Final_Project_Team3

Multiple regression analysis-the weight of the fish

Jinxi Liu & Tingxuan Zhang

Github Link: Jinxi Liu: https://github.com/Jessiexx-27/STAT350-Final-Project.git Tingxuan Zhang: https://github.com/Tingxuan-Zhang/fish-market-analysis.git

Abstract:

In this project, we would investigate the factors which have effects on the fish weight. For the dataset in the study, 159 fishes' data were collected in fish market sales with 7 variables, which were the response variable Weight and the 6 predictors we would like to study. With this dataset, a predictive model can be performed using machine friendly data and estimate the weight of fish can be predicted.

Introduction:

We are interested in building reasonable model between the Weight variable and the different lengths, the height, the width values, the type of fish, then do a prediction on weight.

Data description:

The dataset is about 159 fishes in fish market with 7 variables. There is 1 response variable and 6 explanatory variables.

There are the first 6 rows of the dataset and the structure of a data object

```
mydata <- read.csv("Fish.csv")
head(mydata)
```

```
##   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
```

```
str(mydata)
```

```
## 'data.frame':    159 obs. of  7 variables:
##  $ Species: Factor w/ 7 levels "Bream","Parkki",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ 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 ...
```

Response variable: 1. Weight: Weight of fish in gram

Explanatory variable: 1. Species: Species name of fish 2. Length1: Vertical length in cm 3. Length2: Diagonal length in cm 4. Length3: Cross length in cm 5. Height: Height in cm 6. Width: Diagonal width in cm

There are 7 levels of Species

```
length(unique(mydata$Species))
```

```
## [1] 7
```

Additional datapoint: The introduced rows are the numbers of the means of each variable from the original dataset of the species bream. The unique data points are being chose since we want to avoid extrapolation. Uses mean we can avoid the data point we made are extrapolation or outliers. After defined a new point, we check it is location relative to the RVH to make sure the new data point would fall under interpolation.

```
# specify the species of fish
sub_data = mydata[which(mydata$Species=="Bream"),]
# compute the hat matrix and save the largest diagonal element as hmax
X <- cbind(rep(1, nrow(sub_data)), sub_data$Weight, sub_data$Length1,sub_data$Length2, sub_data$Length3, sub
_data$Height, sub_data$Width)
H <- X %*% solve(t(X) %*% X) %*% t(X)
(h_max <- max(diag(H)))
```

```
## [1] 0.5290972
```

```
new_Weight = round(mean(sub_data$Weight),digits = 1)
new_length1 = round(mean(sub_data$Length1),digits = 1)
new_length2 = round(mean(sub_data$Length2),digits = 1)
new_length3 = round(mean(sub_data$Length3),digits = 1)
new_Height = round(mean(sub_data$Height),digits = 4)
new_Width = round(mean(sub_data$Width),digits= 4)

# define a new point and check it’s location relative to the RVH.
x_0<-data.frame( Weight = new_Weight,
                Length1 = new_length1,
                Length2 = new_length2,
                Length3 = new_length3,
                Height = new_Height,
                Width = new_Width)
x_0 <- as.matrix(cbind(1, x_0), nrow = 1)
(h_00 <- x_0 %*% solve(t(X) %*% X) %*% t(x_0))
```

```
##             [,1]
## [1,] 0.03032347
```

```
# Since 0.03032347 < 0.5290972, we conclude that the new data point would fall under interpolation
# update the data set with new point added
mydata = rbind(data.frame(Species = "Bream",
                Weight = new_Weight,
                Length1 = new_length1,
                Length2 = new_length2,
                Length3 = new_length3,
                Height = new_Height,
                Width = new_Width),mydata)
```

Methods and Results

First load the packages we need to use.

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────────────────────────── tidyverse 1.3.0 ──
```

```
## ✓ ggplot2 3.2.1     ✓ purrr   0.3.3
## ✓ tibble  2.1.3     ✓ dplyr   0.8.4
## ✓ tidyr   1.0.2     ✓ stringr 1.4.0
## ✓ readr   1.3.1     ✓ forcats 0.4.0
```

```
## ── Conflicts ──────────────────────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```r
library(ggplot2)
library(Hmisc)
```

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

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```r
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 objects are masked from 'package:survival':
##
##     rats, solder
```

```
## The following object is masked from 'package:lattice':
##
##     melanoma
```

```
## The following objects are masked from 'package:car':
##
##     logit, vif
```

```r
library(caret)
```

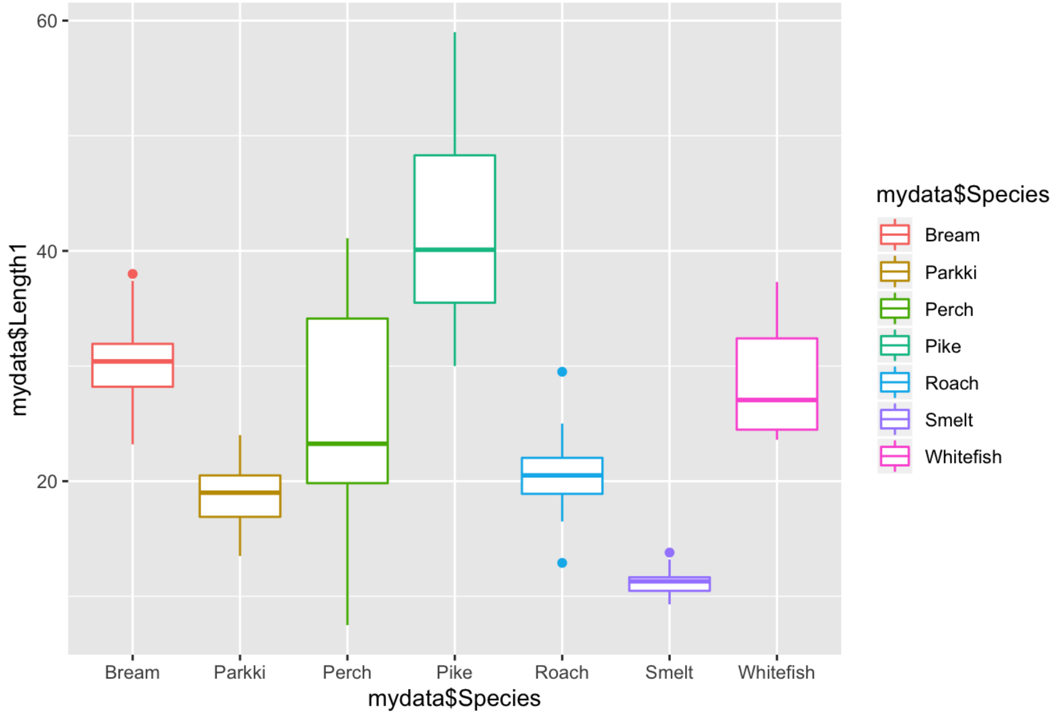
```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:survival':
##
##     cluster
## The following object is masked from 'package:purrr':
##
##     lift
```

We can check the distribution of each predictor by Species.

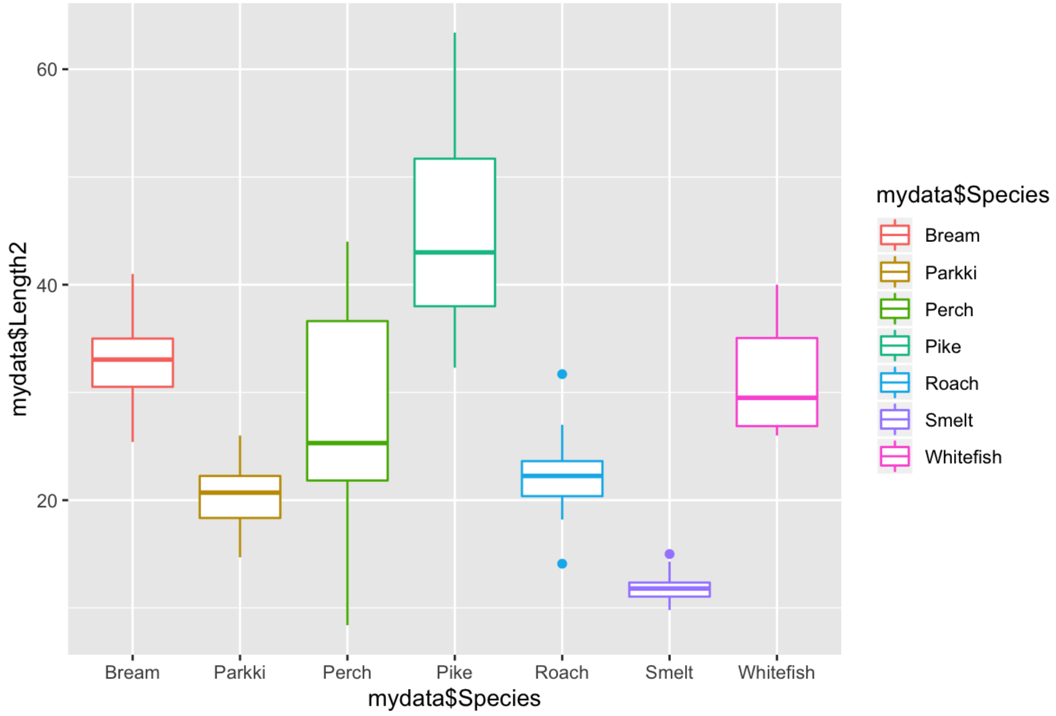
```
ggplot(mydata, aes(x=mydata$Species, y=mydata$Length1, color=mydata$Species)) +
  geom_boxplot() + ggtitle("Figure 1: boxplot of Species and Length1")
```

