Statistical Analysis of Portfolios

Eric Zivot

4/12/2021

The GWN Model and Efficient Portfolios

Let R_{it} denote the return (cc or simple) on asset i in month t and assume that R_{it} follows GWN model:

$$\begin{aligned} R_{it} &\sim iid\ N(\mu_i, \sigma_i^2), \\ i &= 1, \ldots, N \text{ (assets)} \\ t &= 1, \ldots, T \text{ (months)} \\ cov(R_{it}, R_{jt}) &= \sigma_{ij} \end{aligned}$$

We estimate the GWN model parameters using sample statistics giving

$$(\hat{\mu}_i, \hat{\sigma}_i^2, \hat{\sigma}_{ij}) \text{ or } (\hat{\mu}, \hat{\Sigma})$$

Remember, the estimates $\hat{\mu}_i, \hat{\sigma}_i^2$ are $\hat{\sigma}_{ij}$ are random variables and are subject to error

Statistical Properties of Efficient portfolios

- Inputs to portfolio theory are estimates from GWN model $\hat{\mu}$ and $\hat{\Sigma}$
- Sharpe ratios and efficient portfolios are functions of $\hat{\mu}$ and $\hat{\Sigma}$.
- The estimated Sharpe ratio is

$$\widehat{\mathrm{SR}}_i = \frac{\hat{\mu}_i - r_f}{\hat{\sigma}_i}$$

- No easy formula for $\mathrm{SE}(\widehat{\mathrm{SR}}_i)$ because it is a nonlinear function of $\hat{\mu}_i$ and $\hat{\sigma}_i$.
- Can use the bootstrap to estimate $\mathrm{SE}(\widehat{\mathrm{SR}}_i)$

Statistical Properties of Efficient portfolios

- The estimated global minimum variance portfolio is

$$\hat{\mathbf{m}} = \frac{\hat{\Sigma}^{-1}\mathbf{1}}{\mathbf{1}'\hat{\Sigma}^{-1}\mathbf{1}}$$

- $\widehat{\mathbf{m}}$ is estimated with error because we estimate Σ using $\hat{\Sigma}$.
- No easy analytic formulas for the standard errors of the elements of $\widehat{\mathbf{m}} = (\hat{m}_1, \ldots, \hat{m}_n)'$; i.e., no easy formula for $\mathrm{SE}(\hat{m}_i)$ because $\hat{\mathbf{m}}$ is a nonlinear function of $\hat{\Sigma}$.
- Can use the bootstrap to estimate $\mathrm{SE}(\hat{m}_i)$

Statistical Properties of Efficient portfolios

- In addition, the expected return and standard deviation of $R_{p,\hat{m}} = \widehat{\mathbf{m}}'\mathbf{R}$ have additional sources of error due to the error in $\widehat{\mathbf{m}}$. That is,

$$
\begin{aligned}
\hat{\mu}_{p,\hat{m}} &= \widehat{\mathbf{m}}'\widehat{\mu} \\
\hat{\sigma}_{p,\hat{m}} &= (\widehat{\mathbf{m}}'\hat{\Sigma}\hat{\mathbf{m}})^{1/2}
\end{aligned}
$$

- No easy analytic formulas for $\mathrm{SE}(\hat{\mu}_{p,\hat{m}})$ and $\mathrm{SE}(\hat{\sigma}_{p,\hat{m}})$
- Can use the bootstrap to estimate $\mathrm{SE}(\hat{\mu}_{p,\hat{m}})$ and $\mathrm{SE}(\hat{\sigma}_{p,\hat{m}})$

