Heart weight prediction model

Pushkar Shinde

2024-02-12

So Let's get started!!!

Step 1: Loading the dataset!

Using Cat dataset available in MASS package.

```r
install.packages("MASS")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```r
library(MASS)
```

Loading dataset

```r
data("cats")
```

Using head() to view content of data

```r
head(cats,10)
```

```
##    Sex Bwt Hwt
## 1    F 2.0 7.0
## 2    F 2.0 7.4
## 3    F 2.0 9.5
## 4    F 2.1 7.2
## 5    F 2.1 7.3
## 6    F 2.1 7.6
## 7    F 2.1 8.1
## 8    F 2.1 8.2
## 9    F 2.1 8.3
## 10   F 2.1 8.5
```

As the data is inbuilt we don't need to read it separately, we will analyze the data set and then we can split it into two parts train and test.

Step 2: Analyse the data set!

Identifying the number of rows and columns

```r
dim(cats)
```

```
## [1] 144   3
```

```r
nrow(cats)
```

```
## [1] 144
```

```r
ncol(cats)
```

```
## [1] 3
```

Checking for missing values

```r
sum(is.na(cats))
```

```
## [1] 0
```

```r
which(is.na(cats))
```

```
## integer(0)
```

Checking the structure and summary of the dataset

```r
str(cats)
```

```
## 'data.frame':    144 obs. of  3 variables:
##  $ Sex: Factor w/ 2 levels "F","M": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Bwt: num  2 2 2 2.1 2.1 2.1 2.1 2.1 2.1 2.1 ...
##  $ Hwt: num  7 7.4 9.5 7.2 7.3 7.6 8.1 8.2 8.3 8.5 ...
```

```r
summary(cats)
```

```
##  Sex         Bwt             Hwt       
##  F:47   Min.   :2.000   Min.   : 6.30  
##  M:97   1st Qu.:2.300   1st Qu.: 8.95  
##         Median :2.700   Median :10.10  
##         Mean   :2.724   Mean   :10.63  
##         3rd Qu.:3.025   3rd Qu.:12.12  
##         Max.   :3.900   Max.   :20.50  
```

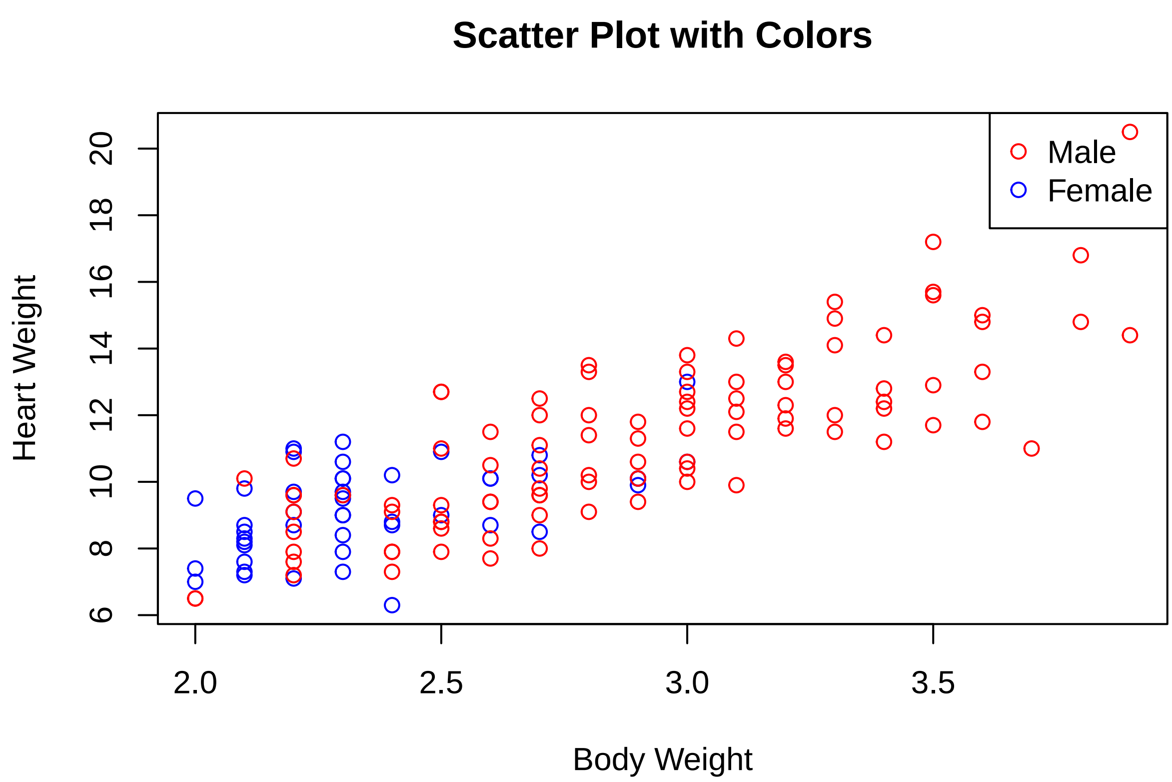
Checking class of variables

```r
sapply(cats,class)
```

```
##       Sex       Bwt       Hwt 
##  "factor" "numeric" "numeric" 
```

