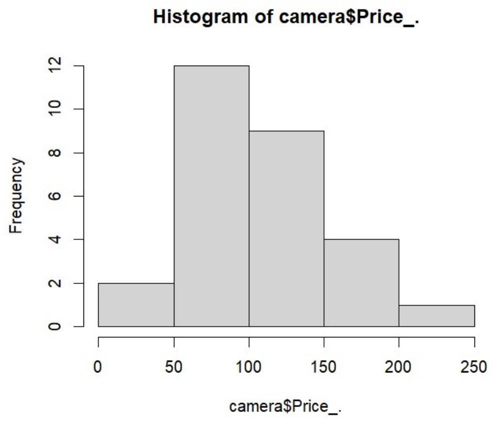
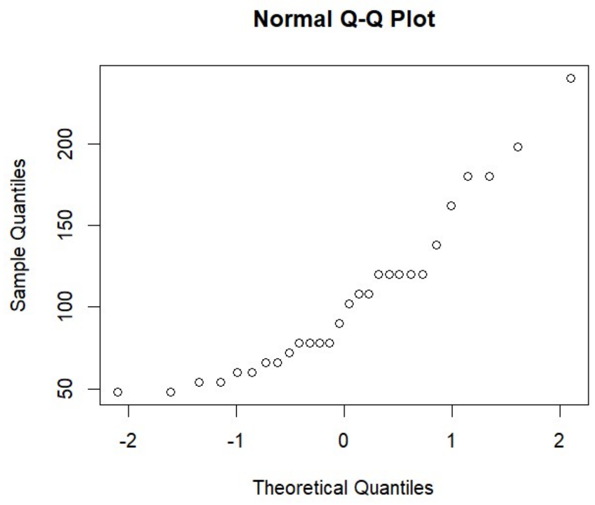
R Assignment – All Output file

```
> #Part b: Loading the data
> camera <- read.csv('C:/Users/musta/Desktop/Coursework/Statistical Inference/
  R assignment/Nikon.csv')


> #Part c:Data structure
> str(camera)
'data.frame':	28 obs. of  7 variables:
 $ Observation: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Brand      : chr  "Canon" "Canon" "Canon" "Canon" ...
 $ Price_.    : int  198 120 180 120 108 120 120 78 78 66 ...
 $ Megapixels : int  10 12 12 10 12 12 14 10 12 16 ...
 $ Weight_oz  : int  7 5 7 6 5 7 5 7 5 5 ...
 $ Score      : int  73 73 72 69 69 68 67 67 66 62 ...
 $ Brand_code : int  1 1 1 1 1 1 1 1 1 1 ...


> summary(camera)
  Observation        Brand               Price_.          Megapixels       Weight_
oz         Score
 Min.   : 1.00   Length:28          Min.   : 48.0   Min.   :10.00   Min.   :4
.000   Min.   :49.00
 1st Qu.: 7.75   Class :character   1st Qu.: 66.0   1st Qu.:12.00   1st Qu.:5
.000   1st Qu.:59.00
 Median :14.50   Mode  :character   Median : 96.0   Median :12.00   Median :6
.000   Median :63.50
 Mean   :14.50                      Mean   :105.2   Mean   :12.86   Mean   :5
.821   Mean   :63.36
 3rd Qu.:21.25                      3rd Qu.:120.0   3rd Qu.:14.00   3rd Qu.:7
.000   3rd Qu.:68.25
 Max.   :28.00                      Max.   :240.0   Max.   :16.00   Max.   :7
.000   Max.   :73.00
   Brand_code
 Min.   :0.0000
 1st Qu.:0.0000
 Median :0.0000
 Mean   :0.4643
 3rd Qu.:1.0000
 Max.   :1.0000


> #Part d(1)
> hist(camera$Price_.)
```