Statistical Properties of Efficient portfolios

The estimated minimum variance portfolio with target mean return μ_p^0, $\hat{\mathbf{x}}$, is the first n elements of the vector $\widehat{\mathbf{z}}_x$ where

$$\widehat{\mathbf{z}}_x = \widehat{\mathbf{A}}_x^{-1}\mathbf{b}_0.$$

where

$$\widehat{\mathbf{A}}_x = \begin{pmatrix} 2\hat{\Sigma} & \widehat{\mu} & \mathbf{1} \\ \hat{\mu}' & 0 & 0 \\ \mathbf{1}' & 0 & 0 \end{pmatrix},\ \hat{\mathbf{z}}_x = \begin{pmatrix} \hat{\mathbf{x}} \\ \lambda_1 \\ \lambda_2 \end{pmatrix} \text{ and } \mathbf{b}_0 = \begin{pmatrix} \mathbf{0} \\ \mu_{p,0} \\ 1 \end{pmatrix}$$

Statistical Properties of Efficient portfolios

- $\hat{\mathbf{x}}$ is estimated with error because the sub-elements Σ and μ of $\mathbf{A}$ are estimated using $\hat{\Sigma}$ and $\hat{\mu}$, respectively.
- No easy analytic formulas for the standard errors of the elements of $\widehat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_n)'$; i.e., no easy formula for $\mathrm{SE}(\hat{x}_i)$ because $\hat{\mathbf{x}}$ is a nonlinear function of $\hat{\Sigma}$ and $\hat{\mu}$.
- Also, $\hat{\mu}_{p,\hat{x}} = \hat{\mathbf{x}}'\hat{\mu}$ and $\hat{\sigma}^2_{p,\hat{x}} = \hat{\mathbf{x}}'\Sigma\hat{\mathbf{x}}$ are estimated with error and there are no easy formulas for standard errors.
- The bootstrap can be used to compute SEs.

Statistical Properties of Efficient portfolios

- The entire frontier of mean-variance efficient portfolios is estimated with error because each portfolio on the frontier is also estimated with error.
- The bootstrap can be used to estimate the magnitude of estimation error in the efficient frontier.

Statistical Properties of Efficient portfolios

- The estimated tangency portfolio is

$$\hat{\mathbf{t}} = \frac{\hat{\Sigma}^{-1}(\hat{\mu} - r_f \cdot \mathbf{1})}{\mathbf{1}'\hat{\Sigma}^{-1}(\hat{\mu} - r_f \cdot \mathbf{1})}$$

- $\hat{\mathbf{t}}$ is estimated with error because Σ and μ are estimated with error.
- No easy analytic formulas for the standard errors of the elements of $\widehat{\mathbf{t}} = (\hat{t}_1, \ldots, \hat{t}_n)'$; i.e., no easy formula for $\mathrm{SE}(\hat{t}_i)$ because $\hat{\mathbf{t}}$ is a nonlinear function of $\hat{\Sigma}$ and $\hat{\mu}$.
- The bootstrap can be used to compute SEs.

Optimizers are Error Maximizers

- From our analysis of the GWN model, μ_i is estimated less precisely than σ_i. That is, there is more estimation error in $\hat{\mu}_i$ than $\hat{\sigma}_i$.
- Large estimation error in $\hat{\mu}_i$ greatly impacts efficient portfolios
  - Large positive errors ($\hat{\mu}_i$ much greater than μ_i) leads to efficient portfolios being concentrated in asset i
  - Large negative errors ($\hat{\mu}_i$ much less than μ_i) leads to efficient portfolios that avoid asset i or shorts asset i
- Constraints on portfolio weights can offset the impact of estimation error in $\hat{\mu}_i$

Three asset example data

Microsoft, Nordstrom and Starbucks monthly simple returns over the 5-year period January 1995 to December 1999.

```
data(msftDailyPrices, jwnDailyPrices, sbuxDailyPrices)
gwnDailyPrices = merge(msftDailyPrices, jwnDailyPrices, sbuxDa
gwnMonthlyPrices = to.monthly(gwnDailyPrices, OHLC = FALSE)
gwnReturns = CalculateReturns(gwnMonthlyPrices)
gwnReturns = gwnReturns["1995::1999"]
colnames(gwnReturns)[2] = "NORD"
```

