, variance, and corresponding weights are quite dissimilar to the sample estimate. Whereas the naive sample estimate allocated over 46% of the portfolio to a single asset, the CAPM weighted portfolio only allocates at most 39% of the portfolio to a single asset. Equally, the expected return is much lower. These two insights lead us to believe that the CAPM portfolio construction should be a more appropriate estimate of the portfolio parameters.

B.4. Estimating E and V using Bayesian/shrinkage techniques.

We set the '`BAYES`' flag to TRUE, and compute the expected returns, variance, and corresponding weights for the tangency portfolio:

Table 9: Tangency (Bayesian) Portfolio

E_rtrn	E_var	XOM	PG	PFE	INTC	WMT
0.507	0.0027	0.239	0.273	0.45	0.147	-0.109

We ostensibly recover an impressive portfolio return and sensible portfolio weights, albeit by shorting WMT. While we expect the Bayesian estimate to be the most robust, we hypothesize this expected return is either the result of a coding error or massive over-fitting. That is, we know post-ante what the optimal Bayesian priors should be, and subsequently average them with the post-ante average CAPM beta loadings. It's therefore unsurprising how well our returns appear to be doing; what will be more interesting, however, is whether or not we realize similar returns out-of-sample.

B.5. Dynamic portfolio rebalancing.

We write the function '`runStrategy`' that dynamically rebalances a portfolio given input parameters to our previously written '`getPortfolio`' function.

```
runStrategy <- function(data, init_period = 5,  ...){
```

```
    ep = endpoints(data, on = "months")

    result <- list()
    k <- 0; while(TRUE){

        period_end = ep[(1 + init_period * 12) + (k * 12)] # augment the data by period(k)
        if(period_end == tail(ep, 1)) break # end run

data_insample = data[1:period_end]
data_outsample = data[ep[(1 + init_period * 12) + (1 + k * 12)]:ep[(1 + 6 * 12) + (k * 12)]]

forecast_weights = getPortfolio(data_insample, ... = ...)[[1]][-(1:2)] # get weights
real_returns = data_outsample[ ,-grep("tbills", colnames(data_outsample))] # actual returns
period_returns = real_returns %*% t(forecast_weights)

        result[[k + 1]] <- period_returns # store the returns for period k
        k <- k + 1
    }

    portfolio_returns = do.call(rbind, result)
    riskfree = data$tbills[-(1:(init_period * 12))] # first 60 periods is init training

    sharpe_ratio = mean(portfolio_returns - riskfree) / sd(portfolio_returns - riskfree)
    mean_return = mean(portfolio_returns)
    out = data.frame(mean_return, sharpe_ratio)
    return(out)
}
```

```
base = runStrategy(data, portfolio = "tangency")
identity = runStrategy(data, portfolio = "tangency", identity = TRUE)
capm = runStrategy(data, portfolio = "tangency", CAPM = TRUE)
bayes = runStrategy(data, portfolio = "tangency", BAYES = TRUE)

out <- rbind(base, identity, capm, bayes)
rownames(out) <- c("Base", "Identity", "CAPM", "Bayes")
kable(t(out), digits = 6, caption = "Strategy Comparison ")
```

Table 10: Strategy Comparison

	Base	Identity	CAPM	Bayes
mean_return	0.00827	0.00643	0.0131	0.0129
sharpe_ratio	0.01601	0.08449	0.2103	0.1949

As expected, portfolio one and two are close in mean return and Sharpe Ratio. However, the methodology used to compute their correlation structure effected the extent to which they differ. Notably, portfolio two has a much higher Sharpe Ratio, albeit a slightly lower mean return. Both the CAPM and Bayesian approaches yield similar results. Earlier, we hypothesized that our seemingly outlandish in-sample returns generated from a Bayesian portfolio construction could have been due to either a coding error or over-fitting. We affirm our conjecture here. Whereas the one period in-sample portfolio construction yielded abnormally high returns, the Bayesian approach—when applied over multiple periods and tested out-of-sample does slightly worse than a CAPM portfolio allocation.