hw06

陆靖磊 22300680221

2024-11-21

8.1

```r
library(bootstrap)

compute_correlation_stats <- function(data, x_col, y_col) {
  n <- nrow(data)
  theta.hat <- cor(data[[x_col]], data[[y_col]])

  theta.jack <- sapply(1:n, function(i) cor(data[-i, x_col], data[-i, y_col]))

  bias <- (n - 1) * (mean(theta.jack) - theta.hat)
  se <- sqrt((n - 1) * mean((theta.jack - mean(theta.jack))^2))

  list(est = theta.hat, bias = bias, se = se)
}

data("law")
results <- compute_correlation_stats(law, "LSAT", "GPA")
print(results)
```

```
## $est
## [1] 0.7763745
##
## $bias
## [1] -0.006473623
##
## $se
## [1] 0.1425186
```

```r
rm(law)
detach(package:bootstrap)
```

8.3

```r
library(boot)
library(bootstrap)

data("law")
attach(law)
```

```
cor.stat <- function(data, indices = 1:nrow(data)) {
  cor(data[indices, 1], data[indices, 2])
}

# 定义 Bootstrap t 区间计算函数
boot.t.ci <- function(data, B = 500, R = 100, level = 0.95, statistic) {
  # 将数据转换为矩阵
  data_matrix <- as.matrix(data)
  n <- nrow(data_matrix)

  # 初始化统计量和标准误向量
  stats <- numeric(B)
  ses <- numeric(B)

  # 定义 Bootstrap 标准误计算函数
  boot.se <- function(data, R, statistic) {
    m <- nrow(data)
    bootstrap_samples <- replicate(R, {
      indices <- sample(1:m, size = m, replace = TRUE)
      statistic(data[indices, ])
    })
    sd(bootstrap_samples)
  }

  # 进行 Bootstrap 抽样
  for (b in 1:B) {
    indices <- sample(1:n, size = n, replace = TRUE)
    bootstrap_sample <- data_matrix[indices, ]
    stats[b] <- statistic(bootstrap_sample)
    ses[b] <- boot.se(bootstrap_sample, R = R, statistic)
  }

  # 计算原始统计量
  original_stat <- statistic(data_matrix)

  # 计算 t 统计量
  t_stats <- (stats - original_stat) / ses

  # 计算标准误
  se0 <- sd(stats)

  # 计算置信水平
  alpha <- 1 - level

  # 计算 t 分布的分位数
  q_t <- quantile(t_stats, c(alpha / 2, 1 - alpha / 2), type = 1)
  names(q_t) <- rev(names(q_t))

  # 计算置信区间
  ci <- rev(original_stat - q_t * se0)

  return(ci)
}
```

```r
result <- boot.t.ci(law, B = 1000, R = 25, statistic = cor.stat)
print(result)
```

```
##       2.5%      97.5%
## -0.2594146  0.9782805
```

```r
detach(law)
detach(package:bootstrap)
detach(package:boot)
```

8.5

```r
library(boot)
x <- aircondit[1]
meant <- function(x, i) return(mean(as.matrix(x[i, ])))
b <- boot(x, statistic = meant, R = 1000)
b
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = x, statistic = meant, R = 1000)
##
##
## Bootstrap Statistics :
##     original  bias    std. error
## t1* 108.0833 0.17775    37.49542
```

```r
boot.ci(b, type = c("norm", "perc", "basic"))
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = b, type = c("norm", "perc", "basic"))
##
## Intervals :
## Level      Normal              Basic              Percentile
## 95%   ( 34.4, 181.4 )   ( 27.1, 168.9 )   ( 47.2, 189.1 )
## Calculations and Intervals on Original Scale
```

```r
detach(package:boot)
```