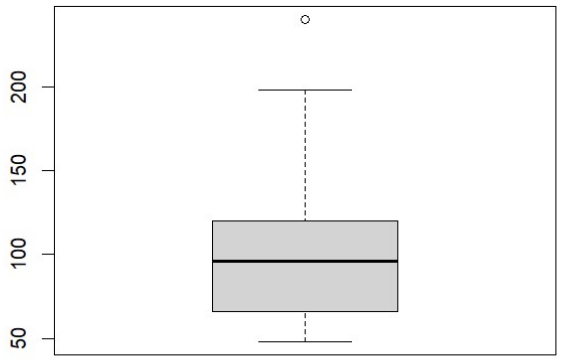


R Assignment – All Output file

```
> qqnorm(camera$Price_.)
```



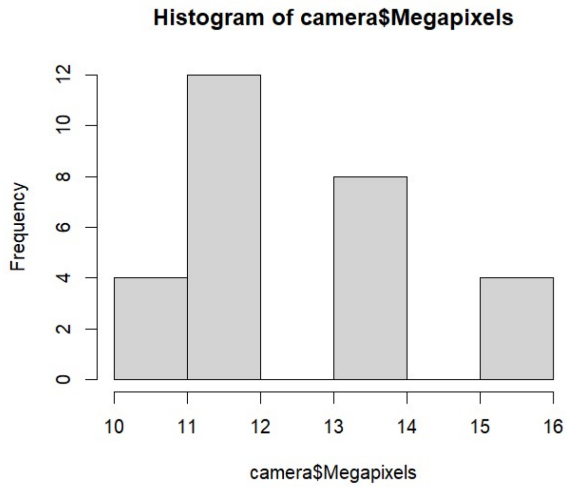
```
> boxplot(camera$Price_.)
```



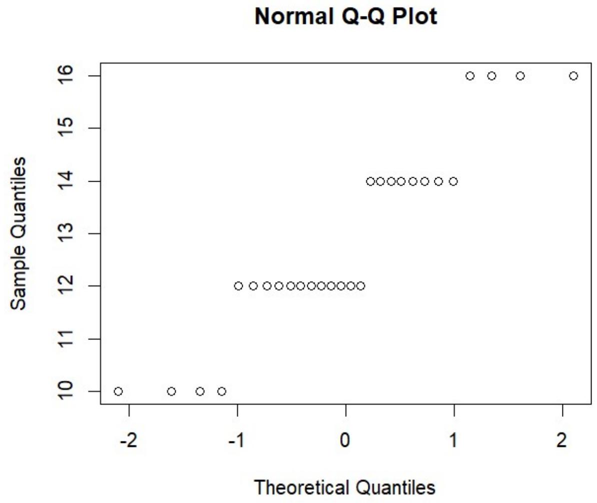
```
> ks.test(camera$Price_.,'pnorm')

        Asymptotic one-sample Kolmogorov-Smirnov test

data:  camera$Price_.
D = 1, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test.default(camera$Price_., "pnorm") :
  ties should not be present for the Kolmogorov-Smirnov test

> shapiro.test(camera$Price_.)

        Shapiro-Wilk normality test

data:  camera$Price_.
W = 0.89739, p-value = 0.009945
```

R Assignment – All Output file

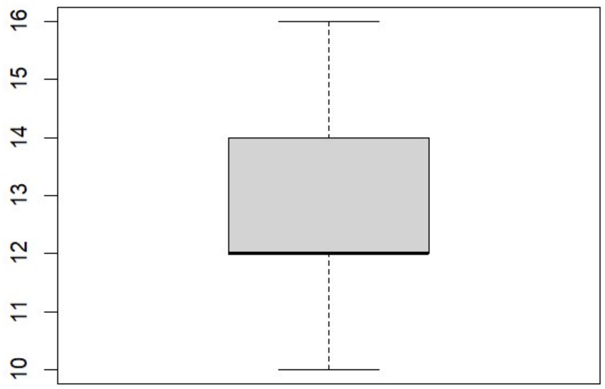
```
> hist(camera$Megapixels)
```



```
> qqnorm(camera$Megapixels)
```

