Santhoshini sree Bolisetty
Alexa Summers
Gireesh kumar Muppalla
Lab #3-CS 5565

(5)
(a)

```
library(ISLR)
attach(Default)
set.seed(1)
fit <- glm(default ~ income + balance, data = Default, family = "binomial")
summary(fit)
```

6:1 (Top Level)

```
Console ~/

    balance, default, income, student

> set.seed(1)
> fit <- glm(default ~ income + balance, data = Default, family = "binomial")
> summary(fit)

Call:
glm(formula = default ~ income + balance, family = "binomial",
    data = Default)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4725  -0.1444  -0.0574  -0.0211   3.7245

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.154e+01  4.348e-01 -26.545  < 2e-16 ***
income       2.081e-05  4.985e-06   4.174 2.99e-05 ***
balance      5.647e-03  2.274e-04  24.836  < 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: 2920.6  on 9999  degrees of freedom
Residual deviance: 1579.0  on 9997  degrees of freedom
AIC: 1585

Number of Fisher Scoring iterations: 8

>
```

(b)
B1

```
train <- sample(dim(Default)[1], dim(Default)[1] / 2)
```

13:1 (Top Level)

Santhoshini sree Bolisetty
Alexa Summers
Gireesh kumar Muppalla
Lab #3-CS 5565

B2

```
train <- sample(dim(Default)[1], dim(Default)[1] / 2)
fit <- glm(default ~ income + balance, data = Default, family = "binomial", subset = train)
summary(fit)
```

```
26:1  (Top Level) ÷                                                          R Scr
```

```
Console ~/

> train <- sample(dim(Default)[1], dim(Default)[1] / 2)
> fit <- glm(default ~ income + balance, data = Default, family = "binomial", subset = train)
> summary(fit)

Call:
glm(formula = default ~ income + balance, family = "binomial",
    data = Default, subset = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1891  -0.1573  -0.0605  -0.0226   3.6623

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.136e+01  5.982e-01 -18.987  < 2e-16 ***
income       2.265e-05  6.947e-06   3.259  0.00112 **
balance      5.530e-03  3.142e-04  17.600  < 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: 1483.83  on 4999  degrees of freedom
Residual deviance:  829.05  on 4997  degrees of freedom
AIC: 835.05

Number of Fisher Scoring iterations: 8
```

B3

```
prob <- predict(fit, new= Default[-train, ], type = "response")
pred <- rep("No", length(prob))
pred[prob > 0.5] <- "Yes"
```

```
22:1  (Top Level) ÷
```

```
Console ~/

> prob <- predict(fit, new= Default[-train, ], type = "response")
> pred <- rep("No", length(prob))
> pred[prob > 0.5] <- "Yes"
>
```

Santhoshini sree Bolisetty
Alexa Summers
Gireesh kumar Muppalla
Lab #3-CS 5565

B4

```
prob <- predict(fit, new= Default[-train, ], type = "response")
pred <- rep("No", length(prob))
pred[prob > 0.5] <- "Yes"

mean(pred != Default[-train, ]$default)
```

```
24:1 (Top Level)
```

```
Console ~/
> prob <- predict(fit, new= Default[-train, ], type = "response")
> pred <- rep("No", length(prob))
> pred[prob > 0.5] <- "Yes"
> mean(pred != Default[-train, ]$default)
[1] 0.028
>
```

(c)

