统计计算与软件包第四次作业

姓名：王韦力　　学号：2019302010009

2022 年 5 月 8 日

7.2

分析过程 仿照例 7.9，把数据集改为 `law`，$\hat{\theta}^{(b)}$ 改为 `LSAT` 与 `GPA` 之间的协方差 R 即可.

Rcode

```
library(bootstrap)

# initialize
data(law,package = "bootstrap")
n <- nrow(law)
LSAT <- law$LSAT
GPA <- law$GPA
B <- 2000
theta.b <- numeric(B)

# set up storage for the sampled indices
indices <- matrix(0, nrow = B, ncol = n)

# run the bootstrap
for (b in 1:B) {
i <- sample(1:n, size = n, replace = TRUE)
LSAT <- law$LSAT[i]
GPA <- law$GPA[i]
theta.b[b] <- cor(LSAT, GPA)
# save the indices for the jackknife
indices[b, ] <- i
}

# jackknife-after-bootstrap to est. se(se(R))
se.jack <- numeric(n)
for (i in 1:n) {
# in i-th replicate omit all samples with x[i]
keep <- (1:B)[apply(indices, MARGIN = 1, FUN = function(k) {!any(k == i)})]
se.jack[i] <- sd(theta.b[keep])
}
```

```
# results
print(sd(theta.b))
print(sqrt((n-1) * mean((se.jack - mean(se.jack))^2)))
```

运行结果 R 输出的结果如下:

```
> print(sd(theta.b))
[1] 0.1336759
> print(sqrt((n-1) * mean((se.jack - mean(se.jack))^2)))
[1] 0.08495476
```

结论 由运行结果,$se(R)$ 的自助法估计为 0.1336759,此估计的标准差的 Jackknife 估计为 0.08495476.

7.3

分析过程 本题只要求给出协方差 R 的一种基于自助法的 t 区间估计，所以我们可以用多种方法解题，如利用 `boot` 包中的 `boot.ci` 函数，或利用例 7.12 中给出的 `boot.t.ci` 函数. 这里我们选择使用百分位数置信区间，借助 `boot` 函数，用百分位数置信区间的定义求解.

Rcode

```
library(bootstrap)
library(boot)

data(law,package = "bootstrap")
theta.boot <- function(dat,ind) { # function to compute statistic
LSAT <- dat[ind,1]
GPA <- dat[ind,2]
cor(LSAT,GPA)
}

LSAT <- law$LSAT
GPA <- law$GPA
dat <- cbind(LSAT,GPA)
boot.obj <- boot(dat, statistic = theta.boot, R = 2000) definitions)
alpha <- c(.025, .975)
print(quantile(boot.obj$t, alpha, type=6))
```

运行结果 Rcode 最后一行的运行结果为

```
> print(quantile(boot.obj$t, alpha, type=6))
2.5%     97.5%
0.4469920 0.9650218
```

结论 由运行结果可知,例 7.2 中 R 的 95% 置信 t 区间的估计(用百分位数置信区间)为 $(0.4469920, 0.9650218)$.

7.4

分析过程 指数分布的概率密度函数为

$$f(x)=\begin{cases}\lambda e^{-\lambda x}, & x>0\\ 0, & x\le 0\end{cases} \tag{1}$$

设有样本 $X_1,X_2,\dots,X_n$，对应的观测值为 $x_1,x_2,\dots,x_n$，则似然函数为 $L(\lambda)=\prod_{i=1}^n \lambda e^{-\lambda x_i}=\lambda^n\exp\{-\lambda\sum_{i=1}^n x_i\}$，对数似然函数为 $l(\lambda)=\ln L(\lambda)=n\ln\lambda-\lambda\sum_{i=1}^n x_i$，求导得 $l'(\lambda)=\dfrac{n}{\lambda}-\sum_{i=1}^n x_i$. 令 $l'(\lambda)=0$，得 $\hat\lambda=\dfrac{n}{\sum_{i=1}^n x_i}$. 此即 λ 的极大似然估计. 在本题中，假设这 12 个观测值是相互独立的，则由以上讨论易得 $\hat\lambda\approx 0.00925$. 用自助法估计该估计的偏差和标准差的方法可参考例 7.2，例 7.4 和例 7.5.