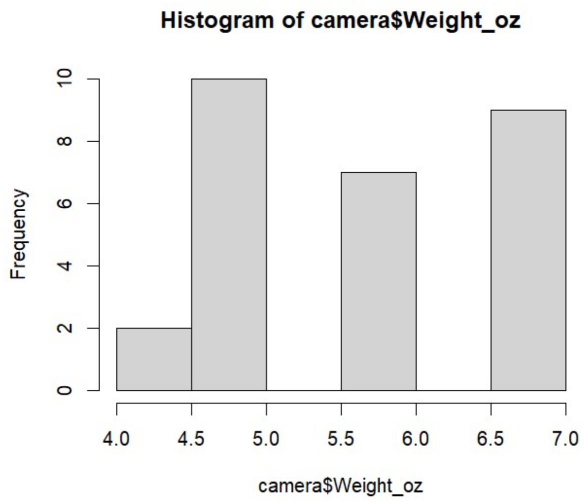


```
> boxplot(camera$Megapixels)
```

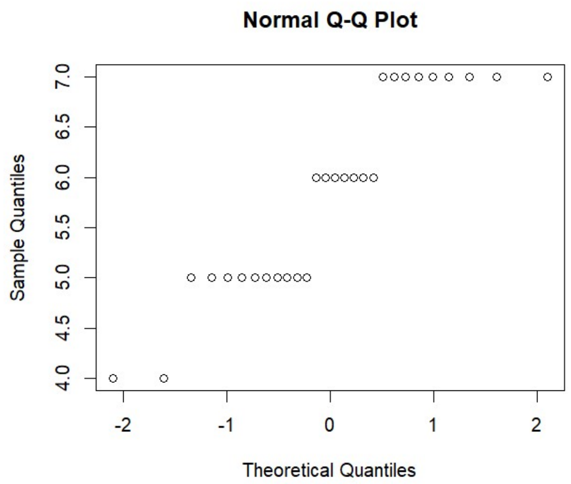


R Assignment – All Output file

```
> ks.test(camera$Megapixels,'pnorm')

        Asymptotic one-sample Kolmogorov-Smirnov test

data:  camera$Megapixels
D = 1, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test.default(camera$Megapixels, "pnorm") :
  ties should not be present for the Kolmogorov-Smirnov test
> shapiro.test(camera$Megapixels)

        Shapiro-Wilk normality test

data:  camera$Megapixels
W = 0.87756, p-value = 0.003549


> hist(camera$Weight_oz)
```

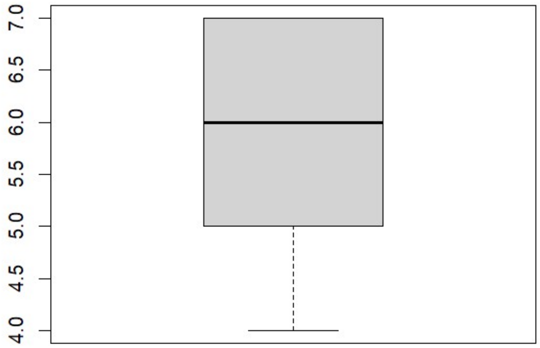


```
> qqnorm(camera$Weight_oz)
```



R Assignment – All Output file

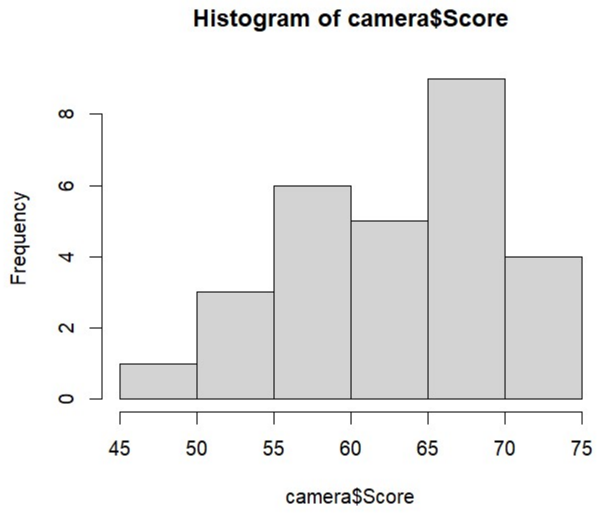
```
> boxplot(camera$Weight_oz)
```



```
> ks.test(camera$Score,'pnorm')

	Asymptotic one-sample Kolmogorov-Smirnov test

data:  camera$Score
D = 1, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test.default(camera$Score, "pnorm") :
  ties should not be present for the Kolmogorov-Smirnov test
```

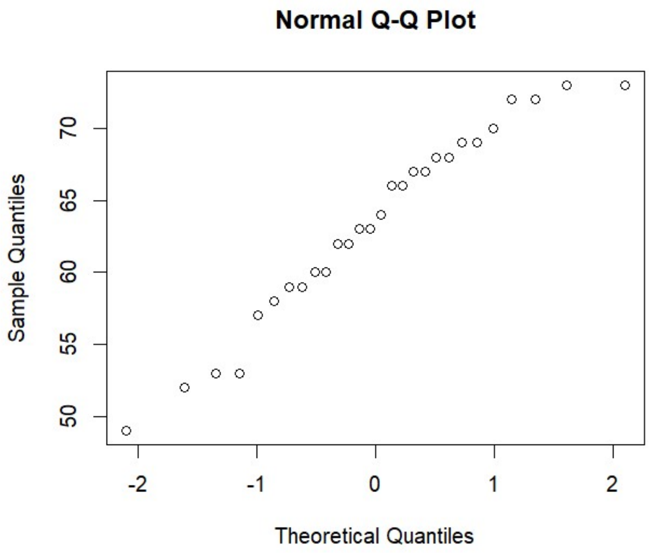
```
> shapiro.test(camera$Score)

	Shapiro-Wilk normality test

data:  camera$Score
W = 0.95719, p-value = 0.2985
```

```
> hist(camera$Score)
```