Estimate GWN model parameters and SEs

Estimate parameters:

```
mu.hat = colMeans(gwnReturns)
sd.hat = apply(gwnReturns, 2, sd)
cov.hat = cov(gwnReturns)
cor.hat = cor(gwnReturns)
```

Estimate SEs:

```
n.obs = nrow(gwnReturns)
se.mu.hat = sd.hat/sqrt(n.obs)
se.sd.hat = sd.hat/sqrt(2*n.obs)
```

Estimate GWN model parameters and SEs

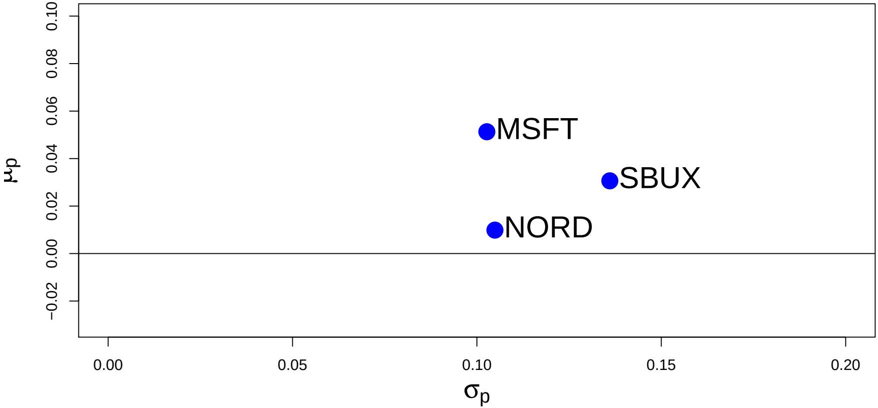
Estimates with standard errors

```r
rbind(mu.hat, se.mu.hat, sd.hat, se.sd.hat)
```

```
##                MSFT    NORD   SBUX
## mu.hat    0.05129 0.00987 0.0306
## se.mu.hat 0.01326 0.01354 0.0176
## sd.hat    0.10269 0.10488 0.1360
## se.sd.hat 0.00937 0.00957 0.0124
```

- Means are not estimated precisely
- Volatilities are estimated more precisely

Risk-Return Characteristics without Estimation Error

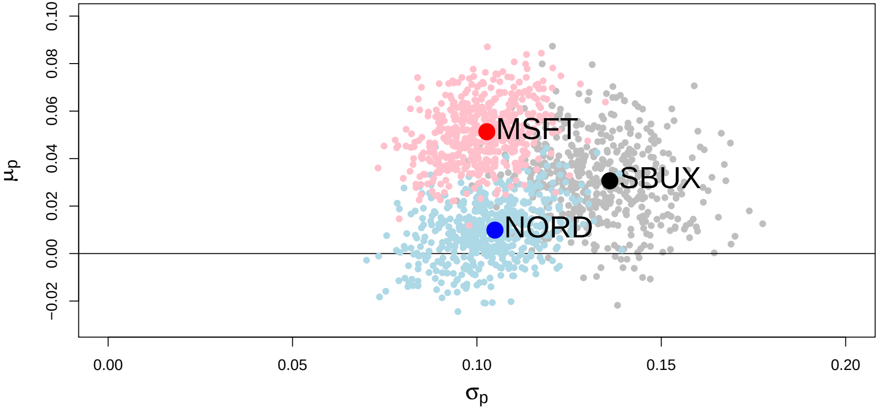


Use Bootstrap to Illustrate Estimation Error

Compute $B = 500$ bootstrap values of $\hat{\mu}_i$ and $\hat{\sigma}_i$.

```
n.boot = 500
mu.boot = matrix(0, n.boot, ncol(gwnReturns))
sd.boot = matrix(0, n.boot, ncol(gwnReturns))
colnames(mu.boot) = colnames(sd.boot) = colnames(gwnReturns)
set.seed(123)
for (i in 1:n.boot) {
  boot.idx = sample(n.obs, replace=TRUE)
  ret.boot = gwnReturns[boot.idx, ]
  mu.boot[i, ] = colMeans(ret.boot)
  sd.boot[i, ] = apply(ret.boot, 2, sd)
}
```

