ABE6933 SML HW2

Yu Chen

2022-09-23

Chap 3 exercise

4a.

```
set.seed(0)
X <- seq(0,10,1)
Y <- rnorm(11,0,1)+X
lm_linear <- lm(Y~X)
lm_cubic <- lm(Y~X+I(X^2)+I(X^3))
summary(lm_linear)$sigma
```

```
## [1] 1.212177
```

```
summary(lm_cubic)$sigma
```

```
## [1] 1.131151
```

When the true relationship between Y and X is linear. Then the training RSS of cubic model is lower than the other, because the cubic model has more parameters which can make model have more flexibility and can explain more variance.

4b.

```
test_x <- seq(11,20,1)
test_y <- rnorm(10,0,1)+test_x
df <- data.frame(Y=test_y,X=test_x)
df$ypred_linear <- predict(lm_linear, df["X"])
df$ypred_cubic <- predict(lm_cubic, df["X"])
myfunc <- function(x,x1,x2) {(x[x1]-x[x2])^2}
sum(apply(df,1,myfunc,x1="ypred_linear",x2="Y"))
```

```
## [1] 10.05541
```

```
sum(apply(df,1,myfunc,x1="ypred_cubic",x2="Y"))
```

```
## [1] 6119.337
```

```
test2_x <- seq(1,4,0.01)
test2_y <- rnorm(301,0,1)+test2_x
df2 <- data.frame(Y=test2_y,X=test2_x)
df2$ypred_linear <- predict(lm_linear, df2["X"])
df2$ypred_cubic <- predict(lm_cubic, df2["X"])
sum(apply(df2,1,myfunc,x1="ypred_linear",x2="Y"))
```

```
## [1] 323.8969
```

```
sum(apply(df2,1,myfunc,x1="ypred_cubic",x2="Y"))
```

```
## [1] 357.6127
```

4c.
For training data, even the true relationship is nonlinear, the cubic model can have a lower RSS because of its flexibility.

4d.
Since nonlinear relationship, if the data is more close to the linear model line, then the linear model RSS will lower, or otherwise.

10a.

```
library(ISLR)
lm <- lm(Sales~Price+Urban+US, data = Carseats)
```

10b.

```
summary(lm)
```

```
##
## Call:
## lm(formula = Sales ~ Price + Urban + US, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9206 -1.6220 -0.0564  1.5786  7.0581
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.043469   0.651012  20.036  < 2e-16 ***
## Price       -0.054459   0.005242 -10.389  < 2e-16 ***
## UrbanYes    -0.021916   0.271650  -0.081    0.936
## USYes        1.200573   0.259042   4.635 4.86e-06 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 2.472 on 396 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2335
## F-statistic: 41.52 on 3 and 396 DF,  p-value: < 2.2e-16
```

From summary table, we can get the coef of ‘Price’ is -0.0546 and it’s significant. That is when the urban and us parameter with the same data, one unit of price increase will lead the sales decrease 0.0545 unit. The coef of ‘Urban’ predictor is -0.0219 and is not significant. When price and US status keep the same, the store is in an urban will decrease 0.0219 unit in sales comparing to the store locates in rural. The coef of ‘US’ preoditor is 1.201 and is significant. When price and the location of store are the same, the store is in the US will increase 1.201 unit in sales, comparing to the store is not in the US.

10c.
$y_i = \beta_0 + \beta_1 * x_{i1} + \beta_2 * x_{i2} + \beta_3 * x_{i3}$

$$x_{i2} = \begin{cases} 1 & \text{if ith store is in an urban location} \\ 0 & \text{if ith store is in an rural location} \end{cases}$$

$$x_{i3} = \begin{cases} 1 & \text{if ith store is in the US} \\ 0 & \text{if ith store is not in the US} \end{cases}$$

10d.
The ‘price’ predictor and the ‘US’ predictor can reject the null hypothesis $\beta_j = 0$.

10e.

```
lm_small <- lm(Sales~Price+US, data = Carseats)
```

10f.

```
summary(lm)$adj.r.squared
```

```
## [1] 0.2335123
```

```
summary(lm_small)$adj.r.squared
```

```
## [1] 0.2354305
```

```
#anova(lm_small,lm)
```

Since they are mutiple models and nested, we cannot simply use R-square to judge the model. Although they are both not well, the adjusted r-square of small model (0.2354305) is a little better than the bigger one (0.2335123). ## I take the anova and can see that there is no significant difference between two model, so the smaller one is better.

