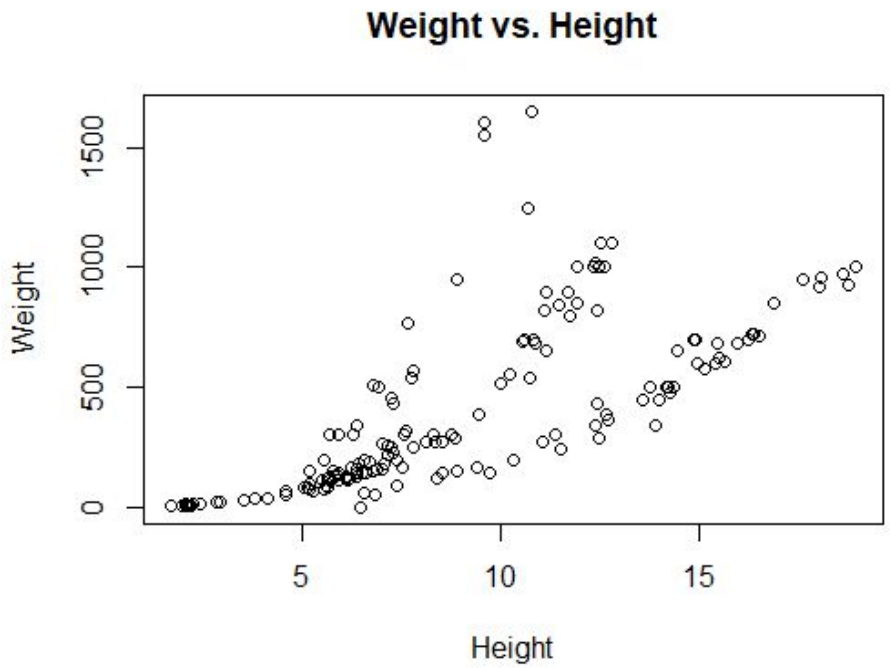
Name: Vedant Puranik
Roll No: 43152
Class: BE-9
Batch No: R9

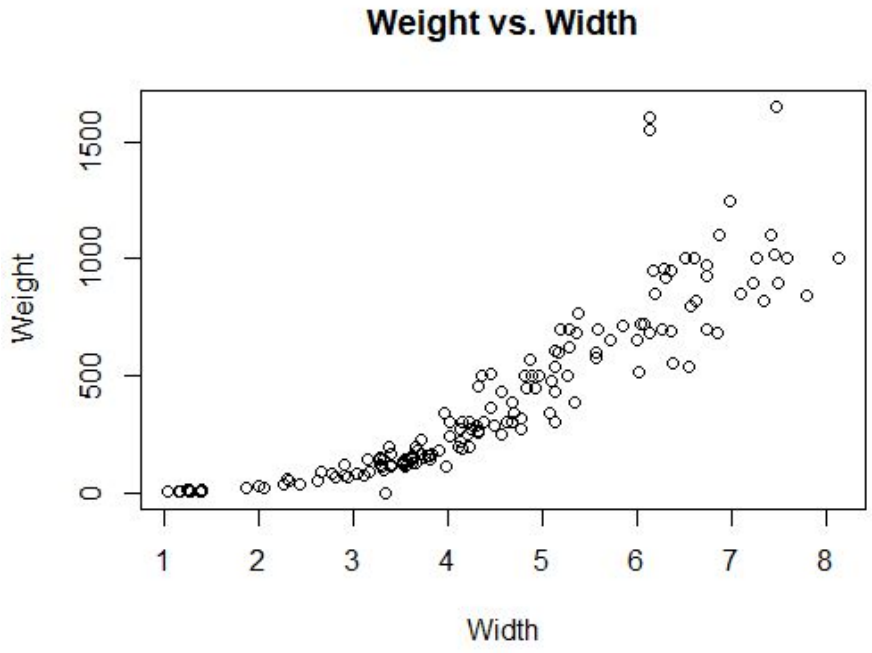
Assignment 6 Output

```
> library("tidyverse")
> library("Metrics")
> library("caret")
> dataset <- read.csv("Fish.csv")
> head(dataset)
  ï..Species Weight Length1 Length2 Length3  Height  Width
1      Bream    242    23.2    25.4    30.0 11.5200 4.0200
2      Bream    290    24.0    26.3    31.2 12.4800 4.3056
3      Bream    340    23.9    26.5    31.1 12.3778 4.6961
4      Bream    363    26.3    29.0    33.5 12.7300 4.4555
5      Bream    430    26.5    29.0    34.0 12.4440 5.1340
6      Bream    450    26.8    29.7    34.7 13.6024 4.9274
> glimpse(dataset)
Rows: 159
Columns: 7
$ ï..Species <chr> "Bream", "Bream", "Bream", "Bream", "Bream", "Bream", "Bream",
"Bream",...