Figure 1: boxplot of Species and Length1



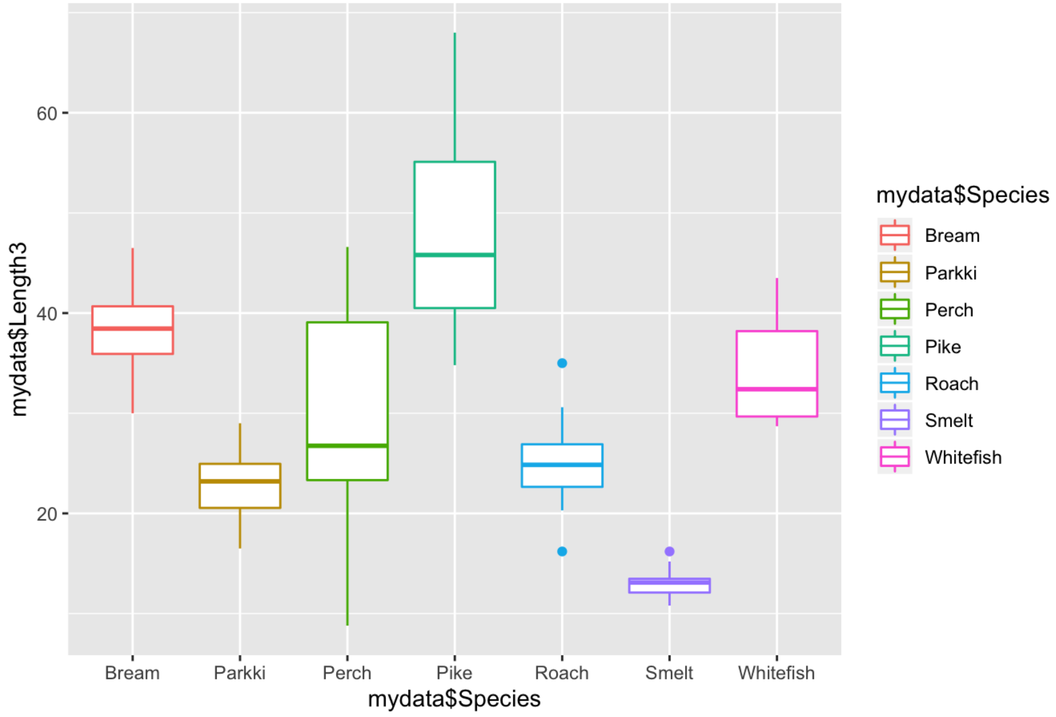
```
ggplot(mydata, aes(x=mydata$Species, y=mydata$Length2, color=mydata$Species)) +
  geom_boxplot() + ggtitle("Figure 2: boxplot of Species and Length2")
```

Figure 2: boxplot of Species and Length2



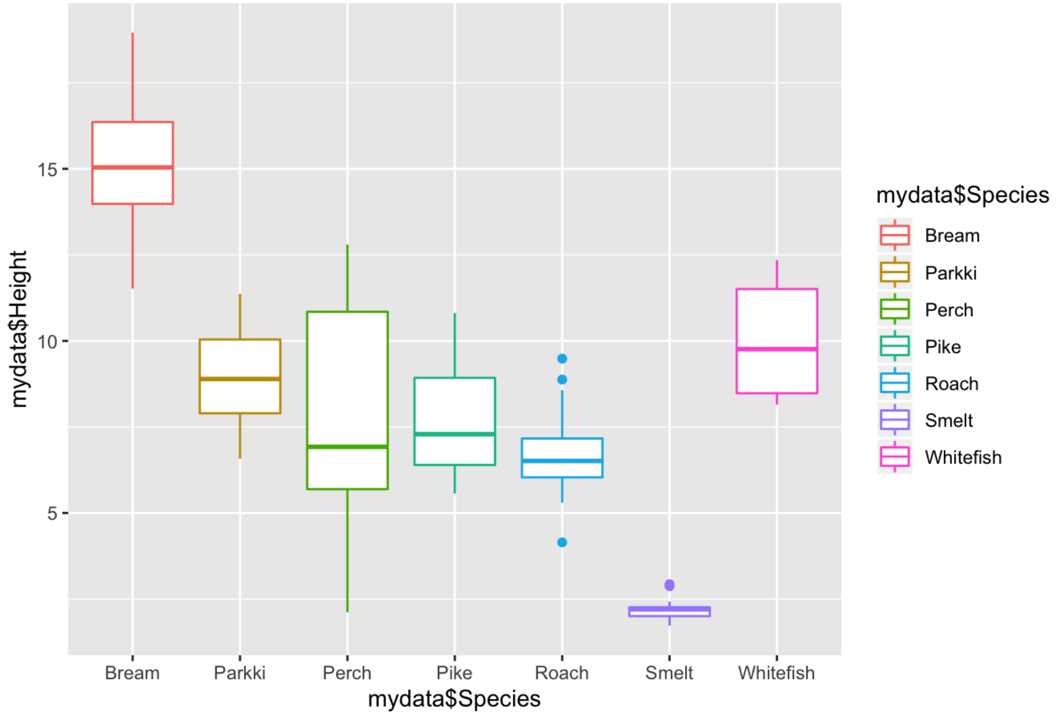
```
ggplot(mydata, aes(x=mydata$Species, y=mydata$Length3, color=mydata$Species)) +
  geom_boxplot() + ggtitle("Figure 3: boxplot of Species and Length3")
```

Figure 3: boxplot of Species and Length3



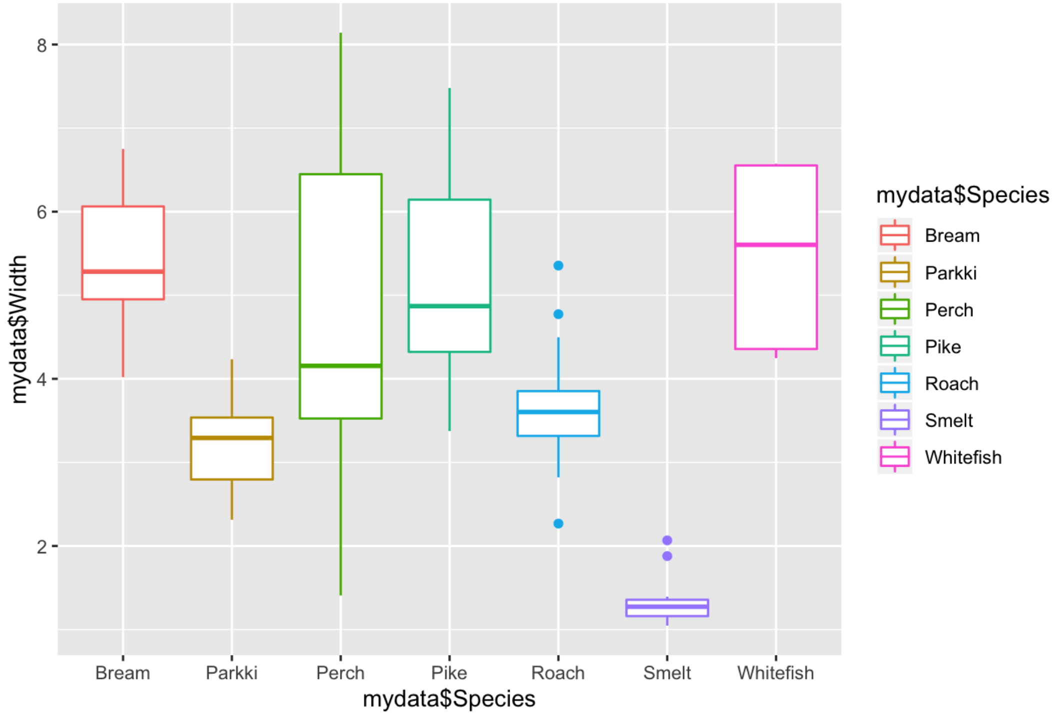
```
ggplot(mydata, aes(x=mydata$Species, y=mydata$Height, color=mydata$Species)) +
  geom_boxplot() + ggtitle("Figure 4: boxplot of Species and Height")
```