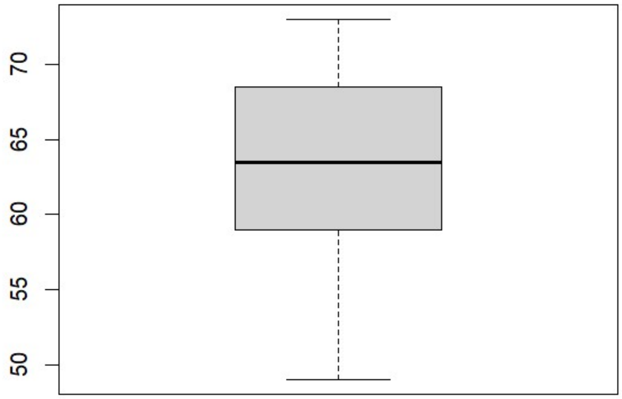


R Assignment – All Output file

```
> qqnorm(camera$Score)
```



```
> boxplot(camera$Score)
```

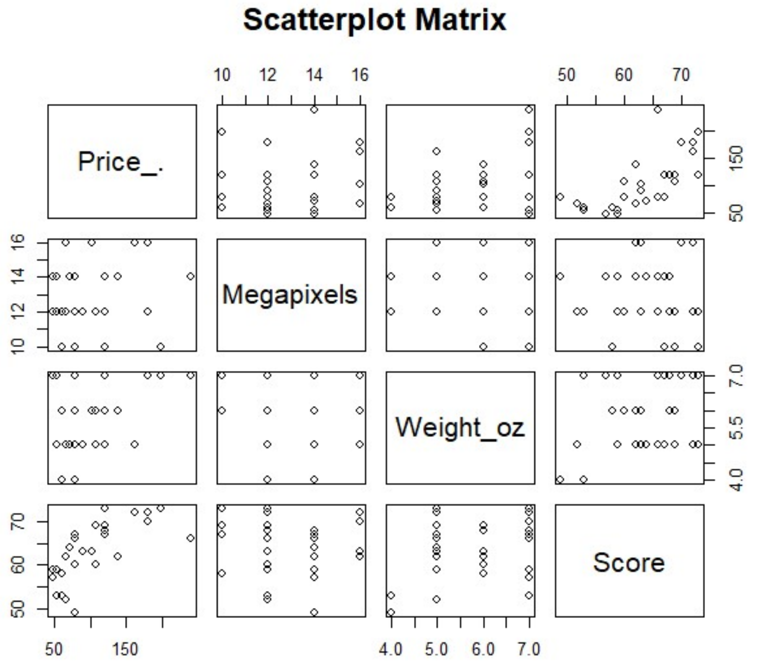


```
> ks.test(camera$Score,'pnorm')

        Asymptotic one-sample Kolmogorov-Smirnov test

data:  camera$Score
D = 1, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test.default(camera$Score, "pnorm") :
  ties should not be present for the Kolmogorov-Smirnov test
```

```
> shapiro.test(camera$Score)

        Shapiro-Wilk normality test

data:  camera$Score
W = 0.95719, p-value = 0.2985
```

R Assignment – All Output file

```
> #Part d - 2)
> camera1 <- camera[ ,-c(1,2,7)] #removing non-numeric variables
> pairs(camera1, main = 'Scatterplot Matrix')
```



```
> #Part d - 3)
> cor_matrix <- cor(camera1)
> cor_matrix
             Price_.   Megapixels  Weight_oz        Score
Price_.    1.0000000  0.138906307  0.3488151  0.683211844
Megapixels 0.1389063  1.000000000 -0.1988338 -0.007729723
Weight_oz  0.3488151 -0.198833809  1.0000000  0.285688204
Score      0.6832118 -0.007729723  0.2856882  1.000000000
>
> install.packages("psych")
> library("psych")
>
> cor_test_mat <- corr.test(camera1)$p
> cor_test_mat
                Price_. Megapixels Weight_oz        Score
Price_.    0.000000e+00  0.9616942 0.3443848 0.0003693039
Megapixels 4.808471e-01  0.0000000 0.9312695 0.9688604750
Weight_oz  6.887697e-02  0.3104232 0.0000000 0.5622358653
Score      6.155065e-05  0.9688605 0.1405590 0.0000000000
```