$ Weight     <dbl> 242, 290, 340, 363, 430, 450, 500, 390, 450, 500, 475, 500, 500, 340, 6...
$ Length1    <dbl> 23.2, 24.0, 23.9, 26.3, 26.5, 26.8, 26.8, 27.6, 27.6, 28.5, 28.4, 28.7,...
$ Length2    <dbl> 25.4, 26.3, 26.5, 29.0, 29.0, 29.7, 29.7, 30.0, 30.0, 30.7, 31.0, 31.0,...
$ Length3    <dbl> 30.0, 31.2, 31.1, 33.5, 34.0, 34.7, 34.5, 35.0, 35.1, 36.2, 36.2, 36.2,...
$ Height     <dbl> 11.5200, 12.4800, 12.3778, 12.7300, 12.4440, 13.6024, 14.1795, 12.6700,...
$ Width      <dbl> 4.0200, 4.3056, 4.6961, 4.4555, 5.1340, 4.9274, 5.2785, 4.6900, 4.8438,...
> dim(dataset)
[1] 159   7
> #Checking null values
> sum_na <- sum(is.na(dataset))
> print(paste("NA Data: ", sum_na))
[1] "NA Data:  0"
> #Renaming Columns
> colnames(dataset)
[1] "ï..Species" "Weight"     "Length1"    "Length2"    "Length3"    "Height"     "Width"
> names(dataset)[names(dataset) == "Length1"] <- "VerticalLen"
> names(dataset)[names(dataset) == "Length2"] <- "HorizontalLen"
> names(dataset)[names(dataset) == "Length3"] <- "DiagonalLen"
> names(dataset)[names(dataset) == "ï..Species"] <- "Species"
> colnames(dataset)
```

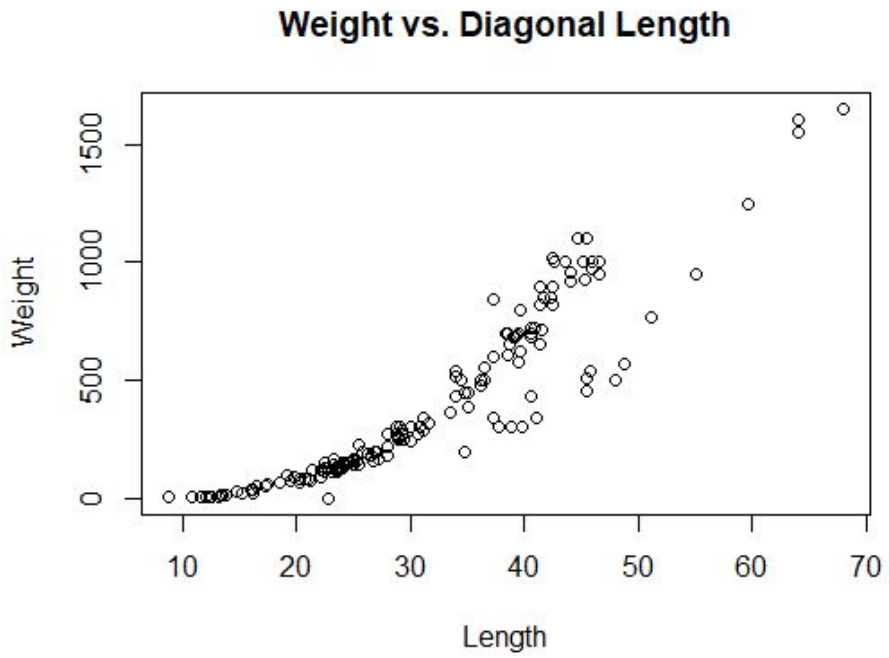
```
[1] "Species"       "Weight"        "VerticalLen"   "HorizontalLen" "DiagonalLen"
[6] "Height"        "Width"
> #Categorical to numeric
> dataset$Species
  [1] "Bream"     "Bream"     "Bream"     "Bream"     "Bream"     "Bream"     "Bream"
  [8] "Bream"     "Bream"     "Bream"     "Bream"     "Bream"     "Bream"     "Bream"
 [15] "Bream"     "Bream"     "Bream"     "Bream"     "Bream"     "Bream"     "Bream"
 [22] "Bream"     "Bream"     "Bream"     "Bream"     "Bream"     "Bream"     "Bream"
 [29] "Bream"     "Bream"     "Bream"     "Bream"     "Bream"     "Bream"     "Bream"
 [36] "Roach"     "Roach"     "Roach"     "Roach"     "Roach"     "Roach"     "Roach"
 [43] "Roach"     "Roach"     "Roach"     "Roach"     "Roach"     "Roach"     "Roach"
 [50] "Roach"     "Roach"     "Roach"     "Roach"     "Roach"     "Roach"     "Whitefish"