10g.

```
confint(lm_small,level=0.95)
```

```
##                   2.5 %      97.5 %
## (Intercept) 11.79032020 14.27126531
## Price       -0.06475984 -0.04419543
## USYes        0.69151957  1.70776632
```

1.2

$$
\begin{aligned}
E(T_1) &= cE(T) \\
&= cE[\sum_{i=1}^{n}(Y_i - \bar{Y})^2] \\
&= cE[\sum_{i=1}^{n}(Y_i - \mu)^2 - 2\sum_{i=1}^{n}(Y_i - \mu)(\bar{Y} - \mu) + \sum_{i=1}^{n}(\bar{Y} - \mu)^2] \\
&= cE[\sum_{i=1}^{n}(Y_i - \mu)^2 - 2n(\bar{Y} - \mu)(\bar{Y} - \mu) + n(\bar{Y} - \mu)^2] \\
&= cE[\sum_{i=1}^{n}(Y_i - \mu)^2 - n(\bar{Y} - \mu)^2] \\
&= c(nvar(Y) - nvar(\bar{Y})) \\
&= c(nvar(Y) - var(Y)) \\
&= c(n-1)var(Y)
\end{aligned}
$$

Given $var(T_1)$ is unbiased for σ^2, so $c(n-1) = 1$. $c = \frac{1}{n-1}$

1.3 a)

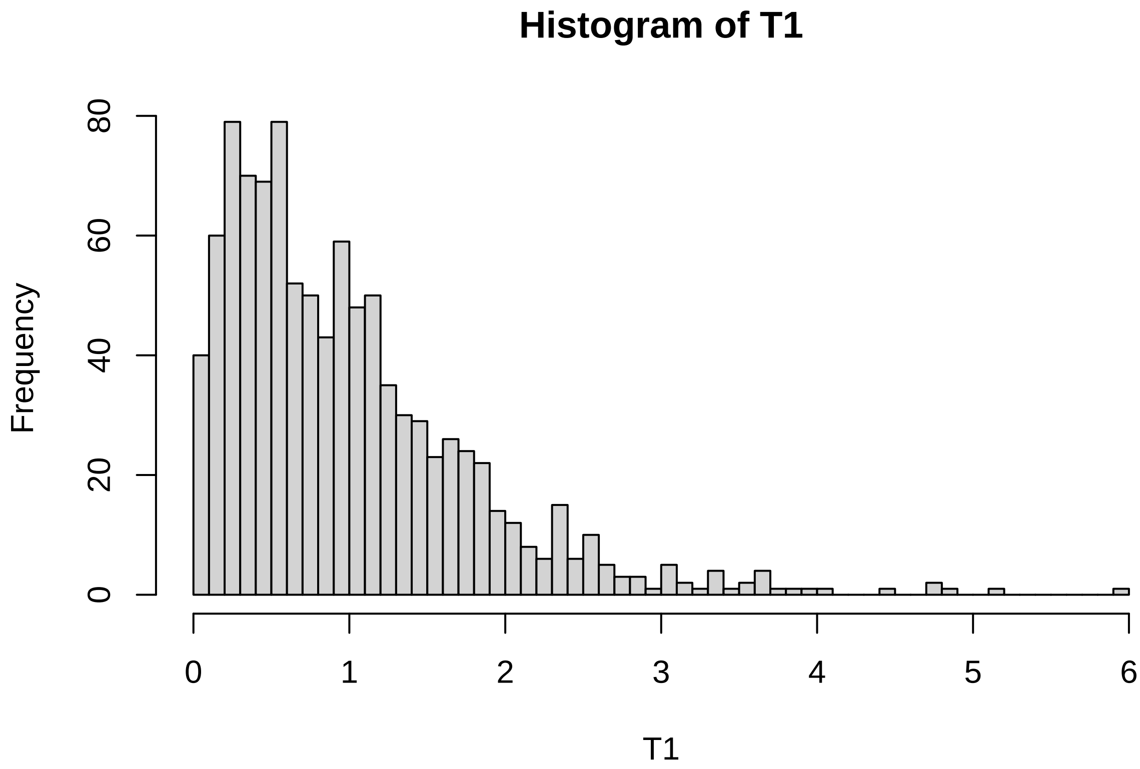
```
m <- 1000 #replications
n <- 4 #sample size
set.seed(0)
M = matrix(rnorm(m*n),nrow=m)
```

b)

