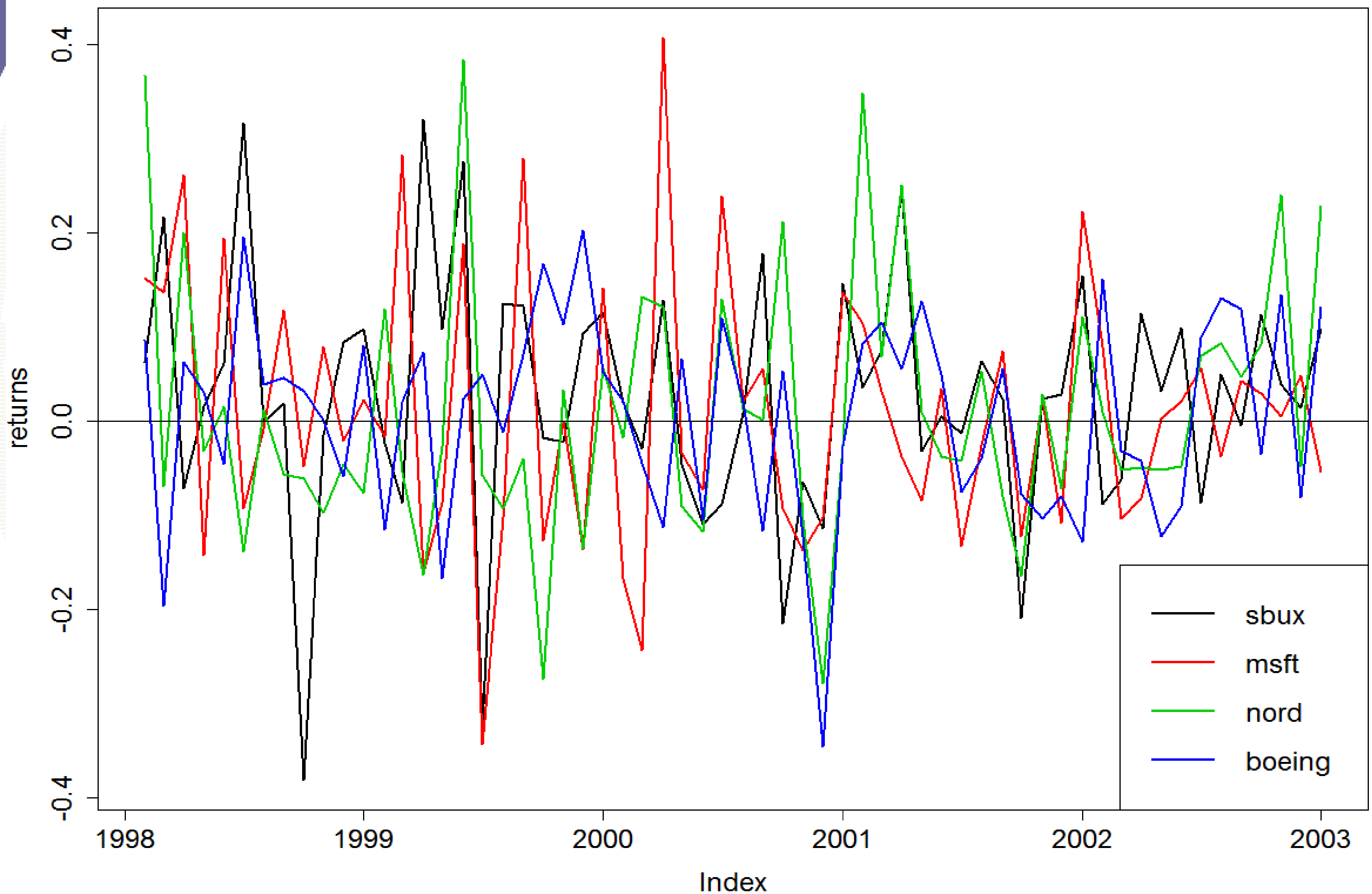
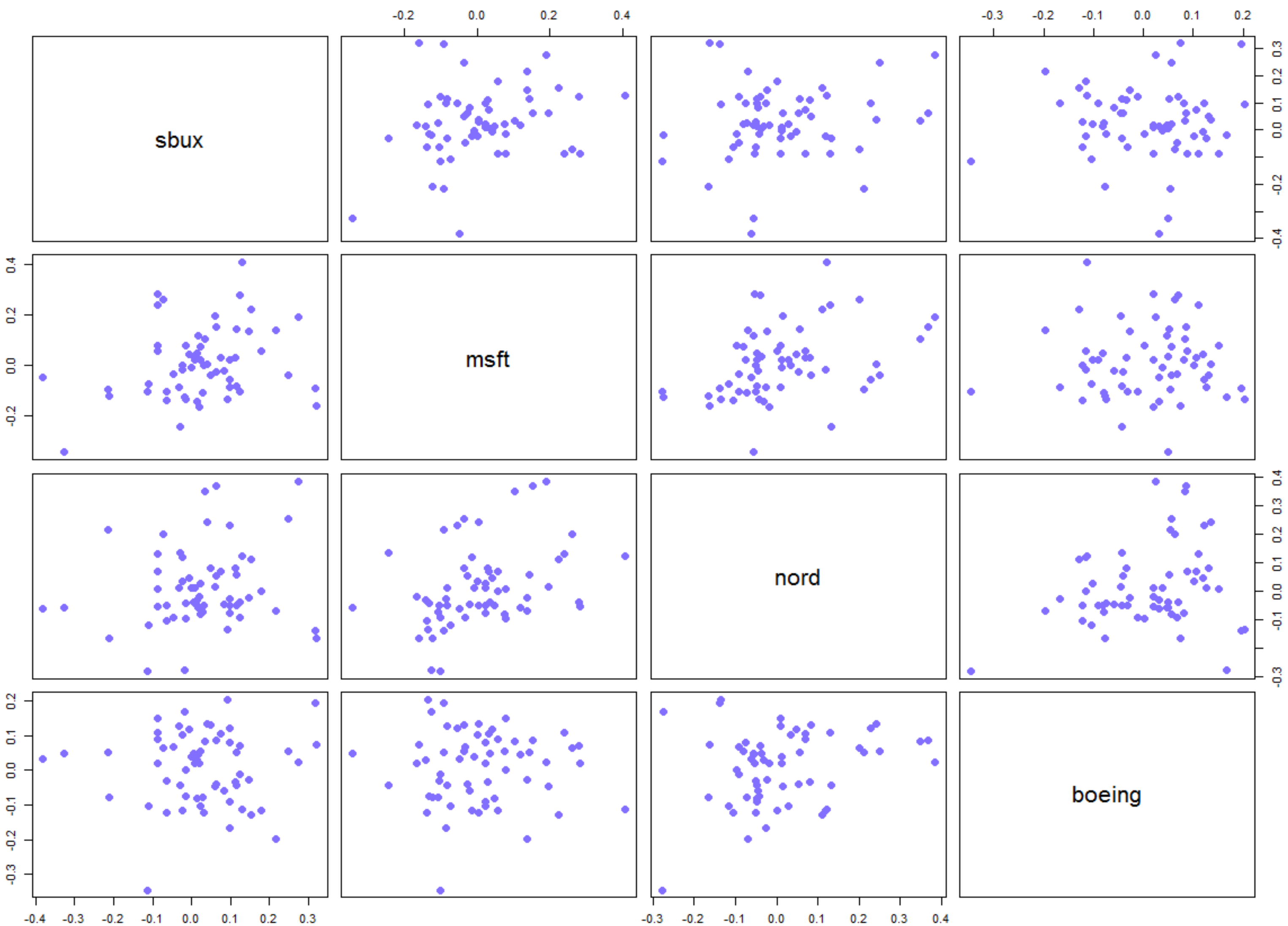
Bootstrapping Efficient Portfolios