Finding the correlation between Body 'weight' and 'Heart weight'

```r
plot(cats$Bwt, cats$Hwt, col = ifelse(cats$Sex == "F", "blue", "red"),
     xlab = "Body Weight", ylab = "Heart Weight", main = "Scatter Plot with Colors")
legend("topright", legend = c("Male", "Female"), col = c("red", "blue"), pch = 1)
```



Replacing blanks or spaces with NA, even though we have none!

```
cats[cats==" "] <- NA
```

Step 3: visualizing the data!

Finding correlations between 'Heart weight' and 'sex' of cats using visual plots. Installing package ggplot2

```
install.packages("ggplot2")
```

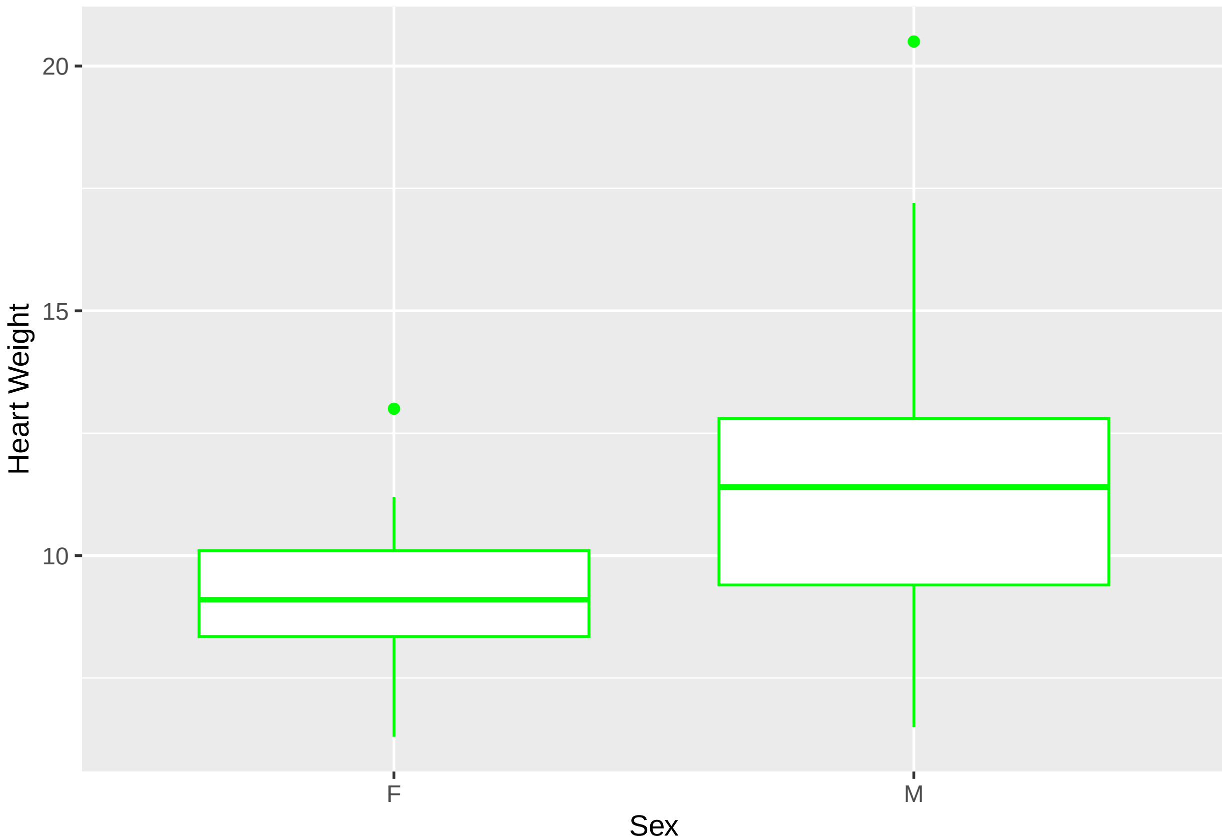
```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

Load the ggplot2 library for plotting.

