Computing_HW5_2020321163_엄상준

Eom SangJun

2020 12 8

9. 1.

$X_1, \cdots, X_n$을 $x_1, \cdots, x_n$의 realized value라 했을 때

$R^* = R(\mathcal{X}^*, \hat{F}) = \hat{\theta}^* - \bar{x}$ $\because X_i^* \sim Ber(\bar{x})$

where $\hat{\theta}^* = \sum_{i=1}^{n} x_i^* / n$

$E(\hat{\theta}^*) = \bar{x}$, $V(\hat{\theta}^*) = \bar{x}(1-\bar{x})/n$

$\therefore E(R^*) = 0$

$V(\hat{\theta}^*) = \bar{x}(1-\bar{x})/n$.

```
###############
##9.4
###############
set.seed(1234)

##dataset
fish <- read.table('salmon.dat', header=TRUE)

##Regression Fitting
x <- 1/fish$spawners
y <- 1/fish$recruits
df <- data.frame(x=x, y=y)
rlm <- lm(y ~ x, data = df)

#point estimate
#R=s=(1-beta2)/beta1
point_est = (1-rlm$coefficients[2])/rlm$coefficients[1]
point_est
```

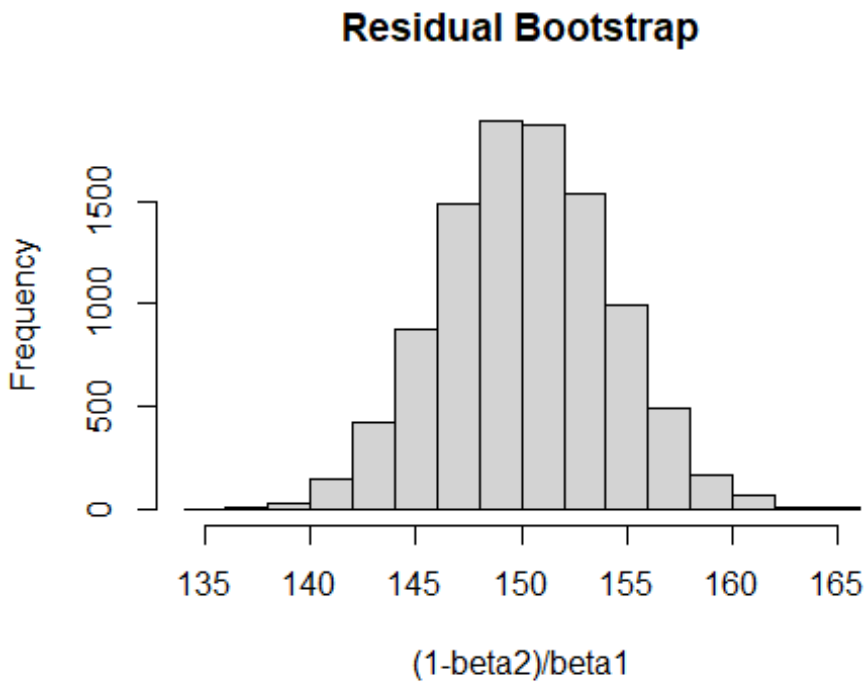
```
##        x
## 150.0976
```

```r
##Bootstrapping the residuals
error <- rlm$residuals
y_hat <- rlm$fitted.values
itr <- 10000
len <- length(y_hat)
est_list1 <- c()
for(i in 1:itr){
  new_error <- error[sample(1:len,len, replace=T)]
  new_y <- y_hat + new_error
  new_df <- data.frame(x=x, y=new_y)
  lm_res <- lm(y ~ x, data=new_df)
  est_list1[i] <- (1-lm_res$coefficients[2])/lm_res$coefficients[1]
}

conf_res <- quantile(est_list1,c(0.025,0.975),na.rm=T)
sd(est_list1)
```

```
## [1] 4.011215
```

```r
hist(est_list1, main='Residual Bootstrap', xlab='(1-beta2)/beta1')
```



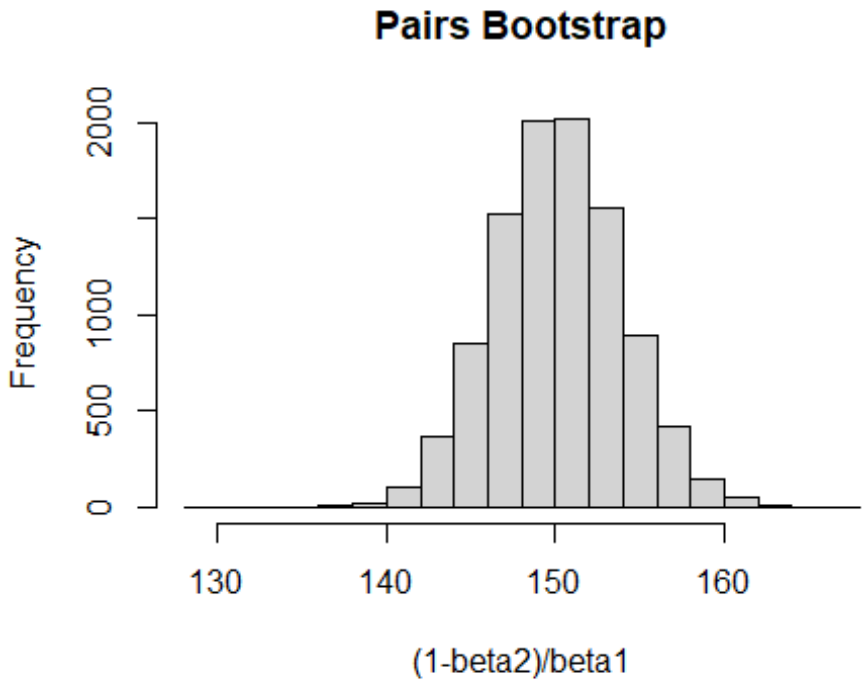

```r
##Bootstrapping the pairs
est_list2 <- c()
for(i in 1:itr){
  ind <- sample(1:len, len, replace=T)
  new_x <- x[ind]
```

```
  new_y <- y[ind]
  new_df <- data.frame(x=new_x, y=new_y)
  lm_pairs <- lm(y ~ x, data=new_df)
  est_list2[i] <- (1-lm_pairs$coefficients[2])/lm_pairs$coefficients[1]
}

conf_pairs <- quantile(est_list2, c(0.025, 0.975), na.rm=T)
sd(est_list2)
```

```
## [1] 3.830465
```

```
hist(est_list2, main='Pairs Bootstrap', xlab='(1-beta2)/beta1')
```




```
#Confidence Interval
conf_res
```

```
##     2.5%    97.5%
## 142.4321 158.0180
```

```
conf_pairs
```

```
##     2.5%    97.5%
## 142.8583 157.8072
```

```
#mean and standard deviation
mean(est_list1)
```

```
## [1] 150.1763
mean(est_list2)
## [1] 150.1353
sd(est_list1)
## [1] 4.011215
sd(est_list2)
## [1] 3.830465
```