Descriptive Analysis of the Data

The purpose of the descriptive analysis is to get a better understanding of the raw data before the actual regression analysis is started. In general, this means determining how clean the data is, its size and scale, and take note of any possible patterns or inconsistencies that may impact the regression analysis.

```
data <- read.csv("fish-market-data.csv", fileEncoding="UTF-8-BOM")
```

Structure

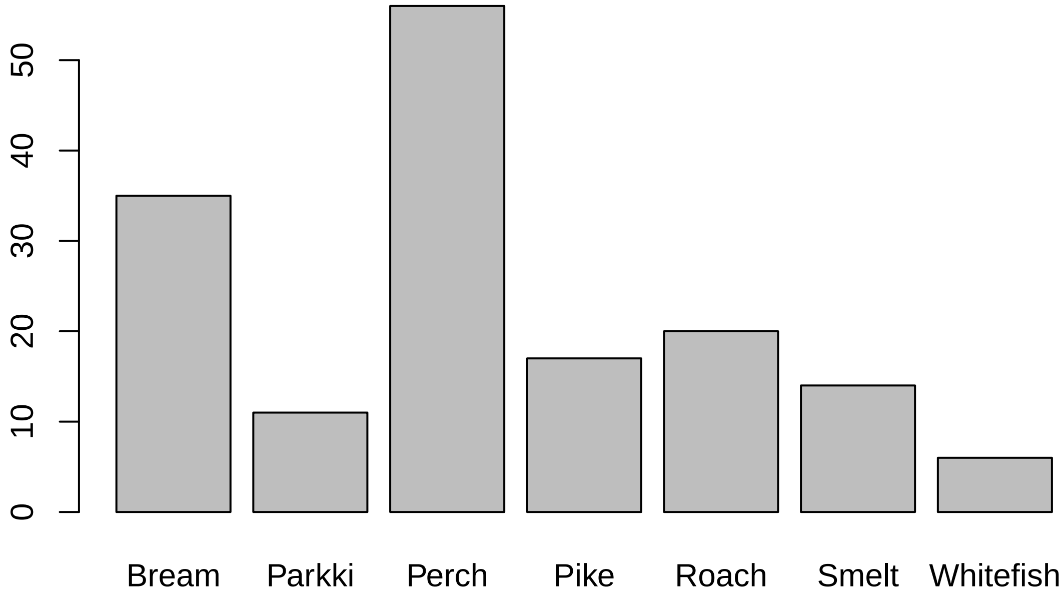
```
str(data)
```

```
## ’data.frame':    159 obs. of  7 variables:
##  $ Species: chr  "Bream" "Bream" "Bream" "Bream" ...
##  $ Weight : num  242 290 340 363 430 450 500 390 450 500 ...
##  $ Length1: num  23.2 24 23.9 26.3 26.5 26.8 26.8 27.6 27.6 28.5 ...
##  $ Length2: num  25.4 26.3 26.5 29 29 29.7 29.7 30 30 30.7 ...
##  $ Length3: num  30 31.2 31.1 33.5 34 34.7 34.5 35 35.1 36.2 ...
##  $ Height : num  11.5 12.5 12.4 12.7 12.4 ...
##  $ Width  : num  4.02 4.31 4.7 4.46 5.13 ...
```

There are 7 total variables, 6 of which are numbers and one of which is an indicator variable.

Qualitative Variables

```
barplot(table(data$Species))
```



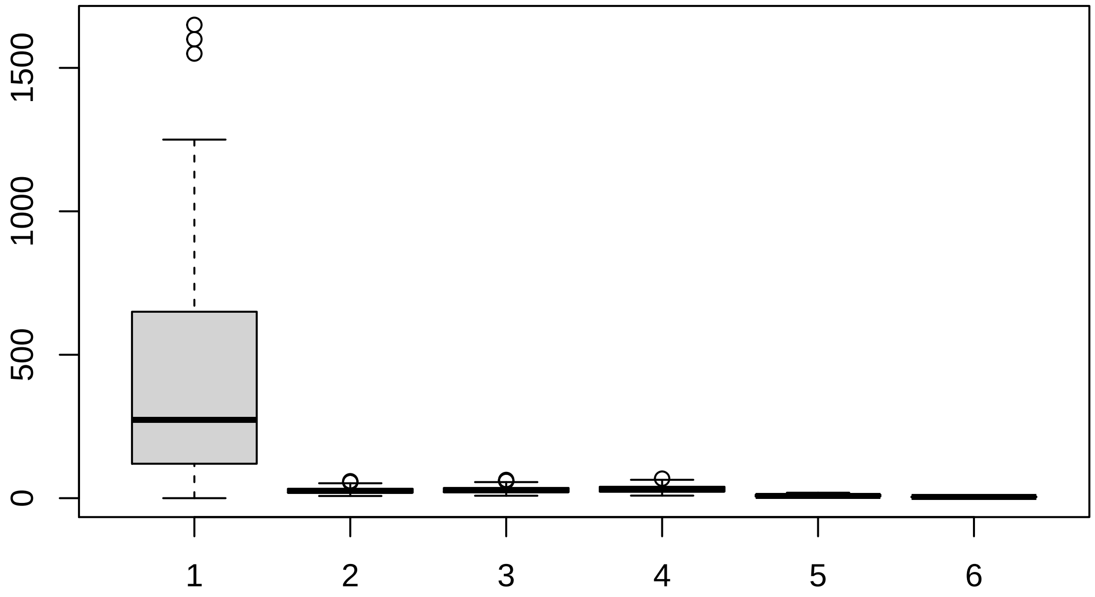
There are more Perch samples than any other type of fish, while there are very few Whitefish samples.

