Problem_Set_2

Sief Salameh

4/16/2023

Load Libraries

```
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 4.1.2
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.4.2     v purrr   1.0.1
## v tibble  3.2.1     v dplyr   1.1.1
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```
## Warning: package 'tibble' was built under R version 4.1.2
```

```
## Warning: package 'tidyr' was built under R version 4.1.2
```

```
## Warning: package 'readr' was built under R version 4.1.2
```

```
## Warning: package 'purrr' was built under R version 4.1.2
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
## Warning: package 'stringr' was built under R version 4.1.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::arrange()   masks plyr::arrange()
## x purrr::compact()   masks plyr::compact()
## x dplyr::count()     masks plyr::count()
## x dplyr::desc()      masks plyr::desc()
## x dplyr::failwith()  masks plyr::failwith()
## x dplyr::filter()    masks stats::filter()
## x dplyr::id()        masks plyr::id()
## x dplyr::lag()       masks stats::lag()
## x dplyr::mutate()    masks plyr::mutate()
## x dplyr::rename()    masks plyr::rename()
## x dplyr::summarise() masks plyr::summarise()
## x dplyr::summarize() masks plyr::summarize()
```

```
library(mgcv)
```

```
## Warning: package 'mgcv' was built under R version 4.1.2
```

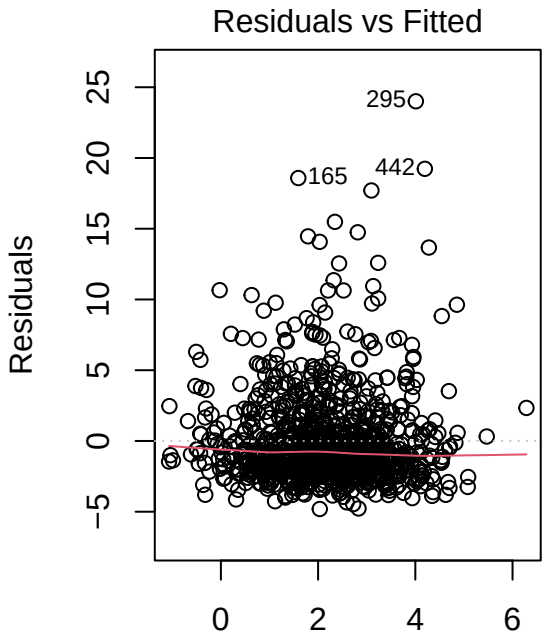
```
## Loading required package: nlme
```

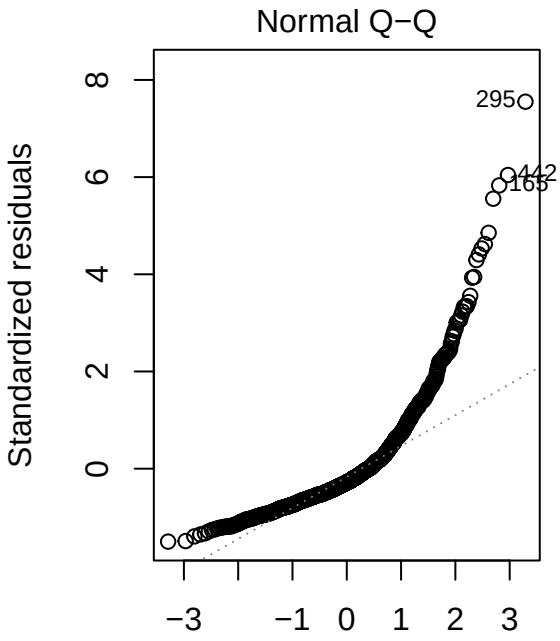
```
## Warning: package 'nlme' was built under R version 4.1.2
##
## Attaching package: 'nlme'
## The following object is masked from 'package:dplyr':
##
##     collapse
## This is mgcv 1.8-39. For overview type 'help("mgcv-package")'.
```

```r
library(wooldridge)

library(boot)
```