Amath 462/Econ 424
Eric Zivot
Summer 2013
Updated: August 21, 2013

Example Data on Four Stocks



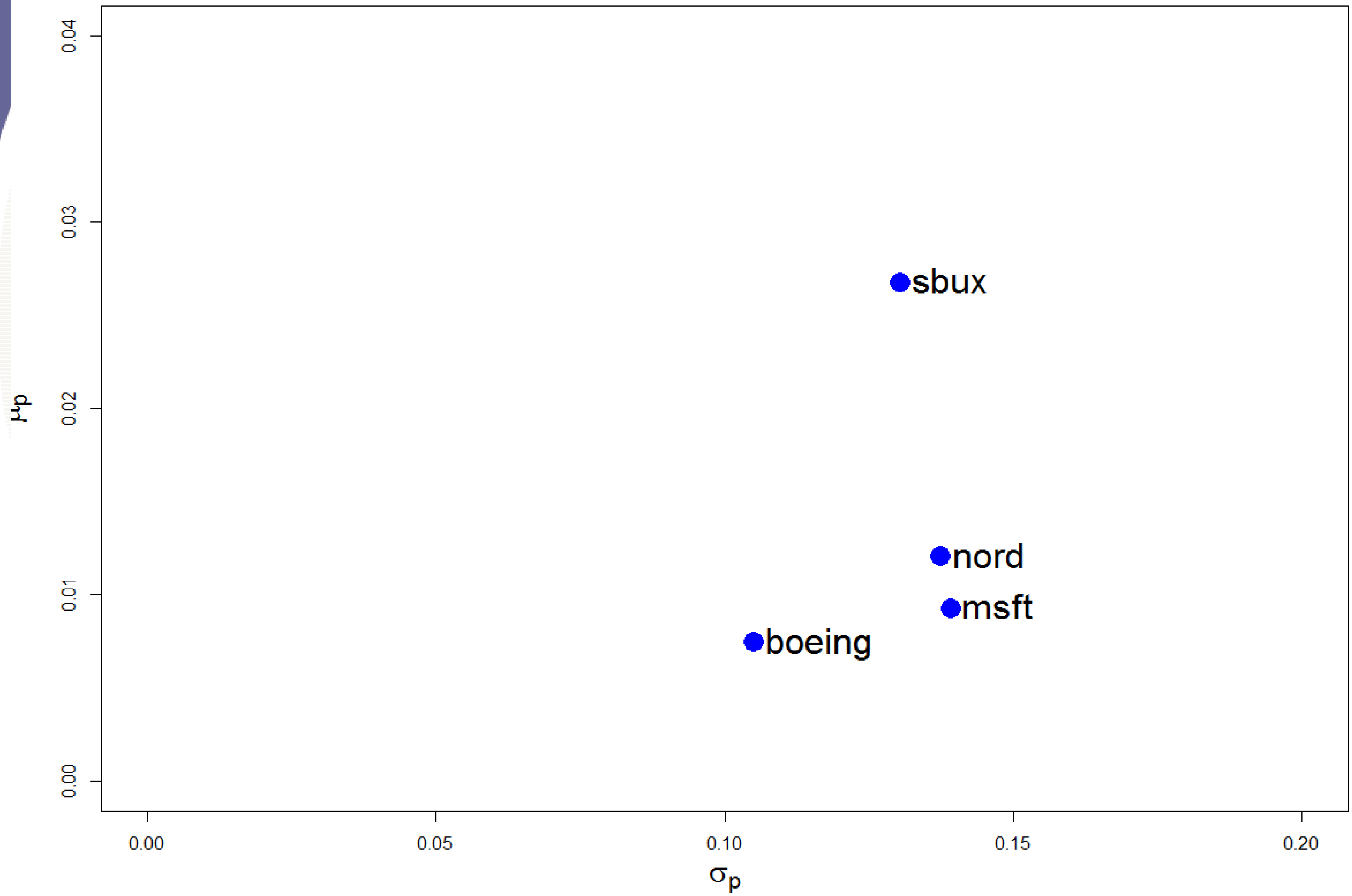
Example Data on Four Stocks



Estimated Inputs

```
# estimated means
> mu.hat
    sbux     msft     nord   boeing
0.026753 0.009256 0.012024 0.007423

# estimated sds
> sd.hat
  sbux   msft   nord boeing
0.1305 0.1391 0.1375 0.1051

# estimated correlations
> cor.hat
          sbux      msft   nord    boeing
sbux   1.00000  0.253079 0.1533  0.016126
msft   0.25308  1.000000 0.3775 -0.006234
nord   0.15327  0.377483 1.0000  0.233900
boeing 0.01613 -0.006234 0.2339  1.000000
```