Summary Statistics

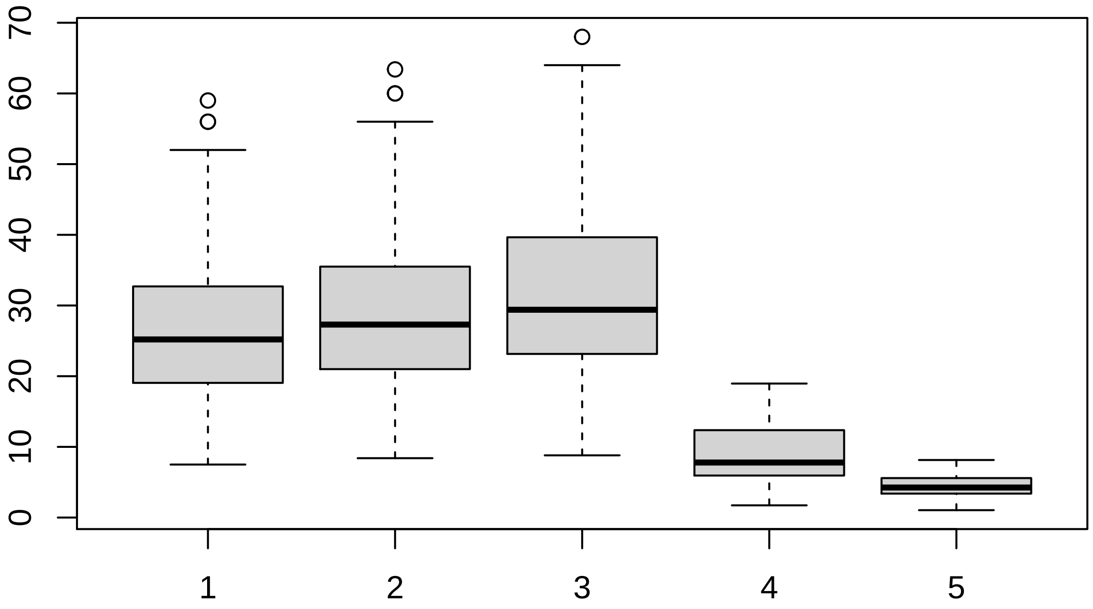
```
summary(data)
```

```
##    Species              Weight          Length1         Length2     
##  Length:159         Min.   :   0.0   Min.   : 7.50   Min.   : 8.40  
##  Class :character   1st Qu.: 120.0   1st Qu.:19.05   1st Qu.:21.00  
##  Mode  :character   Median : 273.0   Median :25.20   Median :27.30  
##                     Mean   : 398.3   Mean   :26.25   Mean   :28.42  
##                     3rd Qu.: 650.0   3rd Qu.:32.70   3rd Qu.:35.50  
##                     Max.   :1650.0   Max.   :59.00   Max.   :63.40  
##     Length3          Height           Width      
##  Min.   : 8.80   Min.   : 1.728   Min.   :1.048  
##  1st Qu.:23.15   1st Qu.: 5.945   1st Qu.:3.386  
##  Median :29.40   Median : 7.786   Median :4.248  
##  Mean   :31.23   Mean   : 8.971   Mean   :4.417  
##  3rd Qu.:39.65   3rd Qu.:12.366   3rd Qu.:5.585  
##  Max.   :68.00   Max.   :18.957   Max.   :8.142
```

```
boxplot(data$Weight, data$Length1, data$Length2, data$Length3, data$Height, data$Width)
```



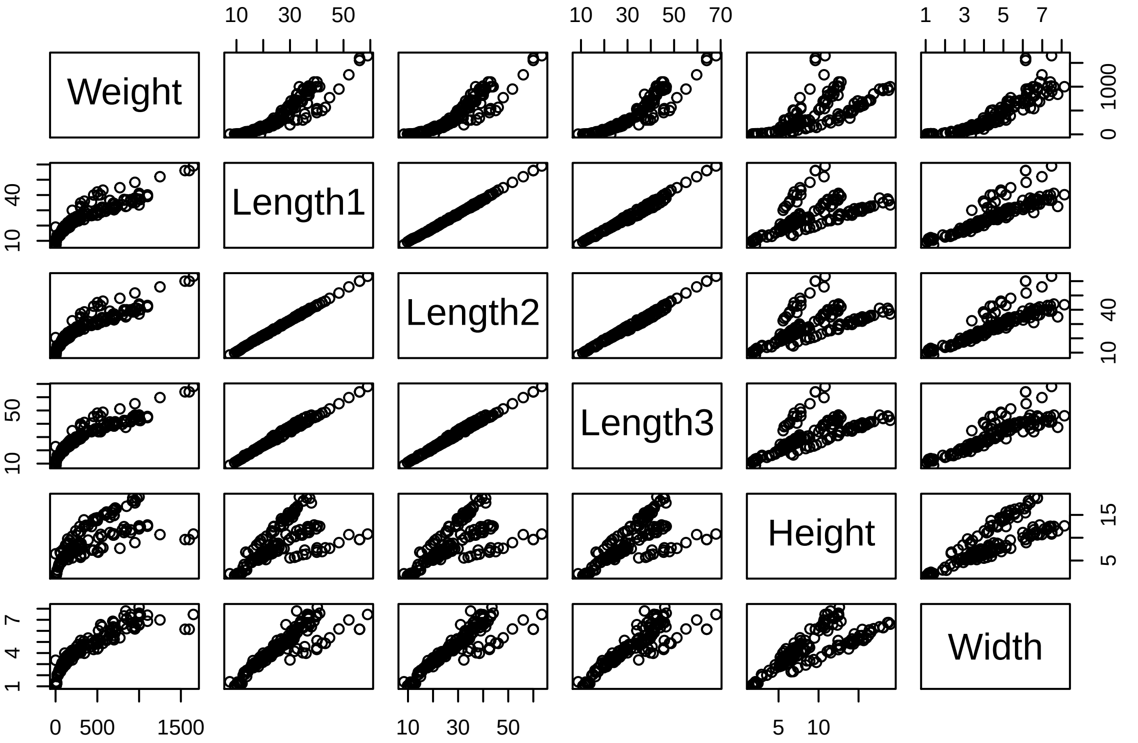
```r
boxplot(data$Length1, data$Length2, data$Length3, data$Height, data$Width)
```



Clearly weight is in a different unit than the other predictors. It will make sense to scale the data for comparability of the regression coefficients. If Length 1 2 and 3 are the same measurement units as Width and Height, then this group may be measuring the length of the fish in slightly different ways. Weight seems to be highly variable and heavily skewed to the right with large outliers

Summary Statistics

```
pairs(data[,c(2:7)])
```



Looking at the pairs plot, there appears to be some challenges ahead. Assuming weight is the response variable variable, it does appear to have a relation with every other predictor. However, the best relationship does not appear to be linear. This means that the data may have to be transformed to reflect a linear relationship between the response and predictors. Length 1, Length 2, and Length 3 all appear to be impacted by multicollinearity. Width and Height may have a the same issue, but have different 'groups' of correlation which may be explained by the categorical variable Species.

```
data <- read.csv("fish-market-data.csv", fileEncoding="UTF-8-BOM")
```

Unique Data Point Introduction

A datapoint can be introduced which is an outlier and has high leverage. The reason is that these data points often exist in real world data due to error or incorrect collection, influence outside of the model, or it may be legitimate. It is up to the statistician to make a judgement call and decide on whether the data point should be kept or discarded.

```
data <- rbind(data.frame(Species = 'Perch',
                         Weight = 2400,
                         Length1= 47.9,
                         Length2 = 46.2,
                         Length3 = 45.8,
                         Height = 17.11,
                         Width = 8.2), data)
```

