ps9

Jared Yu

April 1, 2020

### Problem 1  
# part (a)  
stomach <- c(25, 42, 45, 46, 51, # Load data  
 103, 124, 146, 340, 396, 412, 876, 1112)  
breast <- c(24, 40, 719, 727, 791,  
 1166, 1235, 1581, 1804, 3460, 3808)  
log\_stomach <- log(stomach)  
log\_breast <- log(breast)  
  
par(mfrow=c(1,2))  
boxplot(stomach)  
boxplot(breast)  
dev.off()  
  
par(mfrow=c(1,2))  
boxplot(log(stomach))  
boxplot(log(breast))  
dev.off()  
  
mean(log\_stomach) # 4.96792  
mean(log\_breast) # 6.558603  
  
### Finds the bootstrap t C.I. on log scale  
bootstrap\_t <- function(df, B = 1e3, alpha = 0.05) {  
 log\_df <- log(df) # Take log scale  
 n <- length(df) # Initialize variables  
 T\_F <- mean(log\_df)  
 data\_matrix <- matrix(NA, nrow = B, ncol = length(log\_df))  
 theta\_matrix <- matrix(NA, nrow = B)  
 R\_X\_F\_matrix <- matrix(NA, nrow = B)  
 set.seed(999) # Set seed  
 for (i in 1:B) { # Bootstrap sample  
 data\_matrix[i,] <- sample(log\_df, size = length(log\_df), replace = TRUE)  
 theta\_matrix[i] <- mean(data\_matrix[i,])  
 V\_F\_star <- (1 / (n - 1)) \* sum((data\_matrix[i,] - theta\_matrix[i])^2)  
 R\_X\_F\_matrix[i] <- (theta\_matrix[i] - T\_F) / sqrt(V\_F\_star)  
 }  
   
 theta\_bar <- mean(theta\_matrix) # mean bootstrap theta  
 # variance of bootstrap theta  
 var\_theta\_hat\_star <- sum((theta\_matrix - theta\_bar)^2) / (B - 1)  
   
 # Find quantiles  
 xi <- quantile(R\_X\_F\_matrix, probs = c(alpha / 2, 1 - alpha / 2))  
 names(xi) <- NULL  
 bootstrap\_t <- c(T\_F - sqrt(var\_theta\_hat\_star) \* xi[2],  
 T\_F - sqrt(var\_theta\_hat\_star) \* xi[1])  
 return(bootstrap\_t)  
}  
stomach\_bootstrap\_t\_CI <- bootstrap\_t(df = stomach, B = 1e4, alpha = 0.05) # 4.772891 5.195032  
breast\_bootstrap\_t\_CI <- bootstrap\_t(df = breast, B = 1e4, alpha = 0.05) # 5.891237 6.803640  
  
# part (b)  
# H\_0: no diff. in mean survival times (mu\_log\_breast = mu\_log\_stomach)  
# H\_1: there is a diff. in mean survival times (mu\_log\_breast =/= mu\_log\_stomach)  
n1 <- length(log\_stomach)  
n2 <- length(log\_breast)  
  
t0 <- mean(log\_stomach) - mean(log\_breast)  
m <- choose((n1 + n2), n2)  
m\_combinations <- t(combn(c(log\_stomach, log\_breast), n2)) # m combinations of both data  
stomach\_breast <- c(log\_stomach, log\_breast) # Combined data  
t\_matrix <- matrix(NA, nrow = m) # Empty matrix for t\_i  
  
for (i in 1:m) { # Calculate t\_i for all m combinations of the data  
 ys <- m\_combinations[i,]  
 # Reference: https://stackoverflow.com/questions/5812478/how-i-can-select-rows-from-a-dataframe-that-do-not-match  
 xs <- subset(stomach\_breast, !(stomach\_breast %in% ys))  
 xbar\_new <- mean(xs); ybar\_new <- mean(ys)  
 t\_matrix[i] <- xbar\_new - ybar\_new  
}  
boxplot(t\_matrix)  
k <- which(t0 == sort(t\_matrix, decreasing = TRUE))  
k / m # 0.9925185  
# rej. H\_0 with sig. level. 99.25%  
  
