Name: Shaikh Inamul Hasan

Roll No: 100

Lab 7: Supervised Learning - Regression

1. Below is the sample data representing the observations –

Values of height

151, 174, 138, 186, 128, 136, 179, 163, 152, 131

Values of weight.

63, 81, 56, 91, 47, 57, 76, 72, 62, 48

a. Create height and weight vectors using above data.

Code & Output:

```
> height <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
> weight <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
```

b. Create relationship model & get the coefficients using linear model function of R (lm).

Code & Output:

```
> relation <- lm(weight ~ height)
```

c. Get the summary of the relationship and predict the weight of new persons whose height is 170.

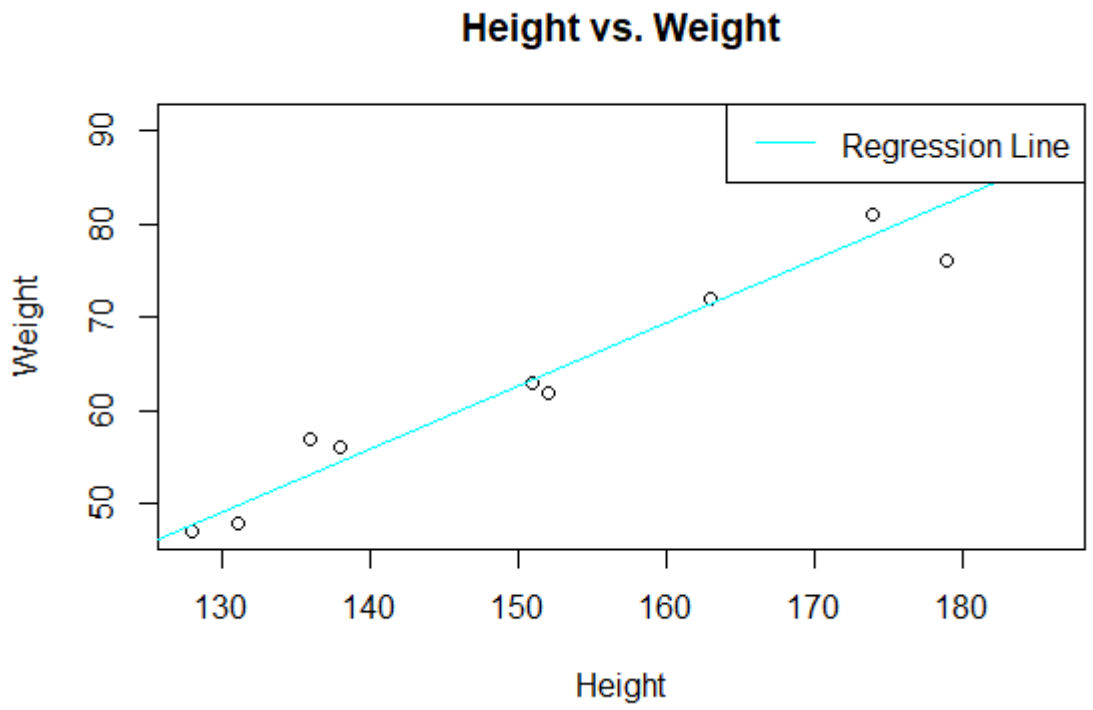
Code & Output:

```
> summary(relation)

Call:
lm(formula = weight ~ height)

Residuals:
    Min      1Q  Median      3Q     Max
-6.3002 -1.6629  0.0412  1.8944  3.9775

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -38.45509    8.04901  -4.778  0.00139 **
height        0.67461    0.05191  12.997 1.16e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.253 on 8 degrees of freedom
Multiple R-squared:  0.9548,    Adjusted R-squared:  0.9491
F-statistic: 168.9 on 1 and 8 DF,  p-value: 1.164e-06

> new_height <- data.frame(height = 170)
> predicted_weight <- predict(relation, newdata = new_height)
> cat("Predicted weight for a height of 170:", predicted_weight, "\n")
Predicted weight for a height of 170: 76.22869
```

d. Visualize the regression graphically.

Code & Output:

```
> plot(height, weight, main = "Height vs. Weight", xlab = "Height", ylab = "Weight")
> abline(relation, col = "cyan")
> legend("topright", legend = "Regression Line", col = "cyan", lty = 1)
```



2. Simple Linear regression

a. Use the dataset Fish.csv for linear regression.

