DSE6111 Module 4 - Ch 5 HW

Nathan Monges

2024-07-28

```r
library(ISLR2)
library(boot)
```

Exercise 6

a) Using the summary() and glm() functions, determine the esti- mated standard errors for the coefficients associated with income and balance in a multiple logistic regression model that uses both predictors.

```r
attach(Default)
set.seed(123)

summary(glm(default ~ income + balance, data = Default, family = binomial))$coef
```

```
##                  Estimate   Std. Error    z value      Pr(>|z|)
## (Intercept) -1.154047e+01 4.347564e-01 -26.544680 2.958355e-155
## income       2.080898e-05 4.985167e-06   4.174178  2.990638e-05
## balance      5.647103e-03 2.273731e-04  24.836280 3.638120e-136
```

b) Write a function, boot.fn(), that takes as input the Default data set as well as an index of the observations, and that outputs the coefficient estimates for income and balance in the multiple logistic regression model.

```r
boot.fn <- function(data, index) {
  def.log <- glm(default ~ income + balance, data = data[index, ], family = binomial)

  return(coef(def.log)[c("income", "balance")])
}
```

c) Use the boot() function together with your boot.fn() function to estimate the standard errors of the logistic regression coefficients for income and balance.

```r
boot(Default, boot.fn, 1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Default, statistic = boot.fn, R = 1000)
##
##
## Bootstrap Statistics :
##         original        bias     std. error
## t1* 2.080898e-05 1.582518e-07 4.729534e-06
## t2* 5.647103e-03 1.296980e-05 2.217214e-04
```

d) Comment on the estimated standard errors obtained using the glm() function and using your bootstrap function.

The standard errors are quite close in both methods, standard formula gives a standard error of 4.985167e-06 and the bootstrap method gives a standard error of 4.729534e-06 for the income variable. For the balance varibale, the standard formula gives a standard error of 2.217214e-0 and the boot strap method gives a slightly smaller standard error of 2.273282e-04. These small differences in standard errors in both approaches could indicate that stnadard formula from the glm() functioin is pretty accurate in this dataset.

Exercise 9

(a) Based on this data set, provide an estimate for the population mean of medv. Call this estimate .

```
attach(Boston)

medv_mean <- mean(medv)
```

b) Provide an estimate of the standard error of . Interpret this result. Hint: We can compute the standard error of the sample mean by dividing the sample standard deviation by the square root of the number of observations.

```
sd_medv <- sd(medv)
n <- length(medv)

se_medv <- sd_medv / sqrt(n)
se_medv
```

```
## [1] 0.4088611
```

c) Now estimate the standard error of using the bootstrap. How does this compare to your answer from (b)?

```
set.seed(123)

dim(Boston)
```

```
## [1] 506  13
```

```
bos.boot.fn <- function(data, index) {
  sample <- data[index, ]
  mean(sample$medv)
}

se_medv_boot <- boot(Boston, bos.boot.fn, R = 506)
se_medv_boot
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Boston, statistic = bos.boot.fn, R = 506)
##
##
## Bootstrap Statistics :
##     original       bias    std. error
## t1* 22.53281 -0.01148784   0.3976881
```

```
se_medv_boot_n <- 0.3976881
```

The bootstrap method gives a standard error of 0.39768 which is slightly smaller than that calculated in part b (0.40886). Since both estimates are relativelt close to each other, this helps to ensure that the standard error estimates from both methods are reliable.

d) Based on your bootstrap estimate from (c), provide a 95 % confidence interval for the mean of medv. Compare it to the results obtained using t.test(Boston$medv).

```
lower <- medv_mean - 2 * se_medv_boot_n
upper <- medv_mean + 2 * se_medv_boot_n

cat(lower, ",", upper)
```

```
## 21.73743 , 23.32818
```

```
t.test(medv)
```

```
##
##  One Sample t-test
##
## data:  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
```

The 95% confidence interval for the mean of medv are very similar in both approaches with an interval of (21.73743,23.32818) in the standard formula, and a interval of (21.72953,23.33608) from the t.test function.

e) Based on this data set, provide an estimate,med, for the median value of medv in the population.

```
median_medv <- median(medv)
median_medv
```

```
## [1] 21.2
```

f) We now would like to estimate the standard error of med.Unfortunately, there is no simple formula for computing the standard error of the median. Instead, estimate the standard error of the median using the bootstrap. Comment on your findings.

```
median_boot.fn <- function(data, index) {
  sample <- data[index, ]
  median(sample$medv)
}

set.seed(123)

median_se <- boot(Boston, median_boot.fn, R = 506)
median_se
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
```

```
## boot(data = Boston, statistic = median_boot.fn, R = 506)
##
##
## Bootstrap Statistics :
##     original      bias    std. error
## t1*     21.2 -0.01373518   0.3586058
```

From using the bootstrap method to calculate the standard error of the median medv, this results in a standard error of 0.3586058 which is a relatively low standard error which represents the estimated variability of the sample median acroos the different bootstrap samples.

g) Based on this data set, provide an estimate for the tenth percentile of medv in Boston census tracts. Call this quantity =ˆ0.1. (You can use the quantile() function.)

```
percentile <- quantile(medv, probs = 0.10)
percentile
```

```
##   10%
## 12.75
```

h) Use the bootstrap to estimate the standard error of. Comment on your findings.

```
percentile_boot.fn <- function(data, index) {
  sample <- data[index, ]
  quantile(sample$medv, probs = 0.10)
}

set.seed(123)

percentile_se <- boot(Boston, percentile_boot.fn, R = 506)
percentile_se
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Boston, statistic = percentile_boot.fn, R = 506)
##
##
## Bootstrap Statistics :
##     original      bias    std. error
## t1*    12.75 -0.02608696   0.5087174
```

From the bootstrap method on the 10th percentile of the medv variable, we get a value of 0.5087174 for the standard error. This has been the highest standard error out of all the variables we used to calculate standard error. Although the value of the standard eror in this case is not significantly high, meaning there would not be much variability in the value of the 10th percentile, if we were to repedeatlty run this process with different samples from the data.