Question 6

a) Here is the output of summary of my glm model. Standard errors for coefficients are shown in the Std. Error column

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

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) Done

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

c)

```
Bootstrap Statistics :
         original        bias     std. error
t1* -1.154047e+01  1.194678e-02 4.230500e-01
t2*  2.080898e-05 -4.698170e-07 4.013535e-06
t3*  5.647103e-03 -4.235789e-06 2.313561e-04
```

d) The estimates are the exact same, but the bootstrap has some errors that are slightly smaller (for example, t1 and t2), with the standard error estimate for t3 being slightly larger. This is with a resample N of 100 times

Question 9

a) I used the mean function for the mu_hat

```
> # a)
> mu_hat = mean(Boston$medv)
> print(mu_hat)
[1] 22.53281
```

b) I used the formula from the hint for the standard error calculation

```
> # b)
> std_err_mu_hat = sd(Boston$medv) / sqrt(nrow(Boston))
> print(std_err_mu_hat)
[1] 0.4088611
```

c) By bootstrapping 100 times, we receive an estimate to the standard error that is the same as the standard error

```
> # c)
> boot.fn <- function(data, index) {
+   sd(data[index,]$medv) / sqrt(nrow(data[index,]))
+ }
> std_err_boot = unname(unlist(boot(Boston, statistic = boot.fn, 100)[1]))
> print(std_err_boot)
[1] 0.4088611
```

d) The bootstrap error interval is slightly different than the 95% confidence interval using all of the Boston data. It is slightly wider than the actual one sample t-test

```
        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

> print('------ Bootstrap error --------')
[1] "------ Bootstrap error --------"
> print(mu_hat - 2 * std_err_boot)
[1] 21.71508
> print(mu_hat + 2 * std_err_boot)
[1] 23.35053
```

e) Median function

```
> # e)
> mu_hat_med = median(Boston$medv)
> print(mu_hat_med)
[1] 21.2
```

f) By doing a bootstrap resample, we receive a range of confidence for the median. In this case, with 100 samples, the range does not cover the true median value, which is very interesting

```
> # f)
> boot.fn <- function(data, index) {
+   1.2533 * (sd(data[index,]$medv) / sqrt(nrow(data[index,])))
+ }
> std_err_boot_med = unname(unlist(boot(Boston, statistic = boot.fn, 100)[1]))
> print('------ Bootstrap error median--------')
[1] "------ Bootstrap error median--------"
> print(mu_hat - 2 * std_err_boot_med)
[1] 21.50795
> print(mu_hat + 2 * std_err_boot_med)
[1] 23.55766
```

g & h) The result of the true data's 10% is 12.75. Using bootstrap (with 100 resamples), we get the same result, however the standard error is around .45. This is due to the uncertainty with using random samples of data from the Boston$medv column.

```
[1] "------Quantiles------"
> print(quantile(Boston$medv, probs = c(.1)))
  10%
12.75
> boot.fn <- function(data, index) {
+   quantile(Boston[index,]$medv, probs = c(.1))
+ }
> quan_boot = boot(Boston, statistic = boot.fn, 100)
> print(quan_boot)

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = Boston, statistic = boot.fn, R = 100)


Bootstrap Statistics :
    original  bias    std. error
t1*    12.75  -0.029   0.4467153
```