Assignment 2

Aly Abdelwahed, Manish Suresh

18/02/2021

Question 1

Part (a)

We are given a value for σ and we know that $\frac{\hat{\beta}_1 - \beta_1}{\frac{\sigma}{\sqrt{S_{XX}}}} \sim N(0,1)$

As such,

$$P(|\hat{\beta}_1 - \beta_1| > 1)$$

$$\therefore P(\left|\frac{\hat{\beta}_1 - \beta_1}{\frac{\sigma}{\sqrt{S_{XX}}}}\right| > \frac{\sqrt{S_{XX}}}{\sigma})$$

$$= P(|Z| > \frac{\sqrt{S_{XX}}}{\sigma})$$

$$= P(Z > \frac{\sqrt{S_{XX}}}{\sigma}) + P(Z < -\frac{\sqrt{S_{XX}}}{\sigma})$$

Therefore, the value of the probability above is:

```
X <- data.frame(X= c(4, 8, 12, 16, 20))
S_XX <- sum((X$X - mean(X$X))^2)
sigma = 5
pnorm(sqrt(S_XX)/sigma, mean = 0,sd = 1, lower.tail = FALSE) + pnorm(-sqrt(S_XX)/sigma,
mean = 0,sd = 1)
```

```
## [1] 0.01141204
```

$$\therefore P(Z > \frac{\sqrt{S_{XX}}}{\sigma}) + P(Z < -\frac{\sqrt{S_{XX}}}{\sigma}) = P(Z > \frac{\sqrt{160}}{5}) + P(Z < -\frac{\sqrt{160}}{5})$$

$$= P(Z > 2.5298221) + P(Z < -2.5298221)$$

$$= 0.005706 + 0.005706$$

$$= 0.011412$$

Part (b)

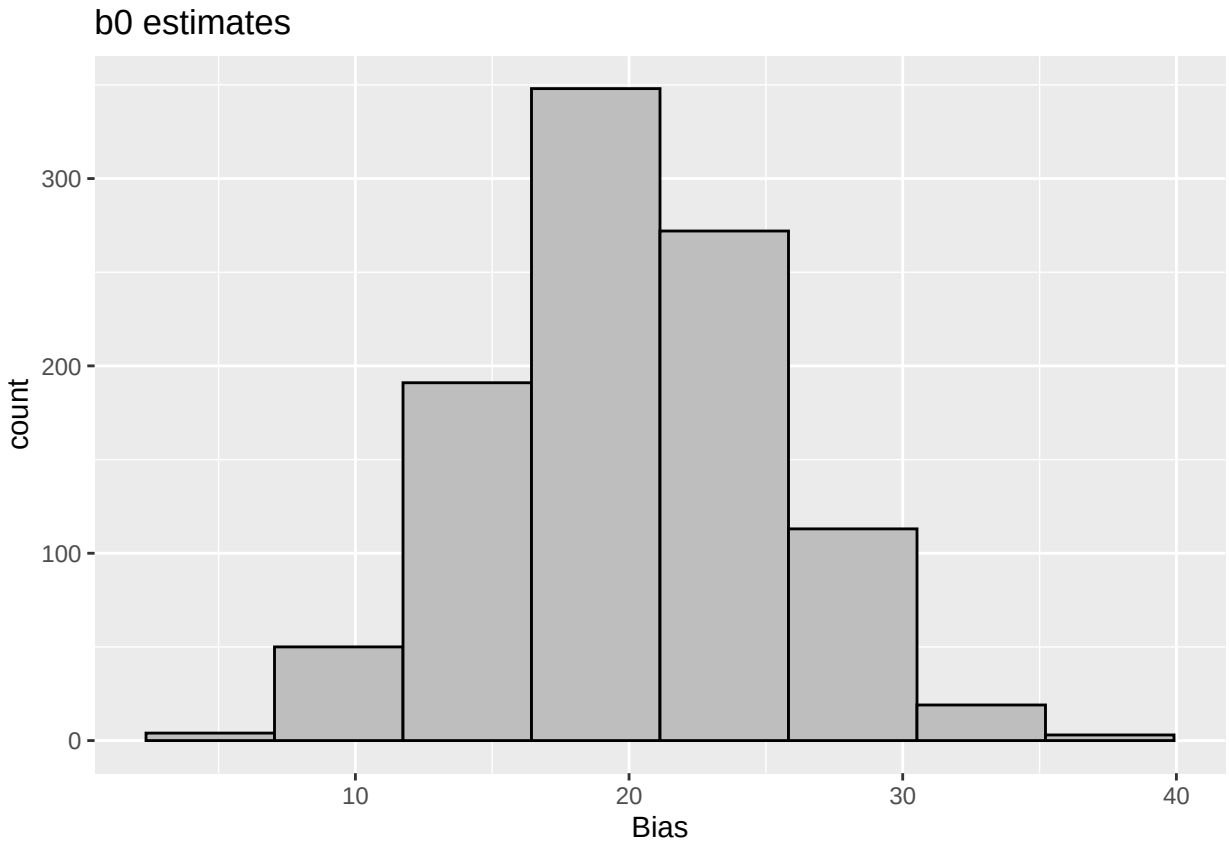
```
# Setting the seed
set.seed(1004269365)

# Generating the random errors
true.errors <- rnorm(5,0,5)

# Storing the values for X
X = c(4,8,12,16,20)

# Calculating the valus for Y
Y = 20 + 4 * X + true.errors

# Performing the linear regression
regression <- lm(Y ~ X)

# Getting the least square estimates
beta0.hat <- regression$coefficients[1]
beta1.hat <- regression$coefficients[2]

# Predicting the value for X = 10
# Y0.hat = beta0.hat + beta1.hat * 10
new.data <- data.frame(X = 10)
prediction <- predict(regression, new.data, interval = "confidence")

Y0.hat <- prediction[1]
interval <- c(prediction[2], prediction[3])
```