Estimated Risk Return Tradeoffs

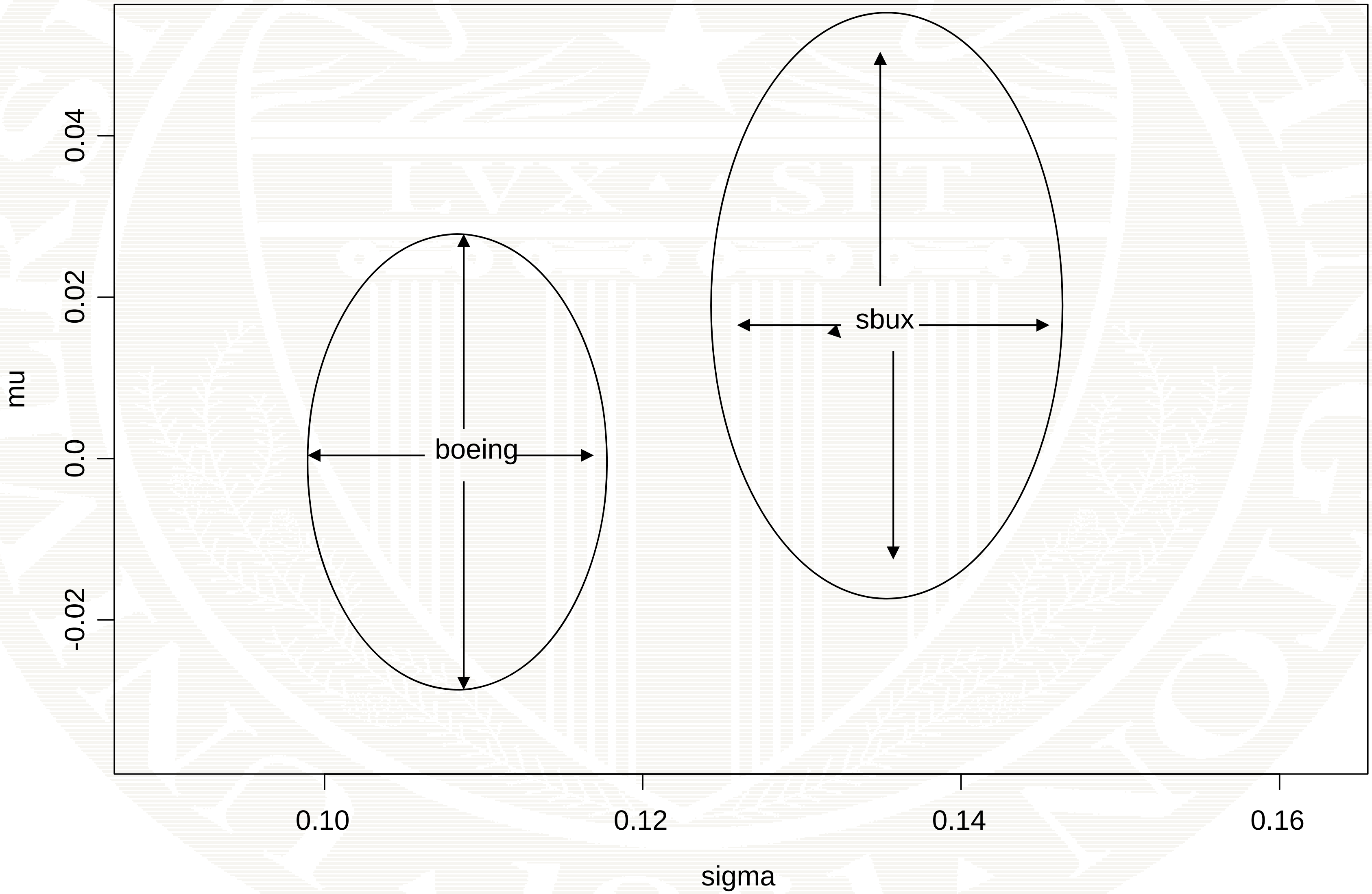


Estimation Error in Inputs

```
# Estimated means with std errors
> rbind(mu.hat, se.mu.hat)
             sbux     msft     nord   boeing
mu.hat    0.02675 0.009256 0.01202 0.007423
se.mu.hat 0.01685 0.017961 0.01775 0.013570

# Estimated sds with std errors
> rbind(sd.hat, se.sd.hat)
             sbux   msft    nord   boeing
sd.hat    0.13052 0.1391 0.13752 0.105111
se.sd.hat 0.01192 0.0127 0.01255 0.009595
```

