STAT 440 Homework 10

Charlie Lu (Cxl5159)

October 2022

```r
bootstrap = function(samples, B, estimator) {
  #' generic bootstrap function
  #' @param samples vector of samples
  #' @param B number of bootstrap estimates
  #' @param estimator estimating function

  # generate matrix of bootstrap resamples
  n = dim(samples)[1]
  # resamples = matrix(
  #   sample(samples, size=n*B, replace=TRUE),
  #   nrow=B
  # )
  bootstrap_ests = c()
  for(i in 1:B){
    index = sample(1:n, n , replace = TRUE)
    sample_new = samples[index, ]
    bootstrap_ests[i] = estimator(sample_new)
  }

  #d <- split(temp,rep(1:B,each=n/2))
  #print(d)
  #print(samples[resamples,])
  # apply to each row
  # bootstrap_ests = apply(resamples, 1, estimator)
  # print(bootstrap_ests)
  # return bootstrap mean and standard error estimates
  c(
    mean(bootstrap_ests),
    sqrt(var(bootstrap_ests))
  )
}

est_function = function(a){
  x = a[,1]
  y = a[,2]
  return(cor(x,y))
}
set.seed(440)
bootstrap_corr = bootstrap(sample_set, 10000, est_function)
bootstrap_corr
c(lower=bootstrap_corr[1] - 2 * bootstrap_corr[2],
  est=bootstrap_corr[1],
  upper=bootstrap_corr[1] + 2 * bootstrap_corr[2])
```

47:1 (Top Level)

Console Terminal Background Jobs

R 4.2.1 · C:/Users/Charlie Lu/Desktop/

```
> bootstrap_corr
[1] 0.7714517 0.1340042
> c(lower=bootstrap_corr[1] - 2 * bootstrap_corr[2],
+   est=bootstrap_corr[1],
+   upper=bootstrap_corr[1] + 2 * bootstrap_corr[2])
    lower       est     upper 
0.5034433 0.7714517 1.0394600 
```

1 B

```r
#Boot function
set.seed(440)
est_function1 = function(a,index){
  x = a[index,1]
  y = a[index,2]
  return(cor(x,y))
}
set.seed(440)
non <- boot(sample_set, est_function1, R = 10000)
non
c(0.7763745 - 2*0.132994,0.7763745+ 2*0.132994)
c(lower=bootstrap_corr[1] - 2 * bootstrap_corr[2],
  est=bootstrap_corr[1],
  upper=bootstrap_corr[1] + 2 * bootstrap_corr[2])
```

Console Terminal Background Jobs

R 4.2.1 · C:/Users/Charlie Lu/Desktop/

```
Bootstrap Statistics :
     original        bias    std. error
t1* 0.7763745 -0.004161978    0.132994
> c(0.7763745 - 2*0.132994,0.7763745+ 2*0.132994)
[1] 0.5103865 1.0423625
> c(lower=bootstrap_corr[1] - 2 * bootstrap_corr[2],
+   est=bootstrap_corr[1],
+   upper=bootstrap_corr[1] + 2 * bootstrap_corr[2])
    lower       est     upper
0.5034433 0.7714517 1.0394600
>
```

Comparing the 2 results we can see that they are very similar which makes sense because it's functionally similar as well.

2 C

```r
#Parametric
mu <- c(mean(data$EntranceExam),mean(data$GPA))
mu
sigma <- cov(data.frame(sample_set))
sigma
```

81:1 (Top Level)

Console Terminal Background Jobs

R 4.2.1 · C:/Users/Charlie Lu/Desktop/

```
> #Parametric
> mu <- c(mean(data$EntranceExam),mean(data$GPA))
> mu
[1] 600.266667   3.094667
> sigma <- cov(data.frame(sample_set))
> sigma
          X1          X2
X1 0.0592981    7.901524
X2 7.9015238 1746.780952
>
>
```

3 D

```
set.seed(440)
binorm <- mvrnorm(10000,mu,sigma)
mv_mu <- c(mean(binorm[,1]),mean(binorm[,2]))

mv_mu
mv_cov <- cov(data.frame(binorm))
mv_cov
```

89:1 (Top Level)

Console Terminal Background Jobs

R 4.2.1 · C:/Users/Charlie Lu/Desktop/

```
> binorm <- mvrnorm(10000,mu,sigma)
> mv_mu <- c(mean(binorm[,1]),mean(binorm[,2]))
> mv_mu
[1] 600.268607   3.415991
> mv_cov <- cov(data.frame(binorm))
> mv_cov
          X1          X2
X1 0.0588713    7.981507
X2 7.9815073 1782.365149
>
```

Compared the previous values we received, we can see that they are almost identical.

4 E

```r
set.seed(440)
# aggregate our estimates using bootstrap rules
bootstrap_cov_est = mean(bootstrap_covs)
bootstrap_cov_se = sqrt(var(bootstrap_covs))
bootstrap_cov_est
bootstrap_cov_se

c(lower=bootstrap_cov_est - 2 * bootstrap_cov_se,
  est=bootstrap_cov_est,
  upper=bootstrap_cov_est + 2 * bootstrap_cov_se)

c(lower=(bootstrap_cov_est - 2 * bootstrap_cov_se) / sqrt(prod(diag(sigma))),
  est=bootstrap_cov_est / sqrt(prod(diag(sigma))),
  upper=(bootstrap_cov_est + 2 * bootstrap_cov_se) / sqrt(prod(diag(sigma))))
```

113:1 (Top Level)

Console Terminal Background Jobs

R 4.2.1 · C:/Users/Charlie Lu/Desktop/

```
> c(lower=bootstrap_cov_est - 2 * bootstrap_cov_se,
+   est=bootstrap_cov_est,
+   upper=bootstrap_cov_est + 2 * bootstrap_cov_se)
     lower        est      upper
 0.7811327  7.8704076 14.9596825
> c(lower=(bootstrap_cov_est - 2 * bootstrap_cov_se) / sqrt(prod(diag(sigma))),
+   est=bootstrap_cov_est / sqrt(prod(diag(sigma))),
+   upper=(bootstrap_cov_est + 2 * bootstrap_cov_se) / sqrt(prod(diag(sigma))))
     lower        est      upper
0.07675121 0.77331713 1.46988305
>
```

Compared with the non-parametric version, the results are different by a significant margin. This makes sense because the parametric version only uses the sample data without any assumptions while the Non parametric assumes the distribution so natually the results would differ as well.