The values of Y_1, Y_2, Y_3, Y_4, Y_5 are 28.8, 58.1, 67.7, 85.5, 89.9
The least Square estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ are 21.11 and 3.74 respectively.
The calculated value of $\hat{Y_0}$ when $X_0 = 10$ is 58.51
The 95% confidence interval for $\mathbb{E}\left(\hat{Y_0}\right)$ is $(47.39, 69.63)$

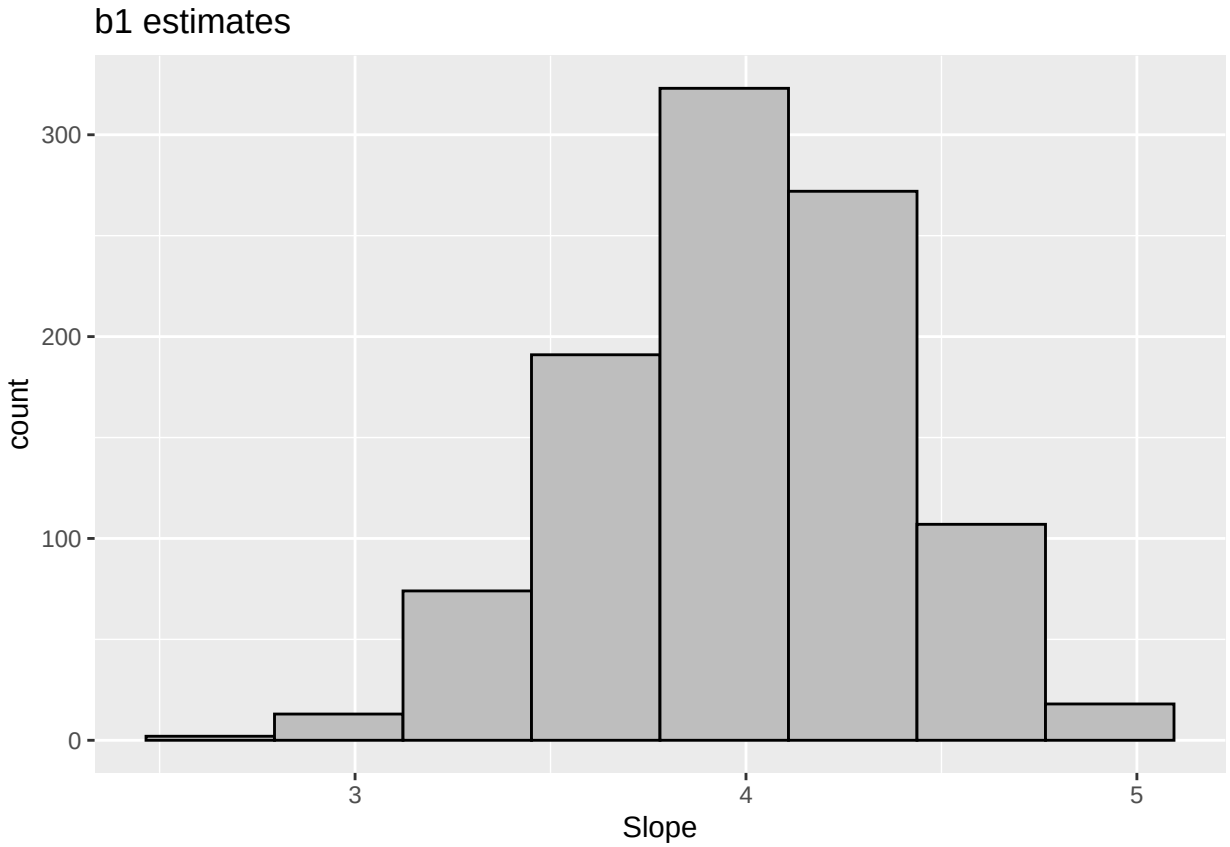
Part (c)

```
# Setting the seed
set.seed(1004269365)

# Stroing the values for X
X = c(4,8,12,16,20)

# Rerun 1000 times
raw.estimates <- rerun(1000, rnorm(5,0,5)) %>%
  map(~ . + 4 * X + 20) %>%
  map(~ lm(. ~ X))

estimates <- raw.estimates %>%
  map_dfr(~ c(.$coefficients[1], .$coefficients[2]))
```

```
names(estimates) <- c("Bias","Slope")

# Plotting a histogram of the least square estimates
ggplot(estimates, aes(x=Bias)) + geom_histogram(bins = 8,
color = "black", fill="grey") + ggtitle("b0 estimates")
```




```
ggplot(estimates, aes(x=Slope)) + geom_histogram(bins = 8,
color = "black", fill="grey") + ggtitle("b1 estimates")
```




```
mean.b0 <- mean(estimates$Bias)
mean.b1 <- mean(estimates$Slope)
sd.b0 <- sd(estimates$Bias)
sd.b1 <- sd(estimates$Slope)
```

The theoretical expectations for the least square estimates as follows

$$\hat{\beta}_0 \sim N\left(\beta_0,\ \sigma^2\left(\frac{1}{n}+\frac{\bar{X}^2}{S_{XX}}\right)\right)$$

and

$$\hat{\beta}_1 \sim N\left(\beta_1,\ \frac{\sigma^2}{S_{XX}}\right)$$

So the theoretical mean and standard deviation of the least square estimates are

	Mean	Standard Deviation
$\hat{\beta}_0$	20	5.2440442
$\hat{\beta}_1$	4	0.3952847

So the analytical mean and standard deviation of the least square estimates are

	Mean	Standard Deviation
$\hat{\beta}_0$	20.0964762	5.1363669
$\hat{\beta}_1$	3.9915116	0.3847291

The theoretical and analytical estimates are consistent.