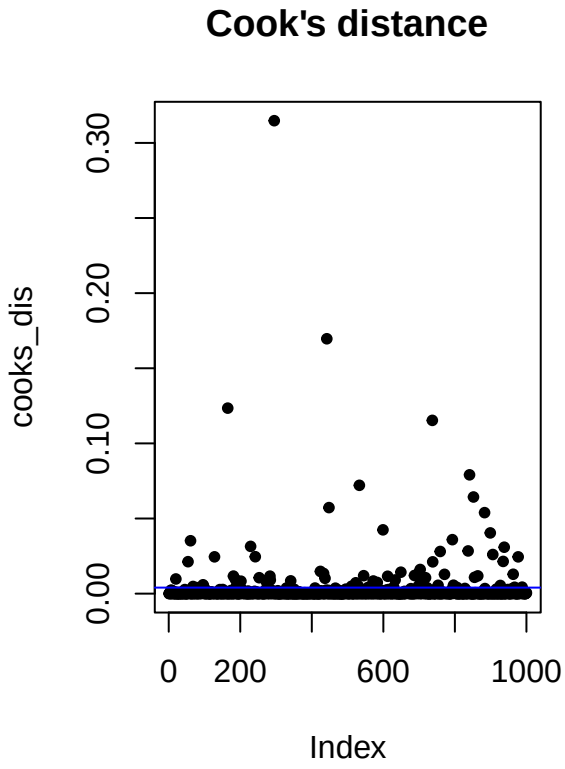
Question One:

```r
set.seed(1)

# Defining the true coefficients and functions

beta_0 <- 0
beta_1 <- 1
beta_2 <- 2
f1 <- function(x) x
f2 <- function(x) x^2

# Simulate the data according to true model

n1 <- 1000
x1 <- rnorm(n1)
x2 <- rnorm(n1)
y1 <- beta_0 + f1(x1) * beta_1 + f2(x2) * beta_2 + rnorm(n1)

# Fitting the true model

true_model <- gam(y1 ~ s(x1) + s(x2))

# Fitting the linear model

linear_model <- lm(y1 ~ x1 + x2)

# Plotting the residual plot and QQ plot for the linear model

par(mfrow = c(1, 2))

plot(linear_model, which = c(1, 2))
```






```r
# Calculating the Cook's distance for each observation

cooks_dis <- cooks.distance(linear_model)

plot(cooks_dis, pch = 20, main = "Cook's distance")

abline(h = 4 / n1, col = "blue")
```



The estimated model is not a good fit because it assumes that the regression is linear. However, based on the plots that I produced, it indicates that the the true model does not exhibit a linear relationship. Since B2 > B1,
the coefficient is likely exponential.

The residual plot that I created displays the standardized residuals vs. the fitted values in the linear regression model. The standardized residuals compute the difference between the observed response value and the predicted response value. The residuals are divided by the estimated standard deviation of the error term. On the other hand, the fitted values are the predicted values of the response variable based on the linear regression model.

Theoretically, we expect to see the following conditions in a well-fitted linear regression model:

1) The residuals should be symmetrically distributed around 0, with no systematic patterns.
2) The residuals should have constant variance across the range of the fitted values (homoscedasticity).
3) The residuals should be independent of the fitted values and the predictor variables.

In the plots I created from the model, we can see that the majority of the
residuals are concentrated around 0, but there is significant distribution above 0. Suggesting that the linear regression model is not well-fitted in terms of the mean of the response variable. Further, the plot shows a curved pattern in the residuals, indicating that the assumption of linearity may not hold true. Lastly, the residuals seem to have increasing variance as the fitted values increase, which suggests that the assumption of homoscedasticity may also be violated. In total, these discrepancies tell us that the linear regression model may not be the best model for representing the true relationship between the predictors and the response variables.

CITATIONS:

https://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/R/R5_Correlation-Regression/R5_Correlation-Regression7.html

https://www.sciencedirect.com/topics/mathematics/regression-diagnostics#:~:text=Regression%20diagnostics%20is%20the

Question Two:

```
data(catholic)
attach(catholic)

female <- as.factor(female)
gender <- revalue(female, c("1" = "female", "0" = "male"))

hsgrad <- as.factor(hsgrad)
hs <- revalue(hsgrad, c("1" = "highschool", "0" = "no_highschool"))
```

Part A:

```
model_a <- lm(read12 ~ gender, data = catholic)

summary(model_a)
```

```
##
## Call:
## lm(formula = read12 ~ gender, data = catholic)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.888  -7.318   1.237   7.660  17.203
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   50.8870     0.1565 325.236  < 2e-16 ***
## genderfemale   1.7114     0.2175   7.867 4.13e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.369 on 7428 degrees of freedom
## Multiple R-squared:  0.008264,   Adjusted R-squared:  0.008131
## F-statistic:  61.9 on 1 and 7428 DF,  p-value: 4.13e-15
```

Interpreting this model, the coefficient intercept tells us that the reading score for individuals who are male, holding all the other variables constant, is equal to 50.8870. This means that if an individual identified as a female, we can predict that their reading score will increase by 1.7114 points (ceteris paribus). This coefficient estimate is statistically significant at the lowest significance/ alpha level of .001. The alpha level represents the probability of rejecting the null hypothesis when the null hypothesis is true (type 1 error). In this case, we can reject the null hypothesis that being female has no influence over reading scores.