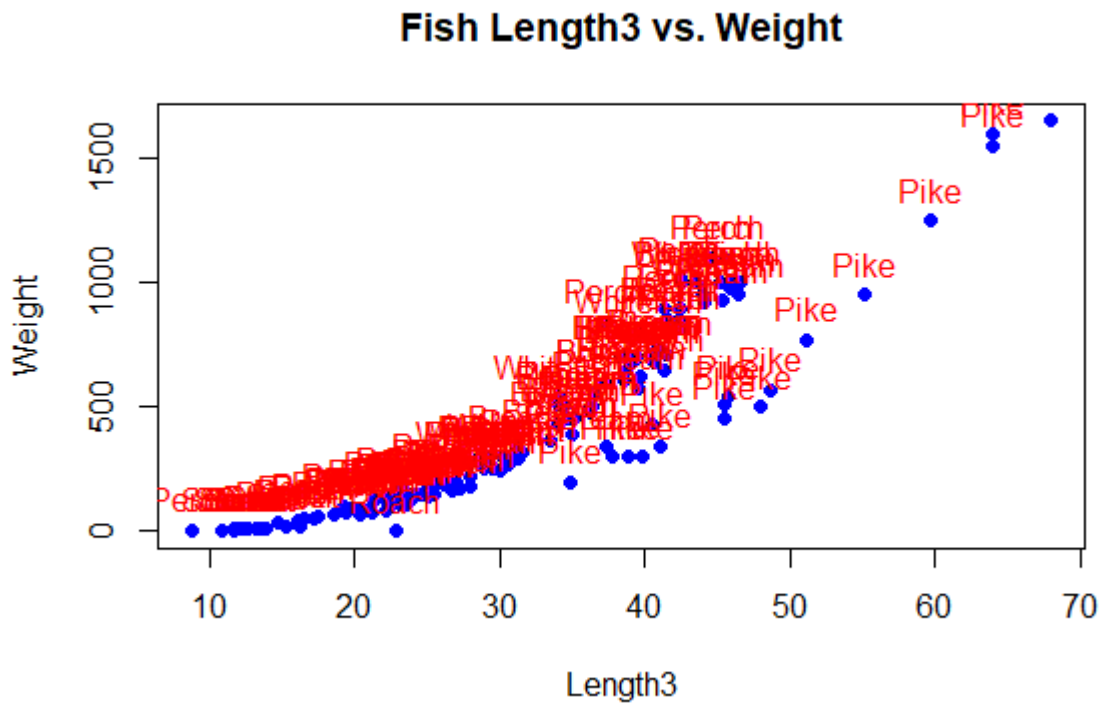
Code & Output:

```
> my_data<-read.csv("Fish.csv")
> my_data
      Species Weight Length1 Length2 Length3  Height  Width
1       Bream  242.0    23.2    25.4    30.0 11.5200 4.0200
2       Bream  290.0    24.0    26.3    31.2 12.4800 4.3056
3       Bream  340.0    23.9    26.5    31.1 12.3778 4.6961
4       Bream  363.0    26.3    29.0    33.5 12.7300 4.4555
5       Bream  430.0    26.5    29.0    34.0 12.4440 5.1340
6       Bream  450.0    26.8    29.7    34.7 13.6024 4.9274
7       Bream  500.0    26.8    29.7    34.5 14.1795 5.2785
8       Bream  390.0    27.6    30.0    35.0 12.6700 4.6900
9       Bream  450.0    27.6    30.0    35.1 14.0049 4.8438
10      Bream  500.0    28.5    30.7    36.2 14.2266 4.9594
```

b. Plot the scatter graphs and check the relationship between Length3 and Weight columns of Fish dataset.

Code & Output:

```
> plot(my_data$Length3, my_data$weight,
+      main = "Fish Length3 vs. Weight",
+      xlab = "Length3",
+      ylab = "Weight",
+      pch = 16,
+      col = "blue"
+ )
> text(my_data$Length3, my_data$weight, labels = my_data$Species, pos = 3, col = "red")
```



c. Randomize the dataset rows.

Code & Output:

```
> randomized_data <- my_data[sample(nrow(my_data)), ]
> head(randomized_data)
    Species Weight Length1 Length2 Length3  Height  Width
32    Bream    955    35.0    38.5    44.0 18.0840 6.2920
45    Roach    145    20.5    22.0    24.3  6.6339 3.5478
139    Pike    567    43.2    46.0    48.7  7.7920 4.8700
143    Pike   1600    56.0    60.0    64.0  9.6000 6.1440
39    Roach     87    18.2    19.8    22.2  5.6166 3.1746
100   Perch    180    23.0    25.0    26.5  6.4395 3.6835
```

d. Split the data set into Training Data set and Test Data set.