Figure 4: boxplot of Species and Height



```
ggplot(mydata, aes(x=mydata$Species, y=mydata$Width, color=mydata$Species)) +
  geom_boxplot() + ggtitle("Figure 5: boxplot of Species and Width")
```

Figure 5: boxplot of Species and Width



```
ggplot(mydata, aes(x=mydata$Species, y=mydata$Weight, color=mydata$Species)) +
  geom_boxplot() + ggtitle("Figure 6: boxplot of Species and Weight")
```

Figure 6: boxplot of Species and Weight

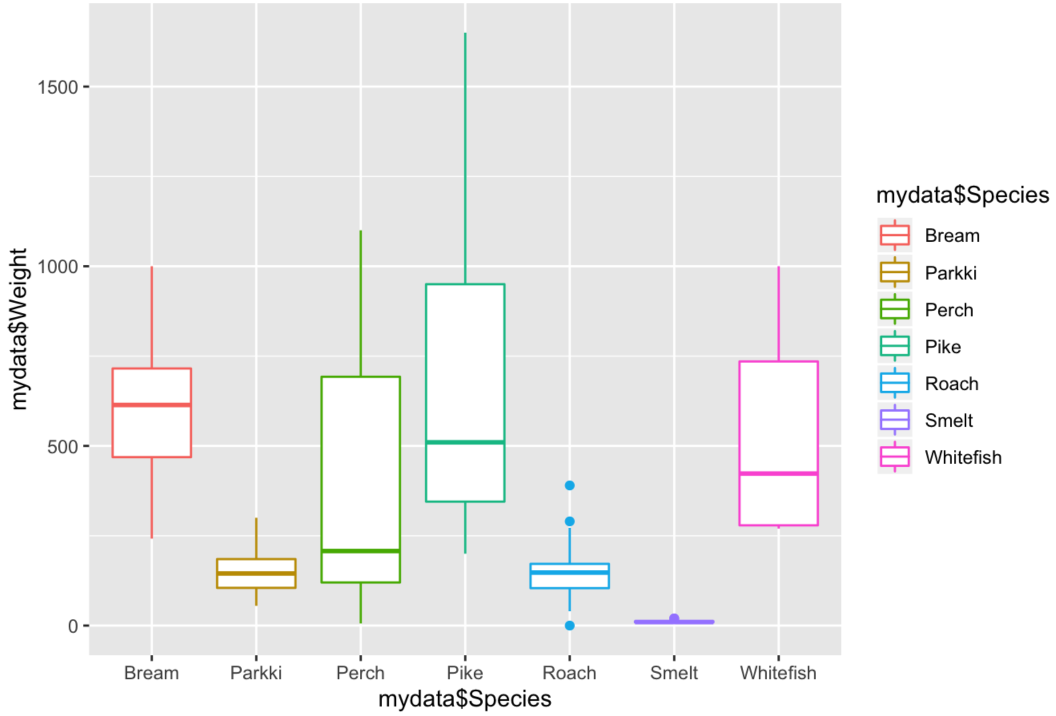


Figure 1-6 shows the boxplot of the distribution between species and Length1, Length2, Length3, Height, Width, Weight respectively from the dataset.

We could see that the value and the range of variables' value are totally different for each species of fish, the mean of predictor's value are different for each specie. So the distribution are different.

Multiple linear regression is a fundamental practice for this dataset. When analyzing the data using the multiple linear regression models, the response variable will be considered as a linear function of the explanatory variables with an error term. If the data are appropriate to use the linear regression, then the relationship between the response variables and the explanatory variables should be approximately linear. The error term will have a mean of zero and a constant variance, which is normally distributed.

So we try to do regression analysis for all species of fish.

We first visualize the data with a scatter plot matrix, and calculate the correlation between these variables. By checking the correlation of the variables, we can see if there is correlation exists.

```
pairs(mydata,main= "Figure 7: scatterplot matrix of the response variable and predictors")
```

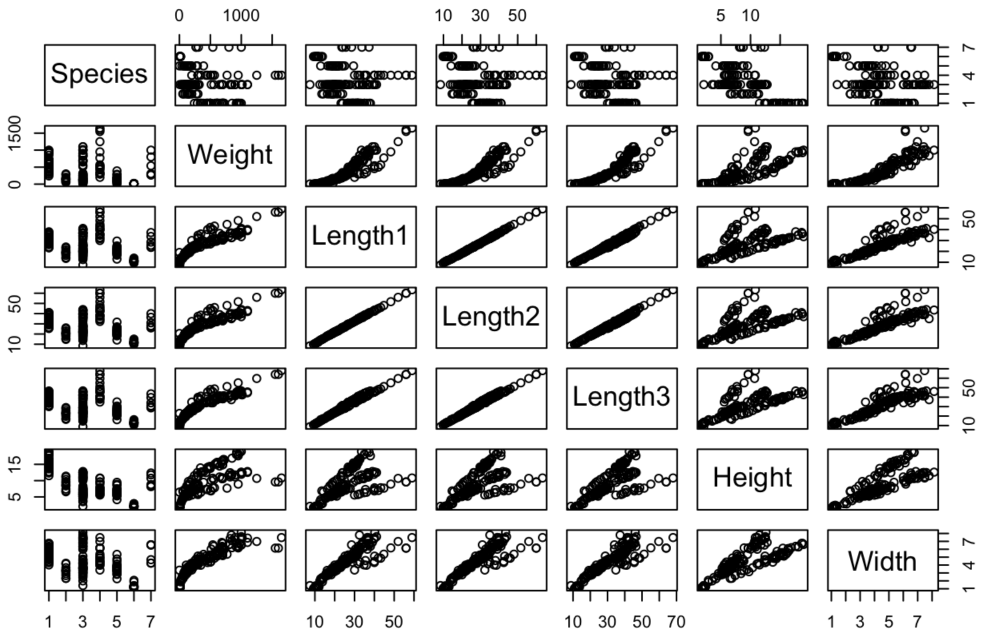


Figure 7: scatterplot matrix of the response variable and predictors

```
# Calculate the correlation between these variables
cor(mydata[,c(2,3,4,5,6,7)])
```

```
##            Weight   Length1   Length2   Length3    Height     Width
## Weight  1.0000000 0.9157195 0.9186655 0.9232264 0.7243038 0.8867677
## Length1 0.9157195 1.0000000 0.9995147 0.9919033 0.6246360 0.8671505
## Length2 0.9186655 0.9995147 1.0000000 0.9940109 0.6398258 0.8736827
## Length3 0.9232264 0.9919033 0.9940109 1.0000000 0.7035588 0.8788014
## Height  0.7243038 0.6246360 0.6398258 0.7035588 1.0000000 0.7922270
## Width   0.8867677 0.8671505 0.8736827 0.8788014 0.7922270 1.0000000
```

Figure 7 shows the scatterplot matrix of the response variable and predictors. From the scatterplot, we observed that the predictors Length1, Length2, Length3, Height, Width might be linearly related to the response variable Weight.

Mainly, we see that a linear regression model seems appropriate. Another thing to take notice of is the multicollinearity present among the explanatory variables, because there are high correlation between all the parameters and the Weight except Species.

Multicollinearity affects the coefficients and p-values, but it does not influence the predictions, precision of the predictions, and the goodness-of-fit statistics.

Our goal is doing a prediction, so we do not have to fix it.

