BoostrappingWithRobustStandardErrors.R

mbair

Thu Nov 15 13:09:32 2018

Chris Daigle

Residual Bootstrapping

```r
# Exercise
#
# Another bootstrap method in a linear regression model is to
# resample residuals instead of (X, Y). The former is called Residual
# Bootstrapping and the latter is called Pairwise Bootstrapping. I beleve that
# boot() is based on the Pairwise Bootstrap. Estimate the standard error and
# bootstrap the 95% confidence intervals for the coeficient of horsepower with
# Residual Bootstrapping.
library(boot)
library(ISLR)

set.seed(1)

df <- Auto
dfBoot <- df[, c(1, 4, 5)]

# Sample Model ####
X1 <- dfBoot[, 2]
X2 <- dfBoot[, 3]
y <- dfBoot[, 1]

lmFit <- lm(y ~ X1 + X2)

beta0Hat <- lmFit$coefficients[1]
beta1Hat <- lmFit$coefficients[2]
beta2Hat <- lmFit$coefficients[3]

# t values of sample model ####
n <- nrow(dfBoot)
num1 <- sum(((X1 - mean(X1)) ^ 2) * (lmFit$residuals^2))/n
den1 <- sum((X1-mean(X1))^2)/(n-1)
robustSE1 <- sqrt(num1)/den1

num2 <- sum(((X2 - mean(X2)) ^ 2) * (lmFit$residuals^2))/n
den2 <- sum((X2-mean(X2))^2)/(n-1)
robustSE2 <- sqrt(num2)/den2

robT1Coef <- sqrt(n) * beta1Hat / robustSE1
robT2Coef <- sqrt(n) * beta2Hat / robustSE2

t1Coef <- summary(lmFit)$coefficients[2,3]
```

```r
t2Coef <- summary(lmFit)$coefficients[3,3]

# Residual Bootstrap Model
# Initialize relevant values ####
R <- 1000

robustT1Star <- NA
robustT2Star <- NA

t1Star <- NA
t2Star <- NA

# Loop for residual bootstrapping with and without robust standard errors ####
for (r in 1:R) {
  uStar <- sample(lmFit$residual, nrow(dfBoot), replace = TRUE)

  yStar <- beta0Hat + beta1Hat * X1 + beta2Hat * X2 + uStar

  lmFitStar <- lm(yStar ~ X1 + X2)
  beta1Star <- lmFitStar$coefficients[2]
  beta2Star <- lmFitStar$coefficients[3]
  seStar1 <- summary(lmFitStar)$coefficients[2,2]
  seStar2 <- summary(lmFitStar)$coefficients[3,2]

  num1Star <- sum(((X1 - mean(X1)) ^ 2) * (lmFitStar$residuals^2))/n
  den1Star <- sum((X1-mean(X1))^2)/(n-1)
  robustSE1Star <- sqrt(num1Star)/den1Star

  num2Star <- sum(((X2 - mean(X2)) ^ 2) * (lmFitStar$residuals^2))/n
  den2Star <- sum((X2-mean(X2))^2)/(n-1)
  robustSE2Star <- sqrt(num2Star)/den2Star

  robustT1Star[r] <- sqrt(n) * (beta1Star - beta1Hat) / robustSE1Star
  robustT2Star[r] <- sqrt(n) * (beta2Star - beta2Hat) / robustSE2Star

  t1Star[r] <- (beta1Star - beta1Hat) / seStar1
  t2Star[r] <- (beta2Star - beta2Hat) / seStar2
}

# Sort the residual moel t values ####
robustT1Star <- sort(robustT1Star)
robustT2Star <- sort(robustT2Star)

t1Star <- sort(t1Star)
t2Star <- sort(t2Star)

# Get the 2.5% and 97.5% levels ####
critT1Star <- c(t1Star[25], t1Star[975])
critT2Star <- c(t2Star[25], t2Star[975])

critRobustT1Star <- c(robustT1Star[25], robustT1Star[975])
critRobustT2Star <- c(robustT2Star[25], robustT2Star[975])
```
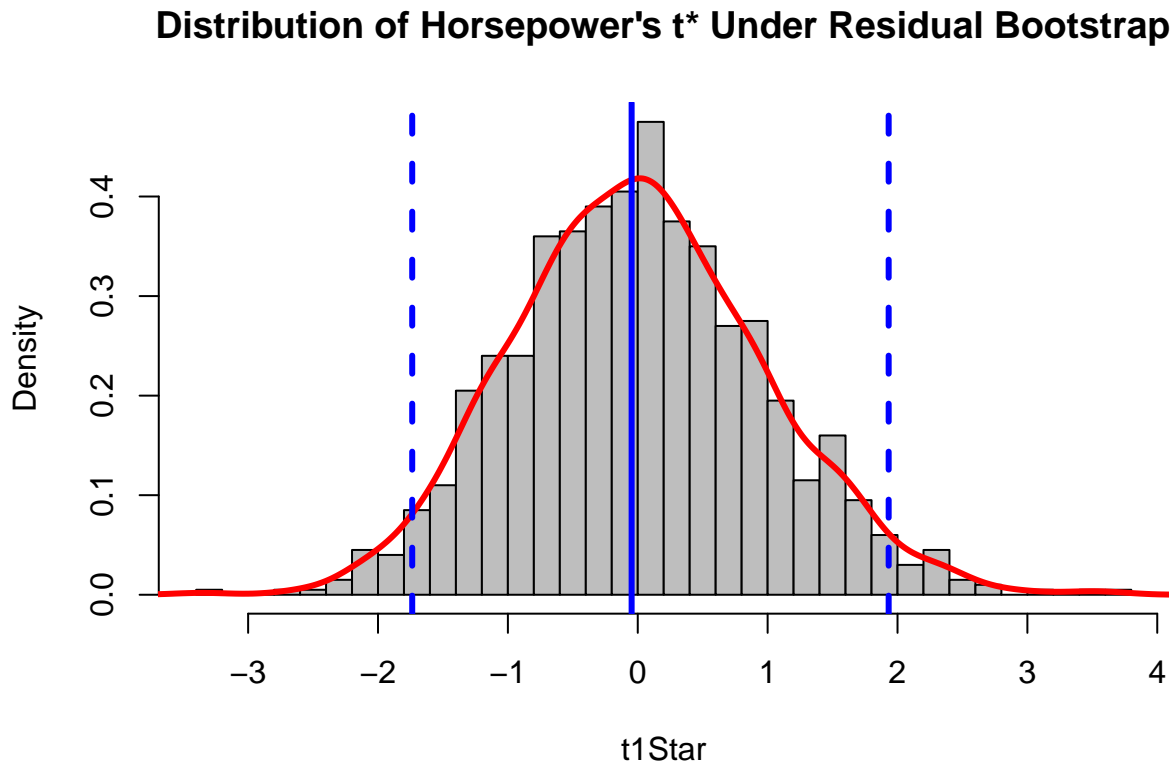
```
# Plot and cut the values ####
hist(t1Star, breaks = 30, probability = TRUE, col= "grey",
     main = "Distribution of Horsepower's t* Under Residual Bootstrap")
lines(density(t1Star), col = "red", lwd = 3)
abline(v = c(critT1Star, beta1Hat), col = c("blue"), lty = c(2, 2, 1), lwd = 3)
```