Data Cleaning

Manually browsing the data, it seems pretty clean. Some issues came up in the descriptive analysis though. In the summary data, one or more observations have zero weight.

```
data[data$Weight == 0, ]
```

```
##    Species Weight Length1 Length2 Length3 Height  Width
## 42   Roach      0      19    20.5    22.8 6.4752 3.3516
```

Clearly, a fish cannot have zero Wight. Since its other attributes seem fine, I don't think removing the observation is necessary. A better option would be to interpolate for Wight given that the Species type is a Roach. Weight has a similar relationship to all other variables, so ordering by one of these and using it as a basis for linear interpolation should be fine. Even though we see currently see that the relationship is not quite linear, interpolating in this way should be fine and not significantly impact the model results.

```
roach_data <- data[data$Species == 'Roach', ]
rownames(roach_data) <- 1:nrow(roach_data)
zero_index <- as.numeric(rownames(roach_data[roach_data$Weight == 0, ]))
pred_val <- roach_data[c(zero_index), 3]

roach_data <- roach_data[-c(zero_index),]
roach_lm <- lm(Weight ~ Length1, data = roach_data)

inter_value <- predict(roach_lm, data.frame(Length1 = pred_val))
data[data$Weight == 0, 2] <- inter_value
```

From the descriptive analysis it was also evident that the data is in different units. To make the regression analysis simpler a unit normal scaling can be performed.

```r
scaled_data <- as.data.frame(scale(data[,-1], center = TRUE, scale = TRUE))
```

This is also a good time to setup our indicator variable species.

```r
indicator_data <- fastDummies::dummy_cols(data[’Species’], select_columns = "Species")
indicator_data$Species <- NULL
indicator_data$Species_Bream <- NULL
```

Combine the scaled and indicator data.

```r
scaled_data <- data.frame(indicator_data, scaled_data)
data <- data.frame(indicator_data, data)
data$Species <- NULL
```

Output the new dataframe as a .csv so it can be used in the regression analysis.

```r
write.csv(data, ’cleaned_data_2.csv’, row.names = FALSE)
write.csv(scaled_data, ’cleaned_data_scaled_2.csv’, row.names = FALSE)
```

Checking for Multicollinearity

Multicollinearity is defined as linear or near linear dependence between predictors. This can impact the usefulness of the linear regression model because it reduces or undermines the statistical significance of the predictor.

```
library(faraway)

data1 <- read.csv("cleaned_data_2.csv", fileEncoding="UTF-8-BOM")
data <- read.csv("cleaned_data_scaled_2.csv", fileEncoding="UTF-8-BOM")
```

From the descriptive analysis (pairs plot), we saw that the predictors Length1, Length2, and Length3 were all highly correlated. Height and Width also looked highly correlated with with he other predictors as well, but formed groups of linear relationships. We can quantify the multicollinearty present in the model using the VIF (Variance Inflation Factor). But first, lets view a summary of our initial model.

```
model <- lm(Weight ~ . - 1, data = data)
summary(model)
```

```
##
## Call:
## lm(formula = Weight ~ . - 1, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.61452 -0.16026  0.00386  0.11097  0.99209
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## Species_Parkki    -0.041129   0.103127  -0.399  0.69060
## Species_Perch     -0.006328   0.062118  -0.102  0.91899
## Species_Pike      -0.565196   0.179374  -3.151  0.00197 **
## Species_Roach      0.054568   0.077854   0.701  0.48445
## Species_Smelt      0.613500   0.097630   6.284 3.45e-09 ***
## Species_Whitefish  0.106220   0.120320   0.883  0.37876
## Length1            3.484141   0.670441   5.197 6.57e-07 ***
## Length2           -0.986473   1.196934  -0.824  0.41116
## Length3           -1.648915   0.688777  -2.394  0.01791 *
## Height             0.453491   0.107867   4.204 4.50e-05 ***
## Width             -0.095669   0.105431  -0.907  0.36566
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.2805 on 149 degrees of freedom
## Multiple R-squared:  0.9263, Adjusted R-squared:  0.9208
## F-statistic: 170.1 on 11 and 149 DF,  p-value: < 2.2e-16
```

Notice that the regression coefficients for Length1 is negative. This goes against our intuition, since it seems reasonable to think that the greater the length of the fish, the more it is going to weigh (regardless of what is actually being measured). Also note that these predictors do not have great significance. These issues both strongly indicate that multicollinearity is present.

```r
vif(model)
```

```
## Warning in vif.lm(model): No intercept term detected. Results may surprise.
```

```
##    Species_Parkki     Species_Perch      Species_Pike     Species_Roach
##          1.486632          2.794973          6.950804          1.540472
##     Species_Smelt Species_Whitefish           Length1           Length2
##          1.695738          1.103812        908.203782       2894.694859
##           Length3            Height             Width
##        958.560486         23.509296         22.459362
```

The VIF for the predictors shows that Length1, Length2, and Length3 are a major concern. They are almost completely linearly dependent. This makes sense given that these measurements are most likely measuring the length of the fish in three, almost identical ways. For this reason, we are justified in only keeping one of the lengths.

```r
data$Length2 = NULL
data$Length3 = NULL
model <- lm(Weight ~ . - 1, data = data)
summary(model)
```

```
##
## Call:
## lm(formula = Weight ~ . - 1, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.65161 -0.17949 -0.03852  0.10274  2.45330
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## Species_Parkki     0.17810    0.10766   1.654   0.1002
## Species_Perch      0.09727    0.05547   1.754   0.0815 .
## Species_Pike      -1.05407    0.17669  -5.966 1.66e-08 ***
## Species_Roach      0.04869    0.07811   0.623   0.5340
## Species_Smelt      0.76853    0.10558   7.279 1.72e-11 ***
## Species_Whitefish  0.05582    0.13874   0.402   0.6880
## Length1            1.20600    0.12182   9.900  < 2e-16 ***
## Height             0.12014    0.05354   2.244   0.0263 *
## Width             -0.10868    0.11825  -0.919   0.3595
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3243 on 151 degrees of freedom
## Multiple R-squared:  0.9001, Adjusted R-squared:  0.8942
## F-statistic: 151.2 on 9 and 151 DF,  p-value: < 2.2e-16
```

```r
vif(model)
```

```
## Warning in vif.lm(model): No intercept term detected. Results may surprise.
##    Species_Parkki     Species_Perch      Species_Pike     Species_Roach 
##          1.212207          1.667420          5.045785          1.159955 
##     Species_Smelt Species_Whitefish           Length1            Height 
##          1.483688          1.097952         22.433565          4.333418 
##             Width 
##         21.136621
```

