Fish Market Data Exploration & Analysis

Keith Kutsuma

9/30/2020

The purpose of this dataset is to predict the weight of fish based on visual cues of the fish. This will allow for farmers to predict the weights of the fish. In addition to biologist to estimate weight of fish to determine the health of the species. For example when salmon cross dams counting and estimating weight may help scientist protect the species.

Step 1: Initialize packages & Directory

```
library(tidyverse)
library(plotly)
library(heatmaply)
library(kableExtra)

current_path = rstudioapi::getActiveDocumentContext()$path
setwd(dirname(current_path ))
# print( getwd() )
```

Step 2: Load Dataset

After loading the dataset we'll look at the variable's, their meanings and types.

```
ds <- read.csv("Fish.csv", stringsAsFactors = T)

# Rename Species column name due to naming issues.
names(ds)[1] <- "Species"

summary(ds)
```

```
##       Species       Weight          Length1         Length2
##  Bream    :35   Min.   :   0.0   Min.   : 7.50   Min.   : 8.40
##  Parkki   :11   1st Qu.: 120.0   1st Qu.:19.05   1st Qu.:21.00
##  Perch    :56   Median : 273.0   Median :25.20   Median :27.30
##  Pike     :17   Mean   : 398.3   Mean   :26.25   Mean   :28.42
##  Roach    :20   3rd Qu.: 650.0   3rd Qu.:32.70   3rd Qu.:35.50
##  Smelt    :14   Max.   :1650.0   Max.   :59.00   Max.   :63.40
##  Whitefish: 6
##     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
##
```

From the Journal of Statistics Education & Kaggle Dataset Creator

Variable	Description	Unit	Type
Species	Species of Fish	NA	character
Weight	Weight of the Fish	grams	double
Length1	Length from nose to beginning of tail	cm	double
Length2	Length from nose to notch of the tail	cm	double
Length3	Length from nose to end of the tail	cm	double
Height	Height of the fish	cm	double
Width	Width of the fish	cm	double

Step 3: Observe Data

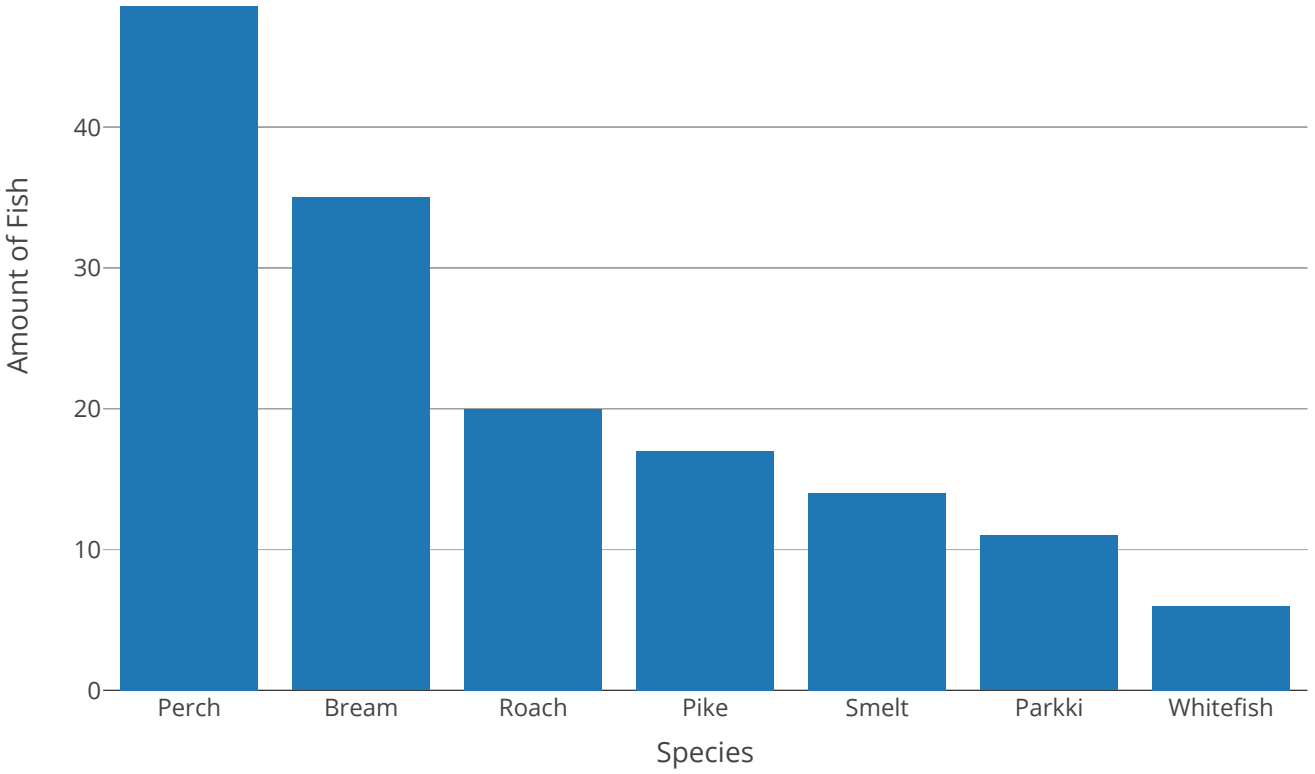
For this step we will look at

- Mean, Median, mode of certain categories
- Observe outliers in certain categories
- Observe graphs of data to Weight