1. Means *are not* estimated precisely
2. Standard deviations are estimated precisely

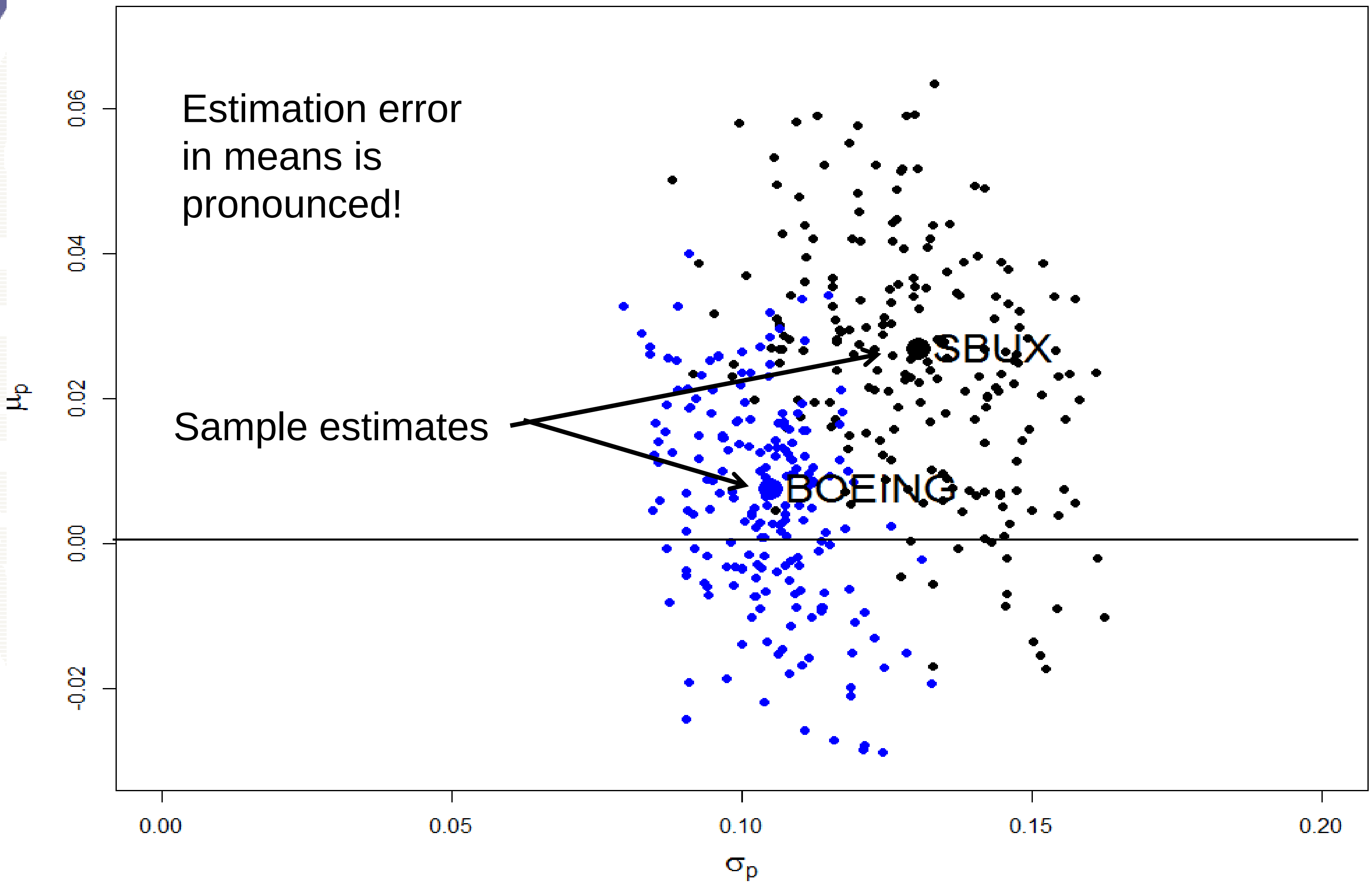
95% Confidence Ellipses for μ and σ



Bootstrapping Means and SD values

```
# re-sample means and sd values
n.boot = 200
mu.boot = matrix(0, n.boot, ncol(ret.mat))
sd.boot = matrix(0, n.boot, ncol(ret.mat))
colnames(mu.boot) = colnames(sd.boot) =
colnames(ret.mat)

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

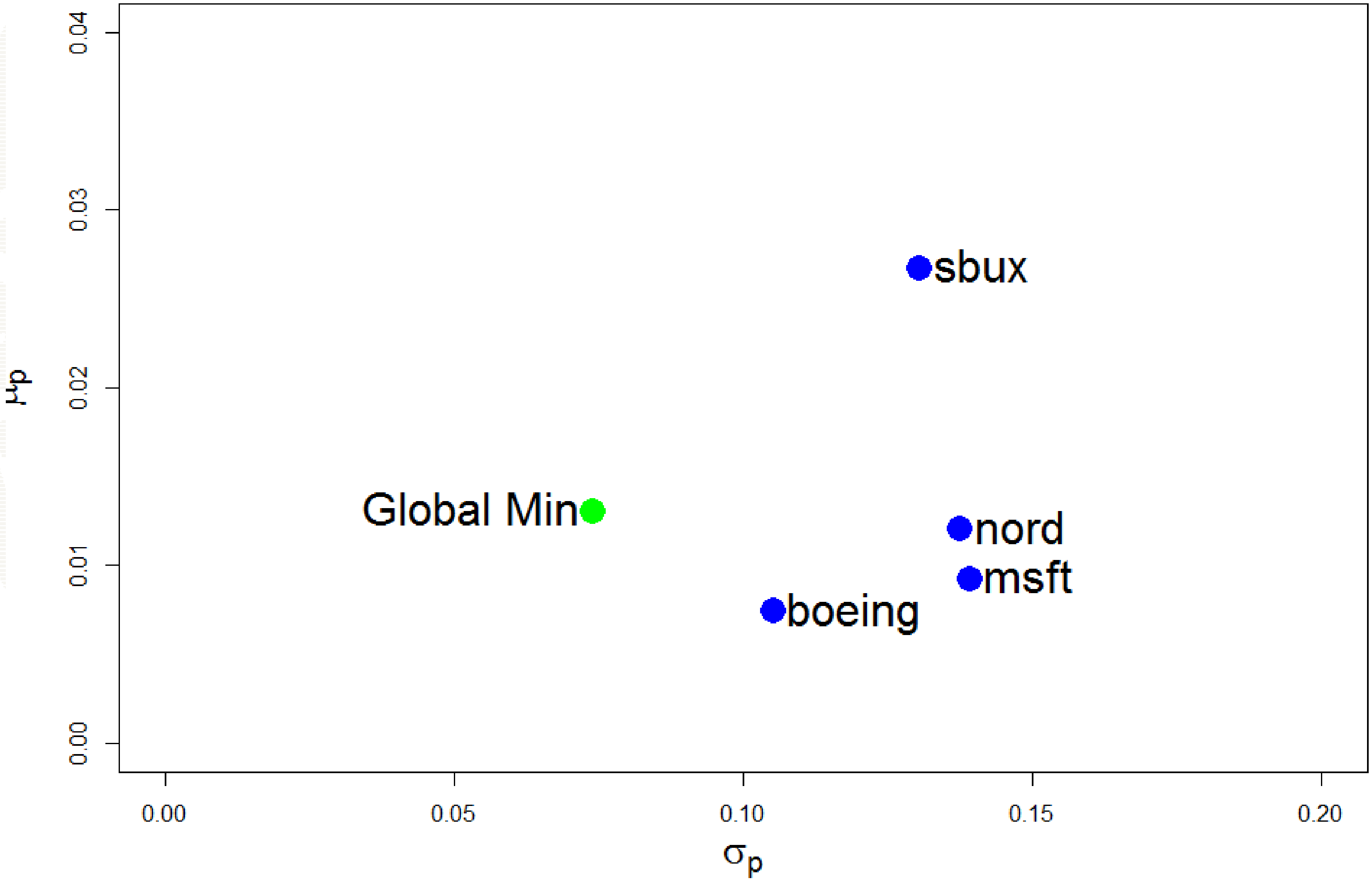
Bootsrapped Means and SD values



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:     0.01303
Portfolio standard deviation:  0.07406
Portfolio weights:
  sbux   msft   nord boeing
0.2489 0.1897 0.0987 0.4628
```