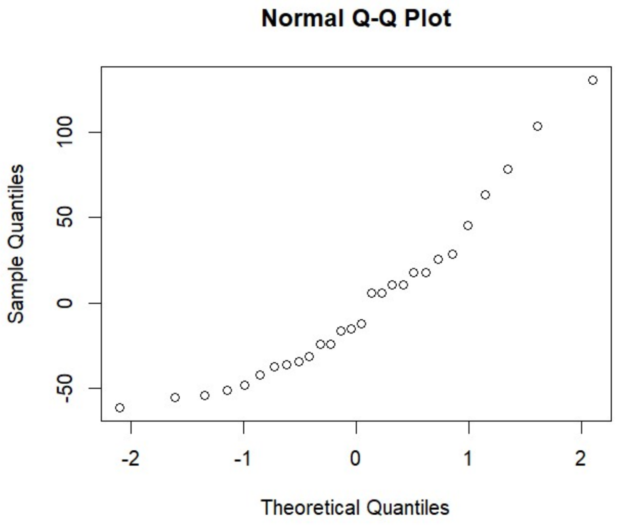
```
> #Part d - 4)
> m1 <- lm(Price_. ~ Megapixels, data=camera1)
```

```
> #Part d - 5)
> m2 <- lm(Price_. ~ Megapixels + Weight_oz, data=camera1)
```

```
> #Part d - 6)
> m3 <- lm(Price_. ~ Megapixels + Weight_oz + Score, data=camera1)
```

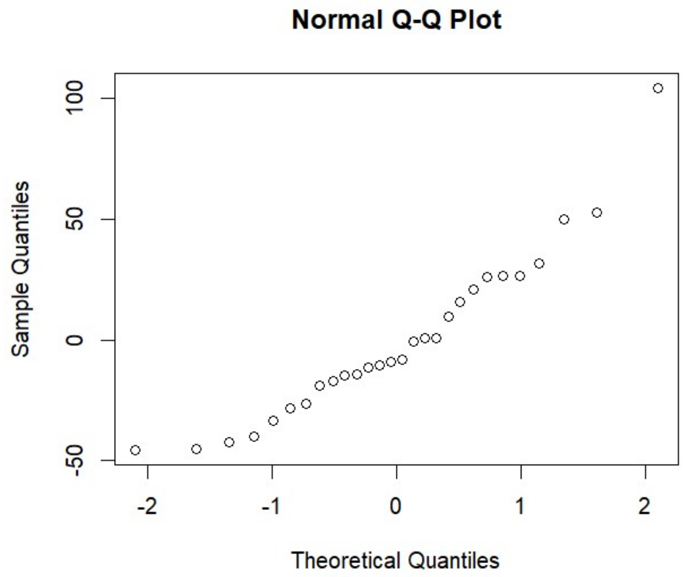
R Assignment – All Output file

```
> #Part d - 7)
> summary(m1)

Call:
lm(formula = Price_. ~ Megapixels, data = camera1)

Residuals:
   Min     1Q Median     3Q    Max 
-61.50 -36.38 -13.50  19.88 130.50 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   57.000     68.074   0.837    0.410
Megapixels     3.750      5.243   0.715    0.481

Residual standard error: 50.13 on 26 degrees of freedom
Multiple R-squared:  0.01929,	Adjusted R-squared:  -0.01842 
F-statistic: 0.5115 on 1 and 26 DF,  p-value: 0.4808

> summary(m3)

Call:
lm(formula = Price_. ~ Megapixels + Weight_oz + Score, data = camera1)

Residuals:
    Min      1Q  Median      3Q     Max 
-45.730 -20.986  -8.589  22.127 104.498 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -313.852     89.606  -3.503 0.001831 ** 
Megapixels     4.991      3.880   1.286 0.210573    
Weight_oz     10.451      7.576   1.379 0.180467    
Score          4.641      1.090   4.256 0.000275 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 36.31 on 24 degrees of freedom
Multiple R-squared:  0.5252,	Adjusted R-squared:  0.4659 
F-statistic:  8.85 on 3 and 24 DF,  p-value: 0.0003961

> qqnorm(m1$residuals)
```