We can see that removing these predictors has fixed the major issues multicollinearity was causing. The estimated regression coefficient for Length1 is now positive and has a significant linear relationship with the response. Note that Length1 and Width still have high VIF (Large VIF). This is likely due to some interaction with the indicator variables. Since Length1 and Width appear to be key variables, they can be left in for now. Hopefully after the model selection process is complete, the VIF's are more reasonable.

```
data1$Length2 <- NULL
data1$Length3 <- NULL

write.csv(data1, 'cleaned_data_3.csv', row.names = FALSE)
write.csv(data, 'cleaned_data_scaled_3.csv', row.names = FALSE)
```

Risidual Analysis

Five important assumptions need to hold so that the regression model can be useful hypothesis testing and predication. These are:

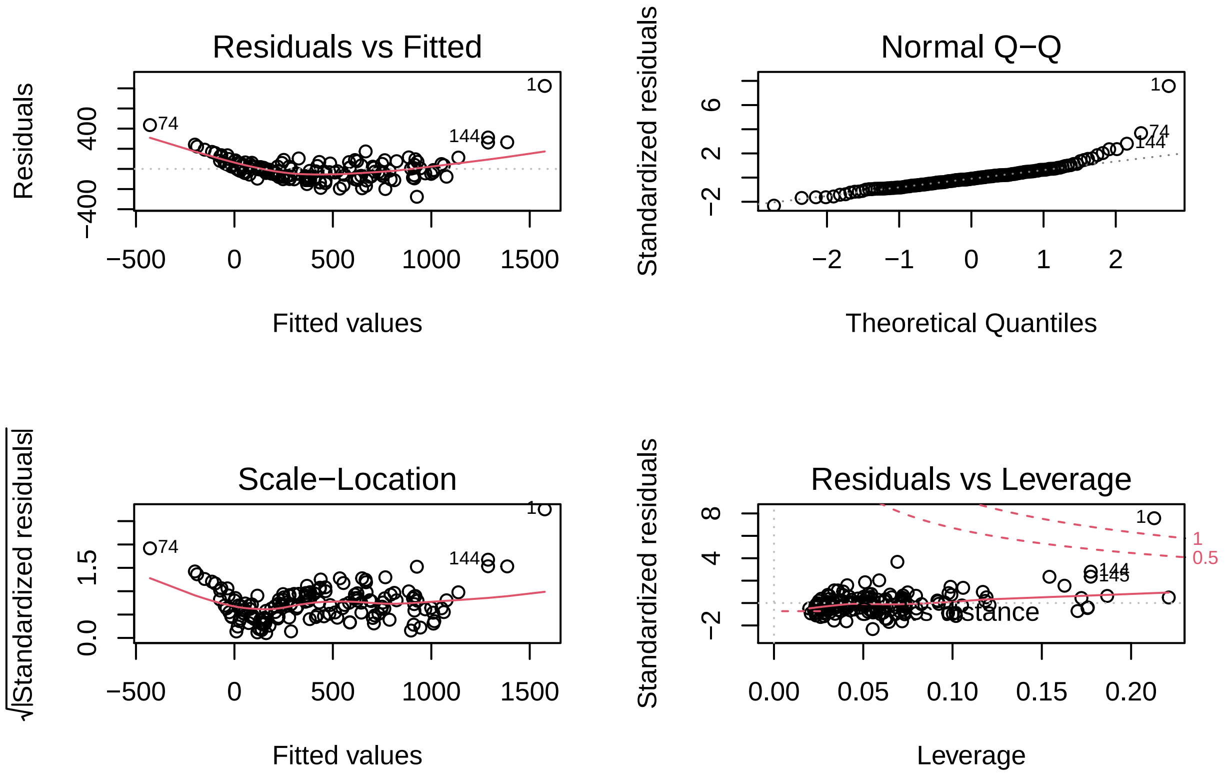
1. The relationship between the response y and the regression is linear (at least approximately).
2. The error term ϵ has zero mean.
3. The error term ϵ has constant variance σ^2.
4. The errors are uncorrelated.
5. The errors are normally distributed.

```
data <- read.csv("cleaned_data_3.csv", fileEncoding="UTF-8-BOM")
```

Lets use the risidual plots using standardized residuals so that we can compare the current state of the model with these assumptions.

```
model <- lm(Weight ~ ., data = data)

par(mfrow = c(2, 2))
plot(model)
```



Addressing Outliers

Observation 1 (the unique data point we added) immediately strikes us as having a huge risidual that makes it an outliar from all other data points. It also has significant leverage, pulling the linear regression line away from the best fit for the other data points. In this case, the fish Weight is disproportionately large from its other attributes like Height and Length. There are many possible causes for this (a faulty measurement scale, a fisherman disproportionately inflating the weight to get a better price, simple data entry error, or even the fish swallowing some object). However, since we have seen how consistent the proportionality of fish dimensions is to weight, it is almost impossible that this could be a legitimate observation. For this reason, removing the observation is justified.

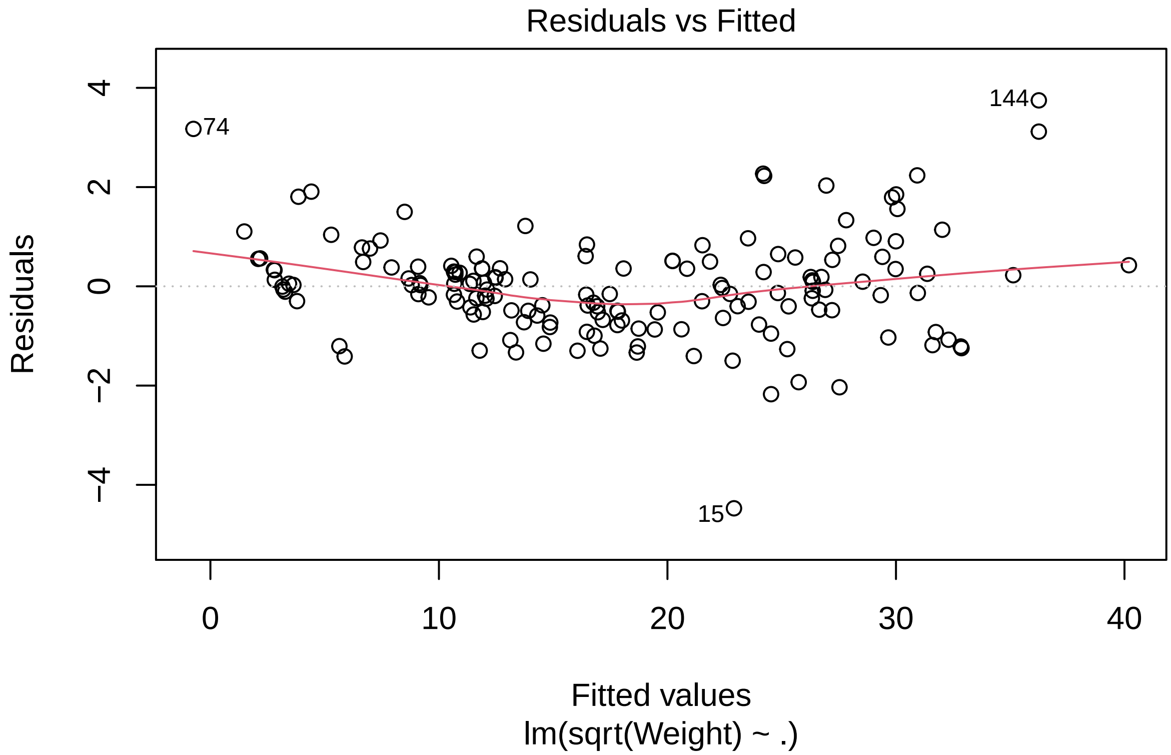
```
data <- data[-c(1),]
model <- lm(Weight ~ ., data = data)
```

