统计计算作业 4

7.2 参考“law”数据（“bootstrap”）。使用基于自助法的水手刀法 se（R）的自助法估计的标准误差。

获取 law 数据，使用 Jackknife-after-Bootstrap 方法估计其标准差，程序运行如下：

```
data(law, package = "bootstrap")
n <- nrow(law)
B <- 1000
COR <- numeric(B)
indices <- matrix(0, nrow = B, ncol = n)
for (b in 1:B) {
  i <- sample(1:n, size = n, replace = TRUE)
  LSAT <- law$LSAT[i]
  GPA <- law$GPA[i]
  COR[b] <- cor(LSAT,GPA)
  indices[b, ] <- i
}
```

```
se.jack <- numeric(n)
for (i in 1:n) {
  keep <- (1:B)[apply(indices, MARGIN = 1,
                      FUN = function(k) {!any(k == i)})]
  se.jack[i] <- sd(COR[keep])
}
print(sd(COR))
print(sqrt((n-1) * mean((se.jack - mean(se.jack))^2)))
```

得到结果：

```
> print(sd(COR))
[1] 0.135925
> print(sqrt((n-1) * mean((se.jack - mean(se.jack))^2)))
[1] 0.0789038
```

7.3 对例 7.2 中的相关性统计量（“bootstrap”中的“law”数据）给出一个自助法 t 置信区间估计。

写一个函数来计算一元或者多元样本下 Bootstrap t 置信区间，默认的置信水平为 95%，Bootstrap 重复数为 500，估计标准差的重复次数默认为 100，程序运行如下：

```r
boot.t.ci <-
  function(x, B = 500, R = 100, level = .95, statistic){
    x <- as.matrix(x)
    n <- nrow(x)
    stat <- numeric(B)
    se <- numeric(B)
    boot.se <- function(x, R, f) {
      x <- as.matrix(x)
      m <- nrow(x)
      th <- replicate(R, expr = {
        i <- sample(1:m, size = m, replace = TRUE)
        f(x[i, ])
      })
      return(sd(th))
    }

    for (b in 1:B) {
      j <- sample(1:n, size = n, replace = TRUE)
      y <- x[j, ]
      stat[b] <- statistic(y)
      se[b] <- boot.se(y, R = R, f = statistic)
    }

    stat0 <- statistic(x)
    t.stats <- (stat - stat0) / se
    se0 <- sd(stat)
    alpha <- 1 - level
    Qt <- quantile(t.stats, c(alpha/2, 1-alpha/2), type = 1)
    names(Qt) <- rev(names(Qt))
    CI <- rev(stat0 - Qt * se0)
    return(CI)
  }

data(law, package = "bootstrap")
stat <- function(x) { cor(x[,1], x[,2]) }
ci <- boot.t.ci(law, statistic = stat, B=1000, R=200)
print(ci)
```

可得到结果：

```
> print(ci)
      2.5%      97.5%
-0.2074181  0.9926069
```

故自助法 t 置信区间估计为[-0.207,0.993]。

7.4 参考“boot”程序包中提供的空调数据集“aircondit”，观测值是空调设备两次故障之间的小时数。假设故障之间的时间服从指数模型 Exp(λ),给出故障率λ的

极大似然估计，并使用自助法对该估计的偏差和标准误差进行估计。
程序运行如下：

```
library(boot)
mle <- function(x,i){ 1/mean(x[i]) }
aircondit.boot <- boot(data = as.matrix(aircondit), statistic = mle, R = 2000)
print(aircondit.boot)
```

可得到结果如下：

```
Bootstrap Statistics :
      original       bias    std. error
t1* 0.00925212 0.001587651   0.0045378
```

偏差的估计为 0.0016，标准差的估计为 0.0045.