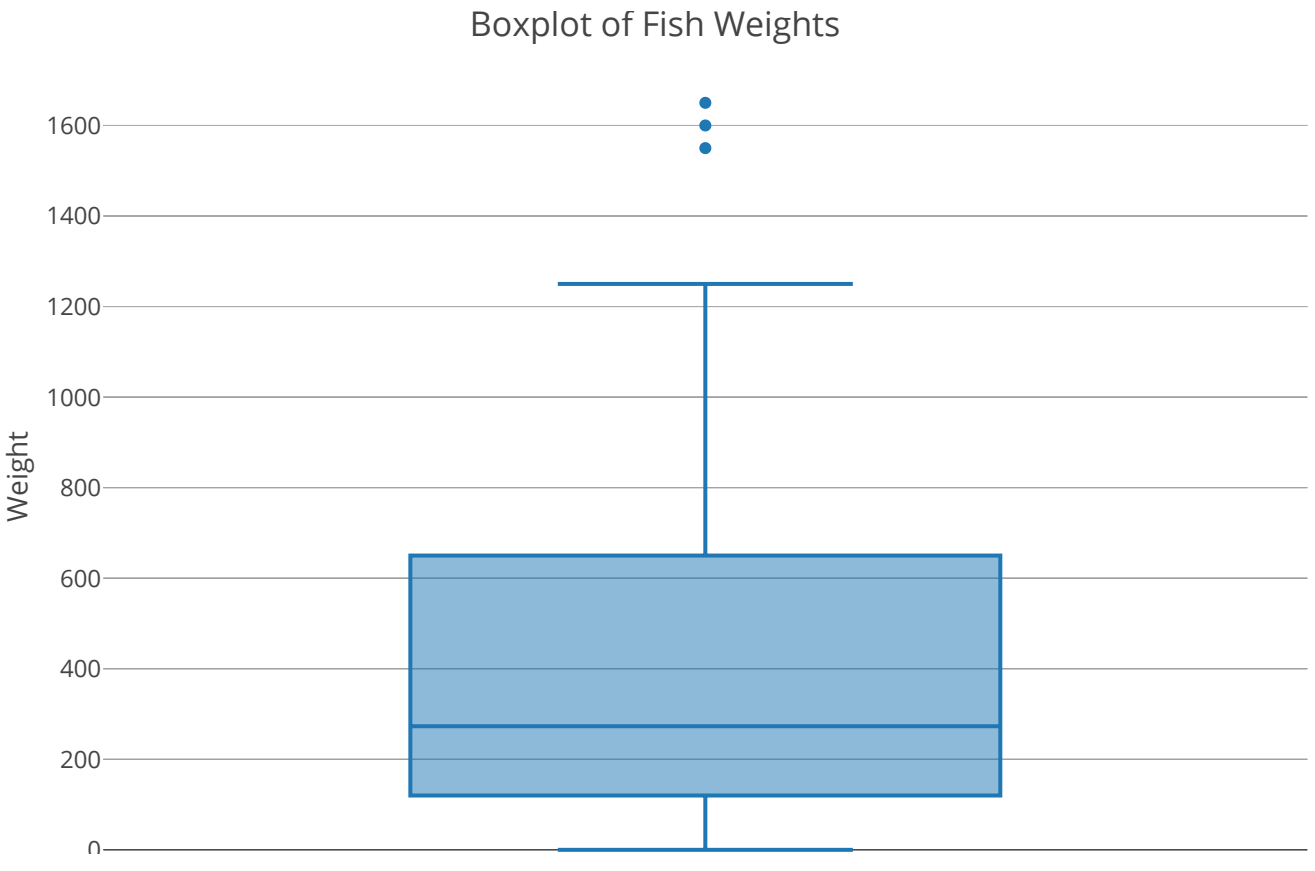
```
# Allow for unique identification of each fish
ds <- ds %>%  mutate(UID = 1:159)
```

```
# Set factor order based on amount in summary
ds$Species <- factor(ds$Species, level = c("Perch", "Bream", "Roach", "Pike", "Smelt", "Parkki",
"Whitefish"))

species_bar <- plot_ly(ds,
                       x = ~Species,
                       type = "histogram") %>%
              layout( title = "Count of Fish by Species",
                      yaxis = list(title = "Amount of Fish"))
species_bar
```