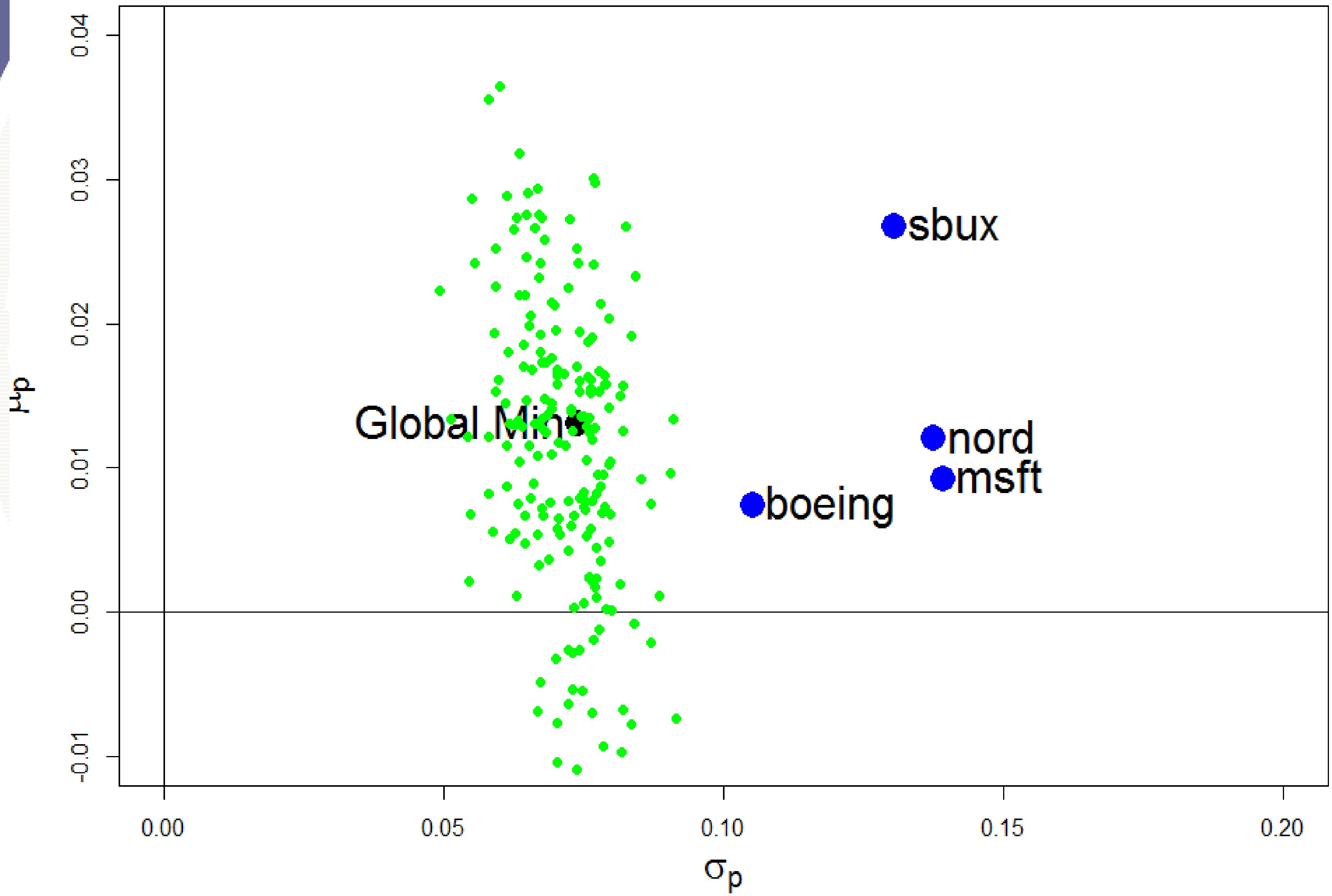
Global Minimum Variance Portfolio



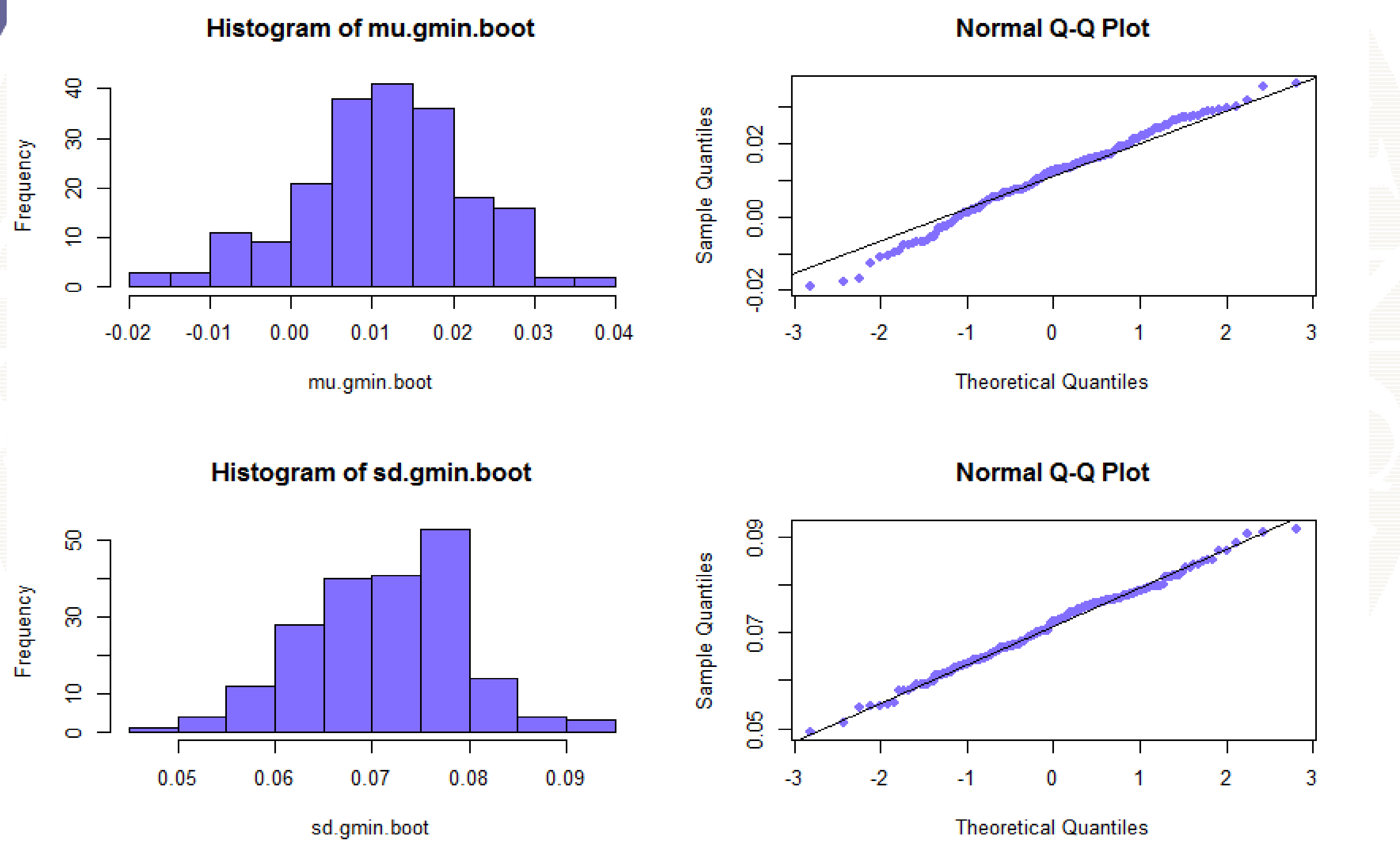
Bootstrapping Global Min Var Portfolio

```
mu.gmin.boot = matrix(0, n.boot, 1)
sd.gmin.boot = matrix(0, n.boot, 1)
w.gmin.boot = matrix(0, n.boot, 4)
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 = ret.mat[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
}
```

