hw 8 solution

Statistical Computing, Jieun Shin

Autumn 2022

문제 1.

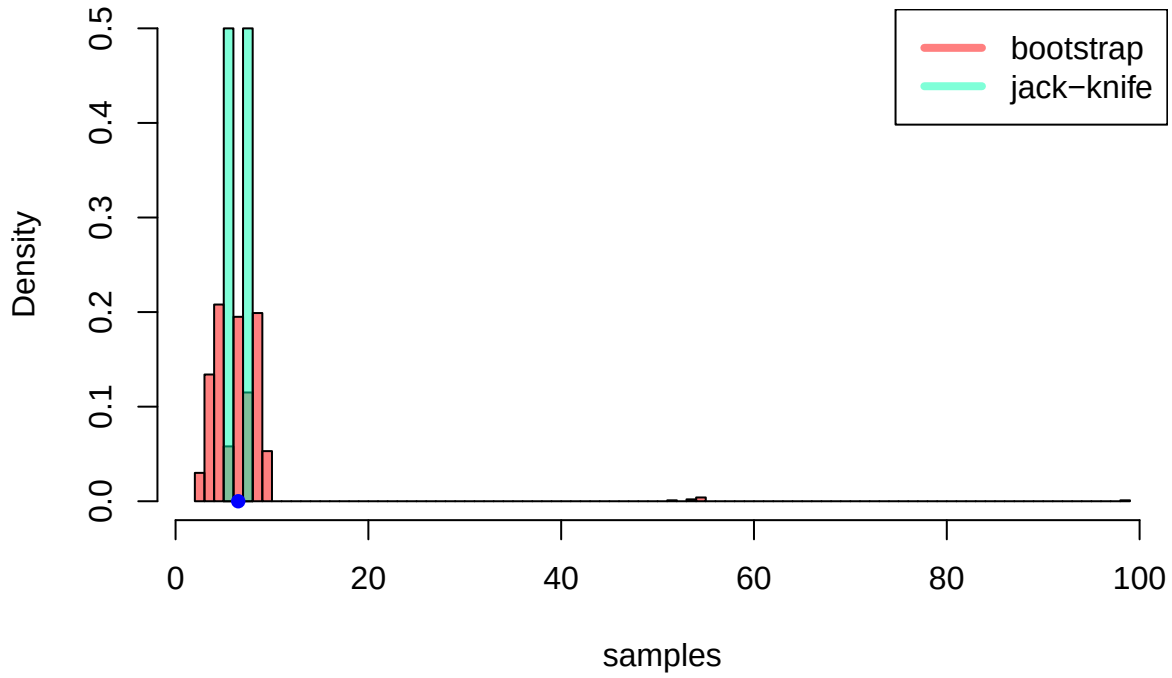
데이터에 99라는 매우 큰 값이 포함되어 있어 붓스트랩의 경우 표본중위수가 커지는 경우가 생기는데, 잭나이프의 경우 5와 8의 값만 나타난다.

R 코드

```
# 붓스트랩
boot_fn = function(){
  x = c(2, 5, 3, 8, 9, 4, 99, 10)
  n = length(x)
  y = sample(x, n, replace = TRUE)
  median(y)
}

B = 1000
b = replicate(B, boot_fn())

# 잭나이프
x = c(2, 5, 3, 8, 9, 4, 99, 10)
n = length(x)
y = J = c()
for(i in 1:n){
  y = x[-i]
  J[i] = median(y)
}

J
```

```
## [1] 8 8 8 5 5 8 5 5
```