```
library(ggplot2)
```

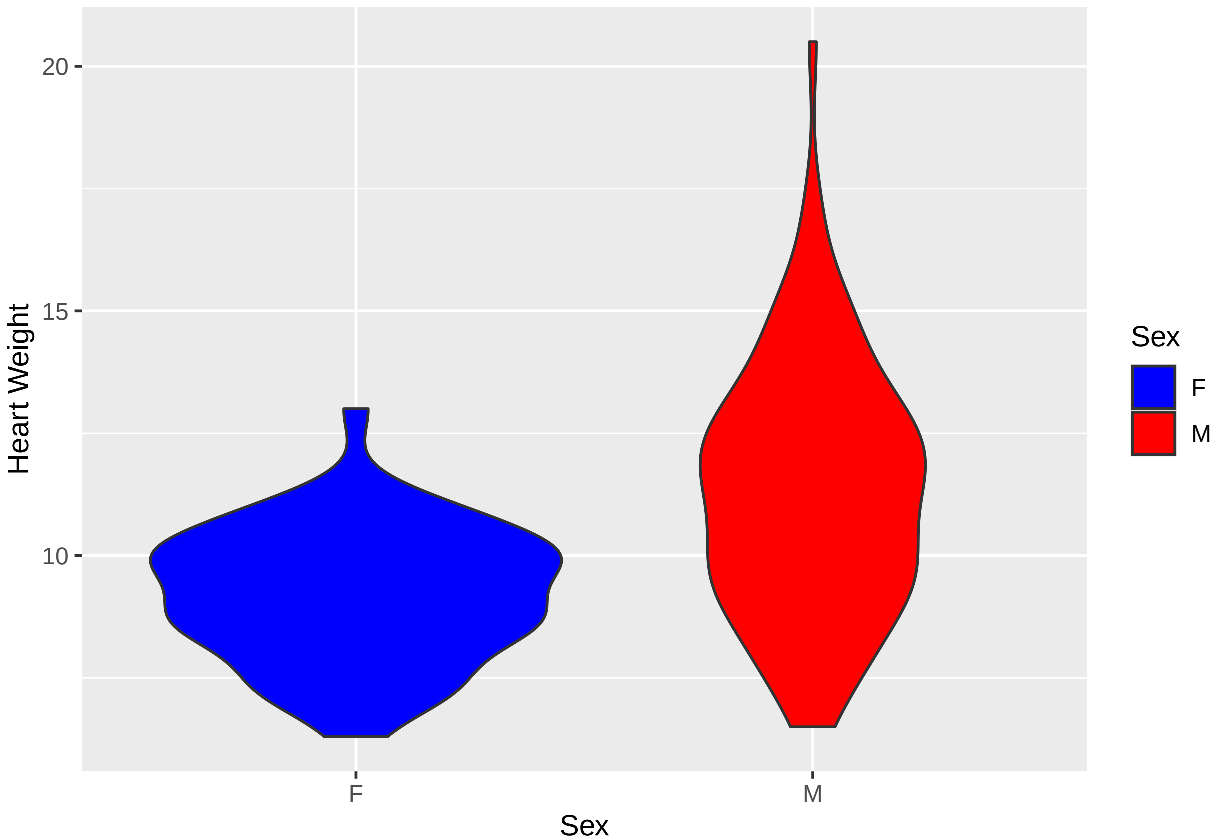
Create a box plot and violin plot to visualize the relationship between 'Hwt' and 'Sex'.

