HW 3

Jared Brotamonte

9/22/2023

Exercise 1

```r
library(ISLR2)
```

```
## Warning: package 'ISLR2' was built under R version 4.1.3
```

```r
data("Auto")
library(ggplot2)
```

(a)

```r
# Perform simple linear regression
model <- lm(mpg ~ horsepower, data = Auto)

# Print the summary of the regression results
summary(model)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

i. Yes, there is a relationship between the predictor "horsepower" and the response "mpg" given the p-value is "<2e-16"

ii. The multiple R-squared value is 0.6059 thus this means that about 60.59% of the variance in "mpg" can be explained by the "horsepower" predictor. This indicates a moderate to strong relationship between the two.

iii. The coefficient for "horsepower" is -0.157845. Since this is less than zero, it indicates a negative relationship between "horsepower" and "mpg" or in otherwords as "horsepower" increases by 1, "mpg" should decrease by about -0.157845.

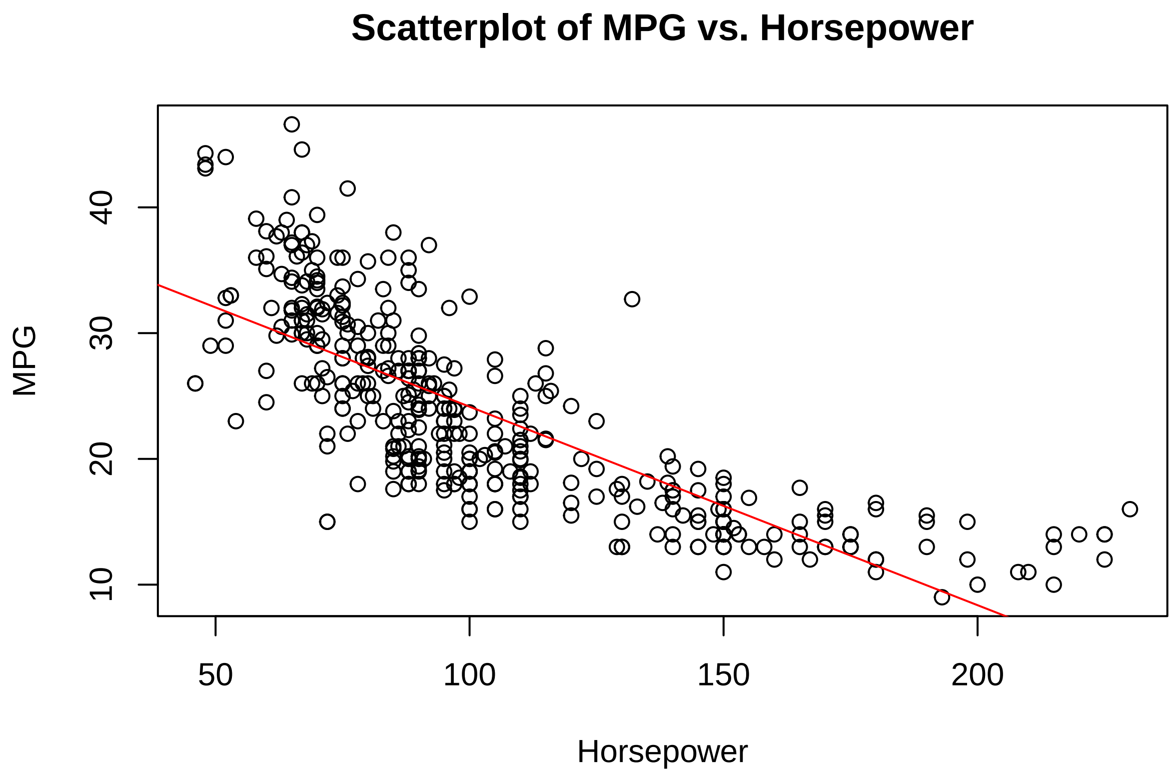
(b)

```
new_data <- data.frame(horsepower = 98)