Risk-Return Characteristics with Estimation Error



Risk-Return Characteristics with Estimation Error

Comments:

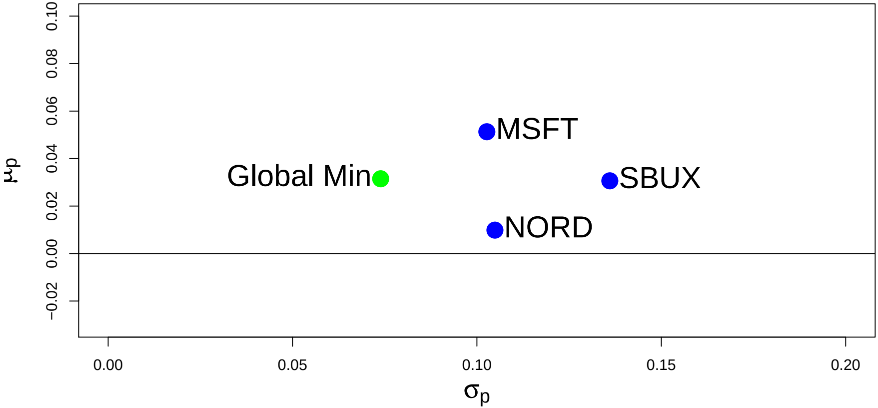
- Sample estimates are big dark circles
- Bootstrap estimates are small light circles
- Estimation error is more prounounced in means than volatilities

Compute Global Minimum Variance Portfolio

```
gmin.port = globalMin.portfolio(mu.hat, cov.hat)
gmin.port
```

```
## Call:
## globalMin.portfolio(er = mu.hat, cov.mat = cov.hat)
##
## Portfolio expected return:     0.0315
## Portfolio standard deviation:  0.0739
## Portfolio weights:
##  MSFT  NORD  SBUX
## 0.428 0.385 0.186
```

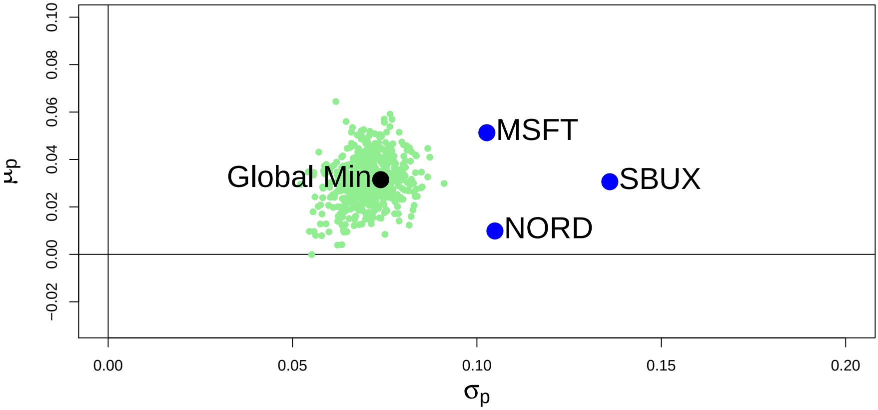
Risk-Return Characteristics without Estimation Error