 [57] "Whitefish" "Whitefish" "Whitefish" "Whitefish" "Whitefish" "Parkki"    "Parkki"
 [64] "Parkki"    "Parkki"    "Parkki"    "Parkki"    "Parkki"    "Parkki"    "Parkki"
 [71] "Parkki"    "Parkki"    "Perch"     "Perch"     "Perch"     "Perch"     "Perch"
 [78] "Perch"     "Perch"     "Perch"     "Perch"     "Perch"     "Perch"     "Perch"
 [85] "Perch"     "Perch"     "Perch"     "Perch"     "Perch"     "Perch"     "Perch"
 [92] "Perch"     "Perch"     "Perch"     "Perch"     "Perch"     "Perch"     "Perch"
 [99] "Perch"     "Perch"     "Perch"     "Perch"     "Perch"     "Perch"     "Perch"
[106] "Perch"     "Perch"     "Perch"     "Perch"     "Perch"     "Perch"     "Perch"
[113] "Perch"     "Perch"     "Perch"     "Perch"     "Perch"     "Perch"     "Perch"
[120] "Perch"     "Perch"     "Perch"     "Perch"     "Perch"     "Perch"     "Perch"
[127] "Perch"     "Perch"     "Pike"      "Pike"      "Pike"      "Pike"      "Pike"
[134] "Pike"      "Pike"      "Pike"      "Pike"      "Pike"      "Pike"      "Pike"
[141] "Pike"      "Pike"      "Pike"      "Pike"      "Pike"      "Smelt"     "Smelt"
[148] "Smelt"     "Smelt"     "Smelt"     "Smelt"     "Smelt"     "Smelt"     "Smelt"
[155] "Smelt"     "Smelt"     "Smelt"     "Smelt"     "Smelt"
> dataset$Species <- as.numeric(as.factor(dataset$Species))
> dataset$Species
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 5 5 5 5 5 5 5 5 5
 [46] 5 5 5 5 5 5 5 5 5 5 7 7 7 7 7 7 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [91] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4
[136] 4 4 4 4 4 4 4 4 4 4 6 6 6 6 6 6 6 6 6 6 6 6 6
> table(dataset$Species)

 1  2  3  4  5  6  7
35 11 56 17 20 14  6
> #Finding relationship
> plot(dataset$Weight~dataset$Height,xlab="Height",ylab="Weight",
+      main="Weight vs. Height")
```




```
> plot(dataset$Weight~dataset$Width,xlab="Width",ylab="Weight",
+      main="Weight vs. Width")
```

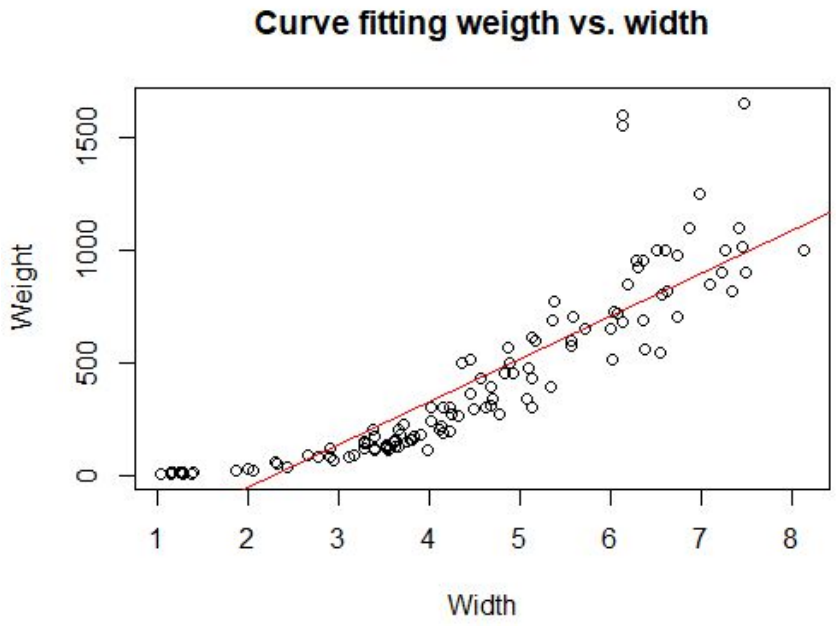