Part (d)

```
# Retreieve all the information about the prediction data on the 1000 regression data
prediction.list <- raw.estimates %>%
  map(~ predict(., data.frame(X = 10), interval = "confidence"))

# Determine the confidence interval information from the prediction data
confidence.intervals <- do.call(rbind.data.frame, prediction.list)

# Determine the Expected value of Y
E.Y0 = 20 + 4 * 10

# Filter the correct ammount of rows
temp.data <- confidence.intervals[confidence.intervals$lwr < E.Y0, ]
temp.data <- temp.data[temp.data$upr > E.Y0,]

# Display the number of rows
nrow(temp.data)
```

```
## [1] 952
```

AS you can see from the number of rows above, at least 952 rows contain the $\mathbb{E}\left(\hat{Y}_0\right)$ which is inline with fact that at least 95% of the intervals contain the $\mathbb{E}\left(\hat{Y}_0\right)$.

Question 2

Part (a)

```r
# Perform the regression on the dataset
NHL.regression <- lm(Weight ~ Height, dataset)

NHL.x0 =  74

# Store the coefficients
NHL.b0 <- NHL.regression$coefficients[1]
NHL.b1 <- NHL.regression$coefficients[2]

# Store the fitted value for a new observation
NHL.Y0.hat <-  NHL.b0 + NHL.b1 * NHL.x0

# Calculate the sigma hat
NHL.sigma.hat <- sqrt(sum(NHL.regression$residuals^2) / (nrow(dataset) - 2))

# Calculate S_XX
NHL.S_XX = (nrow(dataset) - 1) * sd(dataset$Height)^2

# Calculate X_Bar
NHL.X.Bar = mean(dataset$Height)

# Calculate the Standard error of fitted values
NHL.SE.Y0.hat <- NHL.sigma.hat * sqrt(1/nrow(dataset) + (74 - NHL.X.Bar)^2/NHL.S_XX)

# Calculate the confidence interval
NHL.Confidence.interval <- c(NHL.Y0.hat - qt(0.975, nrow(dataset) - 2) *
NHL.SE.Y0.hat, NHL.Y0.hat + qt(0.975, nrow(dataset) - 2) * NHL.SE.Y0.hat)
```

Constructing a confidence interval via hands

To find the confidence interval for the mean weight of all players with $X_0 = 74$, we use the formula

$$\hat{Y_0} \pm t_{\{1-\frac{\alpha}{2};n-2\}} \mathbb{SE}\left(\hat{Y_0}\right)$$

First we calculate the fitted value for the new observation $X_0 = 74$

$$\hat{Y_0} = \hat{\beta_0} + \hat{\beta_1} X_0$$

$$= -163.3483089 + 4.9926656 \times 74$$

$$= 206.1089439$$

$$= 206.11$$

Next we calculate the standard error of $\hat{Y_0}$

$$\mathbb{SE}\left(\hat{Y_0}\right) = \hat{\sigma}\sqrt{\frac{1}{n} + \frac{\left(X_0 - \bar{X}\right)^2}{S_{XX}}}$$

$$= 11.0830595\sqrt{\frac{1}{717} + \frac{(74 - 73.2608089)^2}{3052.2287308}}$$
$$= 0.4396662$$
$$= 0.44$$

Finally we can calculate the confidence interval

$$C_n = \hat{Y}_0 \pm t_{\{1-\frac{\alpha}{2};n-2\}}\mathbb{SE}\left(\hat{Y}_0\right)$$
$$= 206.1089439 \pm 3.1824463 \times 0.4396662$$
$$= (205.2457527, 206.972135)$$
$$= (205.25, 206.97)$$

Constructing a confidence interval via built-R function

```
prediction <- predict(NHL.regression, data.frame(Height=74), interval = "confidence")
R.NHL.confidence.interval <- c(prediction[2], prediction[3])
```

	Confidence Interval
Computed by Hand	(205.25, 206.97)
R's Built in function	(205.25, 206.97)

As you can see they are same

Part (b)

```
# Calculate the standard deviation of the prediction
NHL.SE.pred <- NHL.sigma.hat * sqrt(1 + 1/nrow(dataset) + (74 - NHL.X.Bar)^2/NHL.S_XX)

# Calculate the confidence interval
NHL.Prediction.interval <- c(NHL.Y0.hat - qt(0.975, nrow(dataset) - 2) * NHL.SE.pred,
NHL.Y0.hat + qt(0.975, nrow(dataset) - 2) * NHL.SE.pred)
```