Distribution of Horsepower's t* Under Residual Bootstrap

t1Star

```
cat(paste("The coefficient of intrest is", round(beta1Hat, 3),
          "and the t-statistic is", round(t1Coef, 3), "."))
```

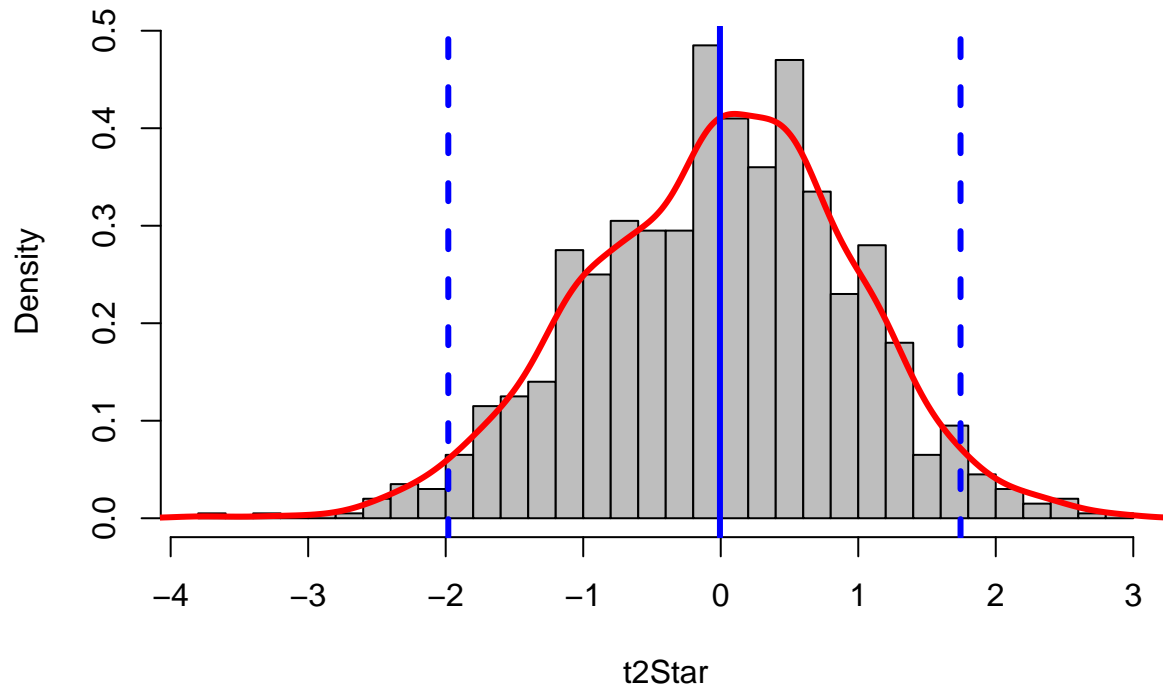```
The coefficient of intrest is -0.047 and the t-statistic is -4.267 .
```