R Assignment – All Output file

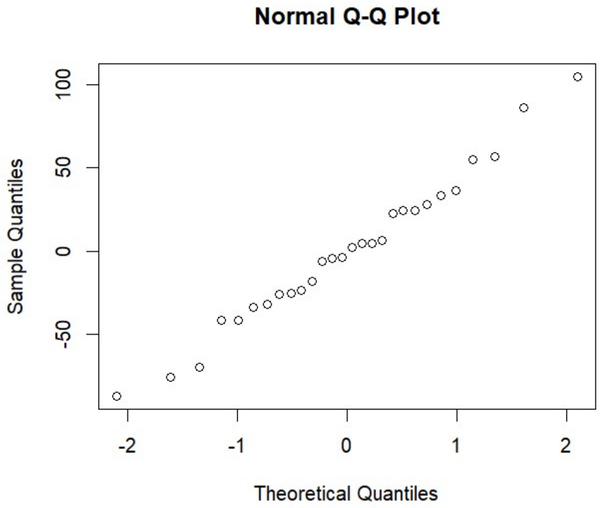
```
> qqnorm(m3$residuals)
```



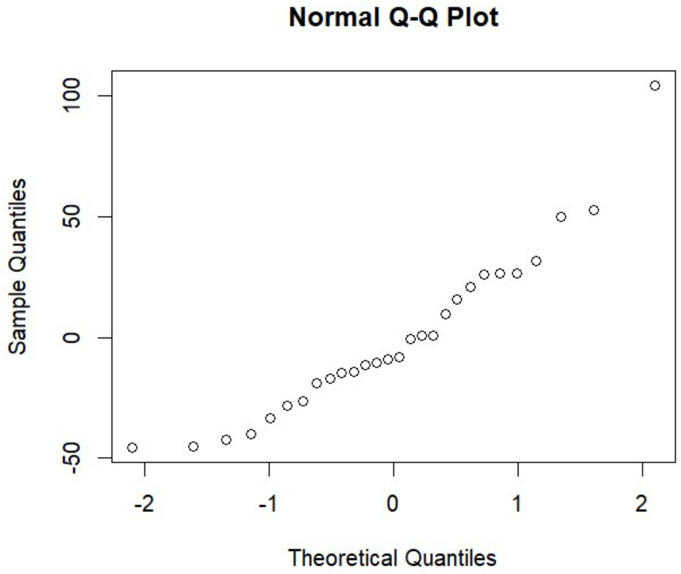
```
> #Part d - 8)
> summary(m2)

Call:
lm(formula = Price_. ~ Megapixels + Weight_oz, data = camera1)

Residuals:
    Min      1Q  Median      3Q     Max 
-87.241 -27.306  -0.686  25.264 104.759 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)  -85.317     93.111  -0.916   0.3683  
Megapixels     5.854      5.029   1.164   0.2554  
Weight_oz     19.801      9.411   2.104   0.0456 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 47.12 on 25 degrees of freedom
Multiple R-squared:  0.1668,	Adjusted R-squared:  0.1002 
F-statistic: 2.503 on 2 and 25 DF,  p-value: 0.1021

> summary(m3)

Call:
lm(formula = Price_. ~ Megapixels + Weight_oz + Score, data = camera1)

Residuals:
    Min      1Q  Median      3Q     Max 
-45.730 -20.986  -8.589  22.127 104.498 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -313.852     89.606  -3.503 0.001831 ** 
Megapixels     4.991      3.880   1.286 0.210573    
Weight_oz     10.451      7.576   1.379 0.180467    
Score          4.641      1.090   4.256 0.000275 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

R Assignment – All Output file

```
Residual standard error: 36.31 on 24 degrees of freedom
Multiple R-squared:  0.5252,	Adjusted R-squared:  0.4659 
F-statistic:  8.85 on 3 and 24 DF,  p-value: 0.0003961

> qqnorm(m2$residuals)
```



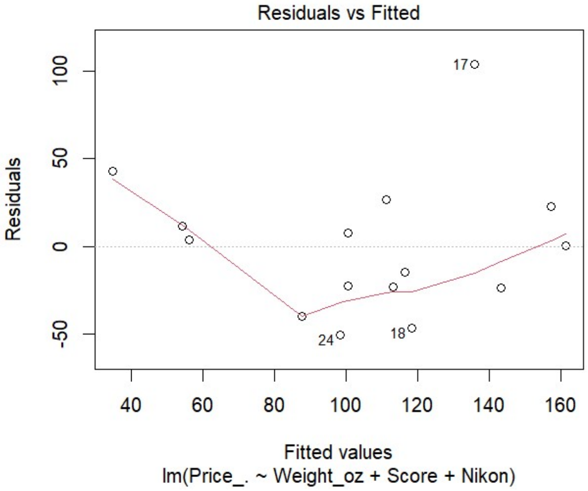
```
> qqnorm(m3$residuals)
```



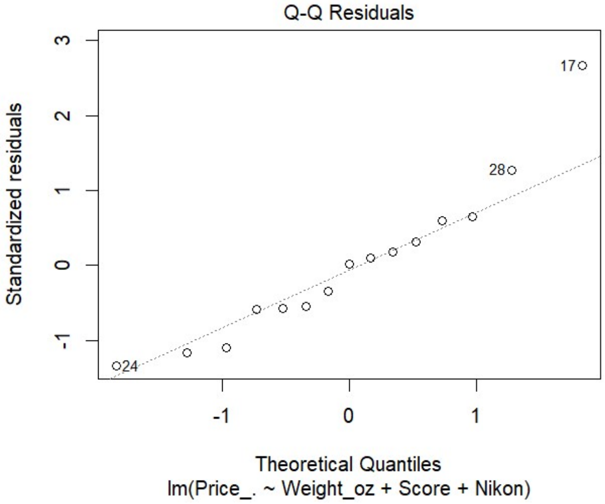
```
> #Part d - 9)
> camera$Nikon <- ifelse(camera$Brand_code == 0,0,1)