# part (c)  
# percentile method  
percentile\_method <- function(df = breast, B = 1e3, alpha = 0.05, take\_log = TRUE) {  
 if (take\_log == TRUE) {  
 log\_df <- log(df) # Take log scale  
 } else {  
 log\_df <- df  
 }  
 data\_matrix <- matrix(NA, nrow = B, ncol = length(log\_df))  
 theta\_matrix <- matrix(NA, nrow = B)  
 set.seed(888) # Set seed  
 for (i in 1:B) { # Bootstrap sample  
 data\_matrix[i,] <- sample(log\_df, size = length(log\_df), replace = TRUE)  
 theta\_matrix[i] <- mean(data\_matrix[i,])  
 }  
   
 percentile\_CI <- quantile(theta\_matrix, probs = c(alpha / 2, 1 - alpha / 2))  
 names(percentile\_CI) <- NULL  
   
 return(percentile\_CI)  
}  
breast\_percentile\_CI <- percentile\_method(df = breast) # 5.617261 7.375876  
exp(breast\_percentile\_CI) # 275.1347 1596.9896  
exp(breast\_bootstrap\_t\_CI) # 361.8524 901.1214  
mean(breast) # 1395.909  
mean(log\_breast) # 6.558603  
breast\_percentile\_CI\_nonlog <- percentile\_method(df = breast, # 778.1273 2145.3841  
 take\_log = FALSE)  
  
stomach\_percentile\_CI <- percentile\_method(df = stomach) # 4.352572 5.631810  
exp(stomach\_percentile\_CI) # 77.67798 279.16689  
exp(stomach\_bootstrap\_t\_CI) # 118.2606 180.3738  
mean(stomach) # 286  
mean(log\_stomach) # 4.96792  
stomach\_percentile\_CI\_nonlog <- percentile\_method(df = stomach, # 133.30 492.25  
 take\_log = FALSE)  
### Problem 2  
# Cauchy distribution  
xs <- seq(-4, 4, length.out = 1e4)  
plot(xs, dnorm(x = xs, mean = 0, sd = 1), type = 'l',  
 main = 'Normal and Cauchy', xlab = 'x', ylab = 'f(x)')  
lines(xs, dcauchy(x = xs, location = 0, scale = 2), type = 'l', col = 'red')  
legend("topleft", legend = c('Normal', 'Cauchy'), col = c('black', 'red'), lty = c(1,1))  
  
par(mfrow=c(2,2))  
cauchy\_bootstrap <- function(sample\_size = 10, B = 1e4, alpha = 0, beta = 2) {  
 set.seed(888)  
 cauchy\_sample <- rcauchy(n = sample\_size, location = alpha, scale = beta) # Original sample  
 data\_matrix <- matrix(NA, nrow = B, ncol = length(cauchy\_sample)) # Initialize variables  
 theta\_matrix <- matrix(NA, nrow = B)  
 set.seed(888) # Set seed  
 for (i in 1:B) { # Bootstrap sample  
 data\_matrix[i,] <- sample(cauchy\_sample, size = length(cauchy\_sample), replace = TRUE)  
 theta\_matrix[i] <- mean(data\_matrix[i,])  
 }  
 hist(theta\_matrix, main = paste('Histogram of the Sample Mean (n=',sample\_size,')'),  
 xlab = 'Bootstrap Sample Mean',  
 sub = paste('Original Sample Mean:', round(mean(cauchy\_sample), 4),  
 ' Bootstrap Sample Mean:', round(mean(theta\_matrix), 4)))  
}  
cauchy\_bootstrap(sample\_size = 1e1)  
cauchy\_bootstrap(sample\_size = 5e1)  
cauchy\_bootstrap(sample\_size = 1e2)  
cauchy\_bootstrap(sample\_size = 1e3)  
  