Bootstrapping Global Min Var Portfolio



Bootstrap Distribution of Mean and SD



Bootstrap Bias, SE and 95% CI

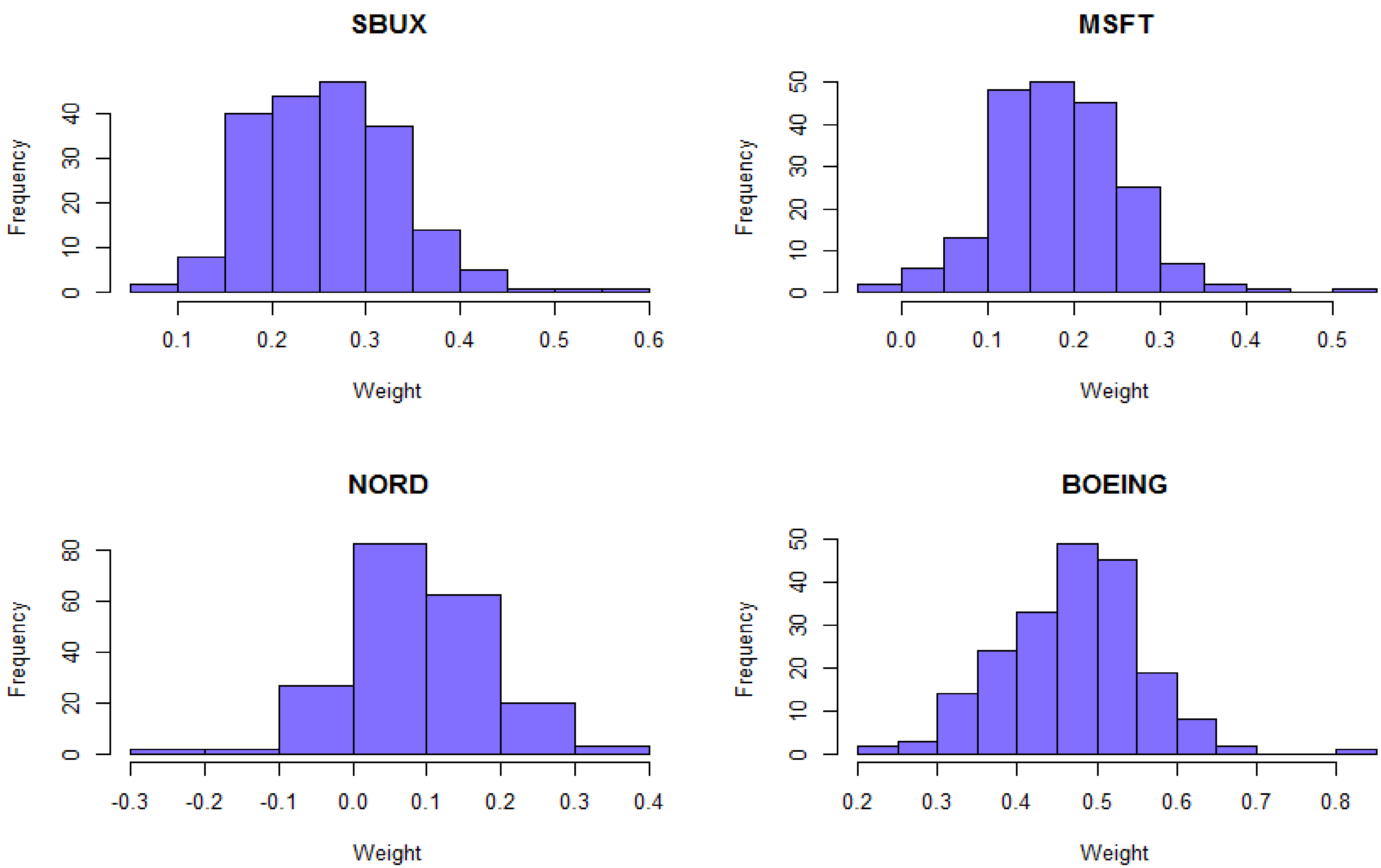
```
> bias.mu.gmin = mean(mu.gmin.boot) - gmin.port$er
> se.mu.gmin = sd(mu.gmin.boot)
> ci.mu.gmin.95 = c(gmin.port$er-2*se.mu.gmin,
+                   gmin.port$er+2*se.mu.gmin)
> bias.mu.gmin
[1] -0.001774
> se.mu.gmin
[1] 0.01051
> ci.mu.gmin.95
[1] -0.007986  0.034056
```

Very wide 95% CI for μ_{gmin}

```
> 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)
> bias.sd.gmin
[1] -0.002672
> se.sd.gmin
[1] 0.007905
> ci.sd.gmin.95
[1] 0.05825 0.08987
```