Rcode

```
data("aircondit",package = "boot")
n <- nrow(aircondit) # sample size
B <- 2000 # number of replicates
theta.b <- numeric(B)
theta.hat <- 1/mean(aircondit$hours)

#bootstrap for bias and se
for(b in 1:B) {
i <- sample(1:n,size=n,replace = TRUE)
hours <- aircondit$hours[i]
theta.b[b] <- 1/mean(hours)
}
bias <- mean(theta.b)-theta.hat
print(bias)
print(se.A <- sd(theta.b))
```

运行结果 Rcode 最后两行的运行结果为

```
> print(bias)
[1] 0.001272751
> print(se.A <- sd(theta.b))
[1] 0.004172741
```

结论 由分析过程的讨论和运行结果可知，λ 的极大似然估计约为 0.00925，用自助法估计该估计的偏差和标准差分别为 0.001272751 和 0.004172741.

7.5

分析过程 用程序包 `boot` 中的 `boot` 和 `boot.ci` 函数来计算置信区间.

Rcode

```
data("aircondit",package = "boot")
theta.boot <- function(dat,ind) {
mean(dat[ind])
}

dat <- aircondit$hours
boot.obj <- boot(dat, statistic = theta.boot, R = 2000)
print(boot.obj)
print(boot.ci(boot.obj, type = c( "norm","basic","perc","bca")))
```

运行结果 Rcode 的输出结果如下:

```
> print(boot.obj)

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = dat, statistic = theta.boot, R = 2000)


Bootstrap Statistics :
original   bias    std. error
t1* 108.0833 0.430625    37.86825
> print(boot.ci(boot.obj, type = c( "norm","basic","perc","bca")))
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 2000 bootstrap replicates

CALL :
boot.ci(boot.out = boot.obj, type = c("norm", "basic", "perc",
"bca"))

Intervals :
Level      Normal              Basic
95%   ( 33.4, 181.9 )   ( 19.6, 170.2 )

Level     Percentile            BCa
95%   ( 46.0, 196.6 )   ( 54.6, 227.8 )
Calculations and Intervals on Original Scale
Some BCa intervals may be unstable
```

结论 由运行结果可知，使用标准正态、基本、百分位数和 BCa 方法计算的故障间隔平均时间 $1/\lambda$ 的 95% 置信区间分别为 $(33.4, 181.9)$、$(19.6, 170.2)$、$(46.0, 196.6)$ 和 $(54.6, 227.8)$.

7.7

分析过程 五门考试的学生的成绩数据大致可以看作服从 5 维正态分布 $N_5(\mu,\Sigma)$. 由多元统计知识，Σ 的极大似然估计为 $\hat{\Sigma}=\dfrac{1}{n}A$, 其中 A 为样本离差阵. 由此可以计算出 $\hat{\Sigma}$ 的特征值 $\lambda_i(i=1,\dots,5)$, 进而得出 $\hat{\theta}$. 参照例 7.4 和例 7.2 的方法估计 $\hat{\theta}$ 的偏差和标准误差.

Rcode

```
data(scor, package = "bootstrap")
n <- nrow(scor)
A <- (n-1)*as.matrix(cov(scor))
Sigma.hat <- A/n
lambda <- eigen(Sigma.hat)$values
theta.hat <- lambda[1]/sum(lambda)

# bootstrap for bias and se
B <- 2000 # number of replicates
theta.b <- numeric(B)

for(b in 1:B) {
i <- sample(1:n,size=n,replace = TRUE)
scores <- scor[i,]
A. <- (n-1)*as.matrix(cov(scores))
Sigma.hat. <- A./n
lambda. <- eigen(Sigma.hat.)$values
theta.b[b] <- lambda.[1]/sum(lambda.)
}
bias <- mean(theta.b)-theta.hat
print(theta.hat)
print(bias)
print(se.A <- sd(theta.b))
```

运行结果 R 输出的结果如下：

```
> print(theta.hat)
[1] 0.619115
> print(bias)
[1] 0.00281215
> print(se.A <- sd(theta.b))
[1] 0.04729762
```

结论 由运行结果可知，$\theta=0.619115$，用自助法估计的 $\hat{\theta}$ 的偏差和标准误差分别为 0.00281215 和 0.04729762.