However, the multiple R-squared value is 0.008264, which means that only about 0.8% of the variance in reading scores is explained by the female gender alone. This means that the linear relationship between the female gender alone and reading scores is not very strong, and that other factors or covariates not included in this model are likely to have a greater influence on reading scores.

Part B:

```
model_b <- lm(read12 ~ gender - 1, data = catholic)

summary(model_b)
```

```
##
## Call:
## lm(formula = read12 ~ gender - 1, data = catholic)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.888  -7.318   1.237   7.660  17.203
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## gendermale    50.8870     0.1565   325.2   <2e-16 ***
## genderfemale  52.5984     0.1511   348.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.369 on 7428 degrees of freedom
## Multiple R-squared:  0.9683, Adjusted R-squared:  0.9683
## F-statistic: 1.135e+05 on 2 and 7428 DF,  p-value: < 2.2e-16
```

Interpreting this model, the coefficient estimate for males tells us that the reading score for individuals who

are male, holding all the other variables constant is equal to 50.8870. This means that if an individual identified as a male, we can predict that their reading score will be 50.8870. This coefficient estimate is statistically significant at the lowest significance/ alpha level of .001. The alpha level represents the probability of rejecting the null hypothesis when the null hypothesis is true (type 1 error).

Likewise, the coefficient estimate for females tells us that the reading score for individuals who are female, holding all the other variables constant is equal to 52.5984. This means that if an individual identified as a female, we can predict that their reading score will be 52.5984. This coefficient estimate is statistically significant at the lowest significance/ alpha level of .001. In this model, we can reject the null hypothesis that being male or female has no effect on reading scores.

The multiple R-squared value is equal to .9683, which means approximately 97% of the variance in reading scores is explained by the female and male gender covariates. This means that the linear relationship between female and male genders and reading scores is very strong, and that other factors or covariates not included in this model are likely not to have a greater influence on reading scores.

Part C:

```
model_c <- lm(read12 ~ gender*hs, data = catholic)

summary(model_c)
```

```
##
## Call:
## lm(formula = read12 ~ gender * hs, data = catholic)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.217  -7.047   1.142   7.382  22.735
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 44.2852     0.6694  66.160   <2e-16 ***
## genderfemale                 0.2790     0.9022   0.309    0.757
## hshighschool                 6.8576     0.6927   9.899   <2e-16 ***
## genderfemale:hshighschool    1.5049     0.9351   1.609    0.108
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.154 on 5966 degrees of freedom
##   (1460 observations deleted due to missingness)
## Multiple R-squared:  0.05128,    Adjusted R-squared:  0.0508
## F-statistic: 107.5 on 3 and 5966 DF,  p-value: < 2.2e-16
```

Interpreting this model, the coefficient intercept tells us that if an individual identified as a male with no high school diploma, then their predicted reading score will be equal to 44.2852. The coefficient intercept is statistically significant at the lowest significance/ alpha level of .001. The alpha level represents the probability of rejecting the null hypothesis when the null hypothesis is true (type 1 error).

Further, the coefficient estimate for genderfemale tells us that if an individual identified as a female with no high school diploma, then we can predict that their reading score will increase by .2790 points holding all the other variables constant. The coefficient estimate is not statistically significant at any of the significance/ alpha levels.

The coefficient estimate for hshighschool tells us that if an individual identified as a male with a high school

diploma, then we can predict that their reading score will increase by 6.8576 points holding all the other variables constant. This coefficient estimate is statistically significant at the lowest significance/ alpha level of .001.

Lastly, the interaction term between gender and highschool tells us that if an individual identified as a female and had a high school diploma, then we can predict that their reading scores will increase by 1.5049 points. This coefficient estimate is not statistically significant at any of the significance/ alpha levels.

In this model, we can reject the null hypothesis that having a highschool diploma while being male has no effect on reading scores. However, we fail to reject the null hypothesis that being female with no high school diploma has an effect on reading scores. Also we fail to reject the null hypothesis that being female with a highschool diploma has no effect on reading scores.

The multiple R-squared value in this case is equal to 0.0508, which means that only about 5% of the variance in reading scores is explained by the female gender and highschool diploma variables. This means that the linear relationship between the variables - female gender, highschool diploma, and reading scores is not very strong. Other factors or covariates not included in this model are likely to have a greater influence on reading scores.