> #Part d - 10)
> m4 <- lm(Price_. ~ Weight_oz + Score + Nikon, data=camera)

> #Part d - 11)
> nikon_df <- subset(camera, Brand=='Nikon')
> canon_df <- subset(camera, Brand=='Canon')

> m4_nikon <- lm(Price_. ~ Weight_oz + Score + Nikon, data=nikon_df)
> m4_canon <- lm(Price_. ~ Weight_oz + Score + Nikon, data=cannon_df)
```

R Assignment – All Output file

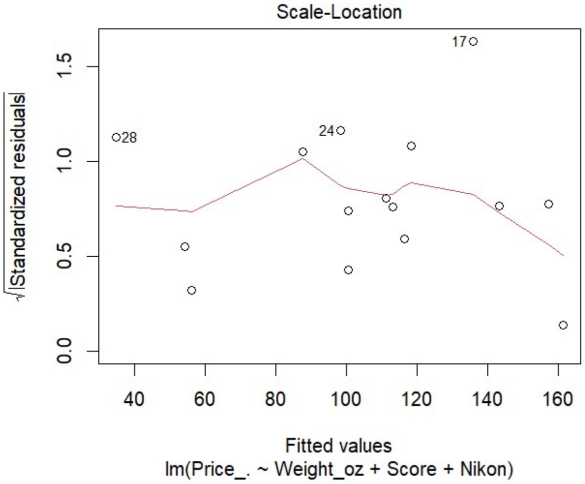
```
> plot(m4_nikon)
```



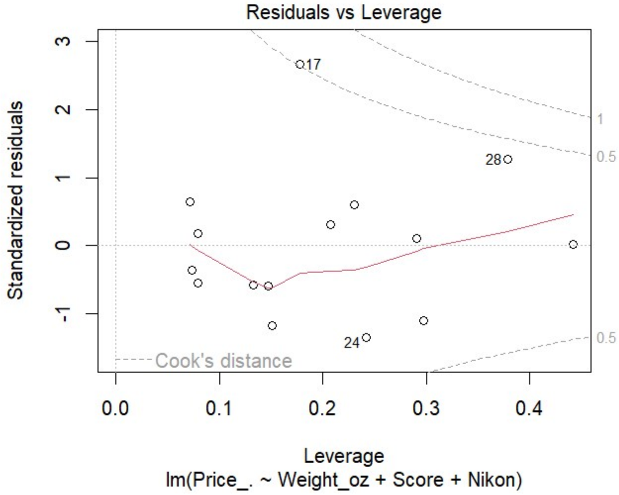
Residuals vs Fitted



Q-Q Residuals



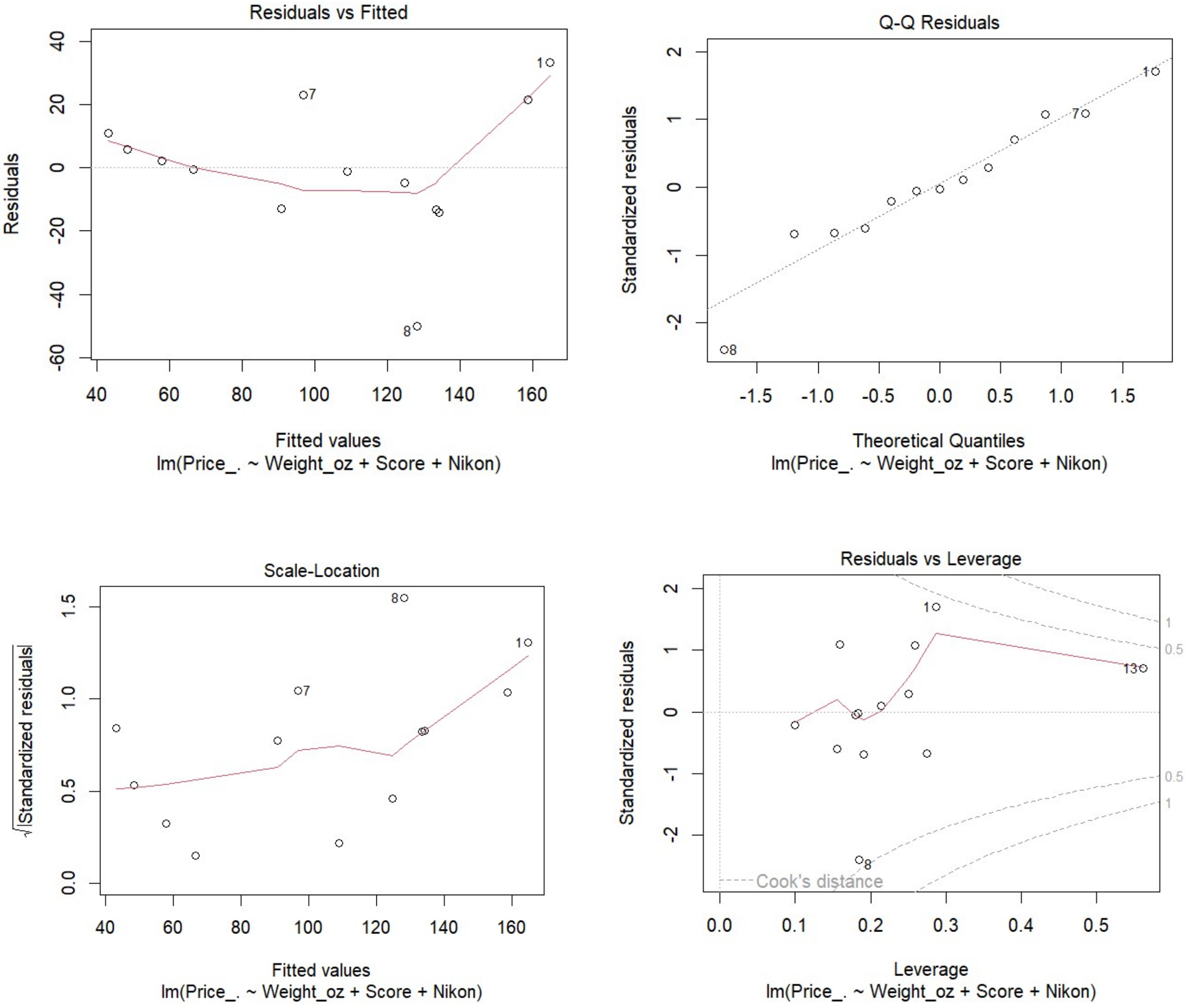
Scale-Location



Residuals vs Leverage

R Assignment – All Output file

