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

#2.b.i
train <- sample(dim(Default)[1], dim(Default)[1] / 2)

#2.b.ii

fit.glm <- glm(default ~ income + balance, data = Default, family =
"binomial", subset = train)
summary(fit.glm)

#2.b.iii
probs <- predict(fit.glm, newdata = Default[-train, ], type = "response")
pred.glm <- rep("No", length(probs))
pred.glm[probs > 0.5] <- "Yes"

#2.b.iv
mean(pred.glm != Default[-train, ]$default)

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

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

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

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

```
pred.glm <- rep("No", length(probs))
probs <- predict(fit.glm, newdata = Default[-train, ], type = "response")
pred.glm[probs > 0.5] <- "Yes"
mean(pred.glm != Default[-train, ]$default)

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


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

#3.c
library(boot)
boot(Default, boot.fn, 1000)

#4.a
library(MASS)
attach(Boston)
mu.hat <- mean(medv)
mu.hat

#4.b
se.hat <- sd(medv) / sqrt(dim(Boston)[1])
se.hat

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

#4.d
t.test(medv)

#4.e
med.hat <- median(medv)
med.hat

#4.f
boot.fn <- function(data, index) {
  mu <- median(data[index])
  return (mu)
}
```

```
boot(medv, boot.fn, 1000)

#4.g
percent.hat <- quantile(medv, c(0.1))
percent.hat

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