Statistical Computing HW3

Wei-Chen Chang r12227118

Due: 2024-05-01

Consider the `scor (bootstrap)` test score data on 88 students who took examinations in five subjects. The five-dimensional scores data have a 5 × 5 covariance matrix Σ, with positive eigenvalues $\lambda_1 > ..., > \lambda_5$. In principal components analysis,

$$\theta = \frac{\lambda_1}{\Sigma_{i=1}^5 \lambda_i}$$

measures the proportion of variance explained by the first principal component. Let $\hat{\lambda}_1 > ..., > \hat{\lambda}_5$ be the eigenvalues of Σ, where $\hat{\Sigma}$ is the MLE of Σ.

Compute the sample estimate

$$\hat{\theta} = \frac{\hat{\lambda}_1}{\Sigma_{i=1}^5 \hat{\lambda}_i}$$

of θ.

1. Use bootstrap to estimate the bias and standard error of $\hat{\theta}$.
2. Obtain the jackknife estimates of bias and standard error of $\hat{\theta}$.
3. Compute 95% percentile and BCa confidence intervals for $\hat{\theta}$.

Ans:

First, the MLE of Σ, $\hat{\Sigma}$ is computed as:

$$\hat{\Sigma} = \frac{1}{n}(\mathbf{x} - \bar{\mathbf{x}})^{\mathbf{T}}(\mathbf{x} - \bar{\mathbf{x}}),$$

where $\mathbf{x}$ is a 88×5 matrix , and $\bar{\mathbf{x}} = \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathbf{T}}\mathbf{x}$, where $\mathbf{1}_{88\times 1} = \begin{pmatrix}1 & 1 & ... & 1\end{pmatrix}^T$.

Here Compute MLE first and the sample estimate first.

```
data(scor, package = "bootstrap")
dta <- as.matrix(scor)
N <- nrow(scor)

get_fPC <- function(mtx){
  N <- nrow(mtx)
  mlecov <-  cov(mtx)*(N-1)/N # give MLE
  eig_val <- eigen(mlecov)$value
```

```r
  first_PC <- eig_val[1]/sum(eig_val)
  return(first_PC)
}
theta <- get_fPC(scor) #theta hat
cat(glue("$\\hat\\theta={theta}$"))
```

$\hat{\theta} = 0.619115038421291$

1. bootstraping

Construct bootstrap samples X^{*b}, $b = 1, .., B$, $B = 2000$; $\hat{\theta}^{*b} = T(X^{*b})$ is the estimate computed from bootstrap sampleX^{*b}.
Bootstrap bias and s.e. can be computed as :

$$\widehat{bias}_{boot}(\hat{\theta}) = \overline{\hat{\theta}^{*}} - \hat{\theta},$$
$$\hat{se}_{boot}(\hat{\theta}) = \sqrt{\frac{1}{B-1}\Sigma_{b=1}^{B}(\hat{\theta}^{*b} - \overline{\hat{\theta}^{*}})^2}$$

```r
set.seed(989)
Nboot <- 2000
boot_theta <- 1:Nboot
for (i in 1:Nboot){
  boot <- dta[sample(1:N, N, replace = T),] #samp w/ replacement
  boot_theta[i] <- get_fPC(boot)
}
# hist(boot_theta, breaks = 100,freq=F);abline(v = theta, col="red",lwd=2, lty=2)
b.bias <- mean(boot_theta)-theta
b.se <- sd(boot_theta)
cat(glue("Bootstrap Bias (B={Nboot}) = {b.bias}, \n
         Bootstrap s.e. (B={Nboot}) = {b.se}"))
```

