STATS 413 Hw5

Shu Zhou

2020/11/15

This is the Assignment 5 of STATS 413 Author: Shu Zhou UMID: 19342932

Exercise 7.1

The estimate of coefficients, standard errors, F-tests are the same between Sue's and Joe's analyses. However, the σ^2 of Joe's analysis is two times of Sue's.

Exercise 7.10

(7.10.1)

```r
fuel2001<-read.csv("fuel2001.csv")
model_OLS <- lm(FuelC ~ Tax+Drivers+Income+log(Miles), data = fuel2001)
#bootstrap
boot1 <- Boot(model_OLS, R=999)
confint(boot1, type="bca")
```

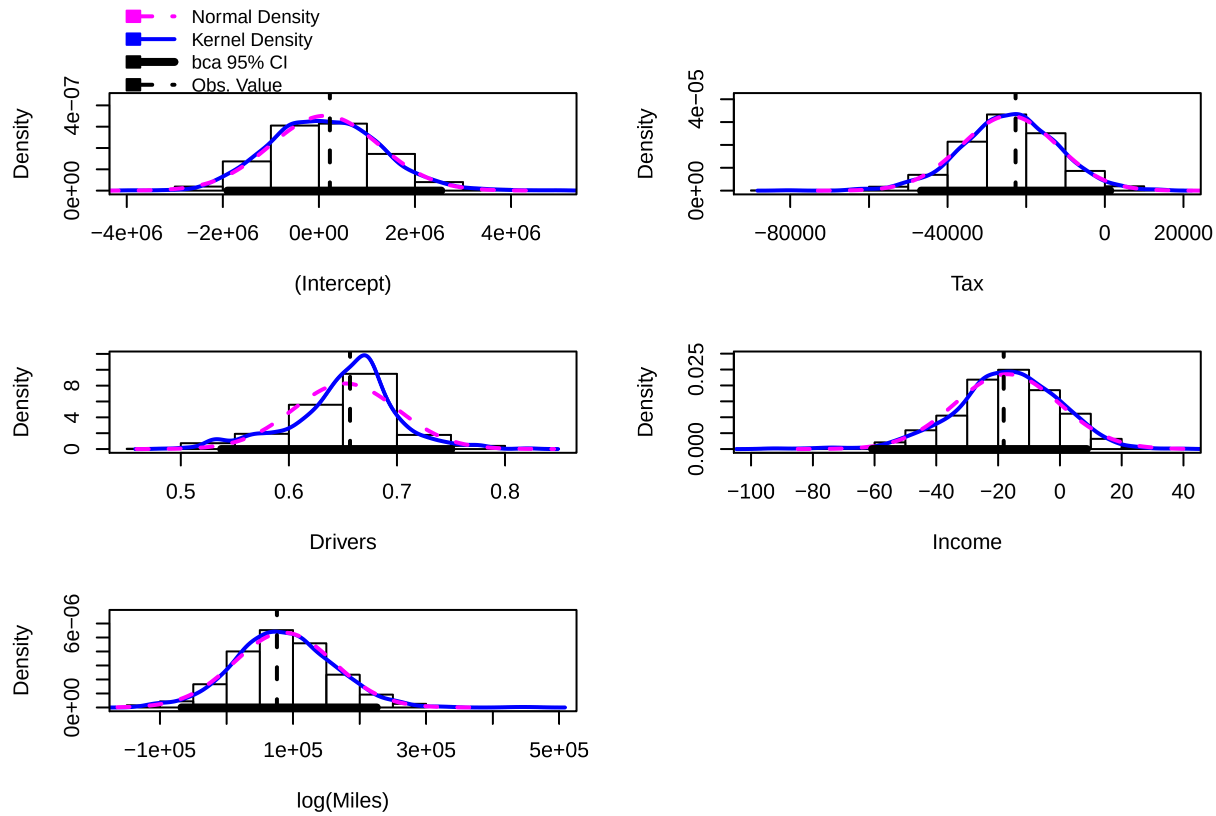
```
## Bootstrap bca confidence intervals
##
##                     2.5 %       97.5 %
## (Intercept) -1.905686e+06 2.544796e+06
## Tax         -4.675186e+04 1.388159e+03
## Drivers      5.373065e-01 7.503208e-01
## Income      -6.077791e+01 8.737136e+00
## log(Miles)  -6.775023e+04 2.258179e+05
```

```r
# Compare with normal
confint(model_OLS)
```

```
##                     2.5 %       97.5 %
## (Intercept) -2.226912e+06 2.681625e+06
## Tax         -5.160861e+04 6.208566e+03
## Drivers      6.123492e-01 7.008500e-01
## Income      -5.331132e+01 1.691998e+01
## log(Miles)  -9.536430e+04 2.469525e+05
```