Code & Output:

```
> TrainData <- my_data[1:111,]
> TestData <- my_data[112:159,]
> TrainData
      Species Weight Length1 Length2 Length3  Height  Width
1       Bream  242.0    23.2    25.4    30.0 11.5200 4.0200
2       Bream  290.0    24.0    26.3    31.2 12.4800 4.3056
3       Bream  340.0    23.9    26.5    31.1 12.3778 4.6961
4       Bream  363.0    26.3    29.0    33.5 12.7300 4.4555
5       Bream  430.0    26.5    29.0    34.0 12.4440 5.1340
6       Bream  450.0    26.8    29.7    34.7 13.6024 4.9274
7       Bream  500.0    26.8    29.7    34.5 14.1795 5.2785
8       Bream  390.0    27.6    30.0    35.0 12.6700 4.6900
9       Bream  450.0    27.6    30.0    35.1 14.0049 4.8438
10      Bream  500.0    28.5    30.7    36.2 14.2266 4.9594
11      Bream  475.0    28.4    31.0    36.2 14.2628 5.1042
12      Bream  500.0    28.7    31.0    36.2 14.3714 4.8146
13      Bream  500.0    29.1    31.5    36.4 13.7592 4.3680
14      Bream  340.0    29.5    32.0    37.3 13.9129 5.0728
15      Bream  600.0    29.4    32.0    37.2 14.9544 5.1708
16      Bream  600.0    29.4    32.0    37.2 15.4380 5.5800
17      Bream  700.0    30.4    33.0    38.3 14.8604 5.2854
18      Bream  700.0    30.4    33.0    38.5 14.9380 5.1975
19      Bream  610.0    30.9    33.5    38.6 15.6330 5.1338
20      Bream  650.0    31.0    33.5    38.7 14.4738 5.7276
21      Bream  575.0    31.3    34.0    39.5 15.1285 5.5695
22      Bream  685.0    31.4    34.0    39.2 15.9936 5.3704
23      Bream  620.0    31.5    34.5    39.7 15.5227 5.2801
24      Bream  680.0    31.8    35.0    40.6 15.4686 6.1306
25      Bream  700.0    31.9    35.0    40.5 16.2405 5.5890
26      Bream  725.0    31.8    35.0    40.9 16.3600 6.0532
27      Bream  720.0    32.0    35.0    40.6 16.3618 6.0900
28      Bream  714.0    32.7    36.0    41.5 16.5170 5.8515
29      Bream  850.0    32.8    36.0    41.6 16.8896 6.1984
30      Bream 1000.0    33.5    37.0    42.6 18.9570 6.6030

> TestData
     Species Weight Length1 Length2 Length3  Height  Width
112   Perch  840.0    32.5    35.0    37.3 11.4884 7.7957
113   Perch  685.0    34.0    36.5    39.0 10.8810 6.8640
114   Perch  700.0    34.0    36.0    38.3 10.6091 6.7408
115   Perch  700.0    34.5    37.0    39.4 10.8350 6.2646
116   Perch  690.0    34.6    37.0    39.3 10.5717 6.3666
117   Perch  900.0    36.5    39.0    41.4 11.1366 7.4934
118   Perch  650.0    36.5    39.0    41.4 11.1366 6.0030
119   Perch  820.0    36.6    39.0    41.3 12.4313 7.3514
120   Perch  850.0    36.9    40.0    42.3 11.9286 7.1064
121   Perch  900.0    37.0    40.0    42.5 11.7300 7.2250
122   Perch 1015.0    37.0    40.0    42.4 12.3808 7.4624
123   Perch  820.0    37.1    40.0    42.5 11.1350 6.6300
124   Perch 1100.0    39.0    42.0    44.6 12.8002 6.8684
125   Perch 1000.0    39.8    43.0    45.2 11.9328 7.2772
126   Perch 1100.0    40.1    43.0    45.5 12.5125 7.4165
127   Perch 1000.0    40.2    43.5    46.0 12.6040 8.1420
128   Perch 1000.0    41.1    44.0    46.6 12.4888 7.5958
129    Pike  200.0    30.0    32.3    34.8  5.5680 3.3756
130    Pike  300.0    31.7    34.0    37.8  5.7078 4.1580
131    Pike  300.0    32.7    35.0    38.8  5.9364 4.3844
132    Pike  300.0    34.8    37.3    39.8  6.2884 4.0198
133    Pike  430.0    35.5    38.0    40.5  7.2900 4.5765
```

e. Perform single linear regression analysis on training dataset columns Length3 as Y and Weight as X, using linear model function (lm).

Code & Output:

```
> fit = lm(Length3 ~ Weight , data=TrainData)
> summary(fit)

Call:
lm(formula = Length3 ~ Weight, data = TrainData)

Residuals:
     Min       1Q   Median       3Q      Max
-11.1397  -1.2017   0.2679   1.5833   7.7128

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 19.769306   0.381054   51.88   <2e-16 ***
Weight       0.028876   0.000907   31.84   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.532 on 109 degrees of freedom
Multiple R-squared:  0.9029,    Adjusted R-squared:  0.902
F-statistic:  1014 on 1 and 109 DF,  p-value: < 2.2e-16
```

f. Predict the Length3 value using Testing dataset.

Code & Output:

```
> preds <- predict(fit, newdata = TestData)
> preds
     112      113      114      115      116      117      118      119      120      121      122      123      124      125      126      127      128
44.02519 39.54940 39.98254 39.98254 39.69378 45.75775 38.53874 43.44767 44.31395 45.75775 49.07850 43.44767 51.53296 48.64536 51.53296 48.64536 48.64536
     129      130      131      132      133      134      135      136      137      138      139      140      141      142      143      144      145
25.54452 28.43212 28.43212 28.43212 32.18601 29.73154 32.93679 34.49609 35.36237 34.20733 36.14203 42.00387 47.20156 55.86437 65.97099 64.52719 67.41479
     146      147      148      149      150      151      152      153      154      155      156      157      158      159
19.96278 19.98588 19.97144 20.04940 20.05229 20.02053 20.05807 20.05518 20.05229 20.12159 20.15625 20.12159 20.33816 20.34394
```

g. Analyze the Testing result using predicted and actual value of the Length3 column data and calculate correlation between them.