Bootstrap Global Minimum Variance Portfolio

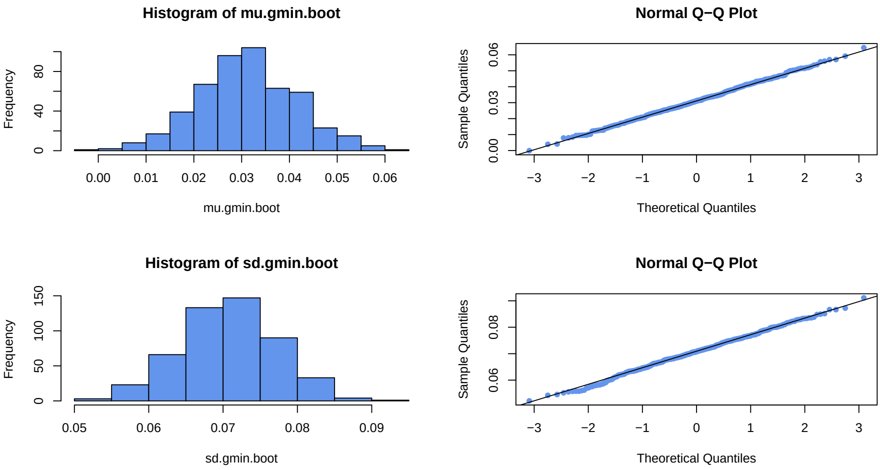
```
mu.gmin.boot = matrix(0, n.boot, 1)
sd.gmin.boot = matrix(0, n.boot, 1)
w.gmin.boot = matrix(0, n.boot, 3)
colnames(mu.gmin.boot) = colnames(sd.gmin.boot) = "global.min'
colnames(w.gmin.boot) = names(mu.hat)
set.seed(123)
for (i in 1:n.boot) {
  boot.idx = sample(n.obs, replace=TRUE)
  ret.boot = gwnReturns[boot.idx, ]
  mu.boot = colMeans(ret.boot)
  cov.boot = cov(ret.boot)
  gmin.boot = globalMin.portfolio(mu.boot, cov.boot)
  mu.gmin.boot[i, ] = gmin.boot$er
  sd.gmin.boot[i, ] = gmin.boot$sd
  w.gmin.boot[i, ] = gmin.boot$weights
}
```

Risk return characteristics with estimation error



- Estimation error is larger in mean than in volatility (cloud is more stretched in vertical direction)

Bootstrap Distribution



Bootstrap bias, SE and 95% CI for mean

```
bias.mu.gmin = mean(mu.gmin.boot) - gmin.port$er
se.mu.gmin = sd(mu.gmin.boot)
ci.mu.gmin.95 = c(gmin.port$er-2*se.mu.gmin,
                  gmin.port$er+2*se.mu.gmin)
names(ci.mu.gmin.95) = c("lower", "upper")
```

Bootstrap bias, SE and 95% CI for mean

Bootstrap estimate of bias for mean is small

```
bias.mu.gmin
```

```
## [1] -0.000316
```

Bootstrap SE for mean is fairly big

```
se.mu.gmin
```

```
## [1] 0.0103
```

Bootstrap 95% CI for mean is pretty wide

```
ci.mu.gmin.95
```

```
##  lower  upper
## 0.0109 0.0521
```

Bootstrap bias, SE and 95% CI for volatility

```
bias.sd.gmin = mean(sd.gmin.boot) - gmin.port$sd
se.sd.gmin = sd(sd.gmin.boot)
ci.sd.gmin.95 = c(gmin.port$sd-2*se.sd.gmin,
                  gmin.port$sd+2*se.sd.gmin)
names(ci.sd.gmin.95) = c("lower", "upper")
```

Bootstrap bias, SE and 95% CI for volatility

Bootstrap estimate of bias for volatility is small

```
bias.sd.gmin
```

```
## [1] -0.00311
```

Bootstrap SE for volatility is fairly small

```
se.sd.gmin
```

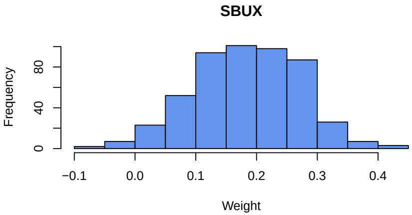
```
## [1] 0.00636
```

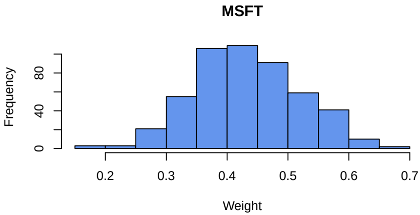
Bootstrap 95% CI for volatility is not so wide