```
ggplot(cats, aes(x = Sex, y = Hwt)) +
  geom_boxplot(color = "green") +  # Adjust outline color
  labs(x = "Sex", y = "Heart Weight")
```



Box plot

```
ggplot(cats, aes(x = Sex, y = Hwt, fill = Sex)) +
  geom_violin() +
  labs(x = "Sex", y = "Heart Weight") +
  scale_fill_manual(values = c("blue", "red"))
```



Violin plot

Creating a new data set 'cats2' so we don't alter the original cats' dataset

```
cats2 <- cats
dim(cats2)
```

```
## [1] 144   3
```

```
head(cats2,10)
```

```
##    Sex Bwt Hwt
## 1    F 2.0 7.0
## 2    F 2.0 7.4
## 3    F 2.0 9.5
## 4    F 2.1 7.2
## 5    F 2.1 7.3
## 6    F 2.1 7.6
## 7    F 2.1 8.1
## 8    F 2.1 8.2
## 9    F 2.1 8.3
## 10   F 2.1 8.5
```

Step 4: Splitting the dataset into Test and Train data

Generating a random sample of indices representing the training set from the dataset 'cats2'

```
train_ind <- sample.int(n = nrow(cats2), size = floor(0.75 * nrow(cats2)), replace = FALSE)
```

Splitting the dataset cats2 into a training set and a test set based on the indices generated earlier

```r
train <- cats2[train_ind,]
test <- cats2[-train_ind,]
```

Observing the test and train data

```r
head(train,10)
```

```
##     Sex Bwt  Hwt
## 33    F 2.4  8.8
## 58    M 2.2 10.7
## 137   M 3.6 13.3
## 13    F 2.2  7.1
## 41    F 2.7 10.2
## 40    F 2.7  8.5
## 31    F 2.4  6.3
## 90    M 2.8 10.2
## 132   M 3.5 12.9
## 139   M 3.6 15.0
```

```r
head(test,10)
```

```
##    Sex Bwt  Hwt
## 4    F 2.1  7.2
## 18   F 2.2 11.0
## 23   F 2.3  9.0
## 24   F 2.3  9.5
## 35   F 2.5  9.0
## 42   F 2.7 10.8
## 45   F 2.9 10.1
## 47   F 3.0 13.0
## 54   M 2.2  8.5
## 56   M 2.2  9.6
```

```r
dim(train)
```

```
## [1] 108   3
```

```r
dim(test)
```

```
## [1] 36  3
```

Step 5: Finding Correlation between variables

Attaching train data

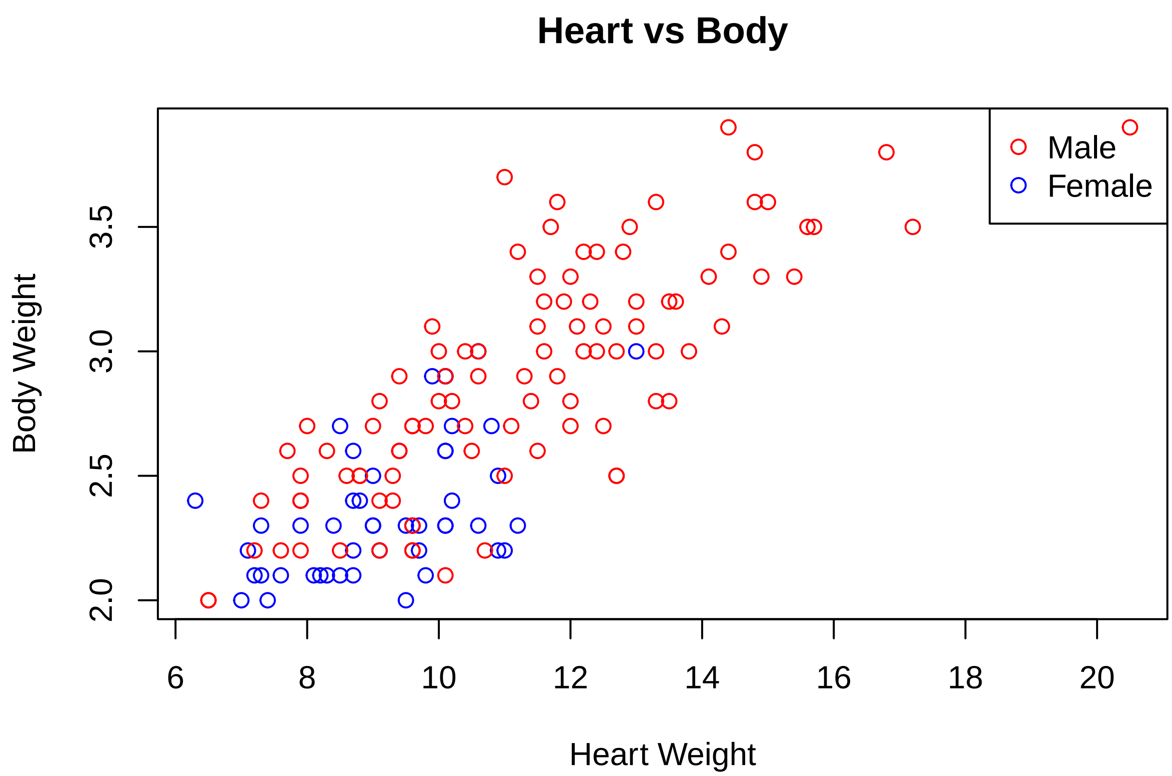
```r
attach(train)
```

Plotting the relationship between 'Hwt' and 'Bwt' and then perform a correlation test between the two variables. Plotting Heart Weight against Body Weight.