Then try to fit the full model, and get the R-squared value to see how it fits.

```
mdl0<- lm(Weight~ factor(Species) +
                Length1+Length2+Length3+Height+Width,
                data = mydata)
sum0<-summary(mdl0)
sum0$r.squared
```

```
## [1] 0.9362356
```

```
plot(x=mydata$Weight,y=predict(mdl0), xlab = "Weight_observed", ylab = "Weight_predicted",pch = 16,
     main = "Figure 8: Weight observed vs Weight predicted")
abline(0,1,col = "Pink",lwd=3)
```

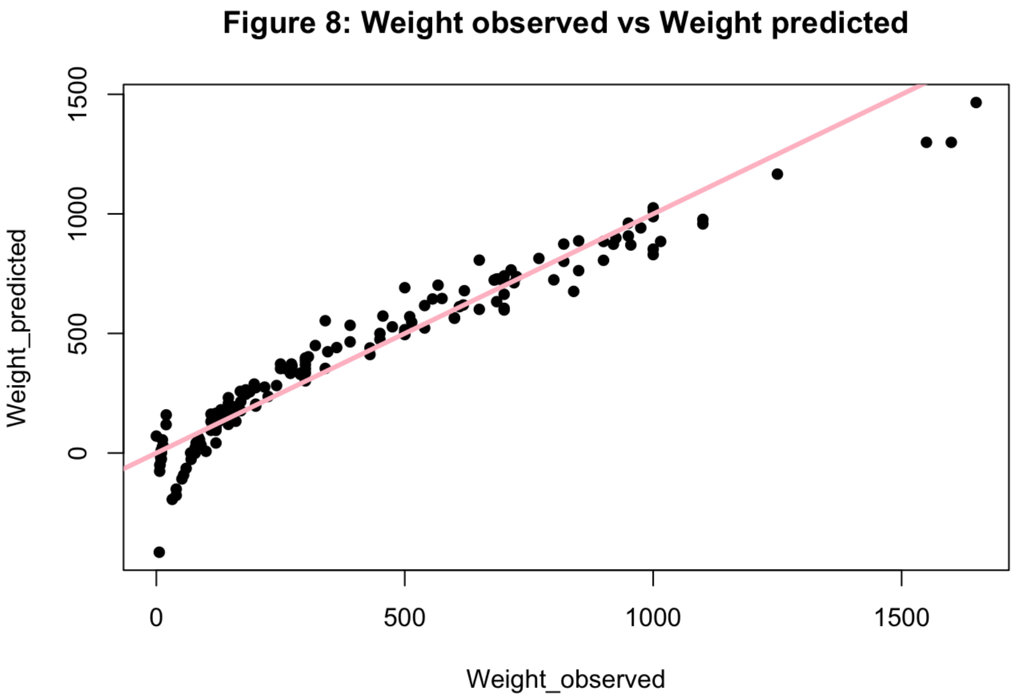


Figure 8 shows linear regression of the full model of the weight observed and the weight predicted.

R squared is 0.9362356, seems like good. But as we noticed, each parameter from data set is a measurement of fish. As far as we know the interaction between body measurements are definitely occur. So we may could do a better model.

Try to fit the full model with interaction terms

```
mdl<- lm(Weight~ factor(Species) +
                Length1*Length2*Length3*Height*Width,
                data = mydata)
sum<-summary(mdl)
sum$r.squared
```

```
## [1] 0.9902657
```

```
plot(x=mydata$Weight,y=predict(mdl), xlab = "Weight_observed", ylab = "Weight_predicted",pch = 16,
     main = "Figure 9: Weight observed vs Weight predicted")
abline(0,1,col = "Pink",lwd=3)
```

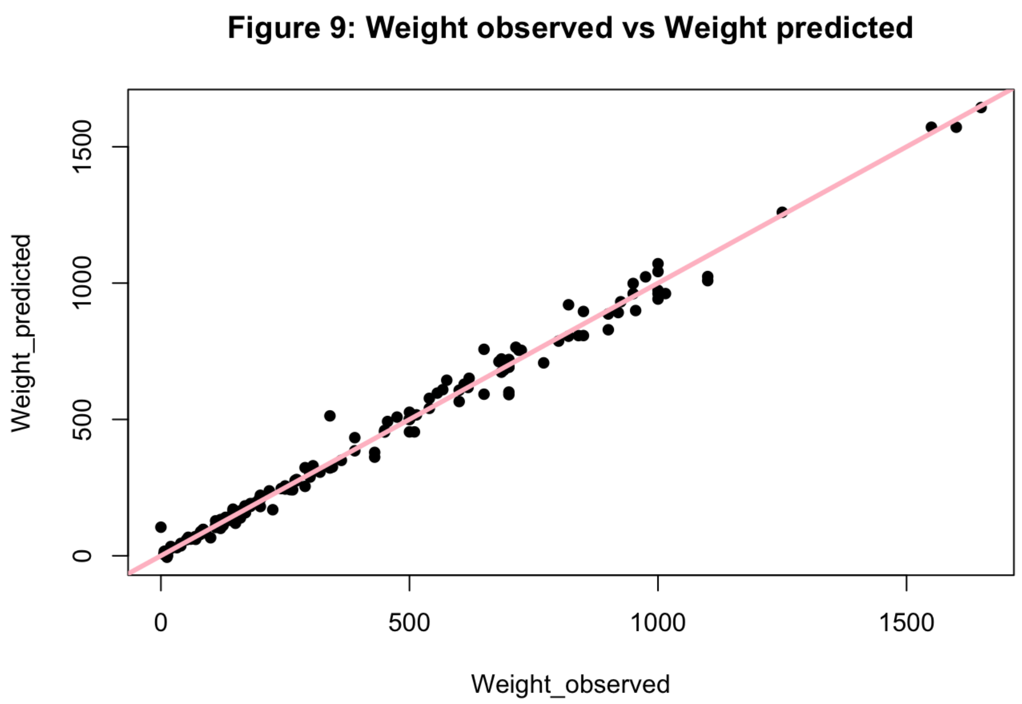
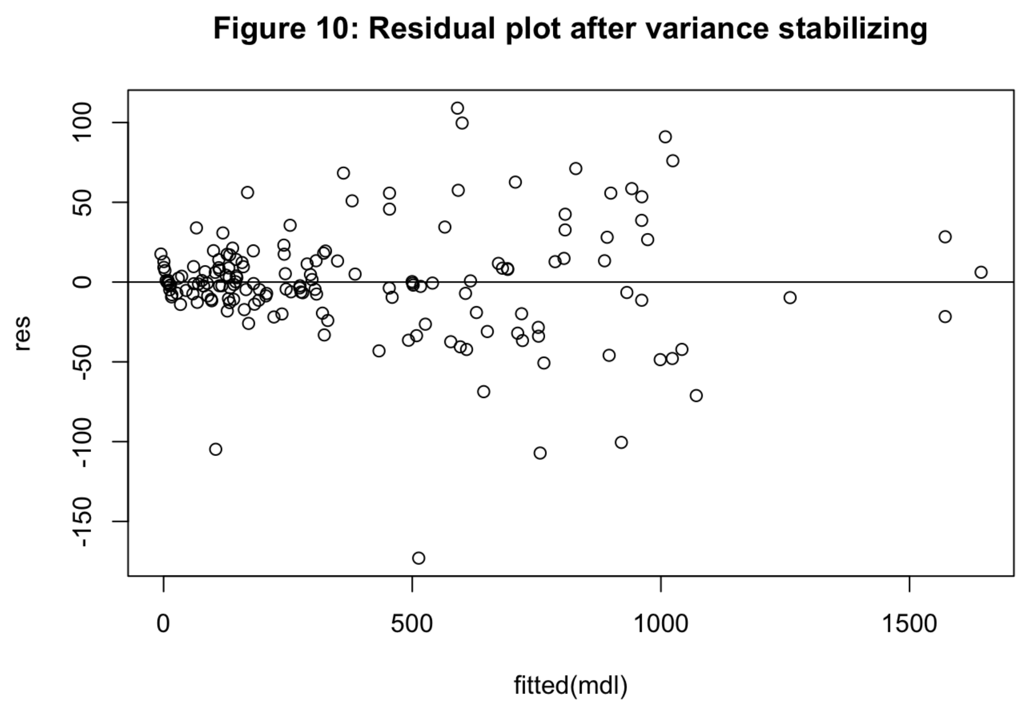


Figure 9 is the plot of the full model with interaction terms. It gives a better linear regression of the weight observed and the weight predicted.

R squared is 0.9902657, it’s better than befor and this value is really close to 1. Then we could plot residual to do Residual Analysis.