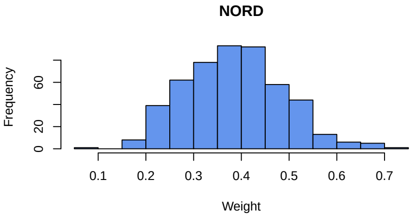
```
ci.sd.gmin.95
```

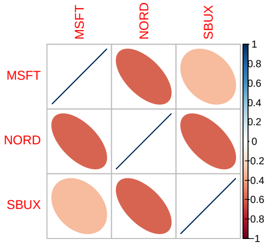
```
##  lower  upper
## 0.0612 0.0866
```

Bootstrap Portfolio Weights









- Weights are negatively correlated (because they sum to one)

Bootstrap bias, SE and 95% CI for weights

```
bias.w.gmin = colMeans(w.gmin.boot) - gmin.port$weights
se.w.gmin = apply(w.gmin.boot, 2, sd)
ci.w.gmin.95 = rbind(gmin.port$weights-2*se.w.gmin,
                     gmin.port$weights+2*se.w.gmin)
rownames(ci.w.gmin.95) = c("lower", "upper")
```

Bootstrap bias, SE and 95% CI for weights

Bootstrap estimates of bias for weights are fairly small

```
bias.w.gmin
```

```
##      MSFT      NORD      SBUX
##  0.003976 -0.000262 -0.003715
```

Bootstrap SEs for weights are moderately large (9%)

```
se.w.gmin
```

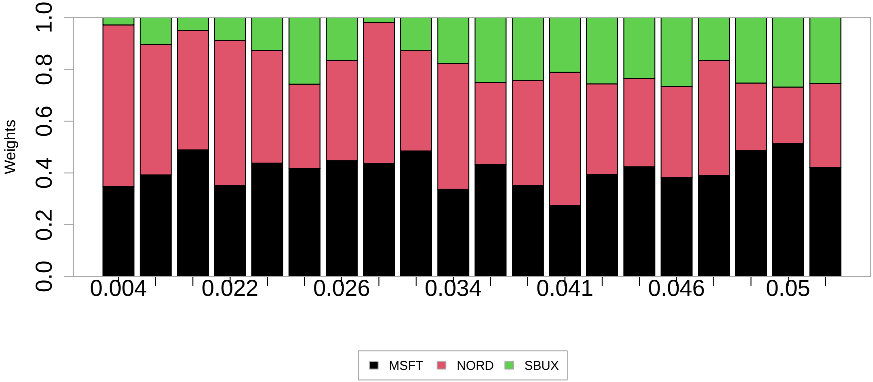
```
##   MSFT   NORD   SBUX
## 0.0864 0.1014 0.0863
```

Bootstrap 95% CIs for weights are wide

```
ci.w.gmin.95
```

```
##         MSFT  NORD   SBUX
## lower 0.256 0.182 0.0139
```

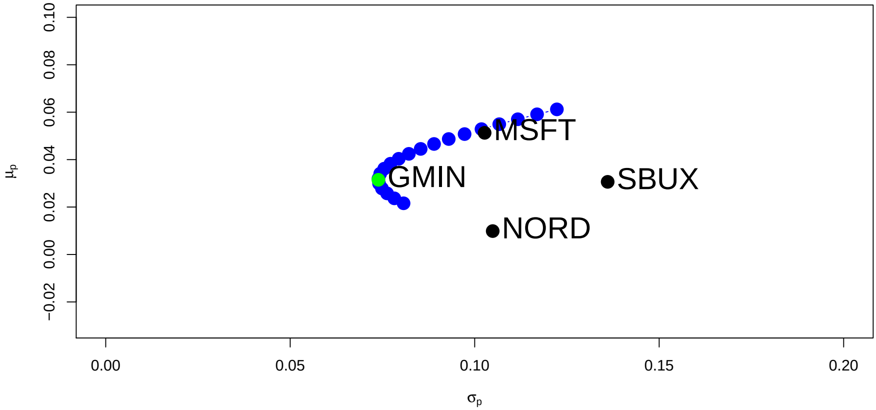
Bootstrap Weights Again



- x-axis is bootstrap mean value of global minimum variance portfolio
- weights vary a bit from portfolio to portfolio - SBUX always has the smallest weight