Not so wide 95% CI for σ_{gmin}

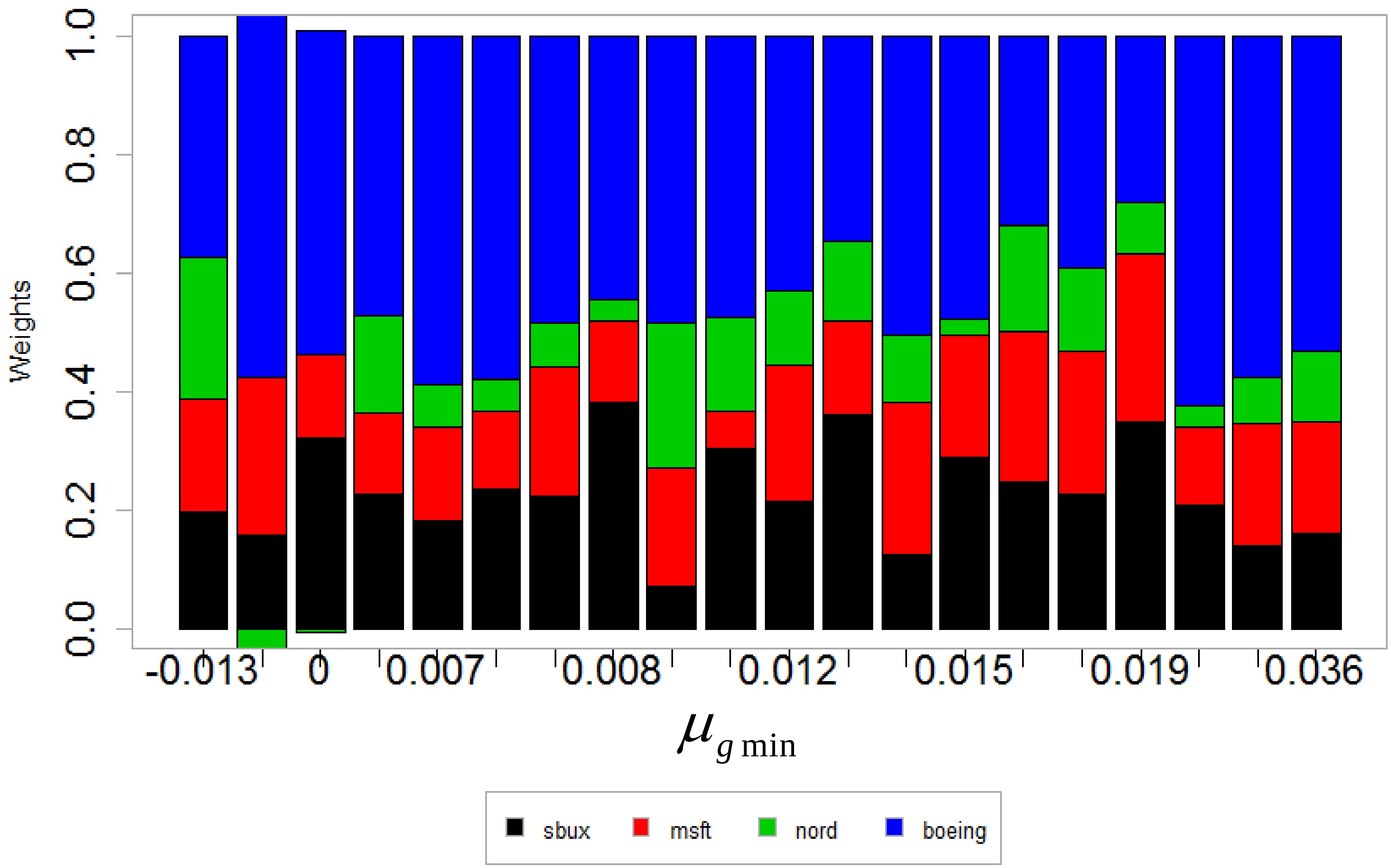
Bootstrap Weights in Min Var Portfolio



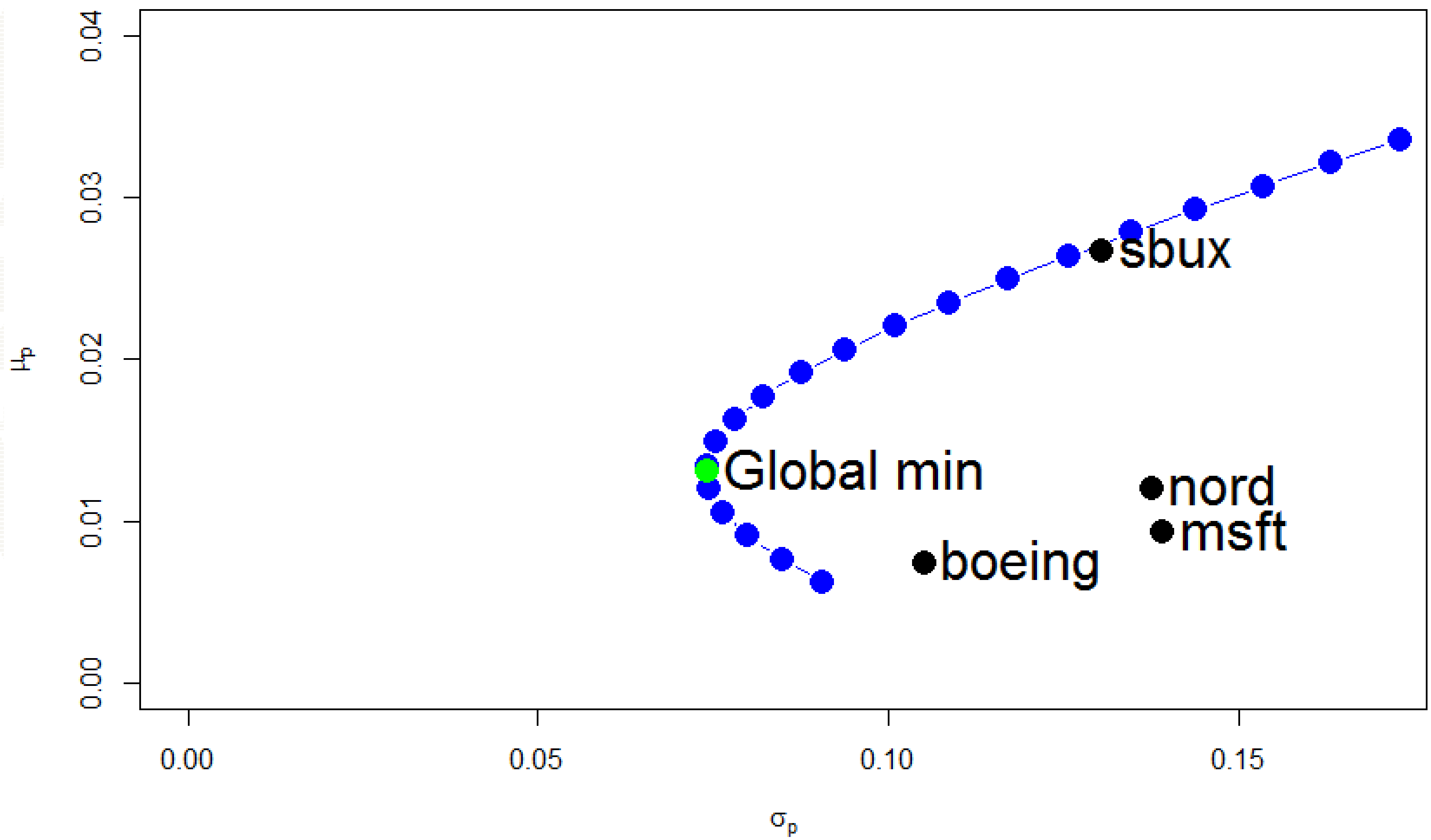
Bias, SE and 95% CI for Weights

```
> 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")
> bias.w.gmin
     sbux      msft      nord    boeing
 0.011067 -0.006630 -0.010480  0.006043

> se.w.gmin
   sbux    msft    nord  boeing
0.07741 0.07742 0.09784 0.08686

> ci.w.gmin.95
         sbux    msft     nord boeing
lower 0.09403 0.03481 -0.09703 0.2891
upper 0.40367 0.34450  0.29433 0.6366
```