Depending on different predictors included in the training and validation set, the validation test error rate is varying.

```
train <- sample(dim(Default)[1], dim(Default)[1] / 2)
fit <- glm(default ~ income + balance, data = Default, family = "binomial", subset = train)
prob <- predict(fit, new= Default[-train, ], type = "response")
pred<- rep("No", length(prob))
pred[prob > 0.5] <- "Yes"
mean(pred != Default[-train, ]$default)
```

```
31:1 (Top Level)
```

```
Console ~/
> train <- sample(dim(Default)[1], dim(Default)[1] / 2)
> fit <- glm(default ~ income + balance, data = Default, family = "binomial", subset = train)
> prob <- predict(fit, new= Default[-train, ], type = "response")
> pred<- rep("No", length(prob))
> pred[prob > 0.5] <- "Yes"
> mean(pred != Default[-train, ]$default)
[1] 0.0252
> train <- sample(dim(Default)[1], dim(Default)[1] / 2)
> fit <- glm(default ~ income + balance, data = Default, family = "binomial", subset = train)
> prob <- predict(fit, new= Default[-train, ], type = "response")
> pred<- rep("No", length(prob))
> pred[prob > 0.5] <- "Yes"
> mean(pred != Default[-train, ]$default)
[1] 0.0246
> train <- sample(dim(Default)[1], dim(Default)[1] / 2)
> fit <- glm(default ~ income + balance, data = Default, family = "binomial", subset = train)
> prob <- predict(fit, new= Default[-train, ], type = "response")
> pred<- rep("No", length(prob))
> pred[prob > 0.5] <- "Yes"
> mean(pred != Default[-train, ]$default)
[1] 0.0266
>
```

Santhoshini sree Bolisetty
Alexa Summers
Gireesh kumar Muppalla
Lab #3-CS 5565

(d)
Including student dummy variable is not showing any decreasing difference in the test error

```r
train <- sample(dim(Default)[1], dim(Default)[1] / 2)
fit <- glm(default ~ income + balance+student, data = Default, family = "binomial", subset = train)
prob <- predict(fit, new= Default[-train, ], type = "response")
pred<- rep("No", length(prob))
pred[prob > 0.5] <- "Yes"
mean(pred != Default[-train, ]$default)
```

31:1 (Top Level) R Script

Console ~/

```
> train <- sample(dim(Default)[1], dim(Default)[1] / 2)
> fit <- glm(default ~ income + balance+student, data = Default, family = "binomial", subset = train)
> prob <- predict(fit, new= Default[-train, ], type = "response")
> pred<- rep("No", length(prob))
> pred[prob > 0.5] <- "Yes"
> mean(pred != Default[-train, ]$default)
[1] 0.0254
>
```

Santhoshini sree Bolisetty
Alexa Summers
Gireesh kumar Muppalla
Lab #3-CS 5565

(6)
(a)
The standard error rates of the coefficients beta0, beta1 and beta2 are 0.4348, 0.000004985, 0.0002274

```
library(ISLR)
attach(Default)
set.seed(1)
fit <- glm(default ~ income + balance, data = Default, family = "binomial")
summary(fit)

7:1   (Top Level) ÷
```

```
Console ~/
> fit <- glm(default ~ income + balance, data = Default, family = "binomial")
> summary(fit)

Call:
glm(formula = default ~ income + balance, family = "binomial",
    data = Default)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4725  -0.1444  -0.0574  -0.0211   3.7245

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.154e+01  4.348e-01 -26.545  < 2e-16 ***
income       2.081e-05  4.985e-06   4.174 2.99e-05 ***
balance      5.647e-03  2.274e-04  24.836  < 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: 2920.6  on 9999  degrees of freedom
Residual deviance: 1579.0  on 9997  degrees of freedom
AIC: 1585

Number of Fisher Scoring iterations: 8

> 
```

(b)

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

37:14   (Top Level) ÷
```

```
Console ~/
> boot.fn <- function(data, index) {
+   fit <- glm(default ~ income + balance, data = data, family = "binomial", subset = index)
+   return (coef(fit))
+ }
```

Santhoshini sree Bolisetty
Alexa Summers
Gireesh kumar Muppalla
Lab #3-CS 5565

(c)

The bootstrap standard error rates of beta0, beta1 and beta2 are 4.285621e-01, 4.950151e-06 and 2.236899e-04

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

37:14 (Top Level)

Console ~/

```
> library("boot", lib.loc="C:/Program Files/R/R-3.5.2/library")
> boot(Default, boot.fn, 1000)

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = Default, statistic = boot.fn, R = 1000)


Bootstrap Statistics :
         original        bias     std. error
t1* -1.154047e+01 -2.099044e-02 4.285621e-01
t2*  2.080898e-05  1.716564e-07 4.950151e-06
t3*  5.647103e-03  6.622474e-06 2.236899e-04
>
```

(d)

The two methods are giving approximately same error rates.

Santhoshini sree Bolisetty
Alexa Summers
Gireesh kumar Muppalla
Lab #3-CS 5565

(9)

(a)