```
> plot(m4_canon)
```



```
> #Part d - 12)
> new_data <- data.frame(
+   Brand = c("Canon", "Canon", "Nikon", "Nikon"),
+   Price_ = c(100, 90, 270, 300),
+   Megapixels = c(10, 12, 16, 16),
+   Weight_oz = c(6, 7, 5, 7),
+   Score = c(51, 46, 65, 63),
+   Brand_code = c(1, 1, 0, 0)
+ )

> pred_m1 <- predict(m1, new_data)
> pred_m2 <- predict(m2, new_data)
> pred_m3 <- predict(m3, new_data)

> new_data1 <- new_data
> new_data1$Nikon <- ifelse(new_data1$Brand_code == 0,0,1)
> pred_m4 <- predict(m4, new_data1)
```

R Assignment – All Output file

```
> #Error values
> error_m1 <- new_data$Price_ - pred_m1
> error_m1
    1     2     3     4 
  5.5 -12.0 153.0 183.0 
> 
> error_m2 <- new_data$Price_ - pred_m2
> error_m2
         1          2          3          4 
  7.975075 -33.533151 162.652792 153.051595 
> 
> error_m3 <- new_data$Price_ - pred_m3
> error_m3
        1         2         3         4 
 64.53223  57.30575 150.05963 168.44028 
> 
> error_m4 <- new_data1$Price_ - pred_m4
> error_m4
        1         2         3         4 
 76.48554  85.50565 149.32050 173.47598 
```