```r
# Defining colors based on the Sex variable
colors <- ifelse(cats2$Sex == "F", "blue", "red")

# Plot with colors
plot(cats2$Hwt, cats2$Bwt, col = colors, xlab = "Heart Weight", ylab = "Body Weight", main = "Heart vs 

# Adding legend
legend("topright", legend = c("Male", "Female"), col = c("red", "blue"), pch = 1)
```



Perform a correlation test

```
correlation_test <- cor.test(cats2$Hwt, cats2$Bwt)
```

Print the correlation test result

```
print(correlation_test)
```

```
##
##  Pearson's product-moment correlation
##
## data:  cats2$Hwt and cats2$Bwt
## t = 16.119, df = 142, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7375682 0.8552122
## sample estimates:
##       cor
## 0.8041274
```

There is a strong positive correlation of ~ 80%

Step 6: Performing regression analysis

Conducting linear regression analysis

```
linear_model1 <- with(train, lm(Hwt ~ Bwt + Sex))
```

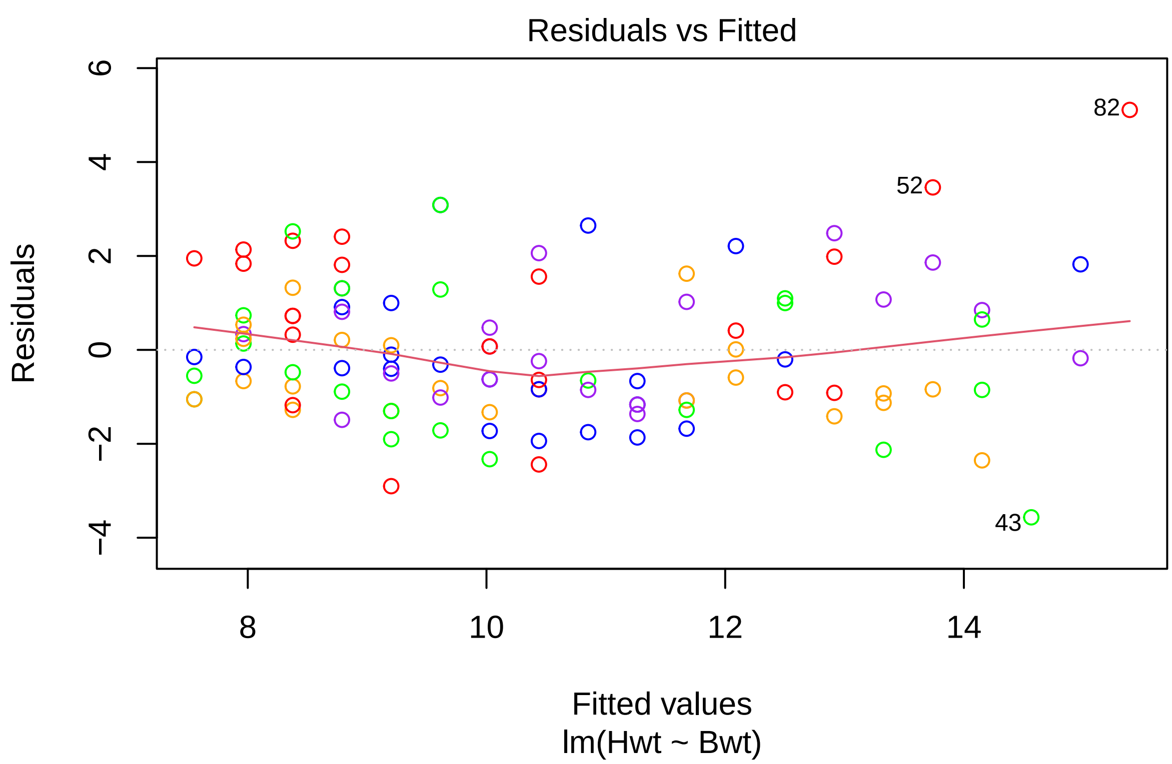
As the variable sex is not significant let's drop it.

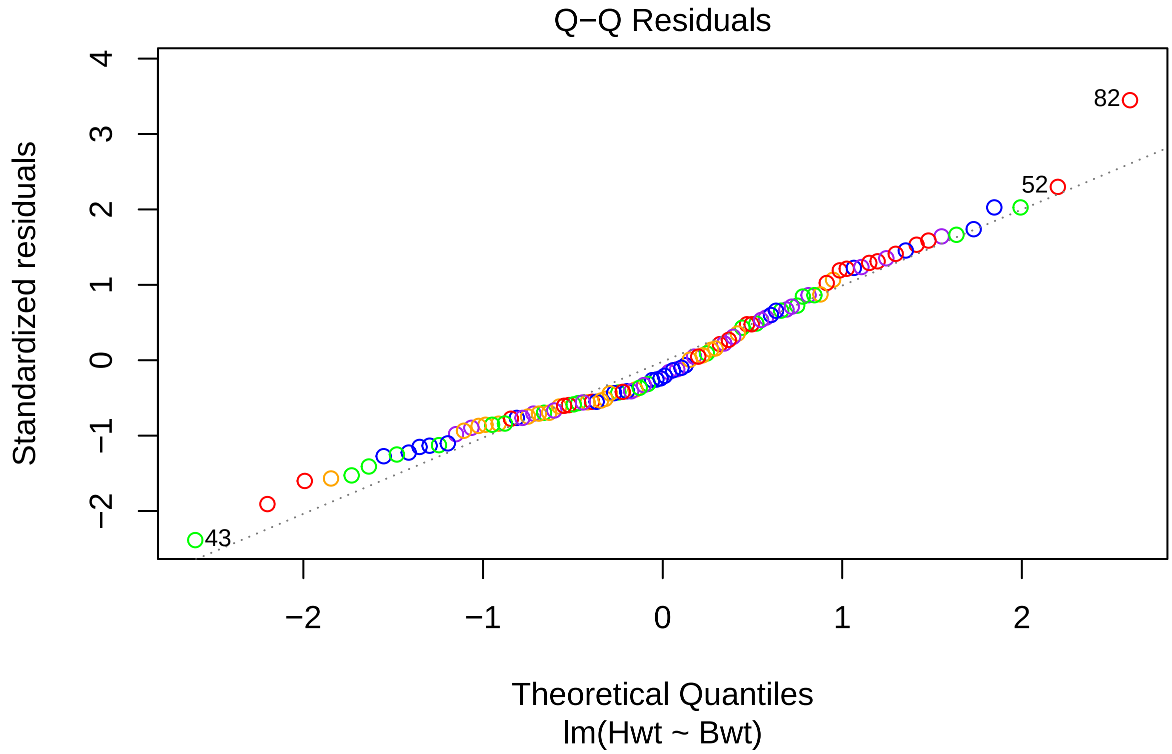
```
linear_model1 <- with(train, lm(Hwt ~ Bwt))
summary(linear_model1)
```

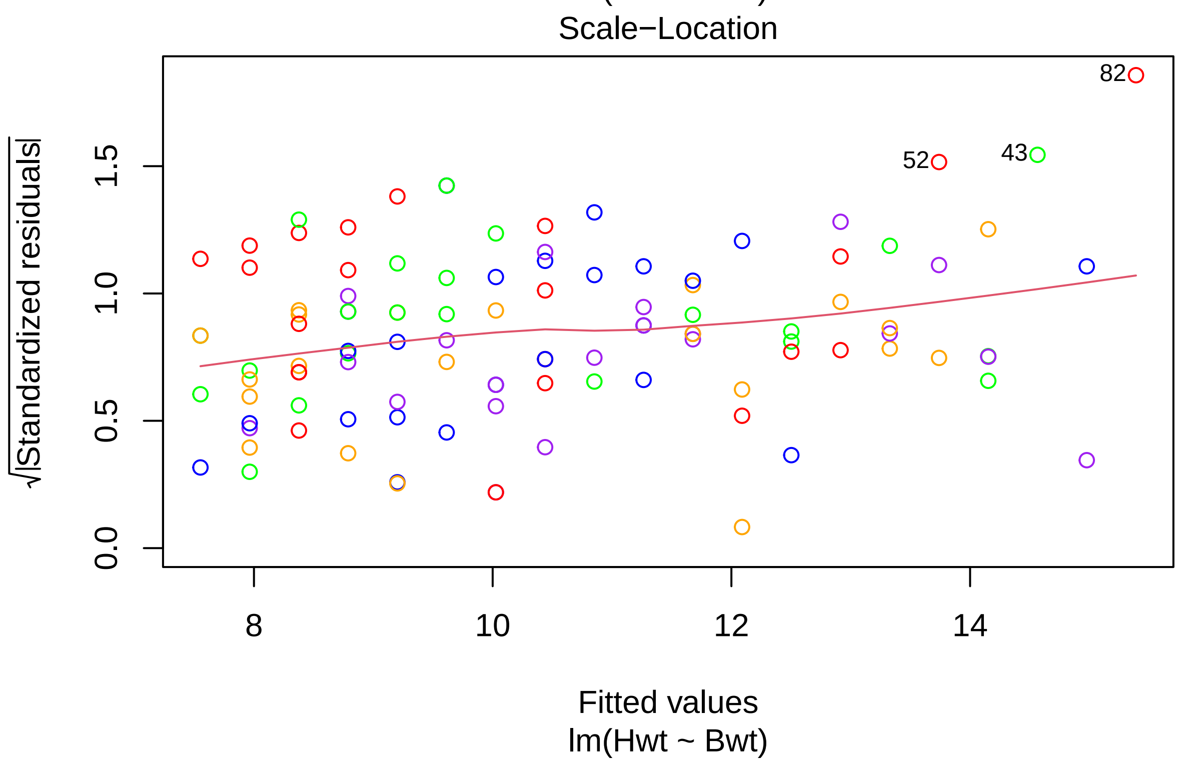
```