Part D:

```
model_d <- lm(read12 ~ gender * lfaminc, data = catholic)

summary(model_d)
```

```
##
## Call:
## lm(formula = read12 ~ gender * lfaminc, data = catholic)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.402  -6.759   1.216   7.151  22.112
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            21.6426     2.0789  10.411   <2e-16 ***
## genderfemale           -3.6048     2.7656  -1.303   0.1925
## lfaminc                 2.8149     0.1996  14.104   <2e-16 ***
## genderfemale:lfaminc    0.5339     0.2662   2.006   0.0449 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.036 on 7426 degrees of freedom
## Multiple R-squared:  0.07784,    Adjusted R-squared:  0.07747
## F-statistic:   209 on 3 and 7426 DF,  p-value: < 2.2e-16
```

Interpreting this model, the coefficient intercept tells us that if an individual identified as a male with no family log income, then their predicted reading score will be equal to 21.6426. The coefficient intercept is statistically significant at the lowest significance/ alpha level of .001. The alpha level represents the probability of rejecting the null hypothesis when the null hypothesis is true (type 1 error).

Further, the coefficient estimate for genderfemale tells us that if an individual identified as a female with no family log income, then we can predict that their reading score will decrease by 3.6048 points holding all the other variables constant. The coefficient estimate is not statistically significant at any of the significance/ alpha levels.

The coefficient estimate for lfaminc tells us that if an individual identified as a male, then we can predict that their reading score will increase by 2.8149 points for every unit increase in family log income holding all the other variables constant. This coefficient estimate is statistically significant at the lowest significance/ alpha level of .001.

Lastly, the interaction term between gender and family income tells us that if an individual identified as a female, then we can predict that their reading scores will increase by .5339 points for every unit increase in family log income holding all the other variables constant. This coefficient estimate is statistically significant at the .05 significance/ alpha level.

In this model, we can reject the null hypothesis that increasing family income while being male has no effect on reading scores. However, we fail to reject the null hypothesis that being female with no family income has an effect on reading scores. Further, we can reject the null hypothesis at the 95% confidence interval that being female and increasing family income has no effect on reading scores.

The multiple R-squared value in this case is equal to 0.07747, which means that only about 7% of the variance in reading scores is explained by the female gender and log family income variables. This means that the linear relationship between the variables - female gender, log family income, and reading scores is not very strong. Other factors or covariates not included in this model are likely to have a greater influence on reading scores.

Text Book Questions

Chapter Five Question Two

Part A:

The probability that the first bootstrap observation is not the jth observation from the original sample is 1 - 1/n. Since we are sampling with replacement, the probability of selecting the jth observation from the original sample on a single draw is 1/n. Therefore, the probability of not selecting the jth observation on a single draw is 1 - 1/n.

Each bootstrap observation is randomly drawn with replacement from the original sample. Therefore, the probability of the first bootstrap observation not being the jth observation from the original sample is the same as the probability of any bootstrap observation not being the jth observation from the original sample, which is equal to the probability of selecting any observation other than the jth observation on a single draw. The probability of the first bootstrap observation not being the jth observation from the original sample is the product of the probabilities that each of the n bootstrap observations is not the jth observation from the original sample, which simplifies to (1 - 1/n)^n.

This probability approaches 1/e as n approaches infinity. Therefore, for large n, the probability that the first bootstrap observation is not the jth observation from the original sample is approximately 1 - 1/n.

CITATIONS:

https://machinelearningmastery.com/a-gentle-introduction-to-the-bootstrap-method/#:~:text=The%20bootstrap%20metho

https://towardsdatascience.com/an-introduction-to-the-bootstrap-method-58bcb51b4d60

Part B:

The probability that the second bootstrap observation is not the jth observation from the original sample is also 1 - 1/n. This is because the first and second bootstrap observations are drawn independently from each other, with replacement. Which means it does not matter if the bootstrap observation was the first drawn,

second drawn, or tenth drawn. The probability of selecting the jth observation from the original sample on a single draw is always 1/n. Therefore, the probability of not selecting the jth observation on a single draw will also always be 1 - 1/n. The probability that both observations are not the jth observation from the original sample is the same as the product of their individual probabilities - that each of the n bootstrap observations is not the jth observation from the original sample. Thus, this simplifies to (1 - 1/n)^n.

Part C:

The probability that the jth observation is not in the bootstrap sample is also equal to the probability that none of the n bootstrap observations is the jth observation from the original sample.