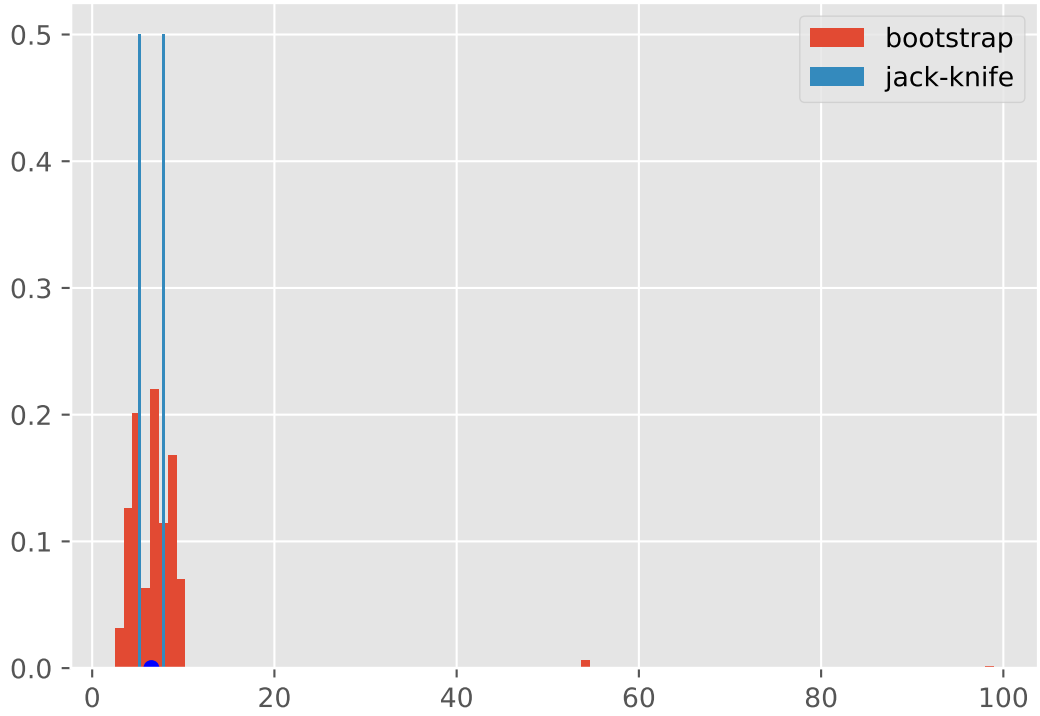
```
hist(b, main = "", breaks = 100, xlab = "samples", freq = F, col = rgb(1,0,0,0.5), ylim = c(0, 0.5))  #
hist(J, col = rgb(0,1,0.7,0.5), freq = F, add = T)  # 잭나이프
points(median(x), 0, col = "blue", pch = 16)  # 원데이터의 표본중위수
legend("topright", legend = c("bootstrap", "jack-knife"),
       col = c(rgb(1,0,0,0.5), rgb(0,1,0.7,0.5)), lwd = 4)
```



파이썬 코드

```
import numpy as np
from numpy import random
import matplotlib.pyplot as plt

N=1000
x = [2, 5, 3, 8, 9, 4, 99, 10]

#붓스트랩

def boot_fn(x):
    y = random.choice(x,len(x))
    return np.median(y)
b = np.array([boot_fn(x) for _ in range(N)])

#잭나이프
def jack_fn(x):
    j = np.empty(len(x))
    for i in range(len(x)):
        y = x[:i]+x[i+1:]
        j[i] = np.median(y)

    return j
j = jack_fn(x)
plt.style.use("ggplot")
c, e, p = plt.hist(b,bins=100,weights=np.zeros_like(b) + 1. / b.size,label="bootstrap")
```

```
c1, b1, p1, = plt.hist(j,weights=np.zeros_like(j) + 1. / j.size,label="jack-knife")
point = plt.plot(np.median(x),0,marker='o',markersize=5,c = 'b')
plt.legend(loc = "upper right")
plt.show()
```



문제 2.

각 셀을 11, 12, 21, 22의 이름으로 해당 셀들의 갯수만큼 생성한다. 붓스트랩 추정치의 분포는 평균적으로 약 0.5521을 중심으로 종 모양을 띠고 있어 주어진 데이터의 로그오즈비 $\log(1.74) = 0.5539$를 중심으로 하여 분포하고 있는 것을 확인할 수 있다.