7.5 分别使用标准正态、基本、百分位数和 BCa 方法计算故障间隔平均时间 1/λ 的 95%自助法置信区间。
程序运行如下：

```
library(boot)
aircondit1 <- as.matrix(aircondit)
aircondit.b <- boot(data = aircondit1, statistic <- function(x,i) {mean(x[i])}, R = 2000)
boot.ci(aircondit.b, type = c("basic","norm",  "perc", "bca"))
```

得到结果如下：

```
Intervals :
Level      Normal              Basic
95%   ( 33.9, 184.0 )   ( 27.6, 171.9 )

Level     Percentile            BCa
95%   ( 44.3, 188.6 )   ( 54.9, 230.8 )
Calculations and Intervals on Original Scale
Some BCa intervals may be unstable
```

7.7 五维成绩数据的 5× 5 协方差矩阵Σ具有正的特征值λ1 > ..... > λ5。在主成分分析中$\theta = \frac{\lambda 1}{\sum_{j=1}^{5} \lambda j}$，给出了第一主成分解释的方差所占比例。
首先进行主成分分析：

```
library(bootstrap)
theta <- function(data, center = TRUE, scale = TRUE) {
  Sigma.hat <-
    cov(scale(as.matrix(data),center = center, scale = scale))
  e.hat <- eigen(Sigma.hat)
  theta.hat <- e.hat$values[1]/sum(e.hat$values)
  return(theta.hat)
}
theta.all <- theta(scor)
```

使用自助法估计偏差和标准误差：

```
B <- 2000
n <- nrow(scor)
theta.B <- numeric(B)
theta.B <- replicate(B, expr = {
  i <- sample(1:n, size = n, replace = TRUE)
  scor.B <- as.matrix(scor)[i,]
  theta.B <- theta(scor.B)
})
bias <- mean(theta.B-theta.all)
std <- sd(theta.B)
result <- c(theta.all,bias,std)
names(result) <- c("original","bias.B", "std.B")
print(result)
```

结果如下：

```
> print(result)
      original         bias.B          std.B
 0.6361960298 -0.0009985678  0.0434058298
```

偏差为-0.000999，标准差为 0.0434。

7.8 参考上题，利用水手刀法估计偏差和标准误差。

```
n <- nrow(scor)
theta.jack <- numeric(n)
for (i in 1:n)
  theta.jack[i] <- theta(as.matrix(scor)[-i,])

bias.jack <- (n-1)*mean(theta.jack-theta.all)
std.jack <- sqrt((n-1)*mean((theta.jack-mean(theta.jack))^2))

result.jack <- c(theta.all,bias.jack,std.jack)
names(result.jack) <- c("original","bias.jack", "std.jack")
print(result.jack)
```

得到结果如下：

```
> print(result.jack)
      original     bias.jack      std.jack
0.6361960298 -0.0003778535  0.0446668393
```

7.10 将双对数模型转换为三次多项式模型并重复分析过程，使用交叉验证法。

程序运行如下：

```r
install.packages("DAAG")
par(ask=TRUE)
library(DAAG)
attach(ironslag)

n <- length(magnetic)
e1 <- e2 <- e3 <- e4 <- numeric(n)
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))
  yhat2 <- J2$coef[1] + J2$coef[2] * chemical[k] +
    J2$coef[3] * chemical[k]^2
  e2[k] <- magnetic[k] - yhat2

  J3 <- lm(y ~ x + I(x^2) + I(x^3))
  yhat3 <- J3$coef[1] + J3$coef[2] * chemical[k] +
    J3$coef[3] * chemical[k]^2 + J3$coef[4] * chemical[k]^3
  e3[k] <- magnetic[k] - yhat3

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

crossv <- c(mean(e1^2), mean(e2^2), mean(e3^2), mean(e4^2))
best.cross.ind <- which(crossv == min(crossv), arr.ind = T)

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

adj.r <-
  c(L1$adj.r.squared,L2$adj.r.squared,L3$adj.r.squared,L4$adj.r.squared)
best.adj.ind <- which(adj.r == max(adj.r),arr.ind = T)

best.ind <- c(best.cross.ind, best.adj.ind)
names(best.ind) <- c('best.cross.ind', 'best.adj.ind')
print(best.ind)
```

得到结果如下：

```
best.cross.ind    best.adj.ind
             2               2
```

故交叉验证法和根据 R^2 的最大调整都会选择第二个模型。