```
> plot(dataset$Weight~dataset$DiagonalLen,xlab="Length",ylab="Weight",
+      main="Weight vs. Diagonal Length")
```



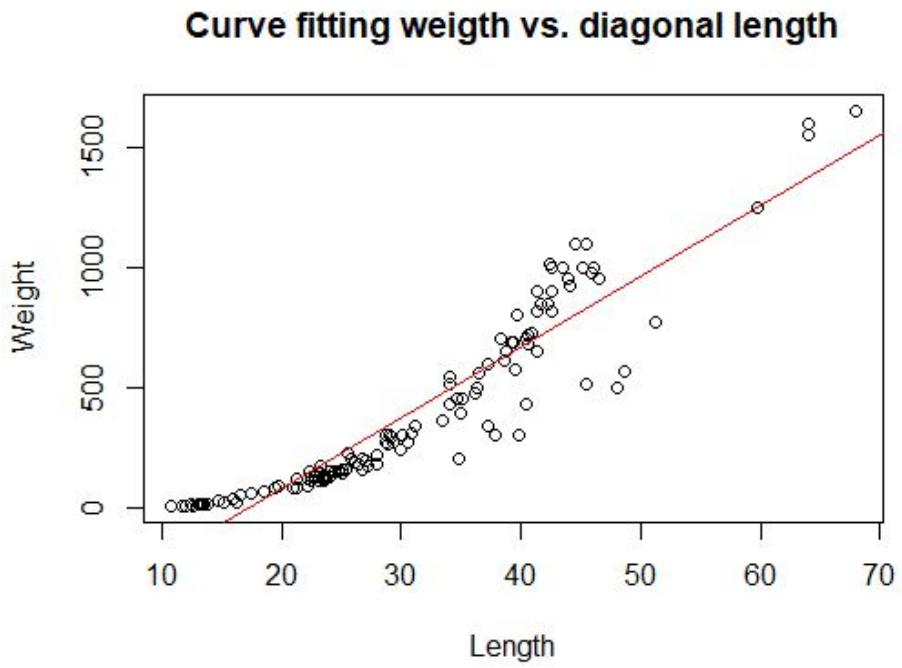

```
> #Split data into train and test
> set.seed(123)
> data.div <- dataset$Weight %>% createDataPartition(p=0.75,list=FALSE)
> train.data <- dataset[data.div,]
> test.data <- dataset[-data.div,]
> dim(train.data)
[1] 121   7
> dim(test.data)
[1] 38  7
> #Multivariate linear regression
> lin.multi.model <- lm(formula = Weight~Species+VerticalLen+HorizontalLen+
+                 DiagonalLen+Height+Width,data=train.data)
> summary(lin.multi.model)

Call:
lm(formula = Weight ~ Species + VerticalLen + HorizontalLen +
    DiagonalLen + Height + Width, data = train.data)

Residuals:
    Min      1Q  Median      3Q     Max
-316.47  -61.64  -15.39   72.76  348.50

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    -687.11      58.50 -11.746  < 2e-16 ***
```

```
Species          35.60       9.72   3.663  0.00038 ***
VerticalLen      33.54      44.60   0.752  0.45363
HorizontalLen    62.73      48.81   1.285  0.20136
DiagonalLen     -64.92      20.11  -3.228  0.00163 **
Height           56.43      11.49   4.912 3.04e-06 ***
Width           -37.83      24.51  -1.543  0.12549
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 118.7 on 114 degrees of freedom
Multiple R-squared:  0.9051,  Adjusted R-squared:  0.9001
F-statistic: 181.2 on 6 and 114 DF,  p-value: < 2.2e-16

> #Univariate Linear regression (width)
> lin.uni.width <- lm(formula=Weight~Width,data=train.data)
> summary(lin.uni.width)

Call:
lm(formula = Weight ~ Width, data = train.data)

Residuals:
   Min      1Q  Median      3Q     Max
-273.82 -112.40  -50.43   74.78  865.63

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -433.45      44.93  -9.647   <2e-16 ***
Width         190.07       9.45  20.113   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 179.8 on 119 degrees of freedom