Risiduals vs Fitted

Here, assumption **1. The relationship between the response y and the regression is linear** is violated (we want to see a linear pattern between the Risiduals and Fitted values). This is not surprising since we observed a non linear pattern between the predictors and response in the pairs plot.

```
model_transformed <- lm(sqrt(Weight) ~ ., data = data)

plot(model_transformed, which = 1)
```



Taking the square root of the response seems to produce the best result compared to other transformations of the response like the natural logarithm. Drawing a horizontal line at 0 seems reasonable. This satisfies 1. The relationship between the response y and the regression is linear. To confirm that we now have an improved linear relationship, we can compare the R^2 of the model.

```r
model <- lm(Weight ~ ., data = data)
summary(model)
```

```
## 
## Call:
## lm(formula = Weight ~ ., data = data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -229.21  -55.05   -7.04   33.25  397.14 
## 
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)    
## (Intercept)       -746.4762    86.2013  -8.660 7.19e-15 ***
## Species_Parkki      64.2364    48.8466   1.315  0.19051    
## Species_Perch       40.6409    82.9492   0.490  0.62489    
## Species_Pike      -254.4729   126.0128  -2.019  0.04524 *  
## Species_Roach       26.4221    77.7535   0.340  0.73447    
## Species_Smelt      299.6009    92.3743   3.243  0.00146 ** 
## Species_Whitefish   42.4582    79.4437   0.534  0.59383    
## Length1             37.8338     4.0004   9.457  < 2e-16 ***
## Height              14.0338    13.2228   1.061  0.29026    
## Width                0.8559    24.4172   0.035  0.97208    
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 96.67 on 149 degrees of freedom
## Multiple R-squared:  0.931,  Adjusted R-squared:  0.9268 
## F-statistic: 223.2 on 9 and 149 DF,  p-value: < 2.2e-16
```

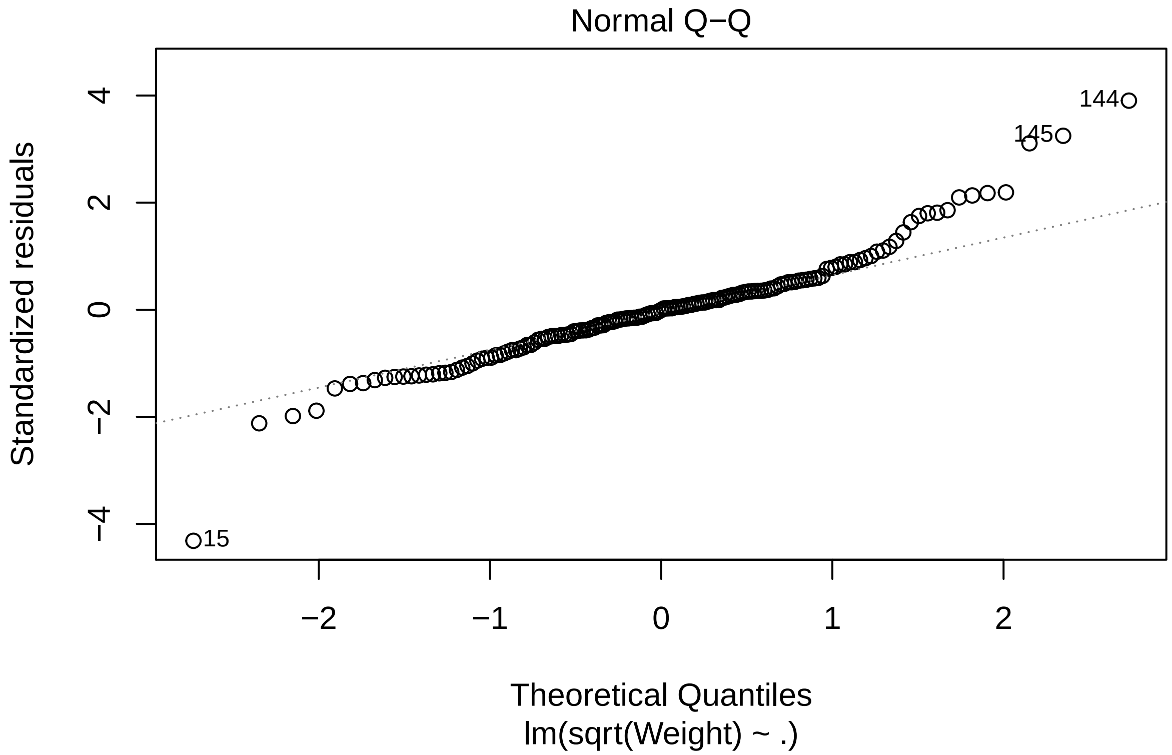
```r
summary(model_transformed)
```

```
## 
## Call:
## lm(formula = sqrt(Weight) ~ ., data = data)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -4.4731 -0.5234 -0.0057  0.4190  3.7476 
## 
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)    
## (Intercept)       -8.88830    0.94432  -9.412  < 2e-16 ***
## Species_Parkki     0.69560    0.53511   1.300 0.195635    
## Species_Perch      0.73702    0.90870   0.811 0.418618    
## Species_Pike      -2.35110    1.38045  -1.703 0.090628 .  
## Species_Roach      0.42972    0.85178   0.504 0.614656    
## Species_Smelt      2.37680    1.01195   2.349 0.020151 *  
## Species_Whitefish  1.55532    0.87029   1.787 0.075951 .  
## Length1            0.64245    0.04382  14.660  < 2e-16 ***
## Height             0.55811    0.14485   3.853 0.000173 ***
## Width              1.00205    0.26749   3.746 0.000256 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.059 on 149 degrees of freedom
## Multiple R-squared:  0.9873, Adjusted R-squared:  0.9865
## F-statistic:  1287 on 9 and 149 DF,  p-value: < 2.2e-16
```

Performing this transformation has significantly improved the R^2 of the model from 0.9309509 to 0.9872994. Overall, we can see that **1. The relationship between the response y and the regression is linear** and **2. The error term ϵ has zero mean.** have been satisfied.