# Use the 'predict' function to make the prediction
predicted_mpg <- predict(model, newdata = new_data)

# Print the predicted 'mpg' value
print(predicted_mpg)
```

```
##        1
## 24.46708
```

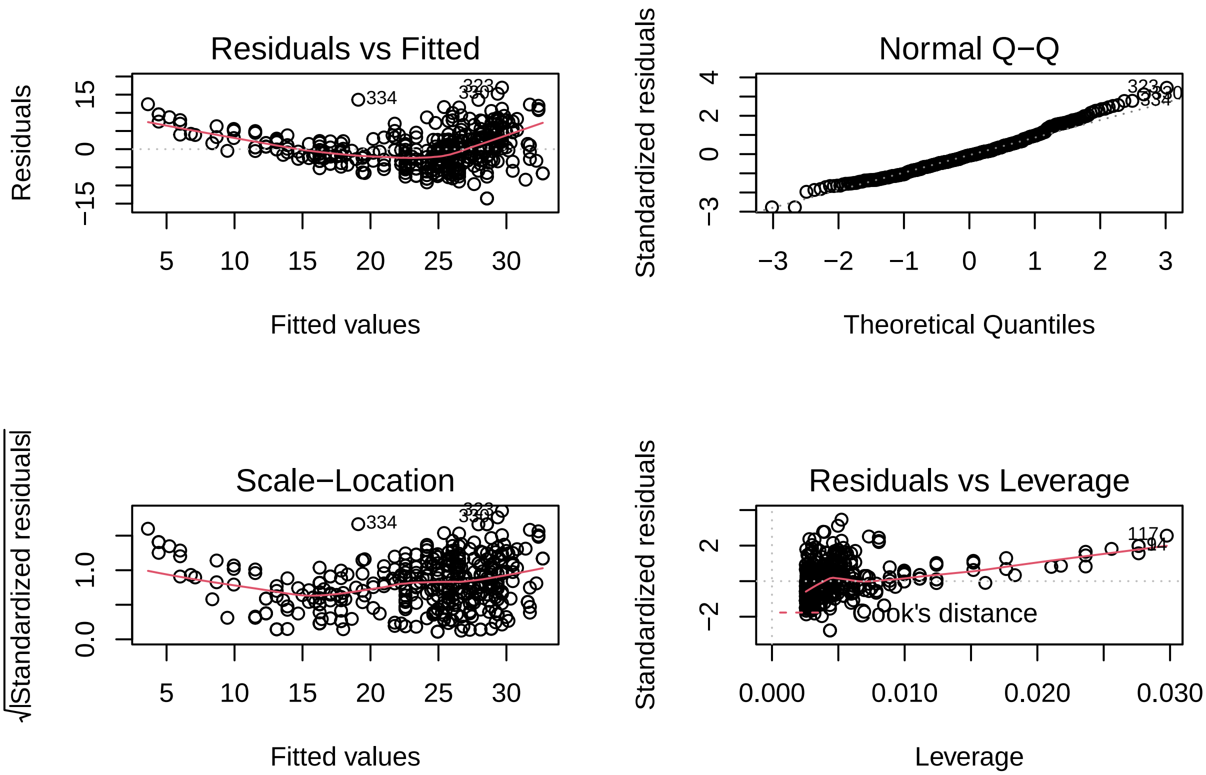
(c)

```
# Create a scatterplot of mpg vs. horsepower
plot(Auto$horsepower, Auto$mpg, xlab = "Horsepower", ylab = "MPG", main = "Scatterplot of MPG vs. Horsep

# Add the least squares regression line
abline(model, col = "red")
```



(d)

```
# Create diagnostic plots for the linear regression model
par(mfrow = c(2, 2))  # Set up a 2x2 grid for the plots
plot(model)
```



Looking at the residuals vs fitted graph, there is a slight curve to the graph thus meaning a more complex model might be better for the data.

Exersize 2

```r
set.seed(1)
```

(a)

```r
X <- rnorm(100, mean = 0, sd = 1)
```

(b)

```r
epsilon <- rnorm(100, mean = 0, sd = 0.25)
```

(c)

```r
# Calculate Y based on the linear model
Y <- -1 + 0.5 * X + epsilon

# (i) Length of vector Y
length_Y <- length(Y)

# (ii) Values of beta
beta_0 <- -1
beta_1 <- 0.5
```

i.

```r
length_Y
```

```
## [1] 100
```

ii.