# Uniform distribution  
xs <- seq(0, 10, length.out = 1e4)  
plot(xs, dunif(xs, 0, 1e4), type = 'l', ylim = c(0, 0.00014))  
segments(x0 = c(0,10), y0 = 0, x1 = c(0,10), y1 = dunif(xs, 0, 1e4))  
abline(h = 0)  
  
theta <- 10  
set.seed(999)  
unif\_sample <- runif(n = 1e1, min = 0, max = theta)  
sample(unif\_sample, size = length(unif\_sample), replace = TRUE)  
theta\_hat <- max(unif\_sample)  
  
unif\_bootstrap <- function(sample\_size = 100, B = 1e4, theta\_max = 10) {  
 set.seed(999) # Set seed  
 unif\_sample <- runif(n = sample\_size, min = 0, max = theta\_max) # Original sample  
 data\_matrix <- matrix(NA, nrow = B, ncol = sample\_size) # Initialize variables  
 theta\_matrix <- matrix(NA, nrow = B)  
 set.seed(999) # Set seed  
 for (i in 1:B) { # Bootstrap sample  
 data\_matrix[i,] <- sample(unif\_sample, size = sample\_size, replace = TRUE)  
 theta\_matrix[i] <- max(data\_matrix[i,])  
 }  
 hist(theta\_matrix, main = paste('Histogram of the Maximum Order Statistic (n=',sample\_size,')'),  
 xlab = 'Bootstrap Maximum Order Statistic',  
 sub = paste('Original Max. Order Statistic:', round(max(unif\_sample), 4),  
 ' Mean Bootstrap Max. Order Statistic:', round(mean(theta\_matrix), 4)))  
}  
par(mfrow = c(2,2))  
unif\_bootstrap(sample\_size = 1e1, B = 1e4)  
unif\_bootstrap(sample\_size = 1e2, B = 1e4)  
unif\_bootstrap(sample\_size = 1e3, B = 1e4)  
unif\_bootstrap(sample\_size = 1e4, B = 1e4)  
dev.off()  
  
### Problem 4  
# part (a)  
set.seed(999)  
sample\_size <- 1e2  
random\_sample <- rnorm(n = sample\_size, mean = 0, sd = 1)  
sample\_mean <- mean(random\_sample)  
  
# part (b)  
B <- 1e1  
set.seed(999) # Set seed  
data\_matrix <- matrix(NA, nrow = B, ncol = sample\_size) # Initialize variables  
theta\_matrix <- matrix(NA, nrow = B)  
set.seed(999) # Set seed  
for (i in 1:B) { # Bootstrap sample  
 data\_matrix[i,] <- sample(random\_sample, size = sample\_size, replace = TRUE)  
 theta\_matrix[i] <- mean(data\_matrix[i,])  
}  
hist(theta\_matrix, main = paste('Histogram of the Sample Mean (n=',sample\_size,')'),  
 xlab = 'Bootstrap Sample Mean',  
 sub = paste('Original Sample Mean:', round(sample\_mean, 4),  
 ' Mean Bootstrap Sample Mean:', round(mean(theta\_matrix), 4)))  
mu\_bar\_star <- mean(theta\_matrix)  
mu\_bar\_star - sample\_mean  
2 \* sample\_mean - mu\_bar\_star  
  
(1 / (B - 1)) \* sum((theta\_matrix - mu\_bar\_star)^2)  
  
# part (c)  
concat\_B <- rep(random\_sample, B) # create n\*B vector  
set.seed(999)  
balanced\_bootstrap <- sample(concat\_B, # permute n\*B vector  
 size = length(concat\_B), replace = FALSE)  
# Reference: https://stackoverflow.com/questions/3318333/split-a-vector-into-chunks-in-r  
balanced\_groups <- split(balanced\_bootstrap, # split into B groups  
 rep\_len(1:B, length(balanced\_bootstrap)))  
balanced\_theta <- unlist(lapply(balanced\_groups, mean)) # calculate theta for each  
names(balanced\_theta) <- NULL  
  
hist(balanced\_theta, main = paste('Histogram of the Balanced Sample Mean (n=',sample\_size,')'),  
 xlab = 'Balanced Bootstrap Sample Mean',  
 sub = paste('Original Sample Mean:', round(sample\_mean, 4),  
 ' Mean Bootstrap Sample Mean:', round(mean(balanced\_theta), 4)))  
mu\_bar\_star <- mean(balanced\_theta)  
mu\_bar\_star - sample\_mean  
2 \* sample\_mean - mu\_bar\_star  
(1 / (B - 1)) \* sum((theta\_matrix - mu\_bar\_star)^2)