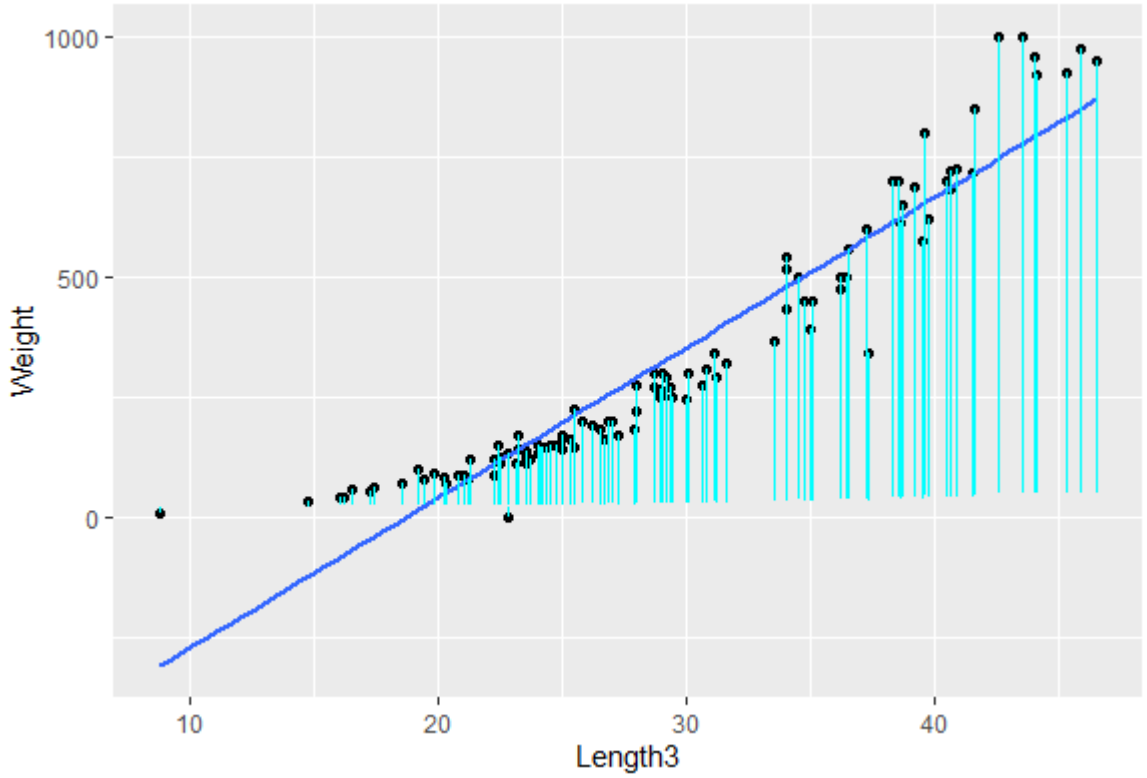
Code & Output:

```
> df1 <- data.frame(preds,TestData$Length3)
> df1
       preds TestData.Length3
112 44.02519             37.3
113 39.54940             39.0
114 39.98254             38.3
115 39.98254             39.4
116 39.69378             39.3
117 45.75775             41.4
118 38.53874             41.4
119 43.44767             41.3
120 44.31395             42.3
121 45.75775             42.5
122 49.07850             42.4
123 43.44767             42.5
124 51.53296             44.6
125 48.64536             45.2
126 51.53296             45.5
127 48.64536             46.0
128 48.64536             46.6
129 25.54452             34.8
130 28.43212             37.8
131 28.43212             38.8
132 28.43212             39.8
133 32.18601             40.5
134 29.73154             41.0
135 32.93679             45.5
136 34.49609             45.5
137 35.36237             45.8
```

h. Analyze the regression line with Residuals(line segment which represents the distance between y-value of the actual scatter plot points and the y values of the regression equation at those points) on a scatter plot.