As I have shown earlier, the probability that any single bootstrap observation is not the jth observation from the original sample is 1 - 1/n. Therefore, the probability that none of the n bootstrap observations is the jth observation from the original sample is the same as the product of the probabilities that each of the n bootstrap observations is not the jth observation from the original sample. Mathematically, this is shown by the following equation ->

(1 - 1/n) * (1 - 1/n) * … * (1 - 1/n) * (n times)

Simplified, this becomes ->

(1 - 1/n)^n

Part D:

Since we are sampling with replacement, the probability of selecting the jth observation from the original sample on a single draw is -> (1/5). Therefore, the probability of not selecting the jth observation on a single draw is -> (1 - 1/5).

Assuming that we take n = 5 bootstrap samples, the probability that the jth observation is not selected in any of the bootstrap samples is: (1 - 1/5)^5 = 0.32768

Therefore, the probability that the jth observation is in at least one of the bootstrap samples is:

(1 - 0.32768) = 0.67232 or 67.23%

Part E:

Since we are sampling with replacement, the probability of selecting the jth observation from the original sample on a single draw is -> (1/100). Therefore, the probability of not selecting the jth observation on a single draw is -> (1 - 1/100).

Assuming that we take n = 100 bootstrap samples, the probability that the jth observation is not selected in any of the bootstrap samples is: (1 - 1/100)^100 = 0.366

Therefore, the probability that the jth observation is in at least one of the bootstrap samples is:

(1 - 0.366) = 0.634 or 63.4%

Part F:

Since we are sampling with replacement, the probability of selecting the jth observation from the original sample on a single draw is -> (1/10,000). Therefore, the probability of not selecting the jth observation on a single draw is -> (1 - 1/10,000).

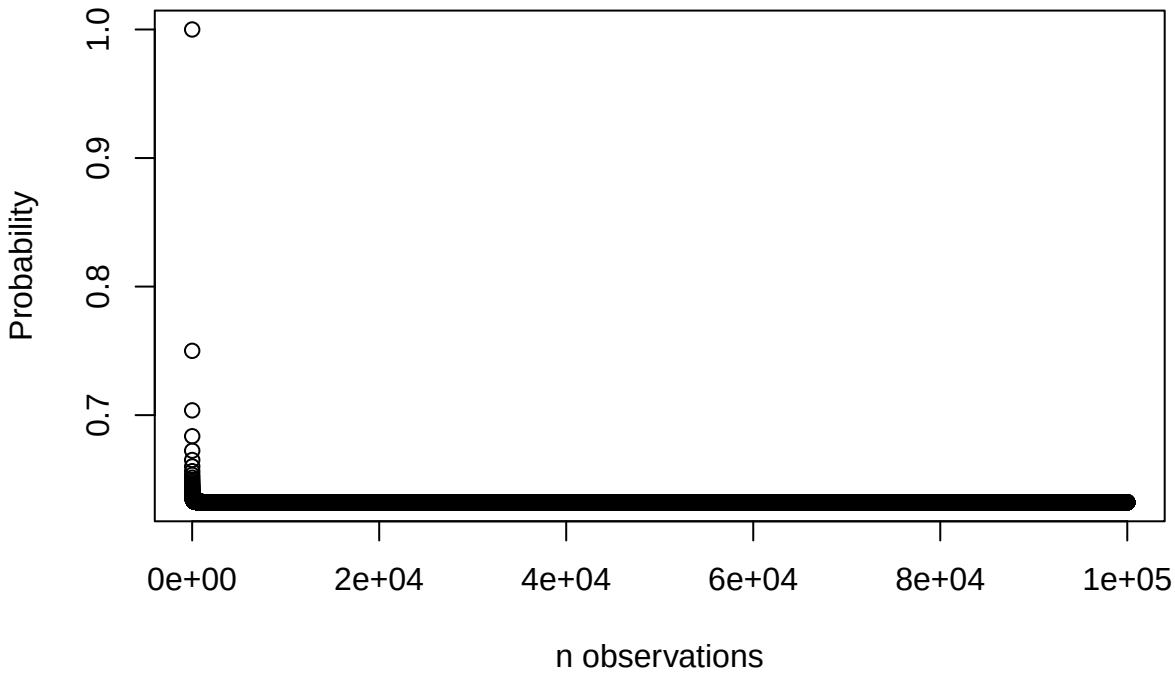
Assuming that we take n = 10,000 bootstrap samples, the probability that the jth observation is not selected in any of the bootstrap samples is: (1 - 1/10,000)^10,000 = 0.3679

Therefore, the probability that the jth observation is in at least one of the bootstrap samples is:

(1 - 0.3679) = 0.6321 or 63.21%

Part G:

```
prob <- numeric(100000)

for (n in 1:100000) {
  prob[n] <- 1 - (1 - 1 / n)^n
}

plot(prob, xlab = "n observations", ylab = "Probability")
```



The plot shows that as we increase n observations, the probability that the jth observation is in the bootstrap sample becomes closer to the mathematical limit we set at 1 - 1/n. In this case, it is equal to 0.6321 or 63.21%. This plot confirms that the probability approaches 1/e as n approaches infinity.