```r
beta_0
```

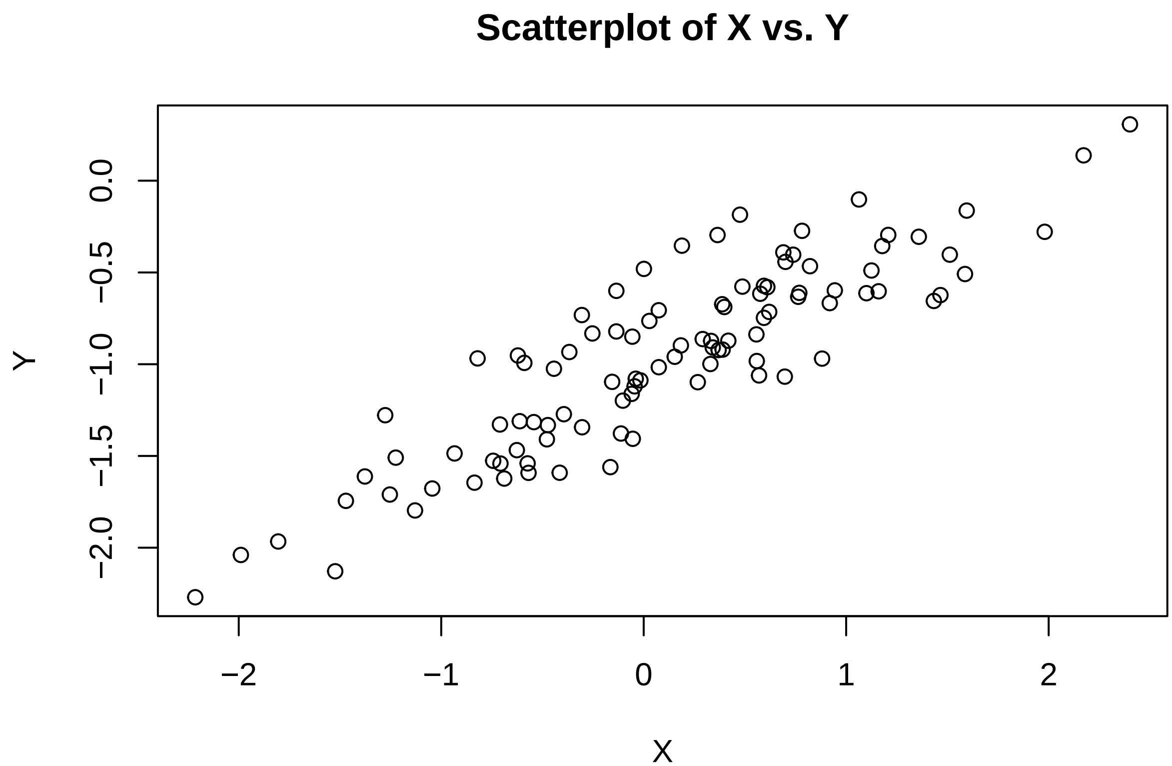
```
## [1] -1
```

```r
beta_1
```

```
## [1] 0.5
```

(d)

```r
# Create a scatterplot of X vs. Y
plot(X, Y, main = "Scatterplot of X vs. Y", xlab = "X", ylab = "Y")
```



Based off the plot above, there seems to be a positive realationship between X and Y, and the pattern is rather linear.

(e)

```r
# Fit a least squares linear model
model <- lm(Y ~ X)

# Print the summary of the model
summary(model)
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46921 -0.15344 -0.03487  0.13485  0.58654
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.00942    0.02425  -41.63   <2e-16 ***
## X            0.49973    0.02693   18.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.2407 on 98 degrees of freedom
## Multiple R-squared:  0.7784, Adjusted R-squared:  0.7762
## F-statistic: 344.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

The actual beta 0 was estimated at about -0.9745 while the actual beta 1 was estimated at about 0.5213. This is extremely close to the predicted beta values from earlier. Another thing to note from this model is that p-value for the relationship between x and y is extremely small thus meaning significance.