Code & Output:

```
> install.packages("ggplot2")
WARNING: Rtools is required to build R packages but is not currently installed. P

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/Admin/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/ggplot2_3.4.4.zip'
Content type 'application/zip' length 4301159 bytes (4.1 MB)
downloaded 4.1 MB

package 'ggplot2' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Admin\AppData\Local\Temp\RtmpWcd1cn\downloaded_packages

> library("ggplot2")
> ggplot(fit, aes(Length3, Weight)) +
+   geom_point() +
+   stat_smooth(method = lm, se = FALSE) +
+   geom_segment(aes(xend = Length3, yend = .fitted), color = "cyan", size = 0.3)
`geom_smooth()` using formula = 'y ~ x'
Warning message:
Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```



3. Multiple Linear regression

a. Use the same training and testing dataset of Fish.csv created in exercise 2.

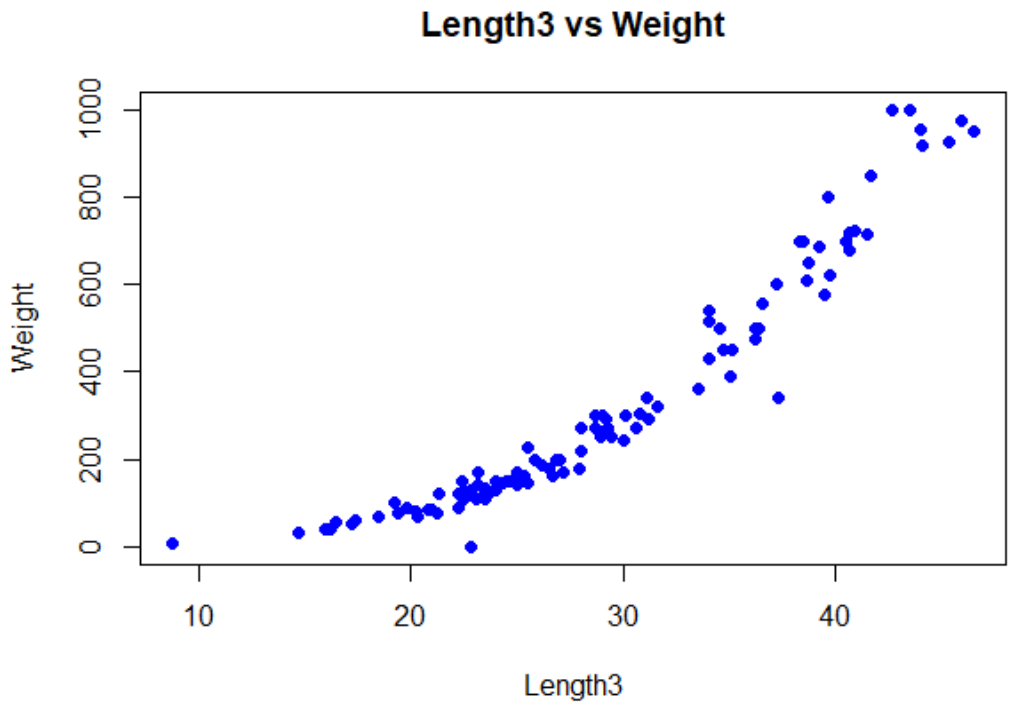
Code & Output:

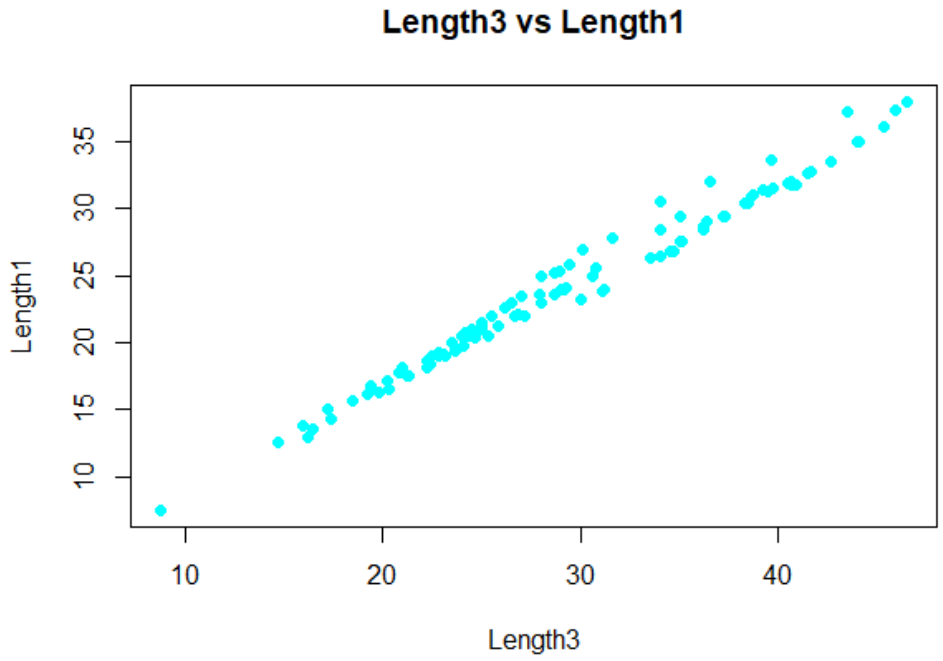
```
> TrainData <- my_data[1:111,]
> TestData <- my_data[112:159,]
> TrainData
      Species Weight Length1 Length2 Length3  Height  Width
1       Bream  242.0    23.2    25.4    30.0 11.5200 4.0200
2       Bream  290.0    24.0    26.3    31.2 12.4800 4.3056
3       Bream  340.0    23.9    26.5    31.1 12.3778 4.6961
4       Bream  363.0    26.3    29.0    33.5 12.7300 4.4555
5       Bream  430.0    26.5    29.0    34.0 12.4440 5.1340
6       Bream  450.0    26.8    29.7    34.7 13.6024 4.9274
7       Bream  500.0    26.8    29.7    34.5 14.1795 5.2785
8       Bream  390.0    27.6    30.0    35.0 12.6700 4.6900
9       Bream  450.0    27.6    30.0    35.1 14.0049 4.8438
10      Bream  500.0    28.5    30.7    36.2 14.2266 4.9594
11      Bream  475.0    28.4    31.0    36.2 14.2628 5.1042
12      Bream  500.0    28.7    31.0    36.2 14.3714 4.8146
13      Bream  500.0    29.1    31.5    36.4 13.7592 4.3680
14      Bream  340.0    29.5    32.0    37.3 13.9129 5.0728
15      Bream  600.0    29.4    32.0    37.2 14.9544 5.1708
16      Bream  600.0    29.4    32.0    37.2 15.4380 5.5800
17      Bream  700.0    30.4    33.0    38.3 14.8604 5.2854
18      Bream  700.0    30.4    33.0    38.5 14.9380 5.1975
19      Bream  610.0    30.9    33.5    38.6 15.6330 5.1338
20      Bream  650.0    31.0    33.5    38.7 14.4738 5.7276
21      Bream  575.0    31.3    34.0    39.5 15.1285 5.5695
22      Bream  685.0    31.4    34.0    39.2 15.9936 5.3704
23      Bream  620.0    31.5    34.5    39.7 15.5227 5.2801
24      Bream  680.0    31.8    35.0    40.6 15.4686 6.1306
25      Bream  700.0    31.9    35.0    40.5 16.2405 5.5890
26      Bream  725.0    31.8    35.0    40.9 16.3600 6.0532
27      Bream  720.0    32.0    35.0    40.6 16.3618 6.0900
28      Bream  714.0    32.7    36.0    41.5 16.5170 5.8515
29      Bream  850.0    32.8    36.0    41.6 16.8896 6.1984
30      Bream 1000.0    33.5    37.0    42.6 18.9570 6.6030