##
## Call:
## lm(formula = Hwt ~ Bwt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5649 -1.0578 -0.3391  1.0045  5.1099
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.6997     0.8108  -0.863     0.39
## Bwt           4.1256     0.2957  13.951   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.531 on 106 degrees of freedom
## Multiple R-squared:  0.6474, Adjusted R-squared:  0.6441
## F-statistic: 194.6 on 1 and 106 DF,  p-value: < 2.2e-16
```

Plot of linear_model,it helps to understand the accuracy and to identify the hidden behavior of data.

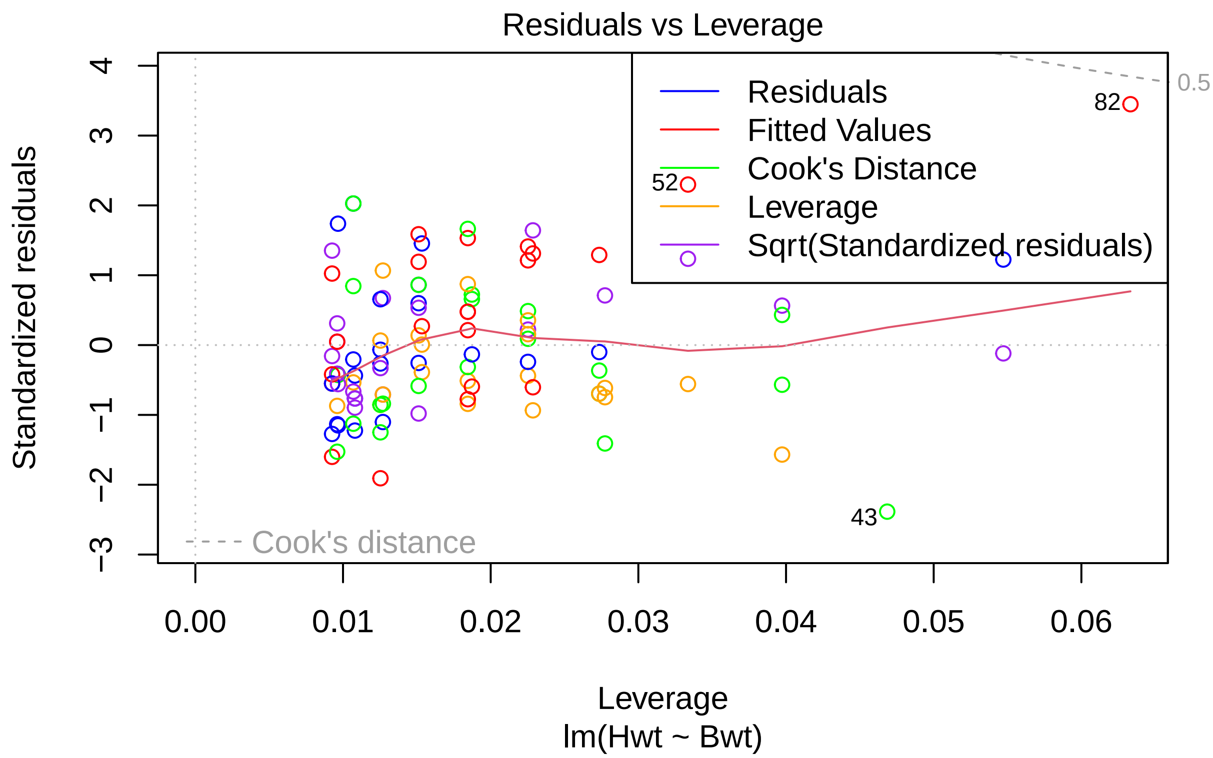
```
plot(linear_model1, col = c("blue", "red", "green", "orange", "purple"))
```