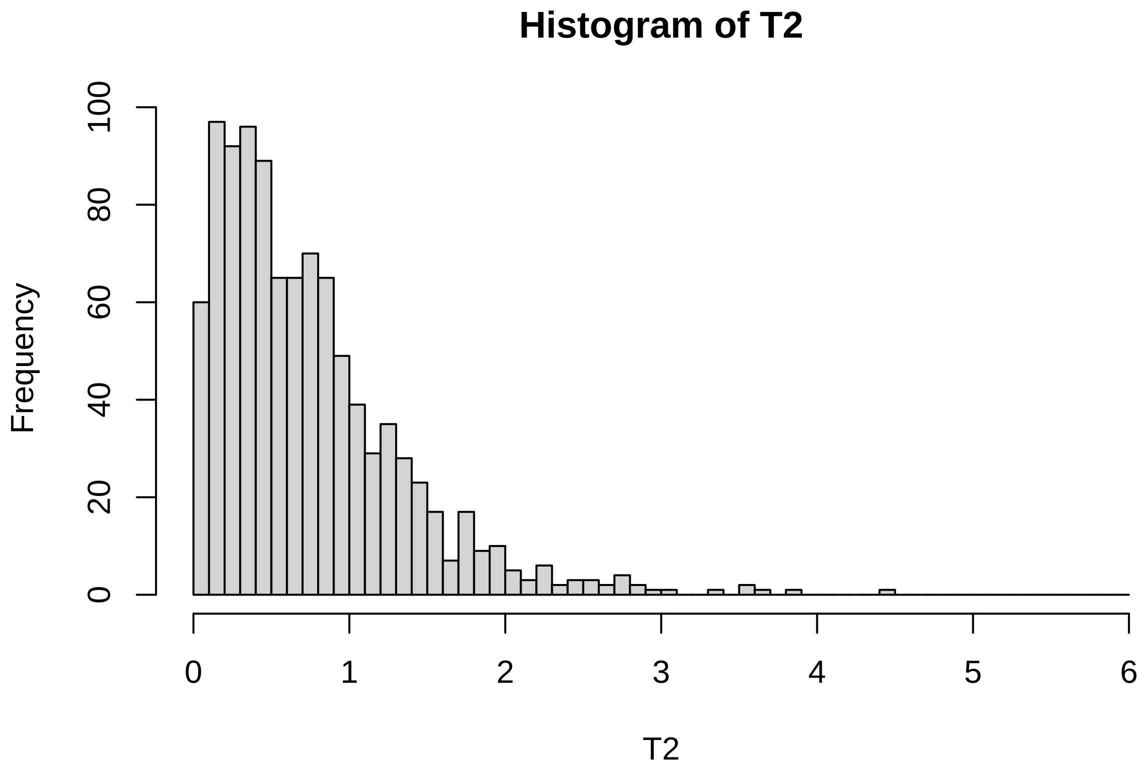
```
fun_T1 <- function(x) {(1/(length(x)-1))*sum((x-mean(x))^2)}
T1 <- apply(M, 1, fun_T1)
fun_T2 <- function(x) {(1/length(x))*sum((x-mean(x))^2)}
T2 <- apply(M, 1, fun_T2)
```

c)

```
breaks <- seq(0,6,0.1)
hist(T1, breaks = breaks)
```



```
hist(T2, breaks = breaks)
```



d)

```
mse <- function(x,true) {mean((x-true)^2)}

mean(T1)
```

```
## [1] 0.9893649
```

```
var(T1)
```

```
## [1] 0.6428409
```

```
T1_bias <- mean(T1)-1 ; T1_bias #bias of T1
```

```
## [1] -0.01063508
```

```
apply(data.frame(T1),2,mse,true=1) #MSE
```

```
##        T1
## 0.6423112
```

```r
mean(T2)
```

```
## [1] 0.7420237
```

```r
var(T2)
```

```
## [1] 0.361598
```

```r
T2_bias <- mean(T2)-(n-1)/n ; T2_bias #bias of T2
```

```
## [1] -0.007976313
```