Constructing a prediction interval via hands

To find the prediction interval for the mean weight of all players with $X_0 = 74$, we use the formula

$$\hat{Y}_0 \pm t_{\{1-\frac{\alpha}{2};n-2\}}\mathbb{SE}\left(\{pred\}\right)$$

First we calculate the fitted value for the new observation $X_0 = 74$

$$\hat{Y}_0 = \hat{\beta}_0 + \hat{\beta}_1 X_0$$
$$= -163.3483089 + 4.9926656 \times 74$$
$$= 206.1089439$$
$$= 206.11$$

Next we calculate the standard error of the new prediction

$$\mathbb{SE}\left(\{pred\}\right) = \hat{\sigma}\sqrt{1 + \frac{1}{n} + \frac{\left(X_0 - \bar{X}\right)^2}{S_{XX}}}$$

$$= 11.0830595\sqrt{1 + \frac{1}{717} + \frac{(74 - 73.2608089)^2}{3052.2287308}}$$

$$= 11.0917768$$

$$= 11.09$$

Finally we can calculate the prediction interval

$$C_n = \hat{Y}_0 \pm t_{\{1-\frac{\alpha}{2};n-2\}}\mathbb{SE}\left(\{pred\}\right)$$

$$= 206.1089439 \pm 3.1824463 * 11.0917768$$

$$= (184.3325985, 227.8852893)$$

$$= (184.33, 227.89)$$

Constructing a prediction interval via built-R function

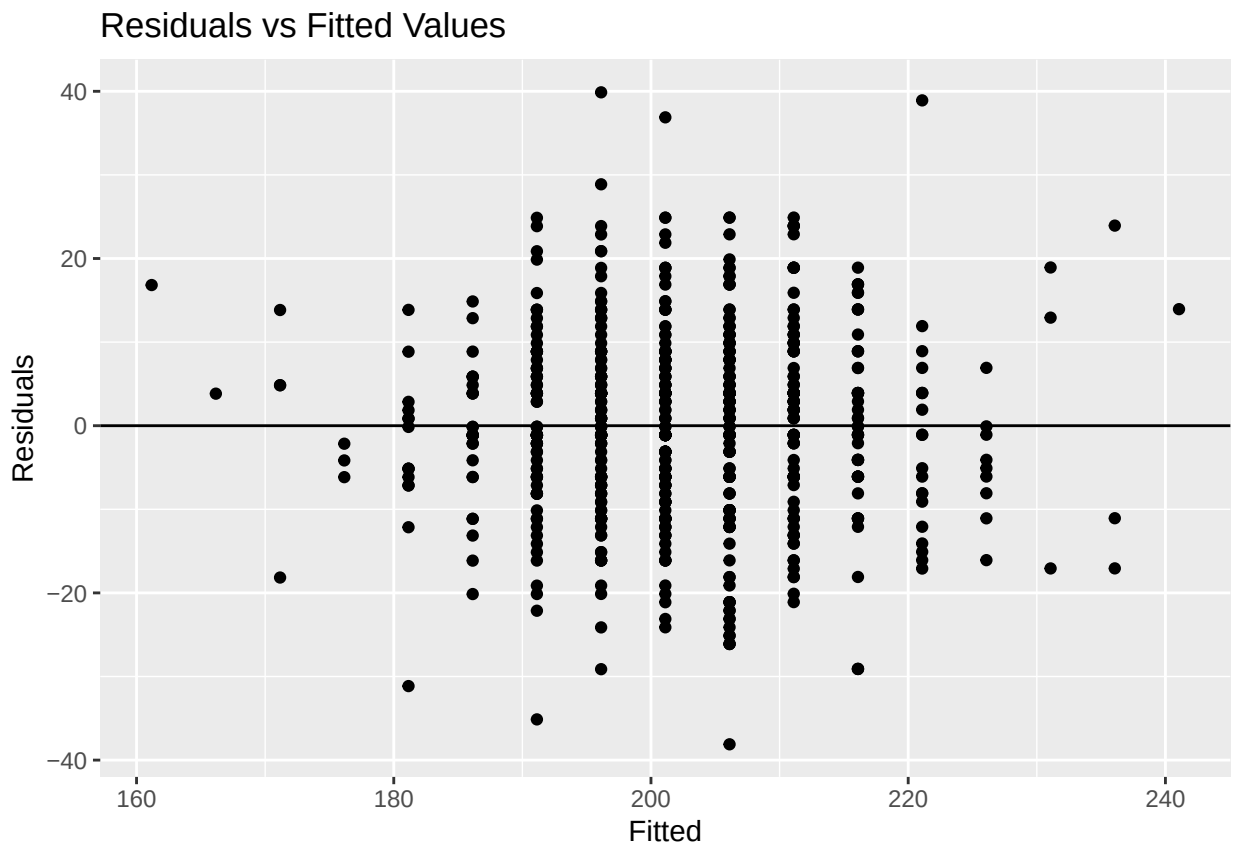
```
prediction <- predict(NHL.regression, data.frame(Height=74), interval = "prediction")
R.NHL.prediction.interval <- c(prediction[2], prediction[3])
```

	Prediction Interval
Computed by Hand	(184.33, 227.89)
R's Built in function	(184.33, 227.89)