Part H:

```
store <- rep(NA, 10000)
for (i in 1:10000) {
  store[i] <- sum(sample(1:100, rep = TRUE) == 4) > 0
}

mean(store)
```

```
## [1] 0.6402
```

The following code creates 10,000 bootstrap samples that each have n = 100 observations. The code then indicates if the 4th observation is contained in each of those samples. The proportion of samples that do contain the 4th observation is then computed by calculating the mean of the resulting samples.

The output of the code tells us that the probability of the 4th observation being included in a bootstrap sample of size n = 100, is approximately 0.63 or 63.4%. This represents the proportion of bootstrap samples that do contain the 4th observation. Conceptually, there is a significantly high probability that the 4th observation is included in a bootstrap sample size of 100.

Chapter Five Question Three

Part A:

The K-fold cross-validation involves dividing the data into k subsets or folds of roughly the same size, then training the model on k-1 of the folds and evaluating the model on the remaining fold. This process is repeated k times, with each fold used exactly once as the validation data.

The following steps outline how the k-fold cross-validation is implemented -

1) Split the data into k roughly equal-sized folds.
2) For each fold, train the model on the remaining k-1 folds.
3) Evaluate the model on the validation fold and record the evaluation metric through the mean squared error.
4) Repeat steps 2 and 3 for each fold.
5) Calculate the average of the evaluation metric (MSE) across all folds. This will give us an estimate of the model's performance on unseen/ testing data.

CITATION:

https://machinelearningmastery.com/k-fold-cross-validation/

Part Bi:

Advantages and disadvantages of k-fold cross-validation using the validation set approach

Advantages

The K-fold cross-validation is a more reliable estimate of the model's performance because it utilizes multiple validation sets instead of one. The validation approach also allows the entire dataset to be used as a training set, rather than splitting the data into in-sample and out of sample data sets. Lastly, this method provides a way to evaluate the variance of the model's performance across different subsets of the data.

Disadvantages

The validation approach requires more time and programming tools since the model needs to be trained and evaluated k times. Additionally, the variance that is estimated through the validation set approach can potentially be higher.

Part Bii:

Advantages and disadvantages of k-fold cross-validation using the LOOCV

Advantages

The K-fold cross-validation is less time consuming and more resource friendly
through the LOOCV, especially for larger datasets It is also less sensitive to outliers because each fold is trained on a different subset of the data.

Disadvantages

K-fold cross-validation through the LOOCV might underestimate the bias of the model because each fold is trained on a smaller subset of the data rather than the full dataset. Also the choice of k can impact the performance of the model because it might not be clear which value of k to choose from.

Chapter Five Question Eight:

Part A:

```
set.seed(123)

a <- rnorm(100)
b <- a - 2 * a^2 + rnorm(100)
```

In the given data set, n = 100, which represents the total number of observations or data points.

Also in the data set, p = 2, which represents the number of predictor variables used to generate the response variable.