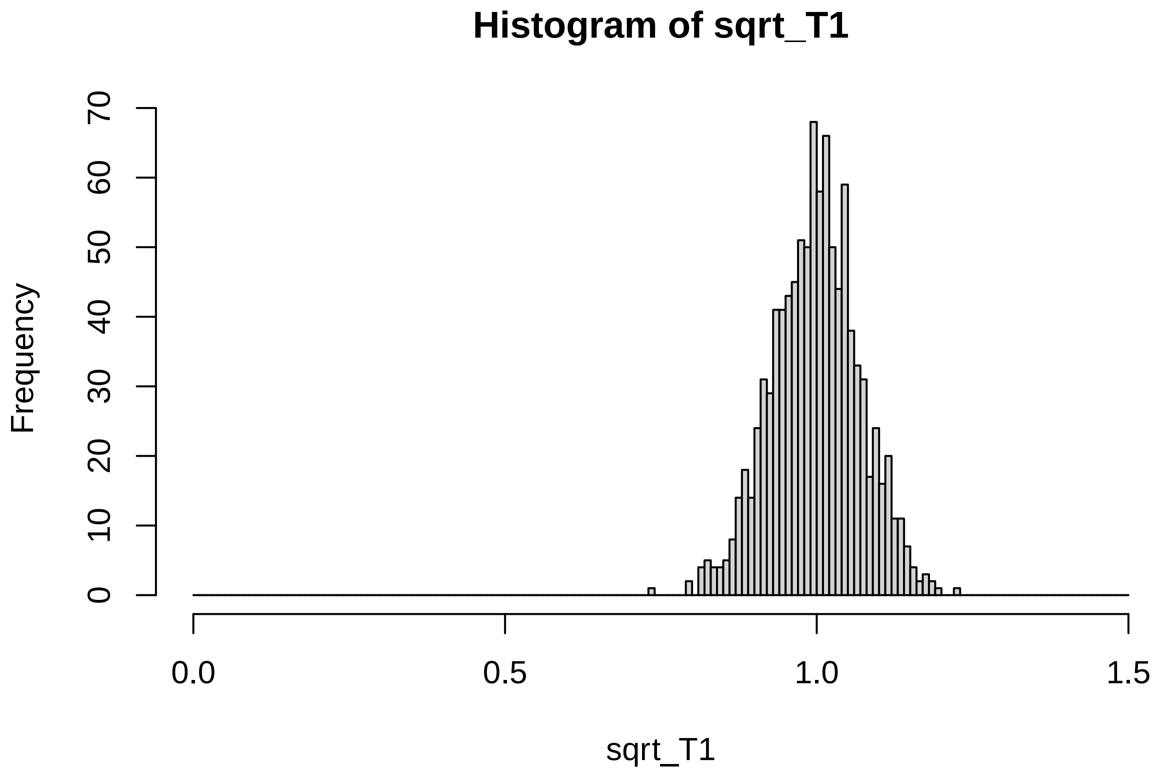
```r
apply(data.frame(T2),2,mse,true=(n-1)/n) #MSE
```

```
##     T2
## 0.3613
```

1.4
In 1.2 we show that T_1 is a unbiased estimator for σ^2. $\sigma = \sqrt{\sigma^2}$, so $\sqrt{T_1}$ is unbiased for estimation of σ.

```r
m_1 <- 1000 #replications
n_1 <- 100 #sample size
M_1 = matrix(rnorm(m_1*n_1),nrow=m_1)

sqrtfun <- function(x) {sqrt(var(x))}
sqrt_T1 <- apply(M_1, 1, sqrtfun)
breaks2 <- seq(0,1.5,0.01)
hist(sqrt_T1,breaks=breaks2)
```



```
mean(sqrt_T1)
```

```
## [1] 0.9985819
```

```
var(sqrt_T1)
```

```
## [1] 0.00494806
```

We get random data from true distribution N(0,1). Then we estimate the σ by using $\sqrt{T_1}$ and we can see that it is unbiased.