Week Four Exercises

Adeline Casali

2023-11-15

# Question 6

Load packages and Data

library(ISLR2)  
library(boot)  
Default <- read.csv("Default.csv")

1. Using glm(), standard error for income was estimated to be 1.274e-07 and the standard error for balance was estimated to be 3.514e-06.

set.seed(123)  
Default$default <- ifelse(Default$default == "Yes", 1, 0)  
default\_lmodel <- glm(default ~ income + balance, data = Default)  
summary(default\_lmodel)

##   
## Call:  
## glm(formula = default ~ income + balance, data = Default)  
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -9.224e-02 5.788e-03 -15.936 < 2e-16 \*\*\*  
## income 4.605e-07 1.274e-07 3.613 0.000304 \*\*\*  
## balance 1.318e-04 3.514e-06 37.511 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.02821664)  
##   
## Null deviance: 321.91 on 9999 degrees of freedom  
## Residual deviance: 282.08 on 9997 degrees of freedom  
## AIC: -7294.7  
##   
## Number of Fisher Scoring iterations: 2

1. Here is a function that will output the coefficient estimates for income and balance from the logistic regression model.

boot.fn <- function(data, index) {  
 coef(glm(default ~ income + balance, data = data, subset = index))  
}

1. Using boot() gives an estimated standard error of 1.297e-07 for income and 6.606e-06 for balance.

set.seed(123)  
boot(Default, boot.fn, 1000)

##   
## ORDINARY NONPARAMETRIC BOOTSTRAP  
##   
##   
## Call:  
## boot(data = Default, statistic = boot.fn, R = 1000)  
##   
##   
## Bootstrap Statistics :  
## original bias std. error  
## t1\* -9.223968e-02 -7.741349e-05 6.549888e-03  
## t2\* 4.604568e-07 3.668321e-09 1.297201e-07  
## t3\* 1.318050e-04 -6.404928e-08 6.606168e-06

1. The estimated standard errors obtained using both the glm() function and the bootstrap method were incredibly similar. Although they do differ slightly, they are so small and so close that it doesn’t actually make a difference.

glm\_std\_errors <- c(1.274e-07, 3.514e-06)  
boot\_std\_errors <- c(1.297e-07, 6.606e-06)  
std\_err\_compare <- data.frame(  
 Method = c("glm()", "boot()"),  
 Std\_Error = c(glm\_std\_errors, boot\_std\_errors),  
 Variable = rep(c("Income", "Balance"), each = 2)  
)  
print(std\_err\_compare)

## Method Std\_Error Variable  
## 1 glm() 1.274e-07 Income  
## 2 boot() 3.514e-06 Income  
## 3 glm() 1.297e-07 Balance  
## 4 boot() 6.606e-06 Balance

# Question 9

1. An estimate of the population mean of median value of owner occupied homes in 1000s.

mu\_hat <- mean(Boston$medv)  
print(mu\_hat)

## [1] 22.53281

1. The standard error represents the typical amount of variability we might expect in the sample mean if we were to take multiple random samples from the population. In this case, the estimate of 0.4088611 indicates the standard deviation of the distribution of sample means.

se\_mu\_hat <- sd(Boston$medv) / sqrt(length(Boston$medv))  
print(se\_mu\_hat)

## [1] 0.4088611

1. The standard error generated by the bootstrap methos is 0.4045557, which is extremely close to the standard error calculated in (b) of 0.4088611.

mean.fn <- function(data, index) {  
 sample\_data <- data$medv[index]  
 return(mean(sample\_data))  
}  
set.seed(123)  
boot(Boston, mean.fn, R = 1000)

##   
## ORDINARY NONPARAMETRIC BOOTSTRAP  
##   
##   
## Call:  
## boot(data = Boston, statistic = mean.fn, R = 1000)  
##   
##   
## Bootstrap Statistics :  
## original bias std. error  
## t1\* 22.53281 -0.01607372 0.4045557

1. The 95% confidence interval calculated from the bootstrap estimate of SE results in a range of 21.72369 to 23.34192. The range obtained from the t-test is 21.72953 to 23.33608. These intervals are extremely close, differing by less than a hundredth on either side.

bs\_se\_mu\_hat <- 0.4045557  
ci <- c(mu\_hat - 2 \* bs\_se\_mu\_hat, mu\_hat + 2 \* bs\_se\_mu\_hat)  
print(ci)

## [1] 21.72369 23.34192

t.test(Boston$medv)

##   
## One Sample t-test  
##   
## data: Boston$medv  
## t = 55.111, df = 505, p-value < 2.2e-16  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## 21.72953 23.33608  
## sample estimates:  
## mean of x   
## 22.53281

1. An estimate of the population median of the median value of owner occupied homes in 1000s.

mu\_hat\_med <- median(Boston$medv)  
print(mu\_hat\_med)

## [1] 21.2

1. The standard error of the median was estimated to be 0.3676453 by the bootstrap method.

median.fn <- function(data, index) {  
 sample\_data <- data$medv[index]  
 return(median(sample\_data))  
}  
set.seed(123)  
boot(Boston, median.fn, R = 1000)

##   
## ORDINARY NONPARAMETRIC BOOTSTRAP  
##   
##   
## Call:  
## boot(data = Boston, statistic = median.fn, R = 1000)  
##   
##   
## Bootstrap Statistics :  
## original bias std. error  
## t1\* 21.2 -0.0203 0.3676453

1. An estimate of the population tenth percentile of the median value of owner occupied homes in 1000s.

mu\_hat\_0.1 <- quantile(Boston$medv, 0.1)  
mu\_hat\_0.1

## 10%   
## 12.75

1. The standard error of the tenth percentile was estimated to be 0.527868 by the bootstrap method.

tenth.fn <- function(data, index) {  
 sample\_data <- data$medv[index]  
 return(quantile(sample\_data, 0.1))  
}  
set.seed(123)  
boot(Boston, tenth.fn, R = 1000)

##   
## ORDINARY NONPARAMETRIC BOOTSTRAP  
##   
##   
## Call:  
## boot(data = Boston, statistic = tenth.fn, R = 1000)  
##   
##   
## Bootstrap Statistics :  
## original bias std. error  
## t1\* 12.75 -0.012 0.527868