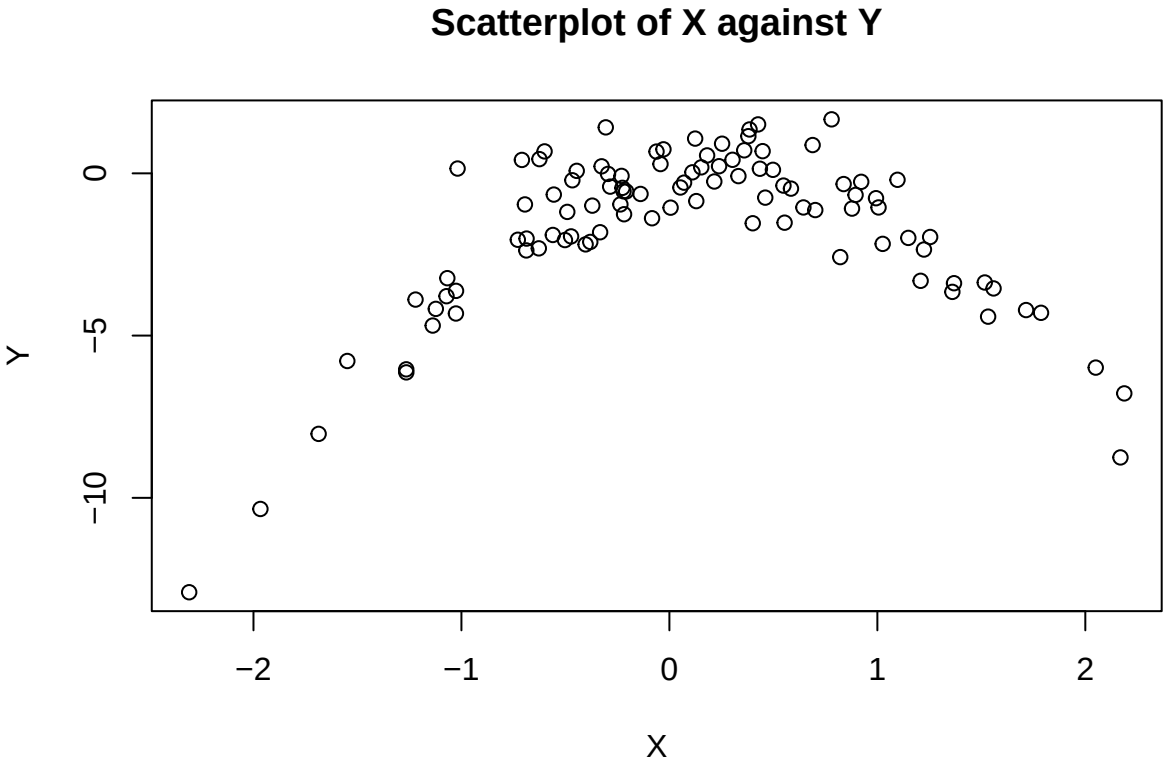
The model used to generate the data in the equation follows this format -

Y = X − 2X^2 +

The second coefficient estimate 2X^2 uses a quadratic term, indicating a nonlinear relationship between x and y.

Part B:

```
plot(a, b, main = "Scatterplot of X against Y", xlab = "X", ylab = "Y")
```



The scatterplot indicates a non-linear relationship between a and b. The relationship is not strictly linear, given that we can observe a curve-like pattern in the scatterplot.

Furthermore, we can also see that the points are spread out and do not form a concentrated cluster around the curve. This suggests that there might be some variability or noise in the relationship between a and b.

Part C:

```
set.seed(123)

df_PartC <- data.frame(a, b)

# Part i -> Y= 0+ 1X+

cv_errors_1 <- glm(b ~ a)
cv.glm(df_PartC, cv_errors_1)$delta[1]
```

```
## [1] 6.975212
```

```
# Part ii -> Y= 0+ 1X+ 2X^2+

cv_errors_2 <- glm(b ~ poly(a, 2))
cv.glm(df_PartC, cv_errors_2)$delta[1]
```

```
## [1] 0.9664678
```

```
# Part iii -> Y= 0+ 1X+ 2X^2+ 3X^3+

cv_errors_3 <- glm(b ~ poly(a, 3))
cv.glm(df_PartC, cv_errors_3)$delta[1]
```

```
## [1] 1.000017
```

```
# Part iv -> Y= 0+ 1X+ 2X^2+ 3X^3+ 4X^4+

cv_errors_4 <- glm(b ~ poly(a, 4))
cv.glm(df_PartC, cv_errors_4)$delta[1]
```

```
## [1] 0.9993215
```

Part D:

```
set.seed(000)

df_PartD <- data.frame(a, b)

# Part i -> Y= 0+ 1X+

cv_errors_1 <- glm(b ~ a)
cv.glm(df_PartD, cv_errors_1)$delta[1]
```

```
## [1] 6.975212
```

```
# Part ii -> Y= 0+ 1X+ 2X^2+

cv_errors_2 <- glm(b ~ poly(a, 2))
cv.glm(df_PartD, cv_errors_2)$delta[1]
```

```
## [1] 0.9664678
```

```
# Part iii -> Y= 0+ 1X+ 2X^2+ 3X^3+

cv_errors_3 <- glm(b ~ poly(a, 3))
cv.glm(df_PartD, cv_errors_3)$delta[1]
```

```
## [1] 1.000017
```

```
# Part iv -> Y= 0+ 1X+ 2X^2+ 3X^3+ 4X^4+

cv_errors_4 <- glm(b ~ poly(a, 4))
cv.glm(df_PartD, cv_errors_4)$delta[1]
```

```
## [1] 0.9993215
```

Yes the results are the same when we use a different seed because the LOOCV method evaluates n folds of a single observation. As the sample size increases, the data set becomes less dependent on the specific values generated by the random seed. Therefore, for a large enough sample size, the LOOCV results should produce relatively identical results across different random seeds.