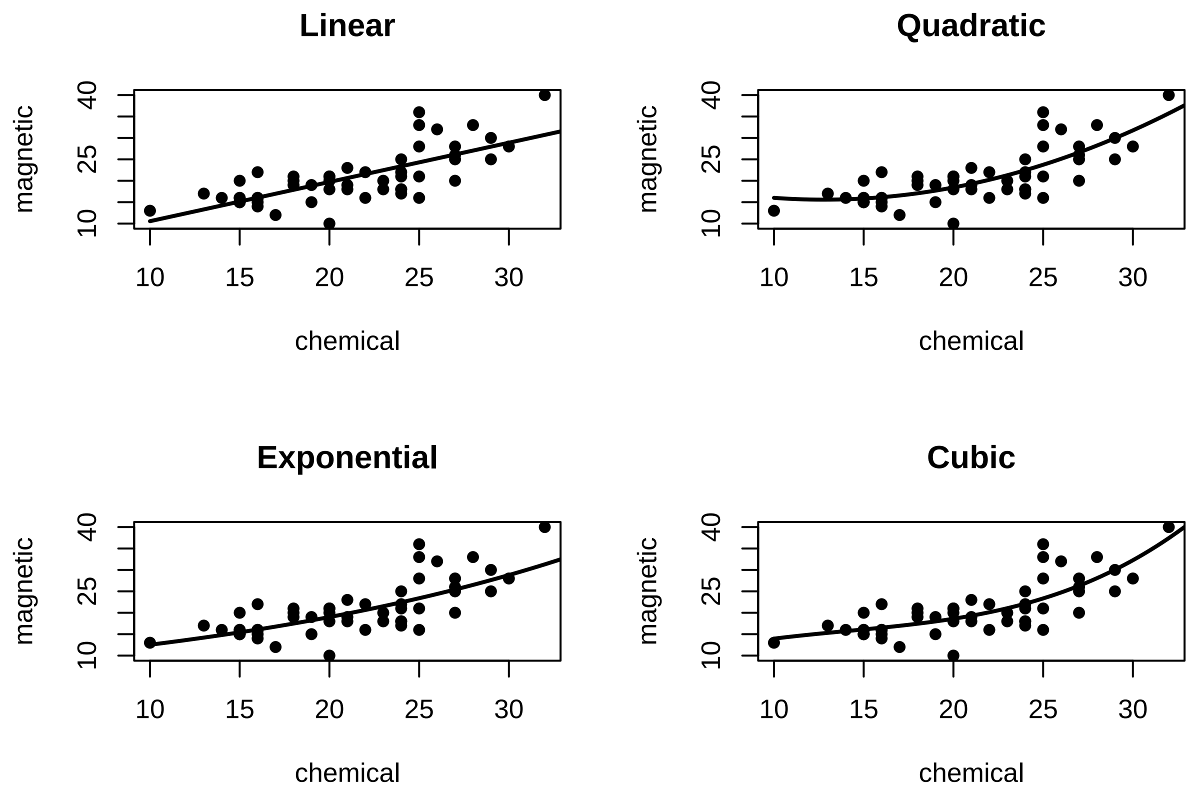
8.10

```r
library(DAAG, warn.conflict = FALSE)
```

```
## Warning: 程序包'DAAG'是用R版本4.4.2 来建造的
```

```r
attach(ironslag)
a <- seq(10, 40, 0.1)

par(mfrow = c(2, 2))
L1 <- lm(magnetic ~ chemical)
plot(chemical, magnetic, main = "Linear", pch = 16)
yhat1<-L1$coef[1]+L1$coef[2]*a
lines(a,yhat1,lwd=2)
L2<-lm(magnetic~chemical+I(chemical^2))
plot(chemical,magnetic,main="Quadratic",pch=16)
yhat2<-L2$coef[1]+L2$coef[2]*a+L2$coef[3]*a^2
lines(a,yhat2,lwd=2)
L3<-lm(log(magnetic)~chemical)
plot(chemical,magnetic,main="Exponential",pch=16)
logyhat3<-L3$coef[1]+L3$coef[2]*a
yhat3<-exp(logyhat3)
lines(a,yhat3,lwd=2)
c2<-chemical^2
c3<-chemical^3
L4<-lm(magnetic~chemical+c2+c3)
plot(chemical,magnetic,main="Cubic",pch=16)
yhat4<-L4$coef[1]+L4$coef[2]*a+L4$coef[3]*a^2+L4$coef[4]*a^3
lines(a,yhat4,lwd=2)
```



Linear

Quadratic

Exponential

Cubic

```
par(mfrow=c(1,1))
Rsq<-numeric(4)
Rsq[1]<-summary(L1)$adj.r.squared
Rsq[2]<-summary(L2)$adj.r.squared
Rsq[3]<-summary(L3)$adj.r.squared
Rsq[4]<-summary(L4)$adj.r.squared
Rsq
```

```
## [1] 0.5281545 0.5768151 0.5280556 0.5740396
```

8.11

```
library(DAAG, warn.conflict = FALSE)
attach(ironslag)
```

```
## The following objects are masked from ironslag (pos = 3):
##
##     chemical, magnetic
```

```
n <- length(magnetic)
N <- choose(n, 2)
e1 <- e2 <- e3 <- e4 <- e5 <- numeric(N)
ij<-1
for (i in 1:(n- 1)) for (j in (i + 1):n) {
 k <- c(i, j)
 y <- magnetic[-k]
 x <- chemical[-k]
 J1 <- lm(y ~ x)
 yhat1 <- J1$coef[1] + J1$coef[2] * chemical[k]
 e1[ij] <- sum((magnetic[k]- yhat1)^2)
 J2 <- lm(y ~ x + I(x^2))
 yhat2 <- J2$coef[1] + J2$coef[2] * chemical[k] +
 J2$coef[3] * chemical[k]^2
 e2[ij] <- sum((magnetic[k]- yhat2)^2)
 J3 <- lm(log(y) ~ x)
 logyhat3 <- J3$coef[1] + J3$coef[2] * chemical[k]
 yhat3 <- exp(logyhat3)
 e3[ij] <- sum((magnetic[k]- yhat3)^2)
 J4 <- lm(log(y) ~ log(x))
 logyhat4 <- J4$coef[1] + J4$coef[2] * log(chemical[k])
 yhat4 <- exp(logyhat4)
 e4[ij] <- sum((magnetic[k]- yhat4)^2)
 c2 <- x^2
 c3 <- x^3
 J5 <- lm(y ~x+c2+c3)
 yhat5 <- J5$coef[1] + J5$coef[2] * chemical[k] +
 J5$coef[3] * chemical[k]^2 + J5$coef[4] * chemical[k]^3
 e5[ij] <- sum((magnetic[k]- yhat5)^2)
 ij<-ij+1
 }
c(sum(e1), sum(e2), sum(e3), sum(e4), sum(e5))/N
```

```
## [1] 39.14455 35.74037 36.90983 40.93436 36.46996
```

```r
detach(ironslag)
detach(package:DAAG)
```