For the full model, the residual plots and Q-Q plots will be study to see whether the assumptions of the linear regression are violated, and whether the variables need transformations.

```
# Get the residual
res<- resid(mdl)
plot(fitted(mdl), res, main = "Figure 10: Residual plot after variance stabilizing")
# add a horizontal line at 0
abline(0,0)
```



```
# create Q-Q plot for residuals
qqnorm(res, main = "Figure 11: Residual plot after variance stabilizing")
```

Figure 11: Residual plot after variance stabilizing

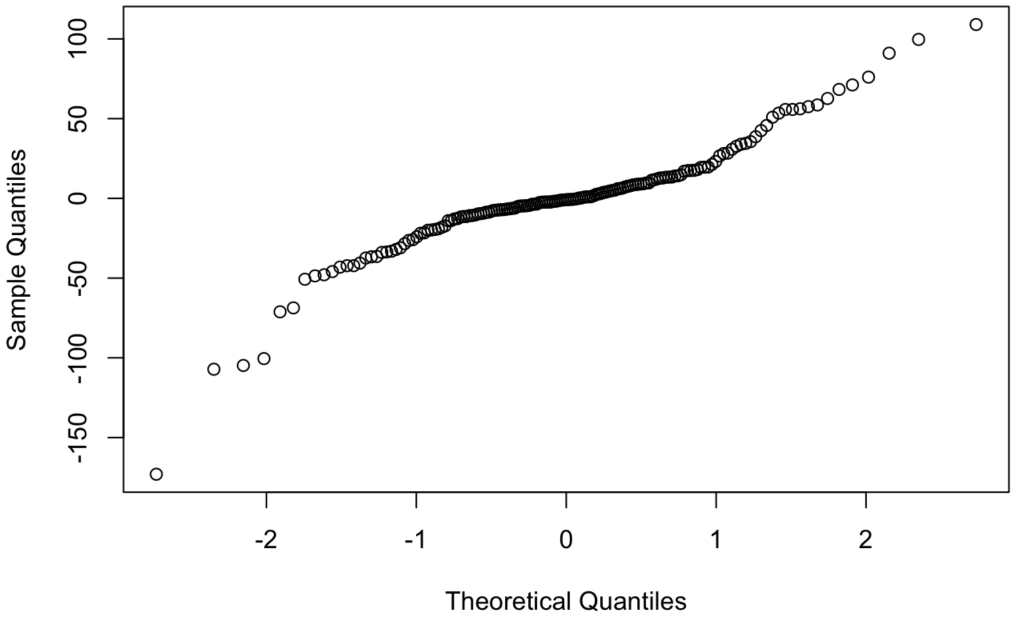
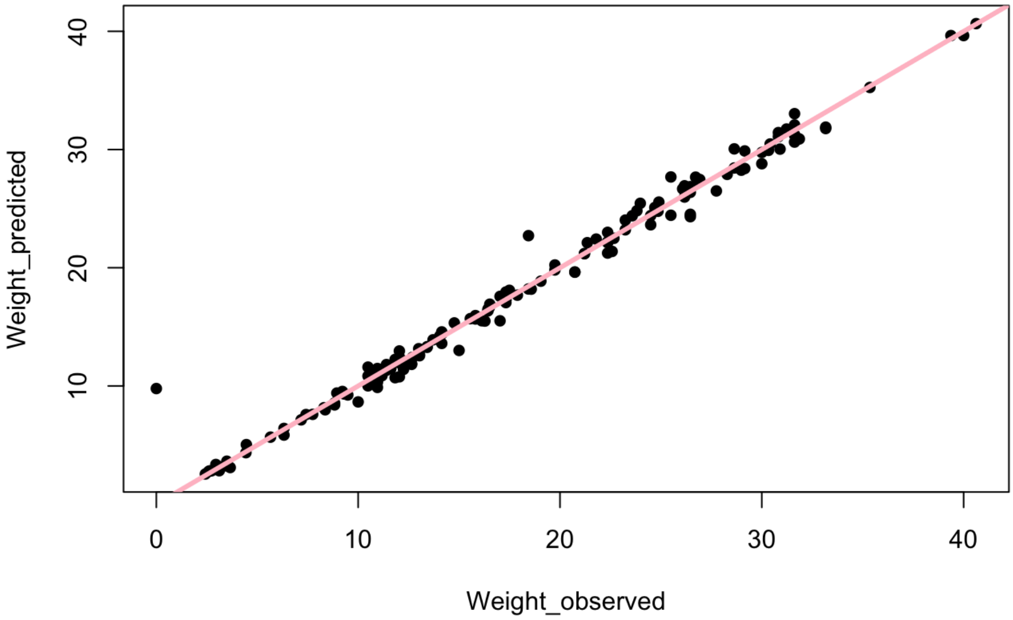


Figure 10 is the residual plot of the regression line, we see that the variability increase as the value of predictor increses. The Figure 11 is the Q-Q plot, there are may points not on the straight line. That violated the constant variance assuption, so we can do a variance stabilizing transformation.

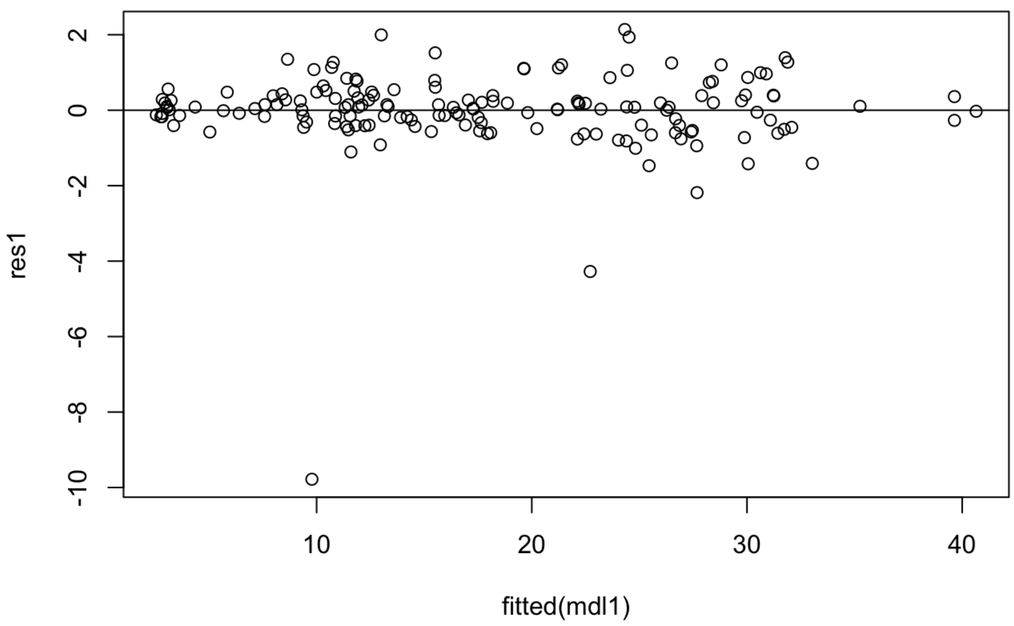
```
# transform y to squre root of y
mydata2 <- transform(mydata, Weight=sqrt(Weight))
mdl1<- lm(Weight~ factor(Species) +
                Length1*Length2*Length3*Height*Width,
                data = mydata2)
plot(x=mydata2$Weight,y=predict(mdl1), xlab = "Weight_observed", ylab = "Weight_predicted",pch = 16,
   main = "Figure 12: Weight observed vs Weight predicted after variance stabilizing transformation")
abline(0,1,col = "Pink",lwd=3)
```

e 12: Weight observed vs Weight predicted after variance stabilizing trans



```
res1 <- resid(mdl1)
plot(fitted(mdl1), res1, main = "Figure 13: Residual plot after variance stabilizing transformation")
abline(0,0)
```

Figure 13: Residual plot after variance stabilizing transformation



```
qqnorm(res1, main = "Figure 14: Normal Q-Q plot after variance stabilizing transformation")
```

Figure 14: Normal Q-Q plot after variance stabilizing transformation

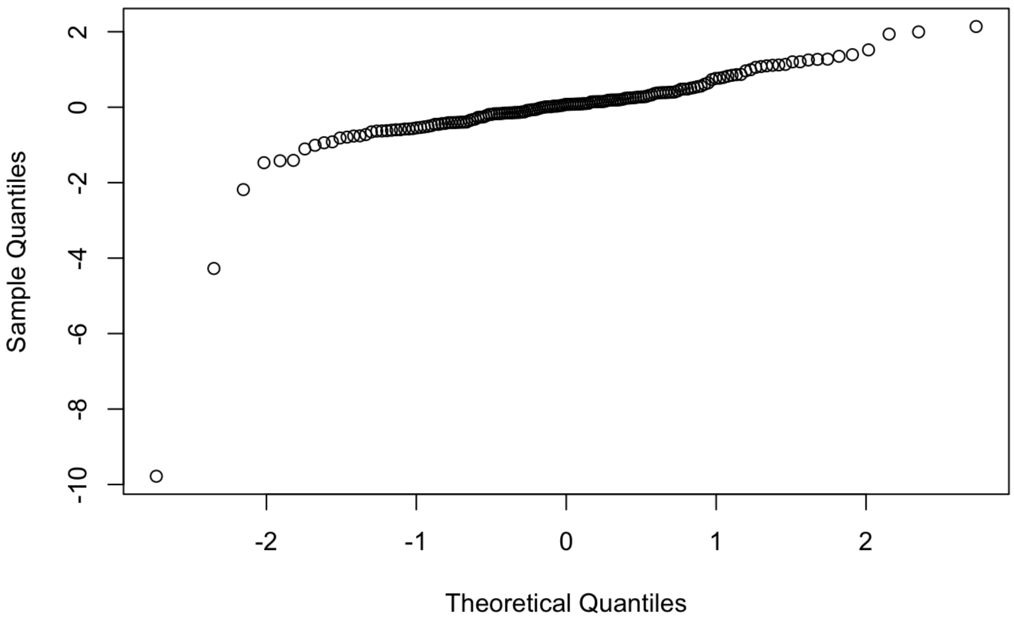


Figure 12-14: are plots of Regression plot, Residual plot and Q-Q plot respectively after we did a variance stabilizing transformation.