CITATION:

https://www.statology.org/leave-one-out-cross-validation/

Part E:

The second model in Part C, Y= 0+ 1X+ 2X^2+, had the smallest LOOCV error. This was expected because the relationship between a and b is not a linear relationship, but instead a quadratic relationship. This was demonstrated in the previous plot, which explains why model two had the lowest LOOCV.

Part F:

```
summary(cv_errors_4)
```

```
##
## Call:
## glm(formula = b ~ poly(a, 4))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8630  -0.7018  -0.1688   0.6240   3.4230
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.68329    0.09742 -17.278  < 2e-16 ***
## poly(a, 4)1   4.32330    0.97424   4.438 2.45e-05 ***
## poly(a, 4)2 -23.34713    0.97424 -23.964  < 2e-16 ***
## poly(a, 4)3   0.29713    0.97424   0.305    0.761
## poly(a, 4)4   1.04583    0.97424   1.073    0.286
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.9491413)
##
##     Null deviance: 655.130  on 99  degrees of freedom
## Residual deviance:  90.168  on 95  degrees of freedom
## AIC: 285.44
##
## Number of Fisher Scoring iterations: 2
```

Based on the p-values of the models, it shows that model one with the linear relationship, and model two with the quadratic relationship were statistically significant at the lowest significance/ alpha level of .001. However, the cubic and 4th power models were not statistically significant at any of the significance/ alpha levels. This makes sense because we previously said that model two had the lowest LOOCV error amongst all the models.

Chapter Five Question Nine

Part A:

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.1.2
```

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

```
## The following object is masked from 'package:wooldridge':
##
##     cement
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
attach(Boston)
```

```
## The following object is masked from catholic:
##
##     black
```

```
mu_hat <- mean(medv)

mu_hat
```

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

Part B:

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

std_error
```

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

The standard error tells us the precision of the sample mean as an estimate of the population mean. It represents the average amount of sampling error that can be expected in the sample mean due to random sampling fluctuations. The standard error decreases as the sample size increases, indicating that larger samples provide more precise estimates of the population mean.

Part C:

```
set.seed(111)

boot_std <- function(data, index) {
  mu <- mean(data[index])
  return(mu)
}
boot(medv, boot_std, 506)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = medv, statistic = boot_std, R = 506)
##
##
## Bootstrap Statistics :
##     original      bias    std. error
## t1* 22.53281 -0.03256144   0.4019346
```

Using the same number of observations in the Boston data set, the boot strap estimated standard error is very close to the estimated standard error of the population mean. .4089 vs .4019

Part D:

```
t.test(Boston$medv)
```

```
##
##  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
```

```
CI_mu_estimate <- c(22.53 - 2 * 0.4019, 22.53 + 2 * 0.4019)

CI_mu_estimate
```

```
## [1] 21.7262 23.3338
```

The mean of medv from the t.test(Boston$medv) equals 22.53. This falls within the range computed by the 95% confidence interval using the bootstrap estimate from Part C - which equals [21.72 - 23.33].

Part E:

```
median_estimate <- median(medv)

median_estimate
```

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

Part F:

```
boot_median <- function(data, index) {
  mu <- median(data[index])
  return(mu)
}

boot(medv, boot_median, 506)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = medv, statistic = boot_median, R = 506)
##
```

```
##
## Bootstrap Statistics :
##     original      bias    std. error
## t1*     21.2 -0.01422925   0.3814058
```

The estimate of the median using the bootstrap method is equal to 21.2. This is the same value as the median computed in Part E. Likewise, the bootstrap method gives us a standard error estimate equal to .3814. The standard error is small compared to the estimate value of 21.2.

Part G:

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

tenth_percentile
```

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

Part H:

```
boot_tenth <- function(data, index) {
  mu <- quantile(data[index], c(0.1))
  return(mu)
}

boot(medv, boot_tenth, 506)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = medv, statistic = boot_tenth, R = 506)
##
##
## Bootstrap Statistics :
##     original      bias    std. error
## t1*    12.75 -0.01956522   0.5115426
```

Using the bootstrap method, we get an estimated value of 12.75 which is the same value we computed in Part G. Also using the bootstrap method to estimate the standard error of the tenth percentile, we get a value equal to .4939. The standard error is small compared to the estimated value of 12.75.

END