As you can see they are same

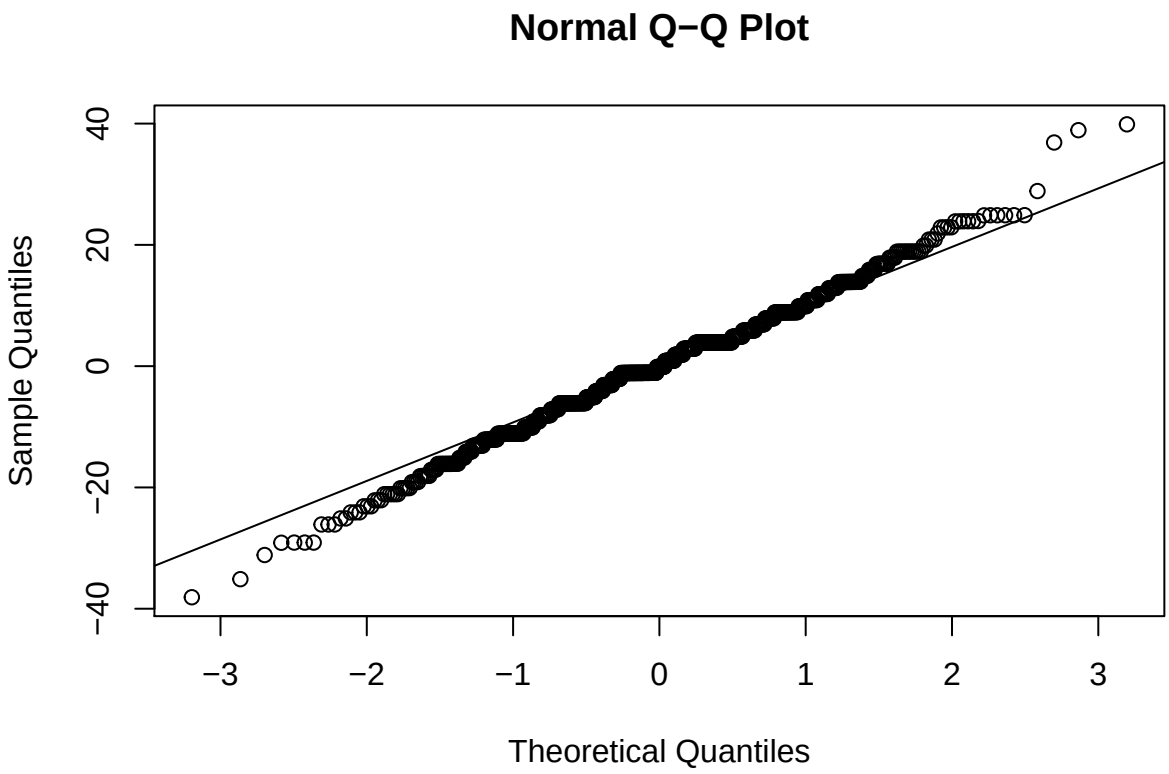
Part (c)

```
# Store the residuals of the dataset
Residuals <- resid(NHL.regression)


# store the fitted values of the regression
Fitted <- fitted.values(NHL.regression)

# Combine them into data frame
graph.data <- cbind.data.frame(Residuals, Fitted)

# PLot the graph
ggplot(graph.data, aes(x=Fitted, y=Residuals)) + geom_point() +
ggtitle("Residuals vs Fitted Values") + geom_hline(yintercept  = 0)
```



Part (d)

```
# Plot the graph to test the normality
qqnorm(Residuals)
qqline(Residuals)
```




```
# Perform the Shapiro test
shapiro.test(Residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Residuals
## W = 0.99407, p-value = 0.006511
```

As we can see by the shapiro-wilk test, the p-value is < 0.05 which means that we reject the null hypothesis H_0 and that the errors are not normally distributed

Part (e)

```
df <- data.frame(resid = c(NHL.regression$residuals), Height = c(dataset$Height))
medianX <- median(dataset$Height)
greaterResid <- subset(df, dataset$Height>medianX)$resid
lowerResid <- subset(df, dataset$Height<=medianX)$resid

D1 <- data.frame(absoluteDeviation1 = abs(greaterResid-median(greaterResid)))
D2 <- data.frame(absoluteDeviation2 = abs(lowerResid-median(lowerResid)))

s1_2 <- var(D1$absoluteDeviation1)
```

```
s2_2 <- var(D2$absoluteDeviation2)

pooledVariance = ((nrow(D1) - 1)*s1_2+(nrow(D2) - 1)*s2_2)/(nrow(dataset)-2)

t_BF = (mean(D1$absoluteDeviation1) - mean(D2$absoluteDeviation2))/(sqrt( pooledVariance*
(1/nrow(D1) + 1/nrow(D2)) ))
t_BF
```

```
## [1] 2.00784
```

```
alpha = 0.05
(abs(t_BF) >= qt(1-alpha/2, nrow(dataset)-2))
```

```
## [1] TRUE
```