Multiple R-squared:  0.7727,  Adjusted R-squared:  0.7708
F-statistic: 404.6 on 1 and 119 DF,  p-value: < 2.2e-16

> plot(train.data$Weight~train.data$Width,xlab='Width',ylab='Weight',
+      main='Curve fitting weigth vs. width')
> abline(lin.uni.width,col='red')
```



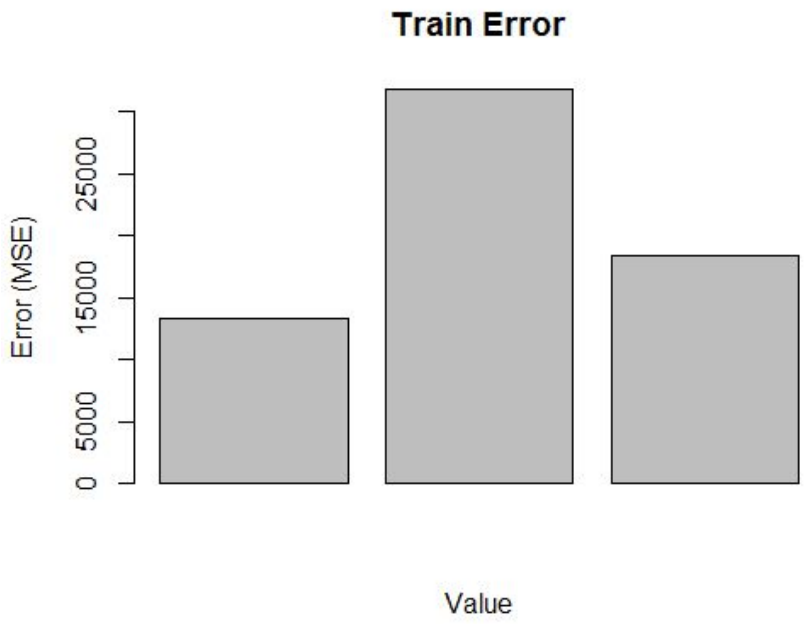

```
> #Univariate Linear Regression (DiagonalLen)
> lin.uni.diagonal <- lm(formula=Weight~DiagonalLen,data=train.data)
> summary(lin.uni.diagonal)

Call:
lm(formula = Weight ~ DiagonalLen, data = train.data)

Residuals:
   Min      1Q  Median      3Q     Max
-399.66  -70.46  -20.98  111.22  300.28

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -511.336     35.018  -14.60   <2e-16 ***
DiagonalLen   29.396      1.047   28.09   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 136.5 on 119 degrees of freedom
Multiple R-squared:  0.8689,  Adjusted R-squared:  0.8678
F-statistic: 788.9 on 1 and 119 DF,  p-value: < 2.2e-16

>#Univariate Linear Regression (DiagonalLen)
> lin.uni.diagonal <- lm(formula=Weight~DiagonalLen,data=train.data)
> summary(lin.uni.diagonal)
```

```
Call:
lm(formula = Weight ~ DiagonalLen, data = train.data)

Residuals:
    Min      1Q  Median      3Q     Max
-399.66  -70.46  -20.98  111.22  300.28

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -511.336     35.018  -14.60   <2e-16 ***
DiagonalLen   29.396      1.047   28.09   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 136.5 on 119 degrees of freedom
Multiple R-squared:  0.8689,  Adjusted R-squared:  0.8678
F-statistic: 788.9 on 1 and 119 DF,  p-value: < 2.2e-16

> plot(train.data$Weight~train.data$DiagonalLen,xlab='Length',ylab='Weight',
+      main='Curve fitting weigth vs. diagonal length')
> abline(lin.uni.diagonal,col='red')
```

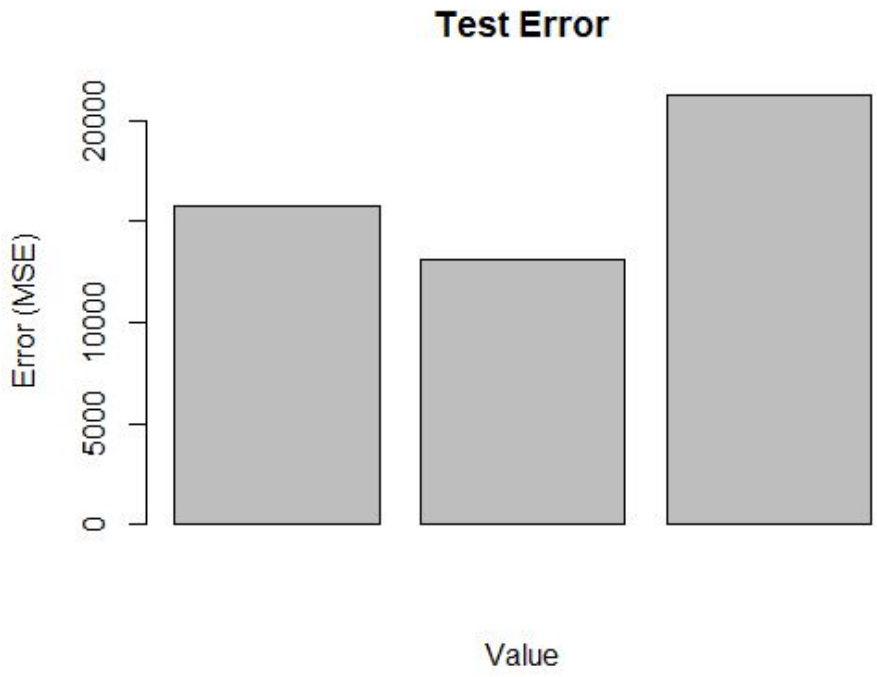



```
> #Prediction (train data)
> train.pred.multi <- predict(lin.multi.model,train.data)
> train.pred.width <- predict(lin.uni.width,train.data)