After transformation, the residual plot looks better because the residuals are almost all around zero. The prediction line still fit the observation points great, but we can see that there is one or two outliers.

Apply Cook’s distance will give us if there are any influential points in the dataset.

```
mdl_cook <- cooks.distance(mdl1)
sort(mdl_cook, decreasing = TRUE)
```

```
##           42          128          142          119          120          134
## 1.935602e-01 1.186596e-01 5.141530e-02 4.096848e-02 4.054933e-02 3.406803e-02
##          146          141           15           35           62          137
## 3.250204e-02 3.230748e-02 3.088504e-02 2.804213e-02 2.781177e-02 2.415966e-02
##          125           36           98          118          140          113
## 2.266394e-02 1.859152e-02 1.607844e-02 1.582130e-02 1.402552e-02 1.272186e-02
##           60           74            6           21           31          127
## 1.251105e-02 1.240680e-02 1.201073e-02 1.168123e-02 1.110052e-02 1.101099e-02
##          139           14           54           37          160          136
## 1.084125e-02 1.014117e-02 9.590049e-03 9.308065e-03 9.225603e-03 8.936256e-03
##          132           19           92           79          129           18
## 8.479280e-03 8.362267e-03 8.317845e-03 8.145770e-03 7.079252e-03 5.394945e-03
##           33           48          123          144          130           41
## 5.072771e-03 4.981766e-03 4.435111e-03 4.290267e-03 3.982773e-03 3.813210e-03
##           29           46            3           45           22          143
## 3.789981e-03 3.773823e-03 3.739976e-03 3.645866e-03 3.370579e-03 3.163553e-03
##          103          121           52          145           56           49
## 2.792501e-03 2.567322e-03 2.557937e-03 2.393340e-03 2.372083e-03 2.354572e-03
##          158           59           24          112           25          131
## 2.301845e-03 2.290924e-03 2.202504e-03 1.945933e-03 1.918577e-03 1.900879e-03
##          126           47          114           61           27           30
## 1.811808e-03 1.623040e-03 1.461933e-03 1.395208e-03 1.375384e-03 1.283167e-03
##          157          135           32           72           28          106
## 1.212278e-03 1.142709e-03 1.128638e-03 1.083125e-03 1.057071e-03 1.031868e-03
##           70          115           63           16          152           66
## 1.024390e-03 9.849194e-04 8.638137e-04 8.212413e-04 7.825493e-04 7.107921e-04
##           26           67          133           44           68           99
## 6.956231e-04 6.582832e-04 6.527391e-04 6.447330e-04 6.338742e-04 6.171349e-04
##           39           80          105           20           81            4
## 5.665246e-04 5.638398e-04 5.516784e-04 5.385391e-04 5.314127e-04 4.792734e-04
##           69          150          111           96           43           12
## 4.709103e-04 4.454141e-04 4.241961e-04 4.236351e-04 3.943388e-04 3.792667e-04
##           51            8           73           88           78            5
## 3.719900e-04 3.703893e-04 3.403760e-04 3.391780e-04 3.356086e-04 3.097901e-04
##          159          147           71           13          122           89
## 2.658244e-04 2.645183e-04 2.585836e-04 2.173857e-04 2.148473e-04 2.067926e-04
##            2           86           11           65          154           64
## 2.055869e-04 2.028005e-04 1.996034e-04 1.994853e-04 1.899048e-04 1.891878e-04
##           84           97          149           34           93          124
## 1.729194e-04 1.647619e-04 1.608574e-04 1.547145e-04 1.442385e-04 1.377228e-04
##           23           38           82           83          110           55
## 1.317638e-04 1.295433e-04 1.237266e-04 1.220454e-04 1.057457e-04 1.012498e-04
##           50          108           76          156          102           57
## 6.879965e-05 6.706083e-05 6.654335e-05 5.974820e-05 5.705676e-05 4.632196e-05
##          155           94          100          107          101           85
## 4.509204e-05 4.473571e-05 4.280144e-05 3.981394e-05 3.851728e-05 3.840584e-05
##           58           17           53          148           90           77
## 3.821648e-05 3.027959e-05 2.494785e-05 2.360717e-05 2.142065e-05 1.941283e-05
##          151           91          116            9          109           87
## 1.787372e-05 1.715077e-05 1.446171e-05 1.378673e-05 1.205452e-05 1.042334e-05
##          104            1           95          138           75           10
## 7.680992e-06 6.152243e-06 5.861947e-06 5.365535e-06 1.844217e-06 1.424715e-06
##            7          153           40          117
## 9.110410e-07 5.717691e-07 1.807217e-07 1.568605e-09
```

```
which(mdl_cook>1)
```

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

There is no observation have a large Cook’s distance that larger than 1, no may be no influence point. But the highest value of cook’s distance is more greater than the second highest one, and it is the 42th observation.

Specify the 42th row to see the data value.

```
mydata2[42,]
```

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

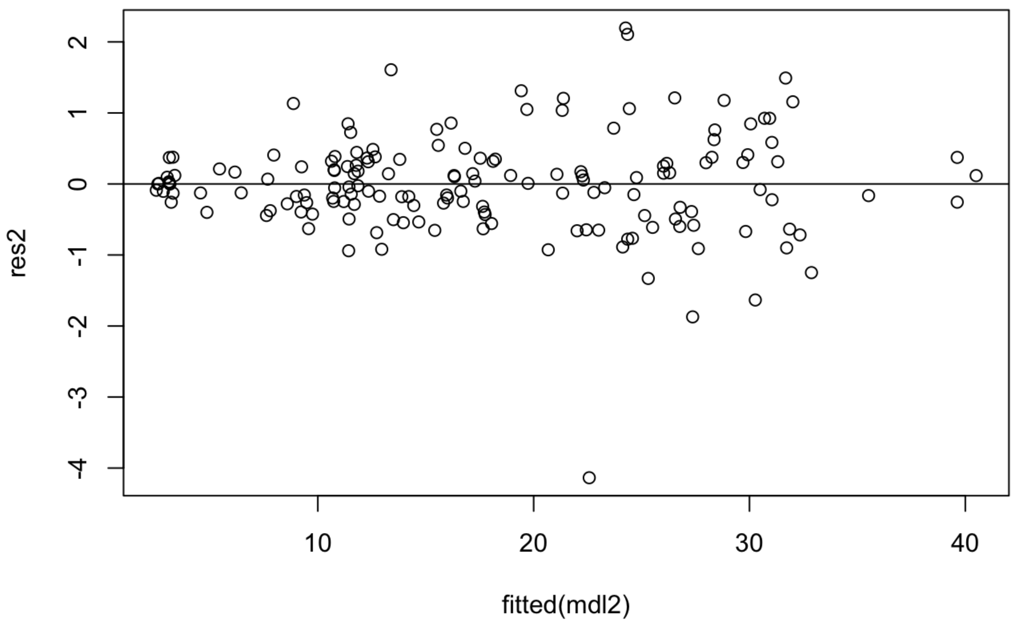
We could see the 42th observation with 0 value for fish weight, it is totally impossible. So we try to move this point to see whether the model is fitting better.

```
mydata3<-mydata2[-42,]
mdl2<- lm(Weight~ factor(Species) +
                Length1*Length2*Length3*Height*Width,
                data = mydata3)
sum2<-summary(mdl2)
sum2$r.squared
```

```
## [1] 0.9939383
```

