Bootstrap

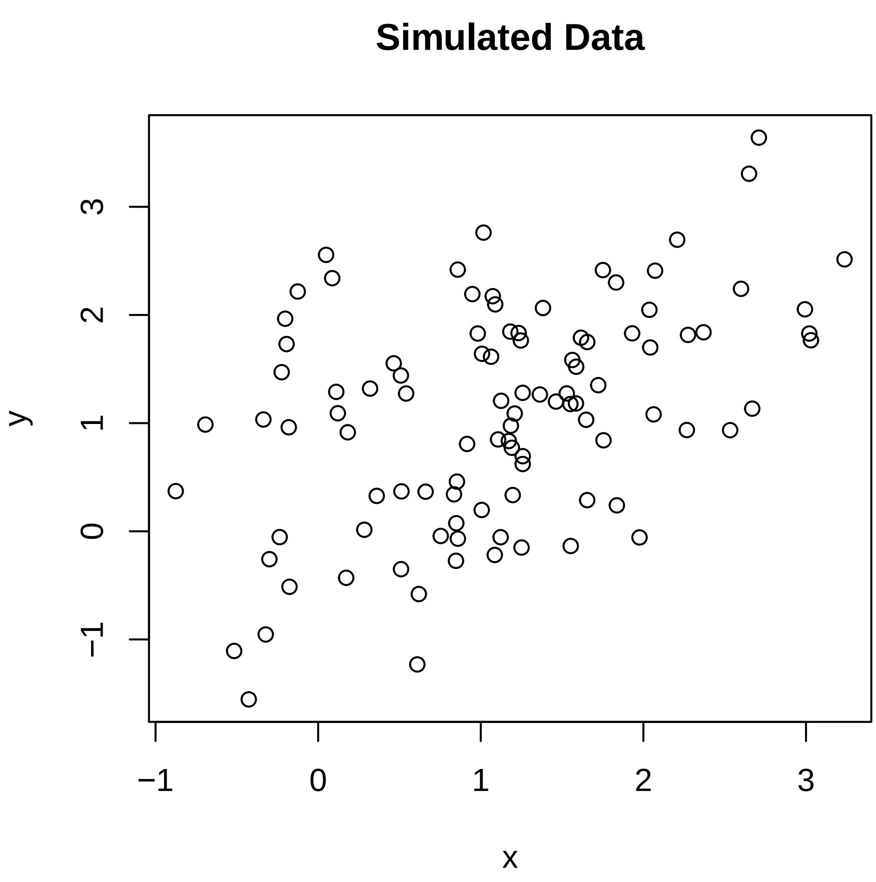
Jason

Tuesday, May 26, 2015

Textbook Practice

```
set.seed(1)
#Package
library(MASS)

mu <- c(1, 1)
Cov <- 0.5*1*sqrt(1.25)
Sigma <- matrix(c(1, Cov, Cov, 1.25), byrow=T, ncol=2)

data <- mvrnorm(100, mu, Sigma)
```

```
plot(data, xlab="x", ylab="y", main="Simulated Data")
```



```r
#Function that outputs alpha
alpha <- function(data){
  sx <- var(data[, 1])
  sy <- var(data[, 2])
  sxy <- cor(data[, 1], data[, 2])
  alpha <- (sy - sxy)/(sx + sy - 2*sxy)
  alpha
}
```

```r
alpha(mvrnorm(100, mu, Sigma, empirical=T))
```

```
## [1] 0.6
```