Bootstrap Weights in Min Var Portfolio



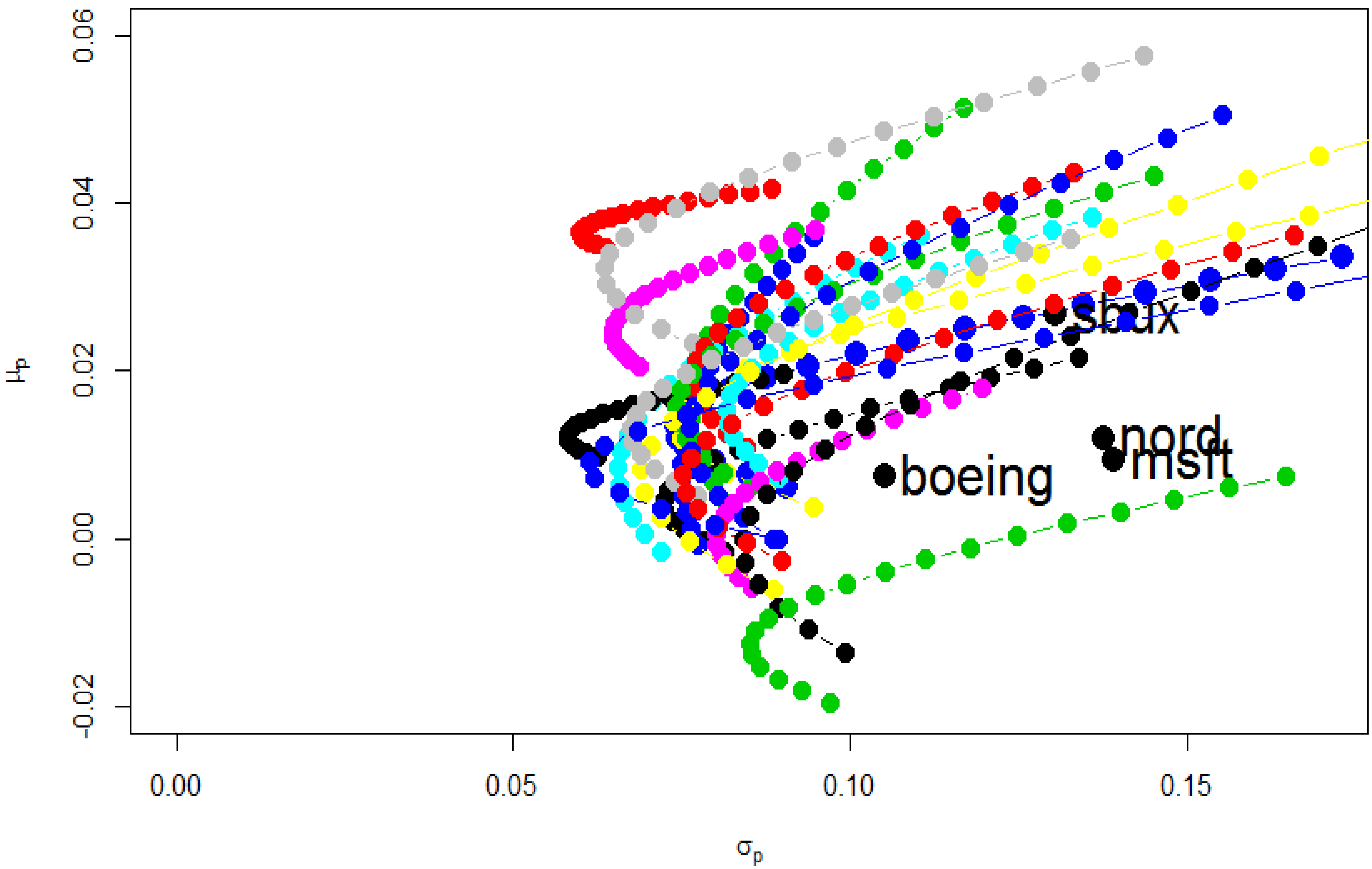
Efficient Frontier of Risky Assets



Bootstrapping the Efficient Frontier

```
# initialize empty list to hold results
> ef.list = list()
> set.seed(123)
> for (i in 1:n.boot) {
+  boot.idx = sample(n.obs, replace=TRUE)
+  ret.boot = ret.mat[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
}
```

Bootstrap Efficient Frontiers



Impacts of Estimation Error in Inputs to Portfolio Theory

- 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 location of efficient frontier of risky assets