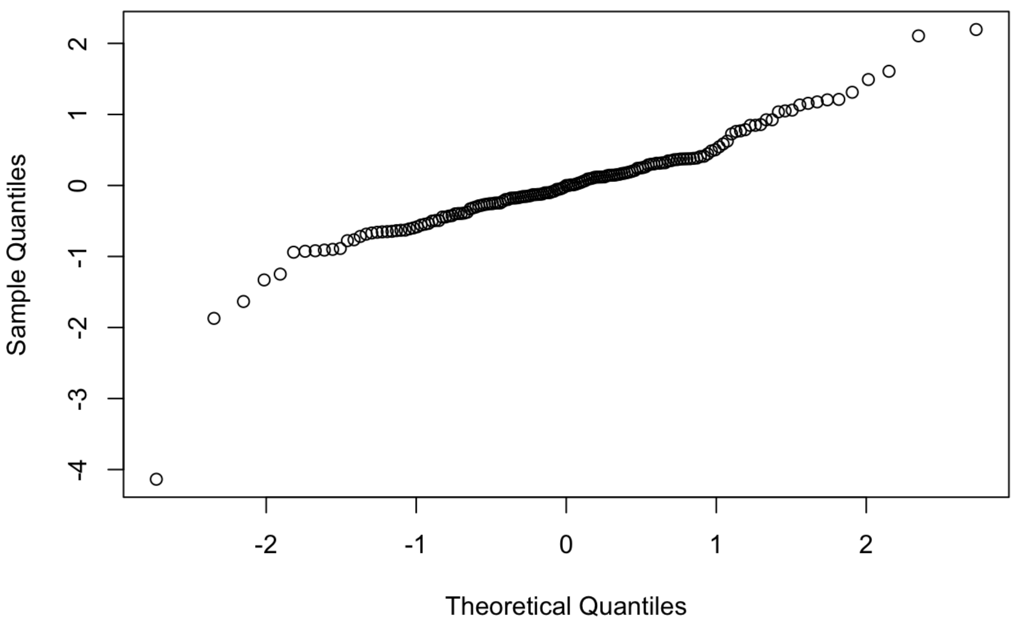
```
res2 <- resid(mdl2)
plot(fitted(mdl2), res2, main = "Figure 15: Residual plot after omitting outliers")
abline(0,0)
```

Figure 15: Residual plot after omitting outliers



```
qqnorm(res2, main = "Figure 16: Normal Q-Q plot after omitting outliers")
```

Figure 16: Normal Q-Q plot after omitting outliers



```
plot(x=mydata3$Weight,y=predict(mdl2), xlab = "Weight_observed", ylab = "Weight_predicted",pch = 16,
     main = "Figure 17: Weight observed vs Weight predicted after omitting outliers")
abline(0,1,col = "Pink",lwd=3)
```

Figure 17: Weight observed vs Weight predicted after omitting outliers

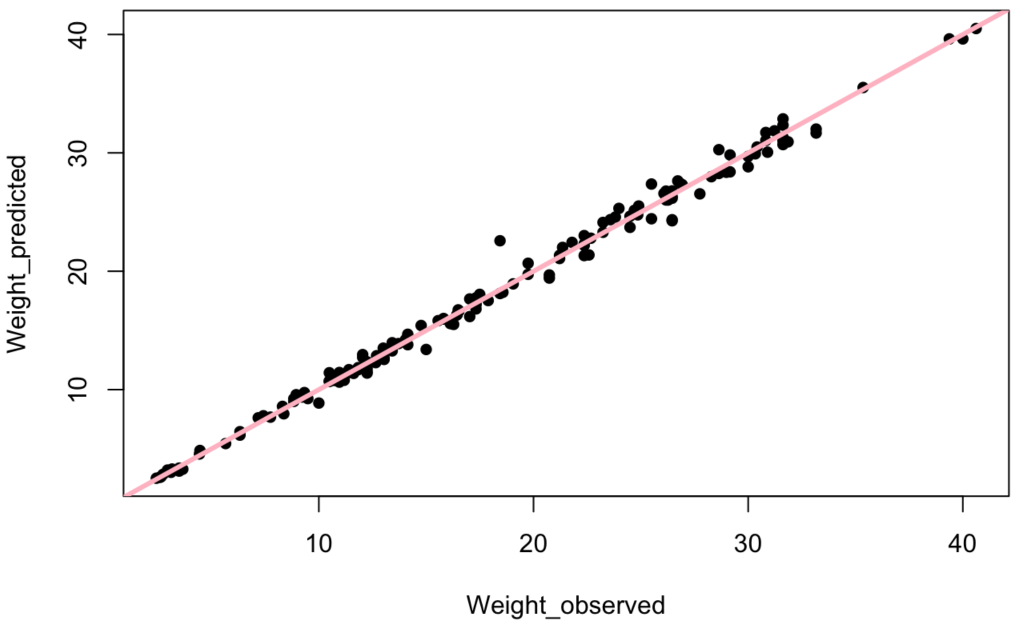


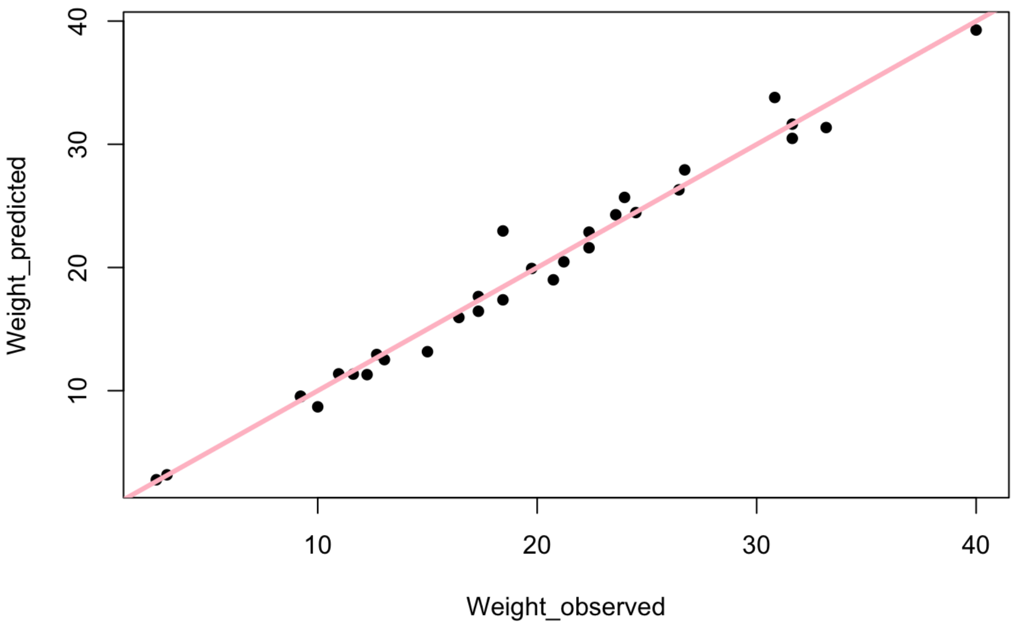
Figure 15-17: are Residual plot, Q-Q plot and Regression plot respectively after omitting outliers. The model fits better after we omit the outlier.

R squared is 0.9939383,That seems pretty better.

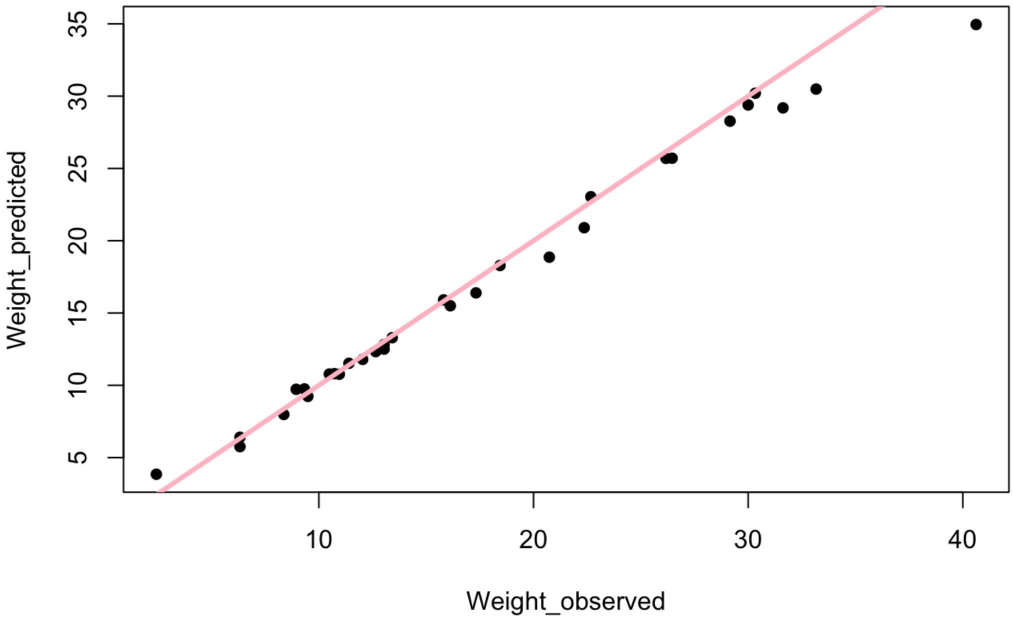
Then do cross validation, Cross Validation estimates the expected level of fit of a model to a data set that is independent of the data that were used to train the model.

```
for (i in 1:5)
{
  nsamp<-ceiling(0.8*length(mydata3$Weight))
  training_samps<-sample(c(1:length(mydata3$Weight)),nsamp)
  train_data<-mydata3[training_samps,]
  test_data<-mydata3[-training_samps,]

  train.lm<-lm(Weight~ factor(Species) +
               Length1*Length2*Length3*Height*Width,
               data = train_data)
  preds<-predict(train.lm,test_data)
  R.sq<-R2(preds, test_data$Weight)
  RMSPE<-RMSE(preds, test_data$Weight)
  MAPE<-MAE(preds, test_data$Weight)
  xx<-RMSPE/sd(test_data$Weight)

  print(c(R.sq,RMSPE,MAPE,xx))
  # Make a plot
  plot(x = test_data$Weight, y = preds,
  xlab = "Weight_observed", ylab = "Weight_predicted",pch = 16)
  abline(0,1,col = "Pink",lwd=3)
}
```