Normal Q-Q

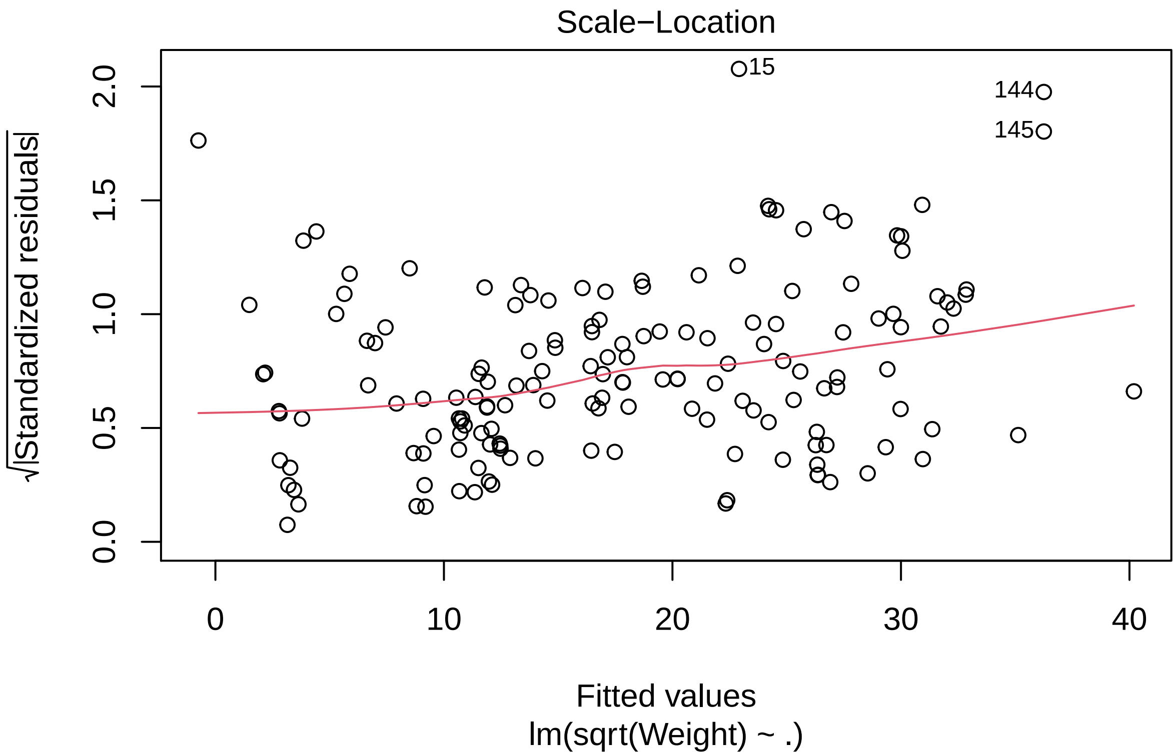
```r
plot(model_transformed, which = 2)
```



The normality assumption is is adequately met after the transformation. The majority of the data follows an x=y diagonal line from +/-2 SD. The one negative result of the square root transformation is that the risidual of observation 15 was magnified. We can see later whether this is a cause for concern based on its Cook's D value. Overall, **5. The errors are normally distributed** is satisfied.

Scale-Location

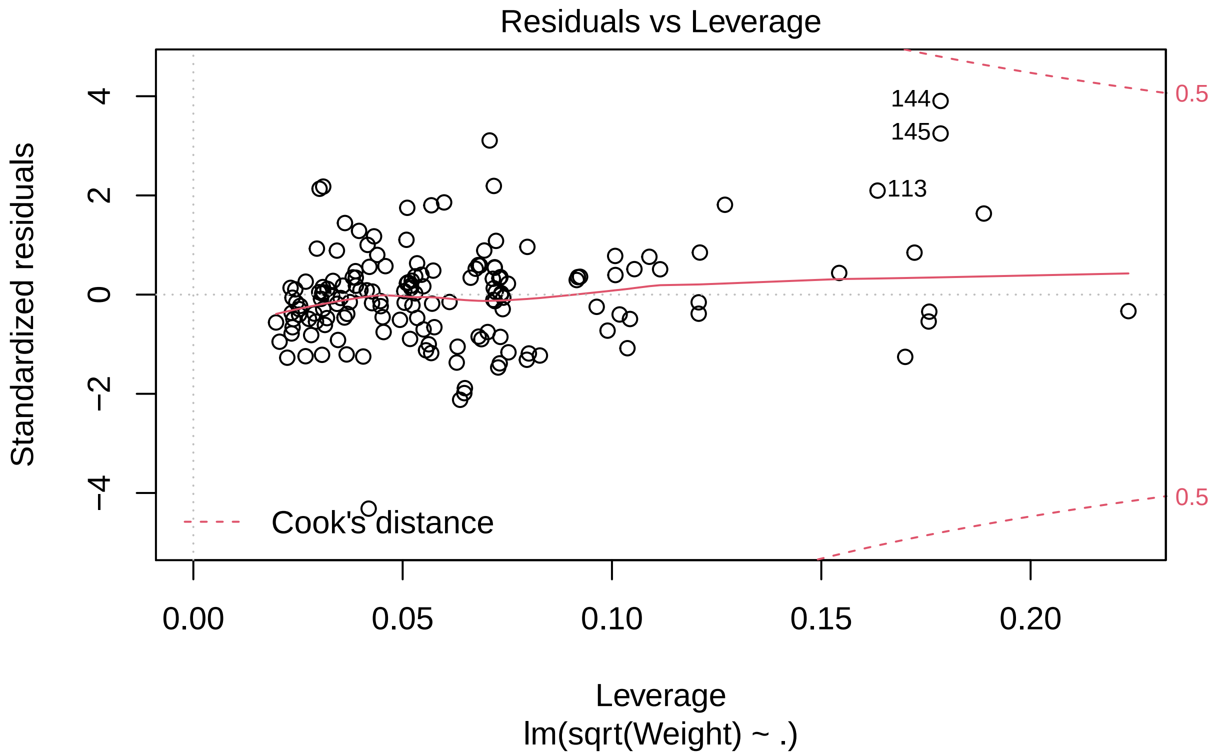
```r
plot(model_transformed, which = 3)
```



The variance is fairly constant. Before the transormation, there was a definite pattern that would lead us to conclude non-constant variance. **3. The error term ϵ has constant variance σ^2.** is satisfied.

