Solution 10: Bootstrap

Yaqi Shi

July 22, 2024

Applying bootstrap

The following code generates (X_i, Y_i) pairs.

```r
library(MASS)
generate_pairs <- function(n) {
  # Generate n pairs of financial returns.
  muX <- 2
  muY <- -1
  CovMx <- matrix(c(1, -.25, -.25, 2), nrow = 2)
  data <- mvrnorm(n = 100, mu = c(muX,muY), Sigma = CovMx)
  return(data.frame('X' = data[, 1],
                    'Y' = data[, 2]))
}
```

```r
fin_pairs <- generate_pairs(100) # Generate 100 (X,Y) pairs.
head(fin_pairs)
```

```
##          X          Y
## 1 2.459012  0.1854205
## 2 2.633453  0.7209364
## 3 3.102875 -0.3329833
## 4 3.765494 -2.6460032
## 5 2.643346  0.4392245
## 6 2.808591  0.5727379
```

We are interested in

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}$$

```r
Sigmahat <- cov(fin_pairs)
Sigmahat
```

```
##            X          Y
## X  0.7730102 -0.1072715
## Y -0.1072715  2.0284446
```

```r
sigma2hatXX <- Sigmahat[1,1]
sigma2hatYY <- Sigmahat[2,2]
sigmahatXY <- Sigmahat[1,2]
```

The $\hat{\alpha}$ is

```
alphahat <- (sigma2hatYY - sigmahatXY)/(sigma2hatXX + sigma2hatYY -2*sigmahatXY)
alphahat
```

```
## [1] 0.7081292
```

While the true value of alpha is

```
sigma2XX <- 1
sigma2YY <- 2
sigmaXY <- -0.25
alpha_true <-(sigma2YY - sigmaXY)/(sigma2XX + sigma2YY -2*sigmaXY)
alpha_true
```

```
## [1] 0.6428571
```

Now, again, we're going to resample with replacement from our data, and compute our statistic $\hat{\alpha}$ on each resample. The hope is that these resampled versions of the statistic will resemble the distribution of the statistic evaluated on the original data.

1. Create a function to compute alphahat from a given data set.
2. Resample the data $B = 200$ times, evaluating $\hat{\alpha}$ on each resample. Then, we'll use those resampled values to estimate the variance.
3. Create the confidence interval at the estimate.

Solution

```
compute_alphahat <- function(data) {
  # We're assuming that data is a data frame with two columns.
  Sigmahat <- cov( data )
  # Extract the variance and covariance estimates from the sample covariance
  sigma2hatXX <- Sigmahat[1,1]
  sigma2hatYY <- Sigmahat[2,2]
  sigmahatXY <- Sigmahat[1,2]
  # plug these into the definition of alpha.
  alphahat <- (sigma2hatYY - sigmahatXY)/(sigma2hatXX + sigma2hatYY -2*sigmahatXY)
  return(alphahat)
}

alphahat <- compute_alphahat(fin_pairs)
```

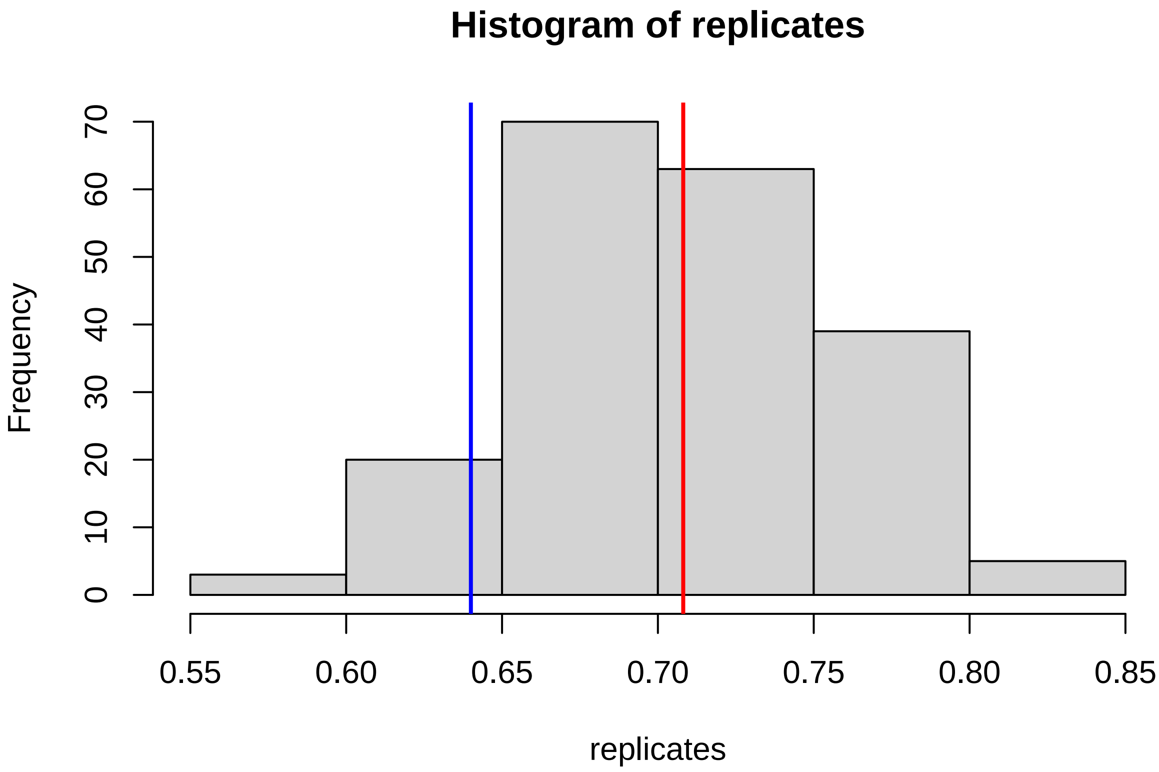
```
B <- 200

replicates <- rep(NA, B)

# number of observations in our data set.
n <- nrow( fin_pairs )

for( i in 1:B ) {
  # To resample the data, we will sample indices, and then grab those rows.
  resample_indices <- sample(1:n, n, replace=TRUE )
  resampled_data <- fin_pairs[resample_indices, ]
  replicates[i] <- compute_alphahat(resampled_data)
}

hist(replicates)
abline(v=alphahat, col='red', lwd=2) # alpha of true data.
abline(v=0.64, col='blue', lwd=2) # True alpha
```



```
# Estimate the variance of alphahat from our bootstrap replicates.
sd_alphahat <- sd(replicates) # estimate of the std dev of alphahat
CI <- c(alphahat - 1.96*sd_alphahat, alphahat + 1.96*sd_alphahat)
CI
```

```
## [1] 0.6133809 0.8028775
```