Bootstrap Bias (B=2000) = 0.000541774547281215,

Bootstrap s.e. (B=2000) = 0.0470853815115956

2. Jackknife

Jackknife bias and s.e. are defined and coumputed as below:

$$\widehat{bias}_{jack} = (n-1)(\overline{\hat{\theta}_{(\cdot)}} - \hat{\theta}),$$
$$\hat{se}_{jack} = \sqrt{\frac{n-1}{n}\Sigma_{i=1}^{n}(\hat{\theta}_{(i)} - \overline{\hat{\theta}_{(\cdot)}})^2}$$

```r
jack_theta <- 1:N
for (i in 1:N){
  jack <- dta[-i,] #LLO
  jack_theta[i] <- get_fPC(jack)
}
j.bias <- (N-1)*(mean(jack_theta)-theta)
j.se <- sqrt(((N-1)/N)*sum((jack_theta- mean(jack_theta))^2))
```

```
cat(glue("Jackknife Bias ={j.bias}, \n
         Jackknife s.e.={j.se}"))
```

Jackknife Bias =0.00106913888650351,

Jackknife s.e.=0.0495523072701255

3. CI

Compute 95% percentile and BCa confidence intervals

```
conf_level <- .05 # Type 1 error rate
alpha <- c(conf_level/2, 1- (conf_level/2))
zalpha <- qnorm(alpha)
# Percentile 95% CI
p.CI <- quantile(boot_theta, alpha, type=1)
# BCa
## bias correction:
bias <- qnorm(mean(boot_theta - theta>0))
## accelaration
delt.jack <- mean(jack_theta)-jack_theta
accer <- sum(delt.jack^3)/(6*(sum(delt.jack^2))^(3/2))
bca.perc <- (bias+ ((bias+zalpha)/(1-accer*(bias+zalpha))))%>%
  pnorm()
#BCa 95% CI
bca.CI <- quantile(boot_theta, bca.perc, type=6)

cat(glue("95% percentile CI: ({p.CI[1]},
               {p.CI[2]}),\n
               95% BCa CI: ({bca.CI[1]},
               {bca.CI[2]})"))
```

95% percentile CI: (0.518983993903049, 0.705246297087152),

95% BCa CI: (0.530840151059053, 0.710474038860443)

`boot` package:

Additionally, here utilized `boot` to attain bootstrap bias, SE and `boot.ci` for Compute 95% percentile and BCa confidence intervals.

```
get_fPC2 <- function(mtx,i){
  N <- nrow(mtx)
  mlecov <-  cov(mtx[i, ])*(N-1)/N # give MLE
  eig_val <- eigen(mlecov)$value
  first_PC <- eig_val[1]/sum(eig_val)
  return(first_PC)
}
# package
library(boot)
set.seed(989)
boot2 <- boot(dta,get_fPC2, R = Nboot)
```

```r
cat(glue("Bias: {mean(boot2$t) -boot2$t0},\n
         s.e.: {sd(boot2$t)}"))
```

Bias: 0.000899669523645397,

s.e.: 0.0477784005293906

```r
ci_result <- boot.ci(boot2, type = c("base", "perc", "bca"))

cat(glue("95% percentile CI: ({ci_result$percent[4]},
               {ci_result$percent[5]}),\n
               95% BCa CI: ({ci_result$bca[4]},
               {ci_result$bca[5]})"))
```

95% percentile CI: (0.52111640764777, 0.706368205477712),

95% BCa CI: (0.518265990949219, 0.704015517729453)

Visualization

The Histogram below visualized the distribution of Bootstrap Estimates, its 95% percentile CI, 95% BCa CI and the sample estimate $\hat{\theta}$.

```r
hist(boot2$t, breaks = 100, main = glue("Bootstrap Estimates (B={Nboot})"),
     xlab= expression(widehat(theta)^b))
abline(v = boot2$t0, col = "red", lwd=2)
abline(v = ci_result$percent[c(4,5)], col = "darkorange", lwd=2, lty=2)
abline(v = ci_result$bca[c(4,5)], col = "darkcyan", lwd=2, lty=3)
legend("topleft",
       legend = c(expression(widehat(theta)), "Percentile CI", "BCa CI"),
       col = c("red", "darkorange", "darkcyan"), lty = 1:3, lwd=2, cex=.5)
```