```
cat(paste("The 95% critical values are", round(critT1Star[1], 3),
          "and", round(critT1Star[2], 3), "."))
```

```
## The 95% critical values are -1.737 and 1.932 .
```

```
hist(t2Star, breaks = 30, probability = TRUE, col= "grey",
     main = "Distribution of Weight's t* Under Residual Bootstrap")
lines(density(t2Star), col = "red", lwd = 3)
abline(v = c(critT2Star, beta2Hat), col = c("blue"), lty = c(2, 2, 1), lwd = 3)
```

Distribution of Weight's t* Under Residual Bootstrap


```r
cat(paste("The coefficient of intrest is", round(beta2Hat, 3),
          "and the t-statistic is", round(t2Coef, 3), "."))
```
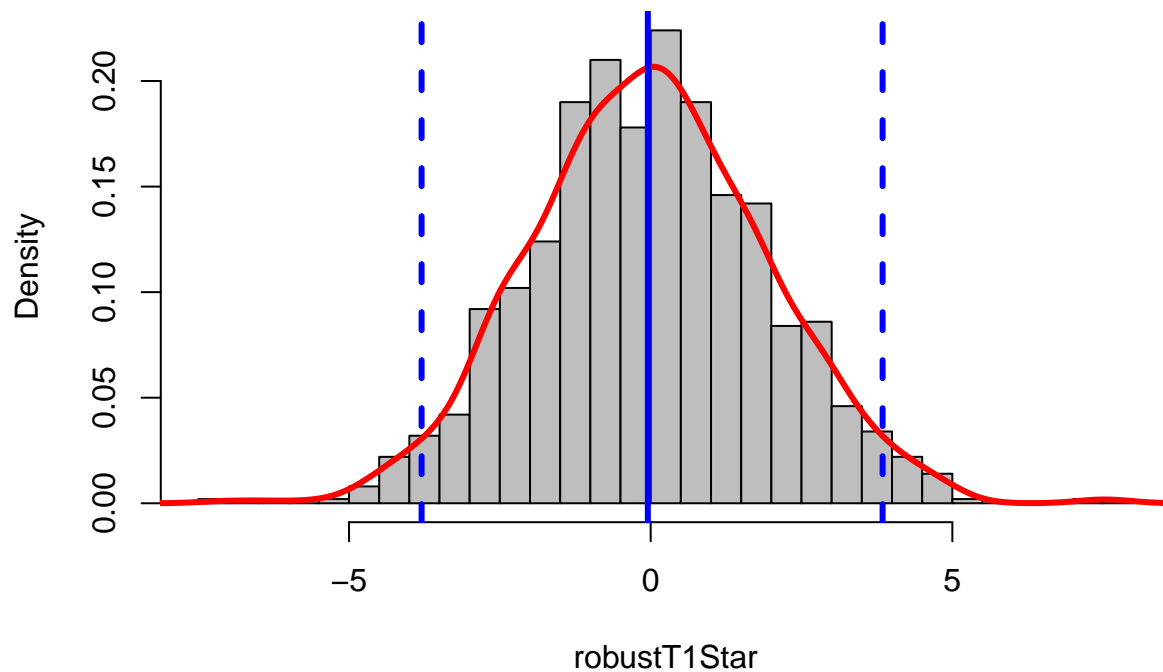
```
## The coefficient of intrest is -0.006 and the t-statistic is -11.535 .
```