(7.10.2)_

```r
hist(boot1)
```



>ISLR:

Exercise 8

Part a)

```
set.seed(1)
y <- rnorm(100)
x <- rnorm(100)
y <- x - 2*x^2 + rnorm(100)
```

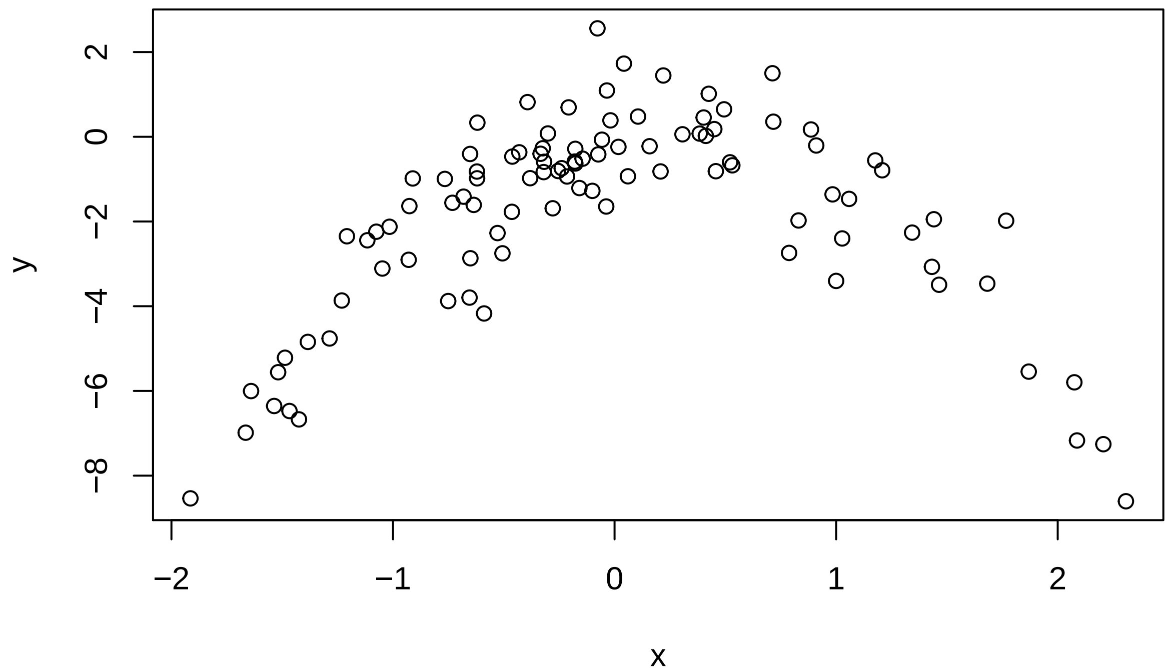
$n = 100$ are observations

$p = 2$ are features

$Y = X - 2X^2 + \epsilon$

Part b)

```
plot(x, y)
```



There is a quadratic relationship between x and y,

Part c)

```
set.seed(1)
df <- data.frame(y, x, x2=x^2, x3=x^3, x4=x^4)
fit1 <- glm(y ~ x, data=df)
cv.err1 <- cv.glm(df, fit1)
cv.err1$delta
```

```
## [1] 5.890979 5.888812
```

```
fit2 <- glm(y ~ x + x2, data=df)
cv.err2 <- cv.glm(df, fit2)
cv.err2$delta
```

```
## [1] 1.086596 1.086326
```

```
fit3 <- glm(y ~ x + x2 + x3, data=df)
cv.err3 <- cv.glm(df, fit3)
cv.err3$delta
```

```
## [1] 1.102585 1.102227
```

```
fit4 <- glm(y ~ x + x2 + x3 + x4, data=df)
cv.err4 <- cv.glm(df, fit4)
cv.err4$delta
```

```
## [1] 1.114772 1.114334
```