7.8

分析过程 参考例 7.6 和例 7.7.

Rcode 我们直接在 7.7 中的 Rcode 的基础上，给出用 Jackknife 法估计 $\hat{\theta}$ 的偏差和标准误差的 Rcode.

```
theta.jack <- numeric(n)
for (i in 1:n) {
scores <- scor[-i,]
A_ <- (n-2)*as.matrix(cov(scores))
Sigma.hat_ <- A_/(n-1)
lambda_ <- eigen(Sigma.hat_)$values
theta.jack[i] <- lambda_[1]/sum(lambda_)
}
bias.jack <- (n-1)*(mean(theta.jack)-theta.hat)
se.jack <- sqrt((n-1)*mean((theta.jack-mean(theta.jack))^2))
print(bias.jack)
print(se.jack)
```

运行结果 R 输出的结果如下:

```
> print(bias.jack)
[1] 0.001069139
> print(se.jack)
[1] 0.04955231
```

结论 由运行结果可知,用 Jackknife 法估计的 $\hat{\theta}$ 的偏差和标准误差分别为 0.001069139 和 0.04955231.

7.10

分析过程 在例 7.18 中，把第二个模型改为三次多项式模型，即添加三次项及其系数，然后用交叉验证法选出模型. 为了选出 adjusted R^2 最大的模型，可以参考例 7.17 的代码，分别对 `L1`，`L2`，`L3` 和 `L4` 取 `summary`，然后比较它们的 `Adjusted R-squared` 的大小即可.

Rcode

```
library(DAAG); attach(ironslag)
n <- length(magnetic)
e1 <- e2 <- e3 <- e4 <- numeric(n)
# for n-fold cross validation
# fit models on leave-one-out samples
for (k in 1:n) {
y <- magnetic[-k]
x <- chemical[-k]

J1 <- lm(y ~ x)
yhat1 <- J1$coef[1] + J1$coef[2] * chemical[k]
e1[k] <- magnetic[k] - yhat1

J2 <- lm(y ~ x + I(x^2) + I(x^3))
```

```
yhat2 <- J2$coef[1] + J2$coef[2] * chemical[k] + J2$coef[3] * chemical[k]^2 +
J2$coef[4] * chemical[k]^3
e2[k] <- magnetic[k] - yhat2

J3 <- lm(log(y) ~ x)
logyhat3 <- J3$coef[1] + J3$coef[2] * chemical[k]
yhat3 <- exp(logyhat3)
e3[k] <- magnetic[k] - yhat3

J4 <- lm(log(y) ~ log(x))
logyhat4 <- J4$coef[1] + J4$coef[2] * log(chemical[k])
yhat4 <- exp(logyhat4)
e4[k] <- magnetic[k] - yhat4
}

L1 <- lm(magnetic ~ chemical)
L2 <- lm(magnetic ~ chemical + I(chemical^2) + I(chemical^3))
L3 <- lm(log(magnetic) ~ chemical)
L4 <- lm(log(magnetic) ~ log(chemical))
S1 <- summary(L1)
S2 <- summary(L2)
S3 <- summary(L3)
S4 <- summary(L4)

c(mean(e1^2), mean(e2^2), mean(e3^2), mean(e4^2))
c(S1$adj.r.squared,S2$adj.r.squared,S3$adj.r.squared,S4$adj.r.squared)
```

运行结果 Rcode 最后两行的运行结果为

```
> c(mean(e1^2), mean(e2^2), mean(e3^2), mean(e4^2))
[1] 19.55644 18.17756 18.44188 20.45424
> c(S1$adj.r.squared,S2$adj.r.squared,S3$adj.r.squared,S4$adj.r.squared)
[1] 0.5281545 0.5740396 0.5280556 0.4870624
```

结论 由运行结果可知，无论是使用交叉验证法还是根据最大 adjusted R^2，均应选择三次多项式模型.