```r
cat(paste("The 95% critical values are", round(critT2Star[1], 3),
          "and", round(critT2Star[2], 3), "."))
```

```
## The 95% critical values are -1.981 and 1.744 .
```

```r
hist(robustT1Star, breaks = 30, probability = TRUE, col= "grey",
     main = "Distribution of Horsepower's robust t* Under Residual Bootstrap")
lines(density(robustT1Star), col = "red", lwd = 3)
abline(v = c(critRobustT1Star, beta1Hat), col = c("blue"), lty = c(2, 2, 1), lwd = 3)
```

Distribution of Horsepower's robust t* Under Residual Bootstrap

robustT1Star

```r
cat(paste("The coefficient of intrest is", round(beta1Hat, 3),
          "and the t-statistic is", round(robT1Coef, 3), "."))
```
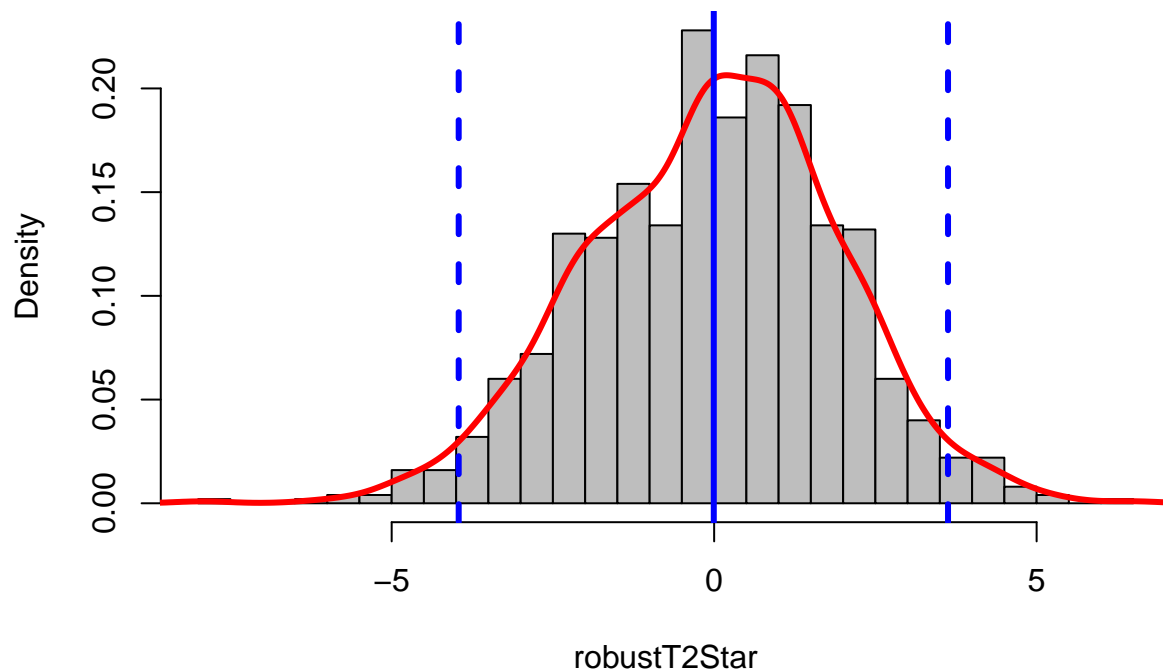
```
## The coefficient of intrest is -0.047 and the t-statistic is -8.49 .
```

```r
cat(paste("The 95% critical values are", round(critRobustT1Star[1], 3),
          "and", round(critRobustT1Star[2], 3), "."))
```

```
## The 95% critical values are -3.797 and 3.844 .
```

```r
hist(robustT2Star, breaks = 30, probability = TRUE, col= "grey",
     main = "Distribution of Weight's robust t* Under Residual Bootstrap")
lines(density(robustT2Star), col = "red", lwd = 3)
abline(v = c(critRobustT2Star, beta2Hat), col = c("blue"), lty = c(2, 2, 1), lwd = 3)
```

Distribution of Weight's robust t* Under Residual Bootstrap

robustT2Star

```r
cat(paste("The coefficient of intrest is", round(beta2Hat, 3),
          "and the t-statistic is", round(robT2Coef, 3), "."))
```

```
## The coefficient of intrest is -0.006 and the t-statistic is -24.852 .
```

```r
cat(paste("The 95% critical values are", round(critRobustT2Star[1], 3),
          "and", round(critRobustT2Star[2], 3), "."))
```

```
## The 95% critical values are -3.961 and 3.626 .
```