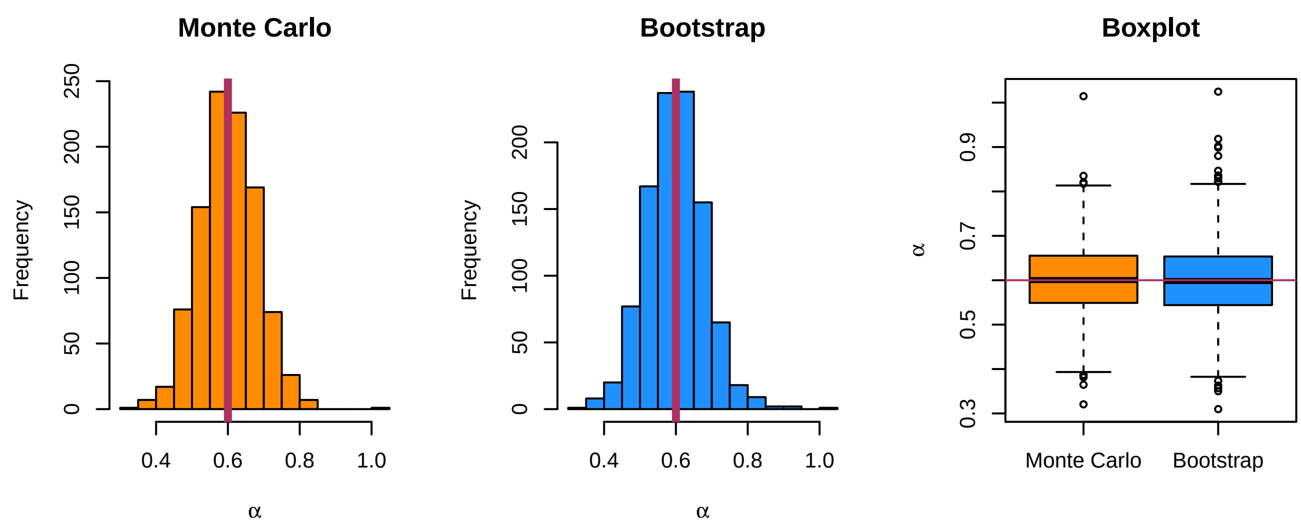
```r
#Know the underlying population
alpha_MC <- c()
for(i in 1:1000){
  alpha_MC[i] <- alpha(mvrnorm(100, mu, Sigma))
}

c(mean(alpha_MC), sd(alpha_MC))
```

```
## [1] 0.60188177 0.08036035
```

```r
#Bootstrap
#Use Simulation data
alpha_BS <- c()
for(i in 1:1000){
  I <- sample(1:100, 100, replace=T)
  alpha_BS[i] <- alpha(data[I, ])
}

c(mean(alpha_BS), sd(alpha_BS))
```

```
## [1] 0.59888314 0.08294589
```

```r
par(mfrow=c(1, 3))
#MC
hist(alpha_MC, main="Monte Carlo", xlab=expression(alpha), col="darkorange")
abline(v=0.6, lwd=4, col="maroon")

#BS
hist(alpha_BS, main="Bootstrap", xlab=expression(alpha), col="dodgerblue")

abline(v=0.6, lwd=4, col="maroon")
#Box plot
boxplot(alpha_MC, alpha_BS, names=c("Monte Carlo", "Bootstrap"), ylab=expression(alpha),
        main="Boxplot", col=c("darkorange", "dodgerblue"))
abline(h=0.6, col="maroon")
```



Lab

```r
#Package
library(boot)
library(ISLR)

#To use boot function, we need to adjust our alpha function

alpha_fn <- function(data, index){
  x <- data[index, 1]
  y <- data[index, 2]
  return((var(y) - cov(x, y))/(var(x) + var(y) - 2*cov(x, y)))
}

boot(Portfolio, alpha_fn, R=1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Portfolio, statistic = alpha_fn, R = 1000)
##
##
## Bootstrap Statistics :
##      original       bias    std. error
## t1* 0.5758321 0.004791436  0.09287509
```

```r
#Coefficient for Regression
coef_fn <- function(data, index){
  coef(lm(mpg ~ horsepower, data=data, subset=index))
```

```r
}

coef_fn(Auto, 1:392)
```

```
## (Intercept)  horsepower 
##  39.9358610  -0.1578447
```

```r
coef_fn(Auto, sample(1:392, 392, replace=TRUE))
```

```
## (Intercept)  horsepower 
##  39.9255340  -0.1545281
```

```r
boot(Auto, coef_fn, R=1000)
```

```
## 
## ORDINARY NONPARAMETRIC BOOTSTRAP
## 
## 
## Call:
## boot(data = Auto, statistic = coef_fn, R = 1000)
## 
## 
## Bootstrap Statistics :
##       original        bias    std. error
## t1* 39.9358610  2.051431e-03 0.850243768
## t2* -0.1578447 -6.952651e-05 0.007388544
```

```r
summary(lm(mpg ~ horsepower, data=Auto))
```

```
## 
## Call:
## lm(formula = mpg ~ horsepower, data = Auto)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -13.5710  -3.2592  -0.3435   2.7630  16.9240 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049 
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

```r
#Categoric example
y <- as.factor(rep(c(0, 2), 50))
x <- rnorm(100, c(0, 2))

glm.fit <- glm(y ~ x, family=binomial)

#Cost function
cost <- function(r, pi){
  mean(abs(r - pi) > 0.5)
}

data <- data.frame(x, y)

cv.glm(data, glm.fit, cost)$delta
```

```
## [1] 0.1700 0.1715
```

```r
cv.glm(data, glm.fit, cost, K=10)$delta
```

```
## [1] 0.180 0.181
```