```
library(MASS)
attach(Boston)
mue<-mean(medv)
mue
```

62:1 (Top Level)

Console ~/

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

> library(MASS)
> attach(Boston)
The following object is masked _by_ .GlobalEnv:

    age

> mue<-mean(medv)
> mue
[1] 22.53281
>
```

Santhoshini sree Bolisetty
Alexa Summers
Gireesh kumar Muppalla
Lab #3-CS 5565

(b)

```
se <- sd(medv) / sqrt(dim(Boston)[1])
se
```

66:1 (Top Level)

Console ~/

```
> se <- sd(medv) / sqrt(dim(Boston)[1])
> se
[1] 0.4088611
>
```

(c)
The bootstrap standard error rate is 0.41193 which is closer to 0.4088 that is the value obtained from b

```
set.seed(1)
boot.fn <- function(data, index) {
  mu <- mean(data[index])
  return (mu)
}
library(boot)
boot(medv, boot.fn, 1000)
```

92:1 (Top Level)

Console ~/

```
> set.seed(1)
> boot.fn <- function(data, index) {
+   mu <- mean(data[index])
+   return (mu)
+ }
> library(boot)
> boot(medv, boot.fn, 1000)

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = medv, statistic = boot.fn, R = 1000)


Bootstrap Statistics :
    original      bias    std. error
t1* 22.53281 0.008517589   0.4119374
>
```

Santhoshini sree Bolisetty
Alexa Summers
Gireesh kumar Muppalla
Lab #3-CS 5565

(d)
The confidence interval of t.test() is closer to the confidence interval given by bootstrap

```
t.test(medv)
ci<- c(22.53 - 2 * 0.4119, 22.53 + 2 * 0.4119)
ci

96:1   (Top Level)
```

```
Console ~/
> 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

> ci<- c(22.53 - 2 * 0.4119, 22.53 + 2 * 0.4119)
> ci
[1] 21.7062 23.3538
>
```

(e)

```
med <- median(medv)
med
<
99:1   (Top Level)
```

```
Console ~/
> med <- median(medv)
> med
[1] 21.2
>
```

\

Santhoshini sree Bolisetty
Alexa Summers
Gireesh kumar Muppalla
Lab #3-CS 5565

(f)The median value is 21.2 which is equal to the value obtained in (e), with a standard error of 0.3745

```
boot.fn <- function(data, index) {
  mu <- median(data[index])
  return (mu)
}
boot(medv, boot.fn, 1000)
```

```
106:1   (Top Level)

Console ~/
> boot.fn <- function(data, index) {
+   mu <- median(data[index])
+   return (mu)
+ }
> boot(medv, boot.fn, 1000)

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = medv, statistic = boot.fn, R = 1000)


Bootstrap Statistics :
    original  bias    std. error
t1*     21.2 -0.0025    0.374358
>
```

(g)

```
quantitymue <- quantile(medv, c(0.1))
quantitymue
```

```
109:1   (Top Level)

Console ~/
> quantitymue <- quantile(medv, c(0.1))
> quantitymue
  10%
12.75
>
```

Santhoshini sree Bolisetty
Alexa Summers
Gireesh kumar Muppalla
Lab #3-CS 5565

(h)
The tenth percentile is 12.75 which is again equal to the value obtained in (g), with a standard error of 0.49122

```
boot.fn <- function(data, index) {
  mu <- quantile(data[index], c(0.1))
  return (mu)
}
boot(medv, boot.fn, 1000)

116:1    (Top Level) ÷
```

```
Console ~/
> boot.fn <- function(data, index) {
+   mu <- quantile(data[index], c(0.1))
+   return (mu)
+ }
> boot(medv, boot.fn, 1000)

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = medv, statistic = boot.fn, R = 1000)


Bootstrap Statistics :
    original  bias    std. error
t1*    12.75 0.0261   0.4912231
> 
```