Part d)

```r
set.seed(2020)
df <- data.frame(y, x, x2=x^2, x3=x^3, x4=x^4)
fit1 <- glm(y ~ x, data=df)
cv.err1 <- cv.glm(df, fit1)
cv.err1$delta
```

```
## [1] 5.890979 5.888812
```

```r
fit2 <- glm(y ~ x + x2, data=df)
cv.err2 <- cv.glm(df, fit2)
cv.err2$delta
```

```
## [1] 1.086596 1.086326
```

```r
fit3 <- glm(y ~ x + x2 + x3, data=df)
cv.err3 <- cv.glm(df, fit3)
cv.err3$delta
```

```
## [1] 1.102585 1.102227
```

```r
fit4 <- glm(y ~ x + x2 + x3 + x4, data=df)
cv.err4 <- cv.glm(df, fit4)
cv.err4$delta
```

```
## [1] 1.114772 1.114334
```

The results are the same. Since the LOOCV methods uses all the other methods as the reference for prediction.

Part e)

Model (ii) using X and X^2 had the lowest error, which shows that our prediction is correct. Since the true model was generated using a quadratic formula.

Part f)

```r
fit0 <- lm(y ~ poly(x,4))
summary(fit0)
```

```
##
## Call:
## lm(formula = y ~ poly(x, 4))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8914 -0.5244  0.0749  0.5932  2.7796
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.8277     0.1041 -17.549   <2e-16 ***
## poly(x, 4)1   2.3164     1.0415   2.224   0.0285 *
## poly(x, 4)2 -21.0586     1.0415 -20.220   <2e-16 ***
## poly(x, 4)3  -0.3048     1.0415  -0.293   0.7704
## poly(x, 4)4  -0.4926     1.0415  -0.473   0.6373
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.041 on 95 degrees of freedom
## Multiple R-squared:  0.8134, Adjusted R-squared:  0.8055
## F-statistic: 103.5 on 4 and 95 DF,  p-value: < 2.2e-16
```

Summary shows that only X and X^2 are statistically significant predictors. Which agrees with our cross-validation results in part (e).

Exercise 9

Part a)

```
data(Boston)
mu <- mean(Boston$medv)
mu
```

```
## [1] 22.53281
```

Part b)

```
sd <- sd(Boston$medv)/sqrt(nrow(Boston))
sd
```

```
## [1] 0.4088611
```

The standard error of the sample mean is equal to the standard deviation of the dataset divided by the number of observations. **Part c)**

```
set.seed(1)
mean.fn <- function(var, id) {
  return(mean(var[id]))
}

boot.res <- boot(Boston$medv, mean.fn, R=200)
boot.res
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Boston$medv, statistic = mean.fn, R = 200)
##
##
## Bootstrap Statistics :
##     original      bias    std. error
## t1* 22.53281 0.04741601   0.3971741
```

From bootstrap with R=200, our estimation of the std.err is 0.43, which is close to 0.41.

Part d)

```
boot.res$t0 - 2*sd(boot.res$t)  # lower bound
```

```
## [1] 21.73846
```

```
boot.res$t0 + 2*sd(boot.res$t)  # upper bound
```

```
## [1] 23.32715
```

```
t.test(Boston$medv)
```

```
##
##  One Sample t-test
##
## data:  Boston$medv
```

```
## t = 55.111, df = 505, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  21.72953 23.33608
## sample estimates:
## mean of x
##  22.53281
```

The 95% confidence interval of the t-test result is approximately equal to the lower and upper bound found by bootstrap. **Part e)**

```r
median <- median(Boston$medv)
median
```

```
## [1] 21.2
```

Part f)

```r
set.seed(1)
median.fn <- function(var, id) {
  return(median(var[id]))
}
boot.res <- boot(Boston$medv, median.fn, R=100)
boot.res
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Boston$medv, statistic = median.fn, R = 100)
##
##
## Bootstrap Statistics :
##     original  bias    std. error
## t1*     21.2  -0.029   0.3461316
```

Estimated standard error is 0.3461 with r = 100.

Part g)

```r
mu0.1 <- quantile(Boston$medv, 0.1)
mu0.1
```

```
##   10%
## 12.75
```

Part h)

```r
set.seed(1)
quantile10 <- function(var, id) {
  return(quantile(var[id], 0.1))
}
(boot.res <- boot(Boston$medv, quantile10, R=100))
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
```

```
## boot(data = Boston$medv, statistic = quantile10, R = 100)
##
##
## Bootstrap Statistics :
##     original  bias    std. error
## t1*    12.75   0.008   0.5370477
```

Estimated standard error is 0.537