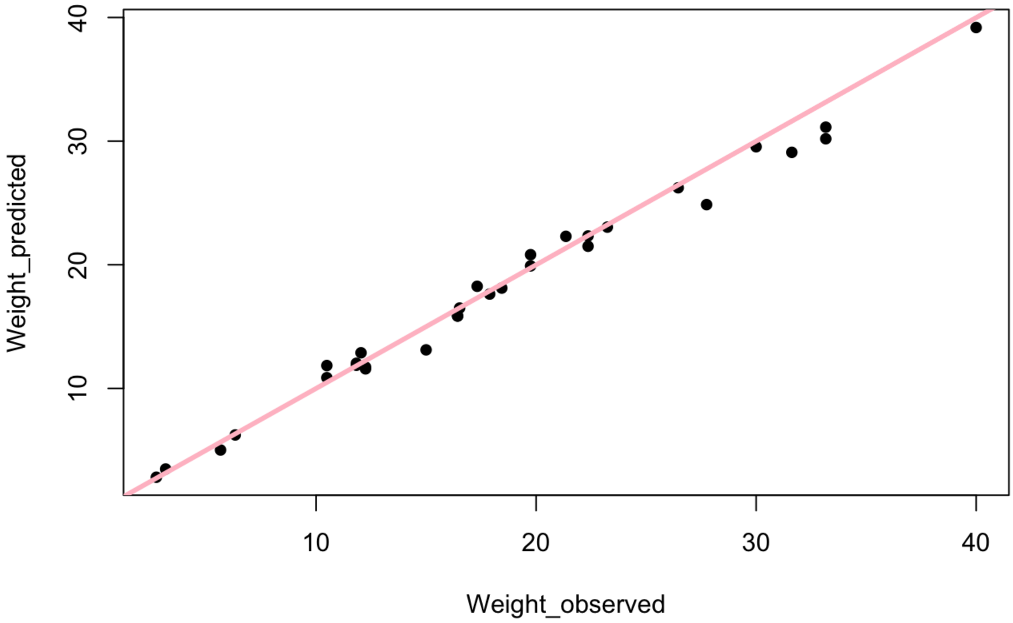
```
## [1] 0.9785510 1.3047550 0.9036460 0.1464519
```



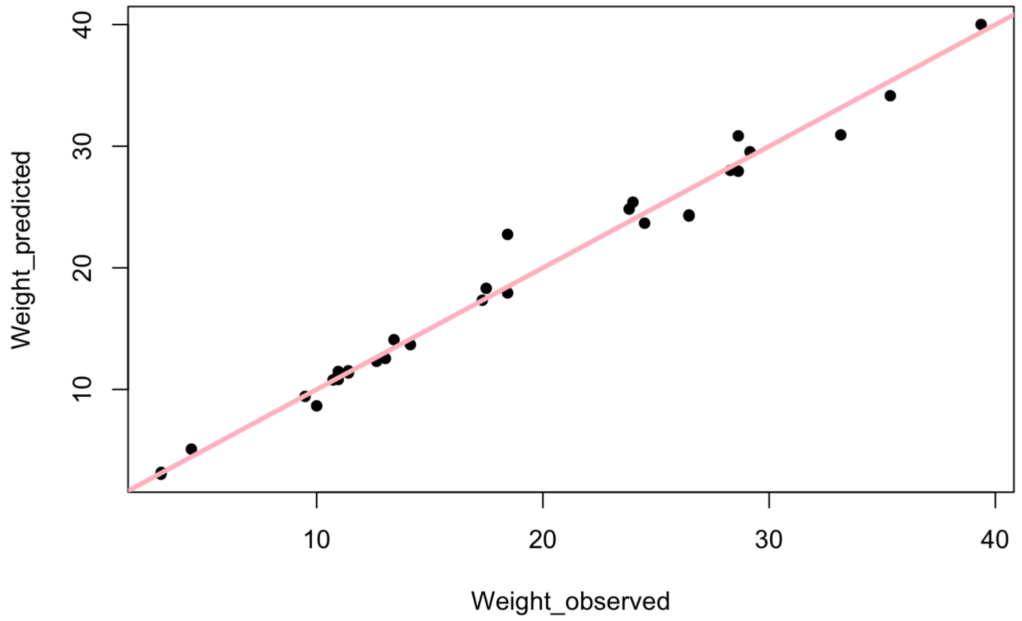
```
## [1] 0.9901570 1.3711072 0.8125829 0.1443306
```



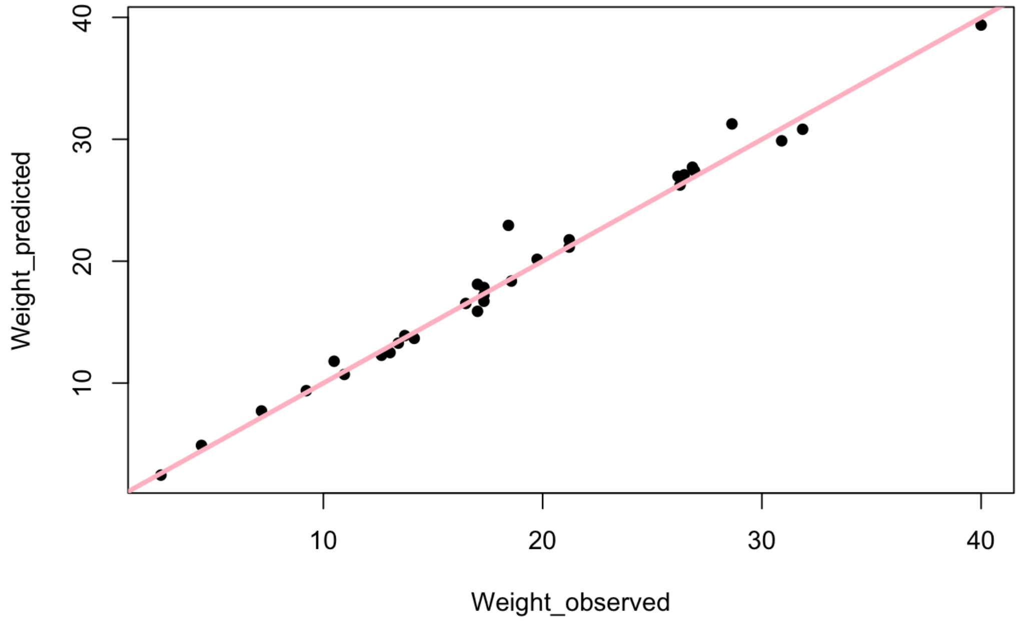
```
## [1] 0.9887623 1.1461729 0.8029870 0.1238623
```



```
## [1] 0.9828694 1.2520063 0.8433513 0.1287560
```



```
## [1] 0.9835722 1.1115565 0.7058304 0.1307944
```



We calculate the r squared-R.sq, root mean squared prediction error-RMSPE, maximum absolute prediction error-MAPE, and the ratio of RMSPE and standard deviation. Also plot the Weight observed vs Weight predicted using training data. Use these values and fitting plot to see how good this prediction performed.

Conclution:

According to plots and these value, the R.sq values all larger than 0.98, and the ratio of RMSPE and standard deviation are so close to 0, they are all around 0.1. That is really good performance, which means the prediction is doing well. We can conclude that the inear regression model is appropriate for this data set, and prediction of weight can be performed very well.