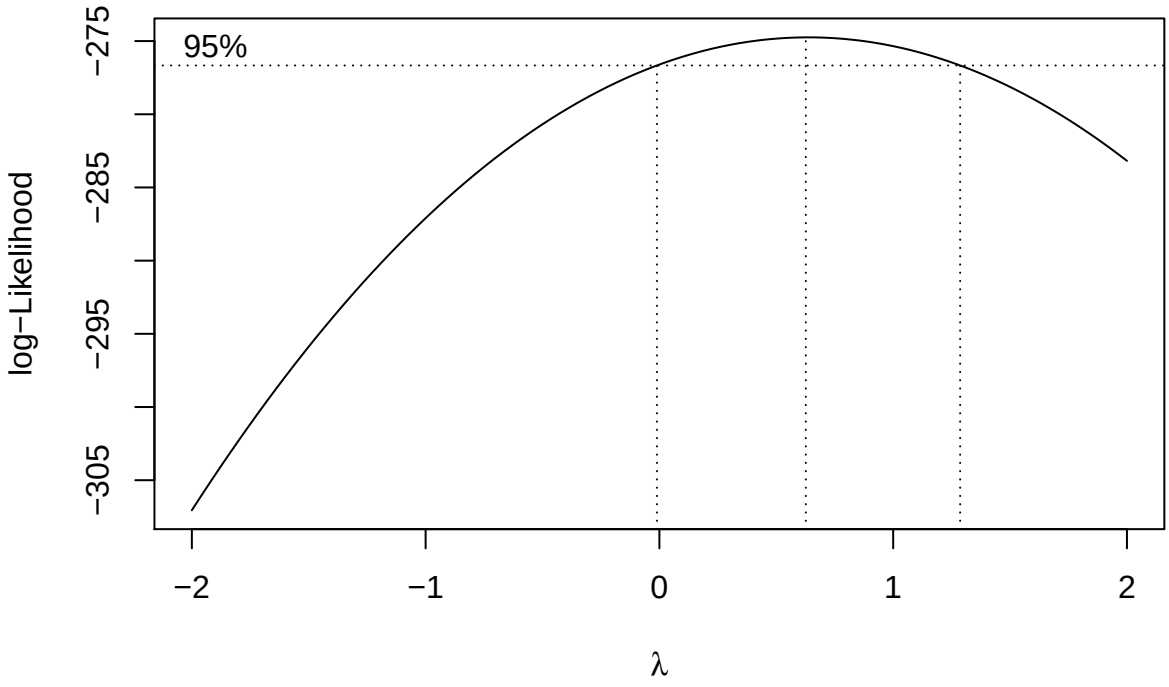
As such, since $|t_{BF}| \geq t_{\{1-\frac{0.05}{2};715\}}$, we reject H_0 and the error variance does not vary with the level of X

Part (f)

```
library(MASS)
```

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

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

```
result <- boxcox(NHL.regression)
```

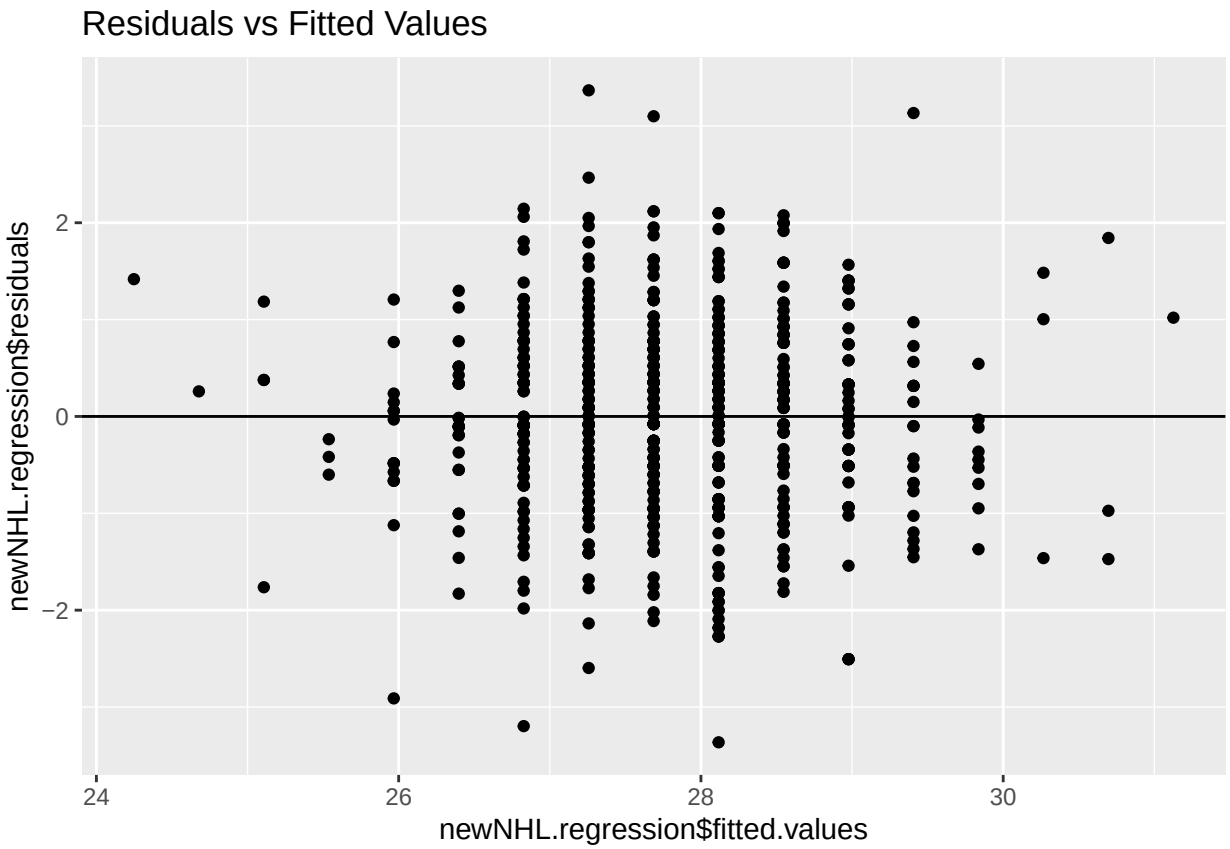



```
mylambda <- result$x[which.max(result$y)]
mylambda
```