Efficient Frontier of Risky Assets

```
ef = efficient.frontier(mu.hat, cov.hat)
```

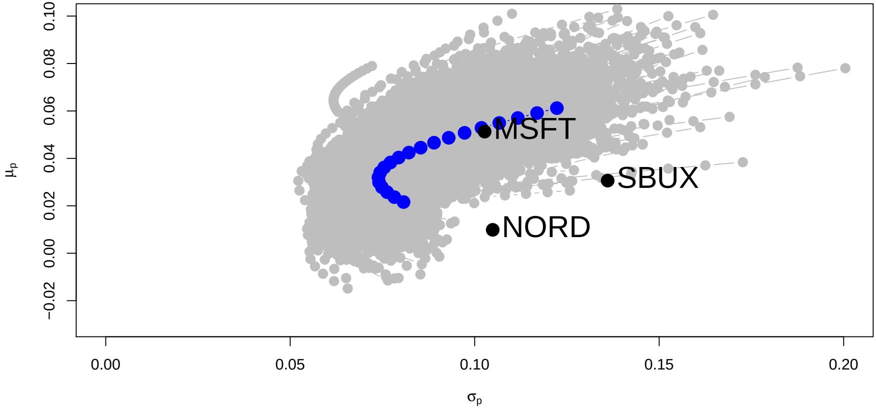


Bootstrap Efficient Frontier

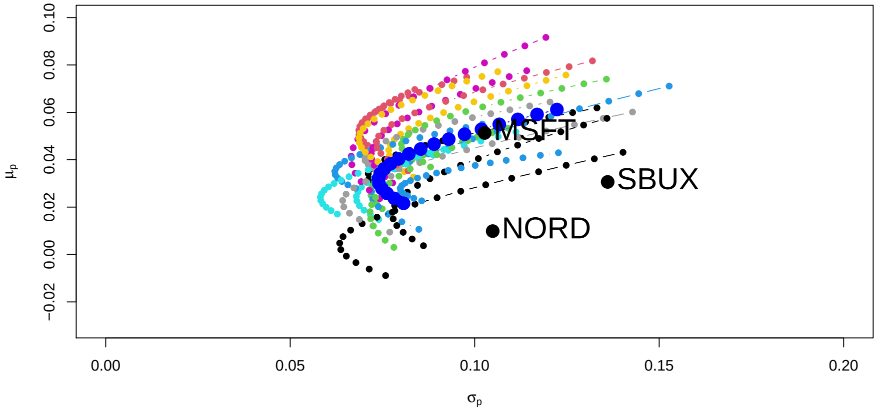
```
ef.list = list()
set.seed(123)
for (i in 1:n.boot) {
  boot.idx = sample(n.obs, replace=TRUE)
  ret.boot = gwnReturns[boot.idx, ]
  mu.boot = colMeans(ret.boot)
  cov.boot = cov(ret.boot)
  ef.boot = efficient.frontier(mu.boot, cov.boot)
  ef.list[[i]] = ef.boot
}
```

- Note: `ef.list` is a list of 500 lists: each component is a different bootstrap efficient frontier.

Bootstrap Frontier Portfolios



A Prettier Picture with 20 Efficient Frontiers



- Sample frontier is the set of large dark blue points

Impacts of Estimation Error for Efficient Portfolios

- Large estimation errors in means of individual assets causes large estimation errors in means of efficient portfolios
- Small estimation errors in standard deviations and correlations does not cause large estimation errors in weights for global minimum variance portfolio
- Large estimation errors in means of individual assets causes large estimation errors in estimated Sharpe ratios
- Large estimation errors in means of individual assets causes large estimation errors in location of efficient frontier of risky assets