> TestData
     Species Weight Length1 Length2 Length3  Height  Width
112    Perch  840.0    32.5    35.0    37.3 11.4884 7.7957
113    Perch  685.0    34.0    36.5    39.0 10.8810 6.8640
114    Perch  700.0    34.0    36.0    38.3 10.6091 6.7408
115    Perch  700.0    34.5    37.0    39.4 10.8350 6.2646
116    Perch  690.0    34.6    37.0    39.3 10.5717 6.3666
117    Perch  900.0    36.5    39.0    41.4 11.1366 7.4934
118    Perch  650.0    36.5    39.0    41.4 11.1366 6.0030
119    Perch  820.0    36.6    39.0    41.3 12.4313 7.3514
120    Perch  850.0    36.9    40.0    42.3 11.9286 7.1064
121    Perch  900.0    37.0    40.0    42.5 11.7300 7.2250
122    Perch 1015.0    37.0    40.0    42.4 12.3808 7.4624
123    Perch  820.0    37.1    40.0    42.5 11.1350 6.6300
124    Perch 1100.0    39.0    42.0    44.6 12.8002 6.8684
125    Perch 1000.0    39.8    43.0    45.2 11.9328 7.2772
126    Perch 1100.0    40.1    43.0    45.5 12.5125 7.4165
127    Perch 1000.0    40.2    43.5    46.0 12.6040 8.1420
128    Perch 1000.0    41.1    44.0    46.6 12.4888 7.5958
129     Pike  200.0    30.0    32.3    34.8  5.5680 3.3756
130     Pike  300.0    31.7    34.0    37.8  5.7078 4.1580
131     Pike  300.0    32.7    35.0    38.8  5.9364 4.3844
132     Pike  300.0    34.8    37.3    39.8  6.2884 4.0198
133     Pike  430.0    35.5    38.0    40.5  7.2900 4.5765
```

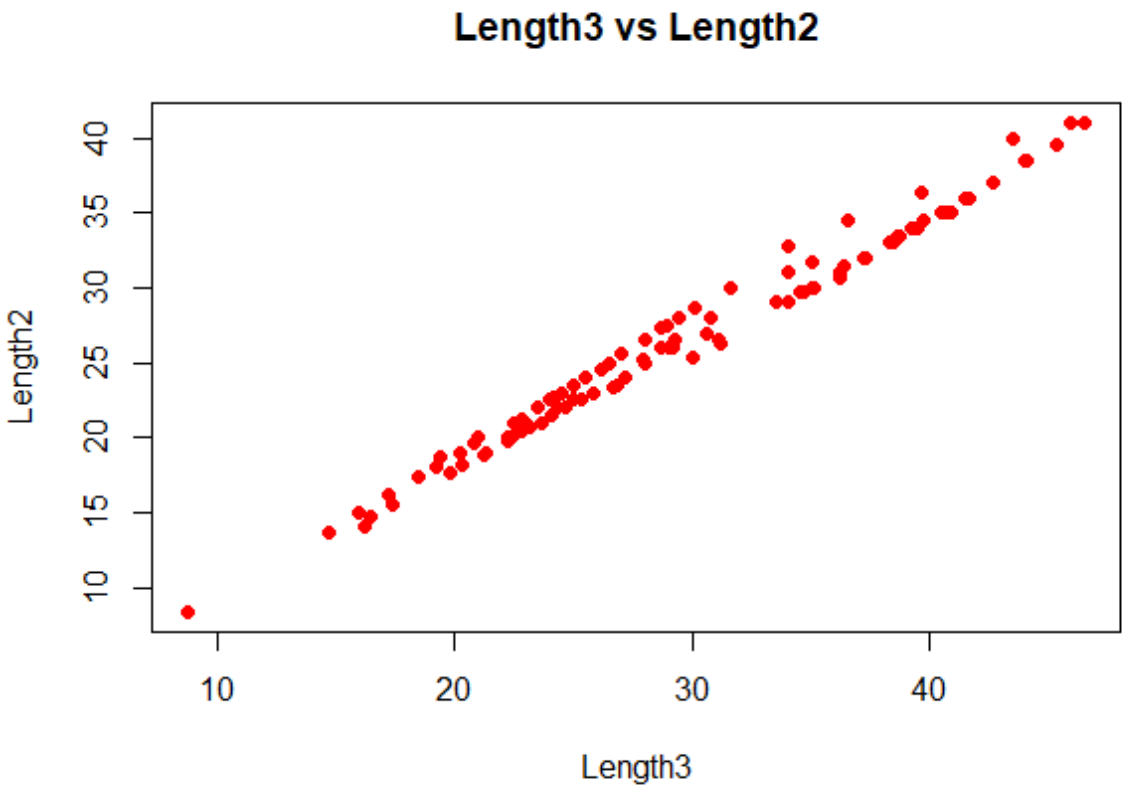
b. Plot the scatter graphs and check the relationship between (Length3) and (Weight, Length1, Length2, Width) columns.

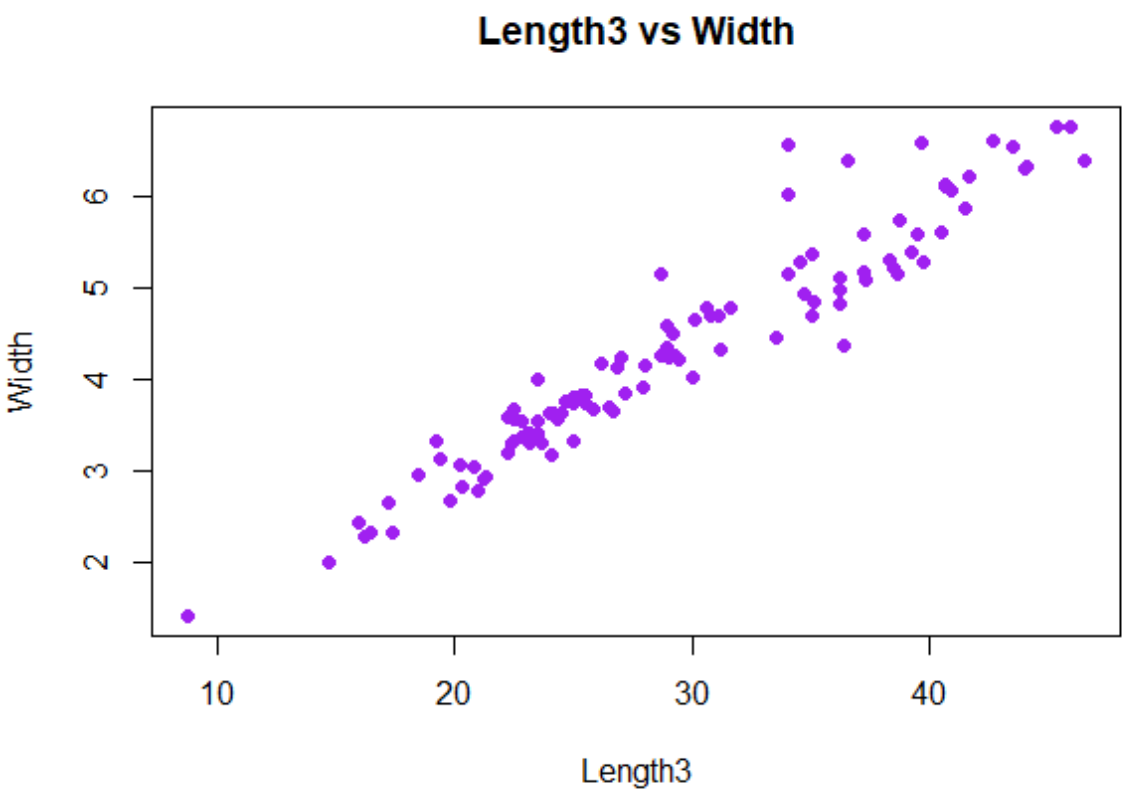
Code & Output:

```
> plot(TrainData$Length3, TrainData$Weight, main = "Length3 vs Weight", xlab = "Length3", ylab = "Weight", pch = 16, col = "blue")
> plot(TrainData$Length3, TrainData$Length1, main = "Length3 vs Length1",xlab = "Length3", ylab = "Length1", pch = 16, col = "cyan")
> plot(TrainData$Length3, TrainData$Length2, main = "Length3 vs Length2",xlab = "Length3", ylab = "Length2", pch = 16, col = "red")
> plot(TrainData$Length3, TrainData$Width, main = "Length3 vs Width",xlab = "Length3", ylab = "Width", pch = 16, col = "purple")
```






Length3 vs Length2
Length2
10
15
20
25
30
35
40
10
20
30
40
Length3


Length3 vs Width
Width
2
3
4
5
6
10
20
30
40
Length3

c. Perform multiple regression analysis on training dataset columns Length3 as Y and Weight, Length2, Length1, Width as X1, X2,X3,X4, using linear model function (lm).

Code & Output:

```
> multiple_lm_model <- lm(Length3 ~ Weight + Length2 + Length1 + Width, data = TrainData)
> summary(multiple_lm_model)

Call:
lm(formula = Length3 ~ Weight + Length2 + Length1 + Width, data = TrainData)

Residuals:
    Min      1Q  Median      3Q     Max
-2.5563 -0.7389  0.2022  0.7034  1.9223

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.596584   0.800584   0.745 0.457809
Weight       0.004177   0.001094   3.817 0.000228 ***
Length2      1.763292   0.381112   4.627 1.06e-05 ***
Length1     -0.648346   0.390316  -1.661 0.099653 .
Width       -0.776338   0.330281  -2.351 0.020596 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9701 on 106 degrees of freedom
Multiple R-squared:  0.9861,    Adjusted R-squared:  0.9856
F-statistic:  1886 on 4 and 106 DF,  p-value: < 2.2e-16
```

d. Predict the Length3 value using Testing dataset

Code & Output:

```
> predicted_length3 <- predict(multiple_lm_model, newdata = TestData)
> cat("Predicted Length3 values:\n", predicted_length3)
Predicted Length3 values:
 38.69708 40.44539 39.72205 41.53086 41.34507 43.64217 43.75499 43.35342 45.23772 45.28966 45.5857 45.35259 48.63178
49.14133 49.25638 49.09226 49.81443 36.31532 38.02102 38.9602 41.9373 42.82857 43.49642 48.15162 48.27121 48.69271 5
1.00171 52.28703 55.23135 59.62082 65.42547 72 71.79115 75.22181 11.06192 11.75847 11.87688 12.21917 12.45755 12.5582
4 13.12223 13.21334 13.51433 13.62182 13.94656 14.75043 15.73094 16.72302
```

e. Analyze the Testing result using predicted and actual value of the Length3 column data and calculate correlation between them.

Code & Output:

```
> actual_length3 <- TestData$Length3
> length3_comparison <- data.frame(Actual = actual_length3, Predicted = predicted_length3)
> print(length3_comparison)
    Actual Predicted
112   37.3  38.69708
113   39.0  40.44539
114   38.3  39.72205
115   39.4  41.53086
116   39.3  41.34507
117   41.4  43.64217
118   41.4  43.75499
119   41.3  43.35342
120   42.3  45.23772
121   42.5  45.28966
122   42.4  45.58570
123   42.5  45.35259
124   44.6  48.63178
125   45.2  49.14133
126   45.5  49.25638
127   46.0  49.09226
128   46.6  49.81443
129   34.8  36.31532
130   37.8  38.02102
131   38.8  38.96020
132   39.8  41.93730
133   40.5  42.82857
134   41.0  43.49642
135   45.5  48.15162
136   45.5  48.27121
137   45.8  48.69271
138   48.0  51.00171
139   48.7  52.28703
140   51.2  55.23135
141   55.1  59.62082
142   59.7  65.42547
143   64.0  72.00000
144   64.0  71.79115

> correlation <- cor(actual_length3, predicted_length3)
> cat("Correlation between Actual and Predicted Length3 values:", correlation, "\n")
Correlation between Actual and Predicted Length3 values: 0.9986846
```

f. Analyze the regression line with Residuals(line segment which represents the distance between y-value of the actual scatter plot points and the y values of the regression equation at those points) on a scatter plot.

Code & Output:

```
> residuals <- actual_length3 - predicted_length3
> plot(predicted_length3, residuals, main = "Residuals vs Predicted Length3",
+      xlab = "Predicted Length3", ylab = "Residuals", pch = 16, col = "turquoise")
> abline(h = 0, col ="purple")
```