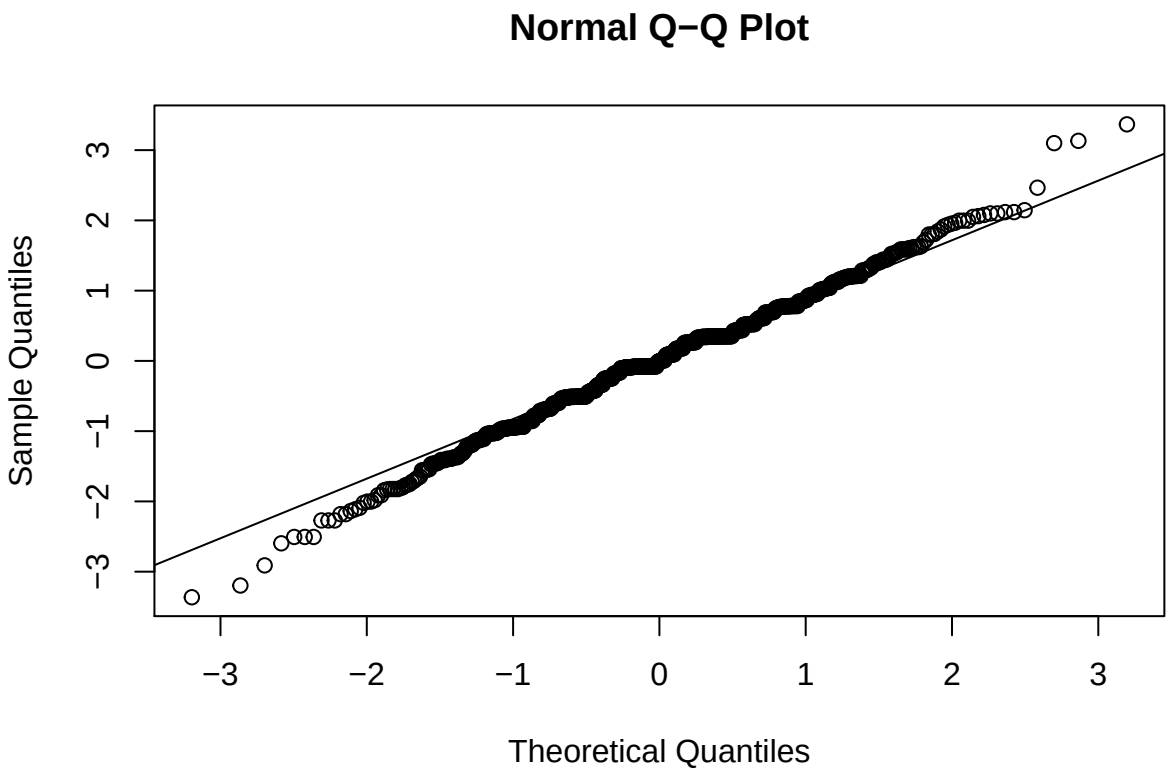
```
## [1] 0.6262626
```

Now, to check the normality of the errors

```
newNHL.regression <- lm(Weight^mylambda ~ Height, dataset)


ggplot(graph.data, aes(x=newNHL.regression$fitted.values, y=newNHL.regression$residuals)) +
geom_point() + ggtitle("Residuals vs Fitted Values") + geom_hline(yintercept  = 0)
```




```
qqnorm(newNHL.regression$residuals)
qqline(newNHL.regression$residuals)
```




```
# Perform the Shapiro test
shapiro.test(newNHL.regression$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  newNHL.regression$residuals
## W = 0.99432, p-value = 0.008735
```

We can see that the fitted values vs the residuals plots before and after the box-cox transformation are almost identical in terms of the spread of the residuals which means the box-cox transformation has not made much of an impact on the spread of the residuals. However, we can see from the QQ-plot, the points on the plot are lying more on a straight line than they were prior to the box-cox transformation which means that the normality of the errors has been improved. That can also be seen from the greater p-value of the shapiro-wilk test as $0.008735 > 0.006511$. However, the new p-value is still less than 0.05 which means that we still reject the null hypothesis H_0 and that the errors are still not normally distributed

Now, to check if the whether or not the error variance varies with the level of X

```
df <- data.frame(resid = c(newNHL.regression$residuals), Height = c(dataset$Height))
medianX <- median(dataset$Height)
greaterResid <- subset(df, dataset$Height>medianX)$resid
```

```
lowerResid <- subset(df, dataset$Height<=medianX)$resid

D1 <- data.frame(absoluteDeviation1 = abs(greaterResid-median(greaterResid)))
D2 <- data.frame(absoluteDeviation2 = abs(lowerResid-median(lowerResid)))

s1_2 <- var(D1$absoluteDeviation1)
s2_2 <- var(D2$absoluteDeviation2)

pooledVariance = ((nrow(D1) - 1)*s1_2+(nrow(D2) - 1)*s2_2)/(nrow(dataset)-2)

t_BF = (mean(D1$absoluteDeviation1) - mean(D2$absoluteDeviation2))/(sqrt( pooledVariance*
(1/nrow(D1) + 1/nrow(D2)) ))
t_BF
```

```
## [1] 1.582621
```

```
alpha <- 0.05
(abs(t_BF) >= qt(1-alpha/2, nrow(dataset)-2))
```

```
## [1] FALSE
```