Residuals vs. Leverage

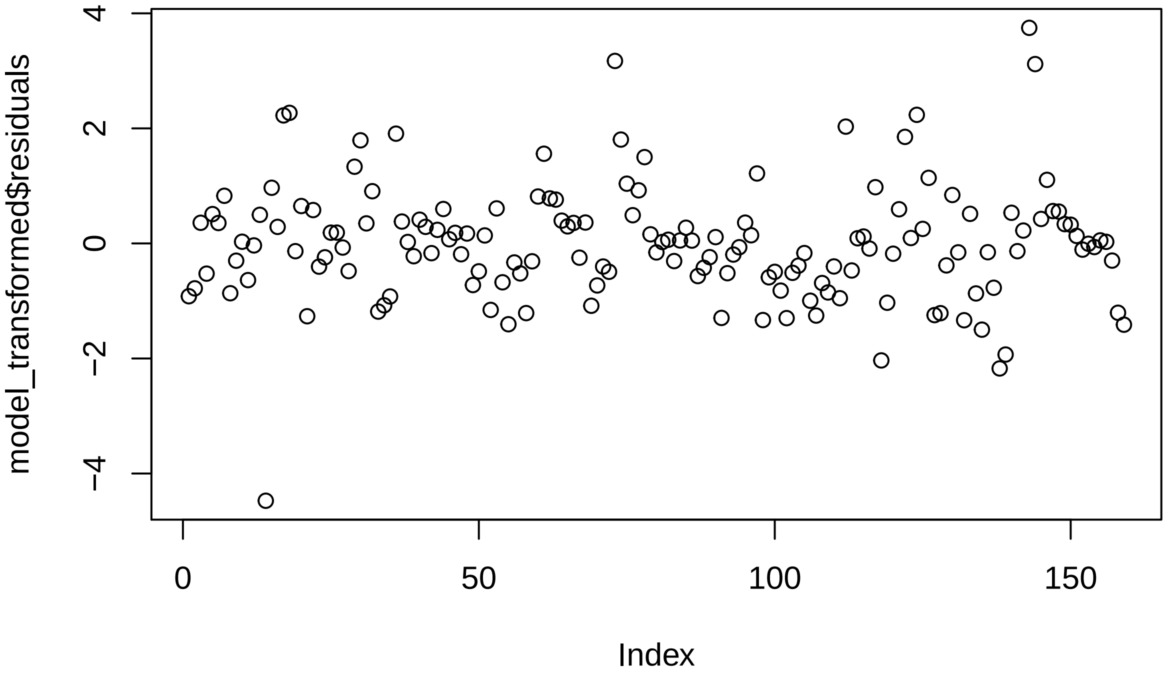
```r
plot(model_transformed, which = 5)
```



Observation 1 was an influential observation negatively impacting the explanatory power of the model. Since it was deemed to be invalid, it was removed. While observation 15 is an outlier, it does not carry much leverage over the regression and should be fine to leave in. It is notable that there are a few observations which exhibit significantly more leverage. However, their Cook's distance still suggest that they are not influential.

Residual vs. Index

```
plot(model_transformed$residuals)
```



The Risidual vs. Index plot shows the observations index on the x-axis and its risidual on the y-axis. We want a random scattering of residuals around $\epsilon = 0$ (i.e. no correlation of the errors). we can clearly see that this is the case, so **4. The errors are uncorrelated.** is satisfied.

Conclusion

Our model fully satisfies all linear regression assumptions, and almost fully explains the variance 0.9872994. Although little in explanatory power can be gained from a model selection process, it will be conducted for completeness.

```
data <- transform(data, Weight = sqrt(Weight))
write.csv(data, 'cleaned_data_4.csv', row.names = FALSE)
```

Variable Selection

Variable selection is a balance between making the model as realistic as possible (include as many regressors as possible) and as simple as possible (including only the variables needed). Forward Selection will be implemented using p-values with a critical value of $\alpha = 0.05$***. All selection criterion will be considered.***

```
library('olsrr')
```

```
## Warning: package 'olsrr' was built under R version 4.0.3
```

```
##
## Attaching package: 'olsrr'
```

```
## The following object is masked from 'package:datasets':
##
##     rivers
```

```
library(faraway)
```

```
## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car
##   influence.merMod                car
##   dfbeta.influence.merMod         car
##   dfbetas.influence.merMod        car
```

```
##
## Attaching package: 'faraway'
```

```
## The following object is masked from 'package:olsrr':
##
##     hsb
```

```
data <- read.csv("cleaned_data_4.csv", fileEncoding="UTF-8-BOM")
```

The forward selection function requires the model and a set p-value.

```
model <- lm(Weight ~ ., data = data)
FWDfit_p <- ols_step_forward_p(model, penter=0.05)
FWDfit_p
```

```
##
##                                   Selection Summary
## -----------------------------------------------------------------------------------
##         Variable                           Adj.
## Step         Entered         R-Square    R-Square      C(p)        AIC        RMSE
## -----------------------------------------------------------------------------------
##    1    Width                  0.9086      0.9081    916.8659    778.8648    2.7671
##    2    Length1                0.9619      0.9614    294.0794    641.8320    1.7928
##    3    Height                 0.9821      0.9817     59.2495     523.8758    1.2334
##    4    Species_Pike           0.9850      0.9846     26.6765     497.3113    1.1311
##    5    Species_Smelt          0.9867      0.9862      9.4981     480.9309    1.0710
##    6    Species_Whitefish      0.9871      0.9866      6.7601     478.0428    1.0582
## -----------------------------------------------------------------------------------
```

The forward selection builds up from no variables in the model. A predictor is added to the model at each iteration based on whether it has the lowest p-value. This continues until there are no more predictors to add, or all the remaining variables have a p-value > 0.05. In this case all selection criteria confirm that Width, Length1, Height, Species_Pike, Species_Smelt, and Species_Whitefish are included in the best model.

```
model <- lm(Weight ~
            Width +
            Length1 +
            Height +
            Species_Pike +
            Species_Smelt +
            Species_Whitefish ,
            data = data)

summary(model)
```

```
##
## Call:
## lm(formula = Weight ~ Width + Length1 + Height + Species_Pike +
##     Species_Smelt + Species_Whitefish, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6278 -0.5128 -0.0424  0.4406  3.7161
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -8.16172    0.31660 -25.779  < 2e-16 ***
## Width              1.11711    0.20841   5.360 3.03e-07 ***
## Length1            0.64943    0.04172  15.565  < 2e-16 ***
## Height             0.46026    0.03642  12.636  < 2e-16 ***
## Species_Pike      -3.20450    0.70686  -4.533 1.17e-05 ***
## Species_Smelt      1.63367    0.37477   4.359 2.39e-05 ***
## Species_Whitefish  0.97924    0.44952   2.178   0.0309 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.058 on 152 degrees of freedom
## Multiple R-squared:  0.9871, Adjusted R-squared:  0.9866
## F-statistic:  1933 on 6 and 152 DF,  p-value: < 2.2e-16
```