> train.pred.diagonal <- predict(lin.uni.diagonal,train.data)
```

```
> #Prediction (test data)
> test.pred.multi <- predict(lin.multi.model,test.data)
> test.pred.width <- predict(lin.uni.width,test.data)
> test.pred.diagonal <- predict(lin.uni.diagonal,test.data)
> #MSE calculation (train)
> train.mse.multi <- mse(train.pred.multi,train.data$Weight)
> train.mse.multi
[1] 13278.22
> train.mse.width <- mse(train.pred.width,train.data$Weight)
> train.mse.width
[1] 31799.81
> train.mse.diagonal <- mse(train.pred.diagonal,train.data$Weight)
> train.mse.diagonal
[1] 18337.1
> #MSE calculation (test)
> test.mse.multi <- mse(test.pred.multi,test.data$Weight)
> test.mse.multi
[1] 15763.67
> test.mse.width <- mse(test.pred.width,test.data$Weight)
> test.mse.width
[1] 13123.48
> test.mse.diagonal <- mse(test.pred.diagonal,test.data$Weight)
> test.mse.diagonal
[1] 21241.92
> trainMSE <- c(train.mse.multi,train.mse.width,train.mse.diagonal)
> testMSE <- c(test.mse.multi,test.mse.width,test.mse.diagonal)
> #Plotting MSE results
> barplot(trainMSE,width=0.01,xlab='Value',ylab='Error (MSE)',main='Train Error')
```




```
> barplot(testMSE,width=0.01,xlab='Value',ylab='Error (MSE)',main='Test Error')
```




```
> #R2 Scores for models
> R2_multi <- R2(test.pred.multi,test.data$Weight)
> R2_multi
[1] 0.8465074
> R2_width <- R2(test.pred.width,test.data$Weight)
```

```
> R2_width
[1] 0.8626647
> R2_diagonal <- R2(test.pred.diagonal,test.data$Weight)
> R2_diagonal
[1] 0.7961032
> #KFold Cross validation
> set.seed(123)
> train.control <- trainControl(method = "cv", number = 7)
> model <- train(Weight ~ ., data = train.data, method = "lm",trControl = train.control)
> print(model)
Linear Regression

121 samples
  6 predictor

No pre-processing
Resampling: Cross-Validated (7 fold)
Summary of sample sizes: 102, 103, 104, 105, 105, 105, ...
Resampling results:

  RMSE      Rsquared  MAE
  121.4829  0.902804  90.3715

Tuning parameter 'intercept' was held constant at a value of TRUE
>
```