As such, after the box-cox transformation, since $|t_{BF}| < t_{\{1-\frac{0.05}{2};715\}}$, we fail to reject the null hypothesis H_0 and as such, we fail to reject the hypothesis that the error variance varies with the level of X

Question 3

Part (a)

```
library(matlib)

X <- matrix(c(1,-9,3,1,-7,-7,1,-5,7,1,-3,-9,1,-1,1,1,1,5,1,3,-1,1,5,-3,1,7,-5,1,9,9),
nrow = 10,byrow=T)

#Method #1: Following Slide 16 on the scanned lecture 8
X.T <- t(X)

XT_X.inv<- inv(X.T %*% X)

Y = matrix(c(34,16,26,35,32,11,24,1,-3,15), nrow = 10,byrow=T)

B_hat <- XT_X.inv %*% X.T %*% Y
B_hat
```

```
##            [,1]
## [1,] 19.1000000
## [2,] -1.5215013
## [3,]  0.4151202
```

```
#Method #2: Using the R linear regression model
fit <- lm(Y~X[,2]+X[,3])
fit
```

```
##
## Call:
## lm(formula = Y ~ X[, 2] + X[, 3])
##
## Coefficients:
## (Intercept)       X[, 2]       X[, 3]
##     19.1000      -1.5215       0.4151
```

As we can see from both methods, the least squares estimate of the model with the given information:

$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon = 19.1 \text{ -}1.5215013 X_1 + 0.4151184 X_2$

Part (b)

```
#Following Slide 17 on the scanned lecture 8
Y_hat <- X %*% B_hat
resid <- Y - Y_hat
resid.T <- t(resid)
errorVarianceSquare <- 1/(nrow(X)-2)*(resid.T %*% resid)
errorVarianceSquare
```

```
##           [,1]
## [1,] 107.2752
```

As we can see, an unbiased estimate of the error variance σ^2 is: 107.2752397

Question 4

Part (a)

Part 1: Design matrix X

```
X <- matrix(c(1,3,1,5,1,4,1,6,1,7), nrow = 5,byrow=T)
X
```

```
##      [,1] [,2]
## [1,]    1    3
## [2,]    1    5
## [3,]    1    4
## [4,]    1    6
## [5,]    1    7
```

Note: In the following questions, I tried to do the tilde symbol under the $\hat{\beta}$ character but I couldn't do it in RStudio. It wasn't working for some reason. Please do not take off marks for that

Part 2: vector of estimated regression coeffcients

```
library(matlib)
Y = matrix(c(4, 6.5, 5, 7, 7.5), nrow = 5,byrow=T)

#Method #1: Following Slide 16 on the scanned lecture 8
X.T = t(X)

XT_X.inv<- inv(X.T %*% X)
B_hat <- XT_X.inv %*% X.T %*% Y
B_hat
```

```
##      [,1]
## [1,]  1.5
## [2,]  0.9
```

```
#Method #2: Using the R linear regression model
fit <- lm(Y~X[,2])
fit
```

```
##
## Call:
## lm(formula = Y ~ X[, 2])
##
## Coefficients:
## (Intercept)       X[, 2]
##         1.5          0.9
```

As such, the vector of estimated regression coefficients is:

$$\hat{\beta} = \begin{bmatrix} 1.5 \\ 0.9 \end{bmatrix}$$

Part 3: variance-covariance matrix

```
#Using slide 18 on lecture 8
varianceCovaraince <- 4*XT_X.inv
varianceCovaraince
```

```
##      [,1] [,2]
## [1,] 10.8 -2.0
## [2,] -2.0  0.4
```

As we can see, the variance-covariance matrix of $\hat{\beta}$ is:

$$4 \times \begin{bmatrix} 2.7 & -0.5 \\ -0.5 & 0.1 \end{bmatrix} = \begin{bmatrix} 10.8 & -2 \\ -2 & 0.4 \end{bmatrix}$$

Part (b)

Therefore, from the matrix above:

$$Cov\{\hat{\beta}_0, \hat{\beta}_1\} = -2$$

$$Var\{\hat{\beta}_0\} = 10.8$$

$$Var\{\hat{\beta}_1\} = 0.4$$

Part (c)

```
H = X %*% XT_X.inv %*% X.T
H
```

```
##               [,1] [,2]         [,3]         [,4]          [,5]
## [1,]  6.000000e-01  0.2 4.000000e-01 4.440892e-16 -2.000000e-01
## [2,]  2.000000e-01  0.2 2.000000e-01 2.000000e-01  2.000000e-01
## [3,]  4.000000e-01  0.2 3.000000e-01 1.000000e-01  3.330669e-16
## [4,]  4.440892e-16  0.2 1.000000e-01 3.000000e-01  4.000000e-01
## [5,] -2.000000e-01  0.2 4.440892e-16 4.000000e-01  6.000000e-01
```

Part (d)

```
varianceErrors = 4*(diag(5) - H)
varianceErrors
```

```
##               [,1] [,2]          [,3]          [,4]          [,5]
## [1,]  1.600000e+00 -0.8 -1.600000e+00 -1.776357e-15  8.000000e-01
## [2,] -8.000000e-01  3.2 -8.000000e-01 -8.000000e-01 -8.000000e-01
## [3,] -1.600000e+00 -0.8  2.800000e+00 -4.000000e-01 -1.332268e-15
## [4,] -1.776357e-15 -0.8 -4.000000e-01  2.800000e+00 -1.600000e+00
## [5,]  8.000000e-01 -0.8 -1.776357e-15 -1.600000e+00  1.600000e+00
```