Annand Module 04 Lab 01

Joseph Annand

2023-11-19

Import Libraries

```
library(ISLR2)
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:ISLR2':
##
##     Boston
```

```
library(boot)
```

Question 6

Import Data

```
default.data <- Default

set.seed(1)
train.default <- sample(nrow(default.data), nrow(default.data) / 2)
```

Part A

```
glm.default <- glm(default ~ income + balance, data = default.data,
                   subset = train.default, family = binomial)

summary(glm.default)
```

```
##
## Call:
## glm(formula = default ~ income + balance, family = binomial,
##     data = default.data, subset = train.default)
##
```

```
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.194e+01  6.178e-01 -19.333  < 2e-16 ***
## income       3.262e-05  7.024e-06   4.644 3.41e-06 ***
## balance      5.689e-03  3.158e-04  18.014  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1523.8  on 4999  degrees of freedom
## Residual deviance:  803.3  on 4997  degrees of freedom
## AIC: 809.3
##
## Number of Fisher Scoring iterations: 8
```

Part B

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

boot.fn(default.data, train.default)
```

```
##   (Intercept)        income       balance
## -1.194413e+01  3.262025e-05  5.689218e-03
```

Part C

```
set.seed(9)
boot(default.data, boot.fn, 1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = default.data, statistic = boot.fn, R = 1000)
##
##
## Bootstrap Statistics :
##          original        bias     std. error
## t1* -1.154047e+01 -2.593696e-02 4.234840e-01
## t2*  2.080898e-05 -1.006081e-07 4.768061e-06
## t3*  5.647103e-03  1.608805e-05 2.237253e-04
```

Part D

The standard error for the parameter estimates are very similar using bootstrap approach and using training data set. The parameter estimates for balance are nearly the same; however, the estimates for income are noticeably different.

Question 9

Import Data

```
boston.data <- Boston
```

Part A

```
u_hat <- mean(boston.data$medv)
u_hat
```

```
## [1] 22.53281
```

Part B

```
se_hat <- sd(boston.data$medv) / sqrt(nrow(boston.data))
se_hat
```

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

Part C

```
u_medv.fn <- function(data, index) {
  z <- data$medv[index]
  sum(z) / length(z)
}

# Bootstrap calcuation for part b
set.seed(8)
boot.b <- boot(boston.data, u_medv.fn, R = 1000)
boot.b
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = boston.data, statistic = u_medv.fn, R = 1000)
##
##
## Bootstrap Statistics :
##     original      bias    std. error
## t1* 22.53281 0.007920751   0.3990518
```

The standard errors form part b and the boostrap are very similar with the bootstrap estimate being only about 0.007 units less.

Part D

```
# Bootstrap from part c estimates mean = 22.53281 and std err = 0.3990518
low_ci <- 22.53281 - 2*0.3990518
upper_ci <- 22.53281 + 2*0.3990518

# Use boot.ci() function to estimate confidence intervals
boot.ci(boot.b, conf = 0.95, type="all")
```

```
## Warning in boot.ci(boot.b, conf = 0.95, type = "all"): bootstrap variances
## needed for studentized intervals

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot.b, conf = 0.95, type = "all")
##
## Intervals :
## Level      Normal              Basic
## 95%   (21.74, 23.31 )   (21.75, 23.33 )
##
## Level     Percentile            BCa
## 95%   (21.74, 23.31 )   (21.74, 23.30 )
## Calculations and Intervals on Original Scale
```

```
# Use t-test to estimate confidence intervals
t.test(boston.data$medv)
```

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

Manually estimating confidence interval and using the t-test yield practically identical results. The estimate from the boot.ci() function is slightly different but more or less the same.

Part E

```
med_hat <- median(boston.data$medv)
med_hat
```

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

Part F

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

set.seed(10)
boot.f <- boot(boston.data, med_medv.fn, 1000)
boot.f
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = boston.data, statistic = med_medv.fn, R = 1000)
##
##
## Bootstrap Statistics :
##     original   bias    std. error
## t1*     21.2 -0.00665   0.3745779
```

The bootstrap approach estimated the same value for the median as in part E. The standard error for the median is slightly lower than that of bootstrap estimation for mean in part C.

Part G

```
# Get tenth percentile of medv
u_10 <- quantile(boston.data$medv, probs = 0.1, na.rm = FALSE)
u_10
```

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

Part H

```
ten_medv.fn <- function(data, index) {
  quantile(data$medv, probs = 0.1, na.rm = FALSE)
}

set.seed(2)
boot.h <- boot(boston.data, ten_medv.fn, 1000)
boot.h
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
```

```
## Call:
## boot(data = boston.data, statistic = ten_medv.fn, R = 1000)
##
##
## Bootstrap Statistics :
##     original  bias    std. error
## t1*    12.75       0           0
```

The bootstrap approach matches the estimate for the tenth percentile of medv from part G exactly.