```r
# Adding legend
legend("topright", legend = c("Residuals", "Fitted Values", "Cook's Distance", "Leverage", "Sqrt(Standa
       col = c("blue", "red", "green", "orange", "purple"), lty = 1)
```



Analyzing all 4 plots…

Step 7: Yay! Let's predict

Predicting values of 'Hwt' using the linear regression mode

```
Hwt_Predicted <- predict(linear_model1, test, method = "class")
```

Inspecting the predicted and actual values of 'Hwt' side by side using a data frame view.

```
head(data.frame(Hwt_Predicted, test$Hwt),10)
```

```
##    Hwt_Predicted test.Hwt
## 4       7.964002      7.2
## 18      8.376560     11.0
## 23      8.789119      9.0
## 24      8.789119      9.5
## 35      9.614237      9.0
## 42     10.439354     10.8
## 45     11.264472     10.1
## 47     11.677030     13.0
## 54      8.376560      8.5
## 56      8.376560      9.6
```

Adding the Predicted Values to the Test dataset.

```
test <- cbind(test, Hwt_Predicted)
head(test,10)
```

```
##    Sex Bwt  Hwt Hwt_Predicted
## 4    F 2.1  7.2      7.964002
## 18   F 2.2 11.0      8.376560
## 23   F 2.3  9.0      8.789119
```

```
## 24   F 2.3  9.5      8.789119
## 35   F 2.5  9.0      9.614237
## 42   F 2.7 10.8     10.439354
## 45   F 2.9 10.1     11.264472
## 47   F 3.0 13.0     11.677030
## 54   M 2.2  8.5      8.376560
## 56   M 2.2  9.6      8.376560
```

Step 8: Let's proceed with calculating the prediction accuracy

Extract Predicted and Actual Values

```r
predicted_values <- test$Hwt_Predicted
actual_values <- test$Hwt
```

Calculating Mean Absolute Error (MAE)

```r
mae <- mean(abs(predicted_values - actual_values))
```

Calculating Root Mean Squared Error (RMSE)

```r
rmse <- sqrt(mean((predicted_values - actual_values)^2))
```

Calculating Root Mean Squared Error (RMSE)

```r
rmse <- sqrt(mean((predicted_values - actual_values)^2))
```

Calculating R-squared (R^2)

```r
rsquared <- cor(predicted_values, actual_values)^2
```

Printing the evaluation metrics

```r
cat("Mean Absolute Error (MAE):", mae, "\n")
```

```
## Mean Absolute Error (MAE): 1.00081
```

```r
cat("Root Mean Squared Error (RMSE):", rmse, "\n")
```

```
## Root Mean Squared Error (RMSE): 1.209546
```

```r
cat("R-squared (R²):", rsquared, "\n")
```

```
## R-squared (R²): 0.6396466
```

Our prediction model has done great!

Thanks for your patiance!!