R 코드

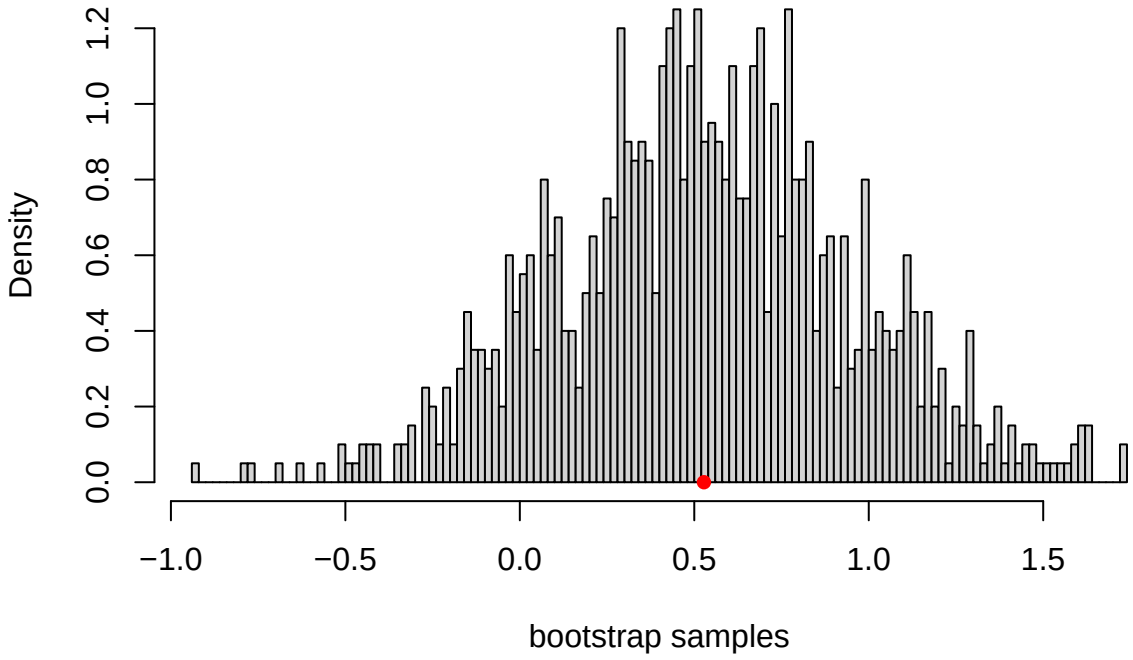
```
# 붓스트랩 함수
boot_fn = function(){
  x = rep(c(11, 12, 21, 22), c(83, 27, 23, 13)) # 각 셀의 갯수만큼 생성
  n = length(x)
  y = sample(x, n, replace = TRUE)
  tb = table(y) # 붓스트랩 표본의 각 셀의 갯수 세기
  log((tb[1]*tb[4])/(tb[2]*tb[3]))  # 로그 오즈비 출력
}

B = 1000
b = replicate(B, boot_fn())

mean(b)-0.5539                    # 편의
```

```
## [1] -0.02625232
```

```
sqrt( sum(b-mean(b))^2 / (B-1))  # 표준오차
```

```
## [1] 3.161331e-16
```

```
hist(b, main = "", breaks = 100, xlab = "bootstrap samples", freq = F)  # 붓스트랩
points(mean(b), 0, col = "red", pch = 16)  # 붓스트랩 샘플의 평균
```



파이썬 코드

```
B=1000
fij = ["11","12","21","22"]
Nij = [83,27,23,13]

def boot_fn2(fij,Nij):
  x = np.repeat(fij,Nij)
  y = random.choice(x,len(x))
  _,count = np.unique(y, return_counts=True)
  return np.log((count[0]*count[3])/(count[1]*count[2]))

b = np.array([boot_fn2(fij,Nij) for _ in range(B)])
np.mean(b)-0.5539                                    # 편의
```

```
## 0.023656037170109778
```

```
np.sqrt( np.power(np.sum(b-np.mean(b)), 2) / (B-1))  # 표준오차
```

```
## 3.3158851516052247e-15
```

```
plt.style.use("ggplot")
c, e, p = plt.hist(b,bins=100,weights=np.zeros_like(b) + 1. / b.size,label="bootstrap")
point = plt.plot(np.mean(b),0,marker='o',markersize=5,c = 'b')
plt.show()
```