We can go back and see that the VIF is roughly the same as it was before the residual analysis and model selection process.

```
vif(model)
```

```
##             Width           Length1            Height      Species_Pike
##         17.418522         24.546980          3.439151          6.774753
##     Species_Smelt Species_Whitefish
##          1.601499          1.041924
```

Width and Length1 still have high VIF. By removing one, we can see that the VIF's improve.

```
model_reduced <- lm(Weight ~
            Length1 +
            Height +
            Species_Pike +
            Species_Smelt +
            Species_Whitefish ,
            data = data)
vif(model_reduced)
```

```
##           Length1            Height      Species_Pike     Species_Smelt
##          4.919206          3.426583          2.972007          1.398381
## Species_Whitefish
##          1.025656
```

However, one of the main issues that multicollinearity causes is reduced statistical significance of the predictor. Clearly multicollinearity isn't causing problems since all the predictors in the model are significant. It should be fine to leave the model as is. The final model seems to be good. In the end, the model has been reduced to a reasonable number of predictors while still maintaining its strong explanatory power.

```
data$Species_Parkki <- NULL
data$Species_Perch <- NULL
data$Species_Roach <- NULL

write.csv(data, 'cleaned_data_5.csv', row.names = FALSE)
```

Model Validation

The goal in Model Validation is to evaluate the quality of the predictions the model generates (the model's performance). This is performed through cross validation. The model's adequacy has already been established through the residual analysis.

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.3
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
data <- read.csv("cleaned_data_5.csv", fileEncoding="UTF-8-BOM")


model <- lm(Weight ~
            Width +
            Length1 +
            Height +
            Species_Pike +
            Species_Smelt +
            Species_Whitefish ,
            data = data)

summary(model)
```

```
##
## Call:
## lm(formula = Weight ~ Width + Length1 + Height + Species_Pike +
##     Species_Smelt + Species_Whitefish, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6278 -0.5128 -0.0424  0.4406  3.7161
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -8.16172    0.31660 -25.779  < 2e-16 ***
## Width              1.11711    0.20841   5.360 3.03e-07 ***
## Length1            0.64943    0.04172  15.565  < 2e-16 ***
## Height             0.46026    0.03642  12.636  < 2e-16 ***
## Species_Pike      -3.20450    0.70686  -4.533 1.17e-05 ***
## Species_Smelt      1.63367    0.37477   4.359 2.39e-05 ***
## Species_Whitefish  0.97924    0.44952   2.178   0.0309 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.058 on 152 degrees of freedom
## Multiple R-squared:  0.9871, Adjusted R-squared:  0.9866
## F-statistic:  1933 on 6 and 152 DF,  p-value: < 2.2e-16
```

This is the model determined to be the best through variable selection.

```
set.seed(101)
n_train <- ceiling(0.8 * length(data$Weight))
train_sample <- sample(c(1:length(data$Weight)), n_train)
train_data <- data[train_sample, ]
test_data <- data[-train_sample, ]
```

This randomly splits the data set into a traing data set and a testing data set. To measure the performance of the model and the quality of its predictions, cross validation is used. This involves fitting fitting the model using the training data, and using this model to predict the responses on the test set.

```
model <- lm(Weight ~
            Width +
            Length1 +
            Height +
            Species_Pike +
            Species_Smelt +
            Species_Whitefish ,
            data = train_data)

summary(model)
```

```
##
## Call:
## lm(formula = Weight ~ Width + Length1 + Height + Species_Pike +
##     Species_Smelt + Species_Whitefish, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6995 -0.5485 -0.0137  0.4101  3.2264
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -8.16271    0.35105 -23.252  < 2e-16 ***
## Width              0.98164    0.22311   4.400 2.35e-05 ***
## Length1            0.67223    0.04552  14.766  < 2e-16 ***
## Height             0.46653    0.04023  11.596  < 2e-16 ***
## Species_Pike      -3.21867    0.80021  -4.022 0.000101 ***
## Species_Smelt      1.50787    0.43260   3.486 0.000685 ***
## Species_Whitefish  1.00205    0.45615   2.197 0.029944 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.064 on 121 degrees of freedom
## Multiple R-squared:  0.9869, Adjusted R-squared:  0.9862
## F-statistic:  1516 on 6 and 121 DF,  p-value: < 2.2e-16
```

```
predictions <- predict(model, test_data)

R_sq <- R2(predictions, test_data$Weight)
RMSE <- RMSE(predictions, test_data$Weight)
MAE <- MAE(predictions, test_data$Weight)
```

```r
print(c(R_sq, RMSE, MAE))
```

```
## [1] 0.9878644 1.0691612 0.8428491
```

The quality of the predictions is about the same using the test data. The root mean square error (RMSE) and mean absolute error (MAE) should be small on a well performing model. To give context to these values, we can divide RMSE by the mean of the response variable to give a prediction error rate.

```r
pred_error <- RMSE / mean(test_data$Weight)
pred_error
```

```
## [1] 0.05897247
```

This means that there is only a 0.0589725 error on average (in the unit of measure weight is). Since the data set is not huge, there could be a risk that observations with a large absolute value could have all been partitioned exclusively to the test or training set (creating bias). Even though the prediction error is low, we have no basis of comparison.

```r
R_sq <- 0
RMSE <- 0
MAE <- 0

for(i in 1:20){

  n_train <- ceiling(0.8 * length(data$Weight))
  train_sample <- sample(c(1:length(data$Weight)), n_train)
  train_data <- data[train_sample, ]
  test_data <- data[-train_sample, ]


  model <- lm(Weight ~
              Width +
              Length1 +
              Height +
              Species_Pike +
              Species_Smelt +
              Species_Whitefish ,
              data = train_data)

  summary(model)

  predictions <- predict(model, test_data)

  R_sq <- R_sq + R2(predictions, test_data$Weight)
  RMSE <- RMSE + RMSE(predictions, test_data$Weight)
  MAE <- MAE + MAE(predictions, test_data$Weight)

}

R_sq = R_sq / 20
RMSE = RMSE / 20
```

```r
MAE = MAE / 20

print(c(R_sq, RMSE, MAE))
```

```
## [1] 0.9866953 1.0384956 0.7575819
```

The average of R^2, RMSE, and MAE is almost the same as what was initially computed. This shows that the inital cross validation was not a one off.