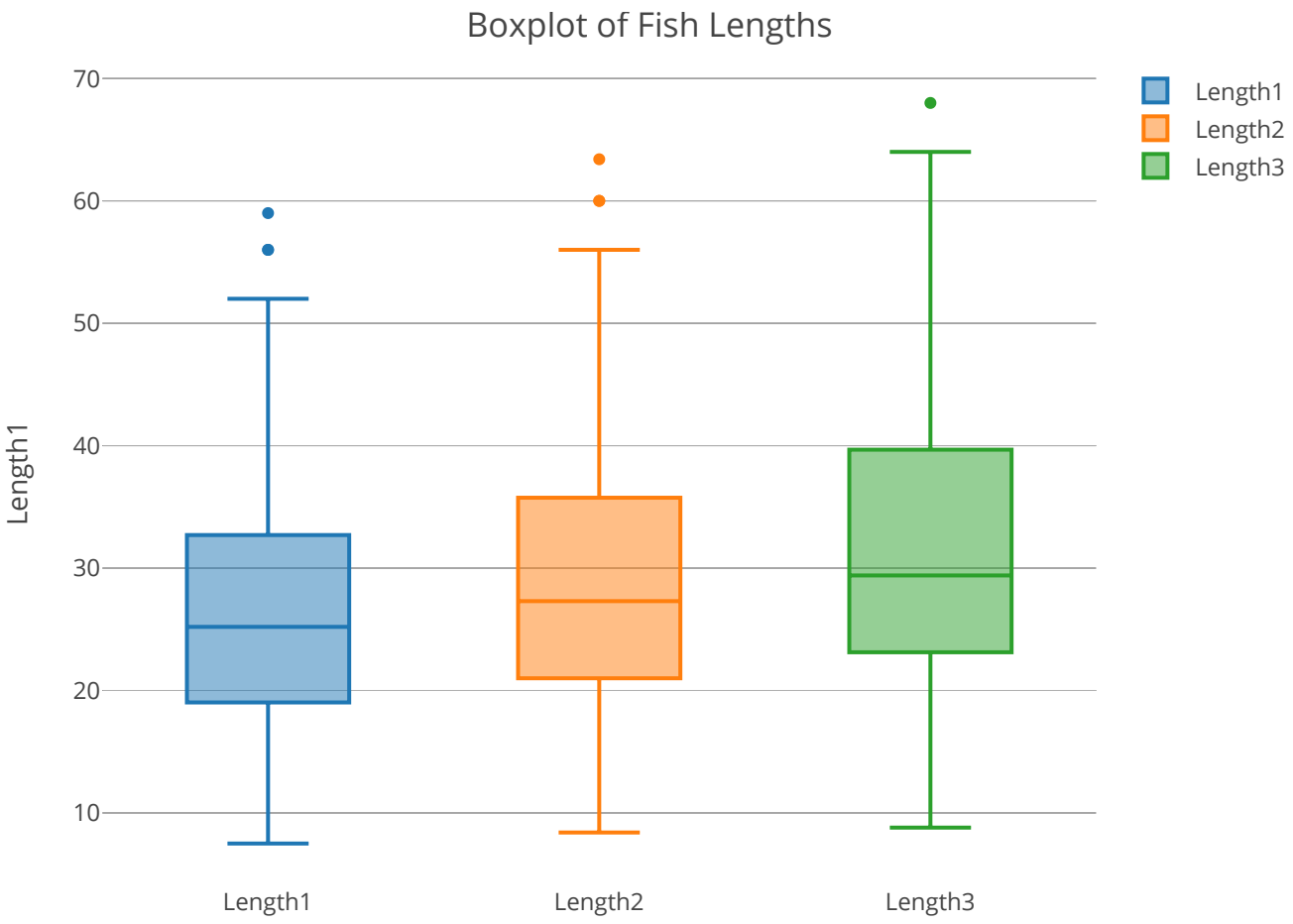



```
weight_box <- plot_ly(ds,
                      y = ~Weight,
                      type = "box",
                      text = paste0("UID: ", ds$UID,"\nWeight: ", ds$Weight," (g)"),
                      name = "Weight") %>%
              layout(title = "Boxplot of Fish Weights")
weight_box
```



Weight

```
length_box <- plot_ly(ds,
                      y = ~Length1,
                      type = "box",
                      text = paste0("UID: ", ds$UID,"\nLength: ", ds$Length1," (cm)"),
                      name = "Length1") %>%
              add_trace(y = ~Length2,
                        name = "Length2",
                        text = paste0("UID: ", ds$UID,"\nLength: ", ds$Length2," (cm)"),
                        name = "Length2") %>%
              add_trace(y = ~Length3,
                        name = "Length3",
                        text = paste0("UID: ",
                                      ds$UID,"\nLength: ",
                                      ds$Length3," (cm)"),
                        name = "Length3") %>%
              layout(title = "Boxplot of Fish Lengths")
length_box
```

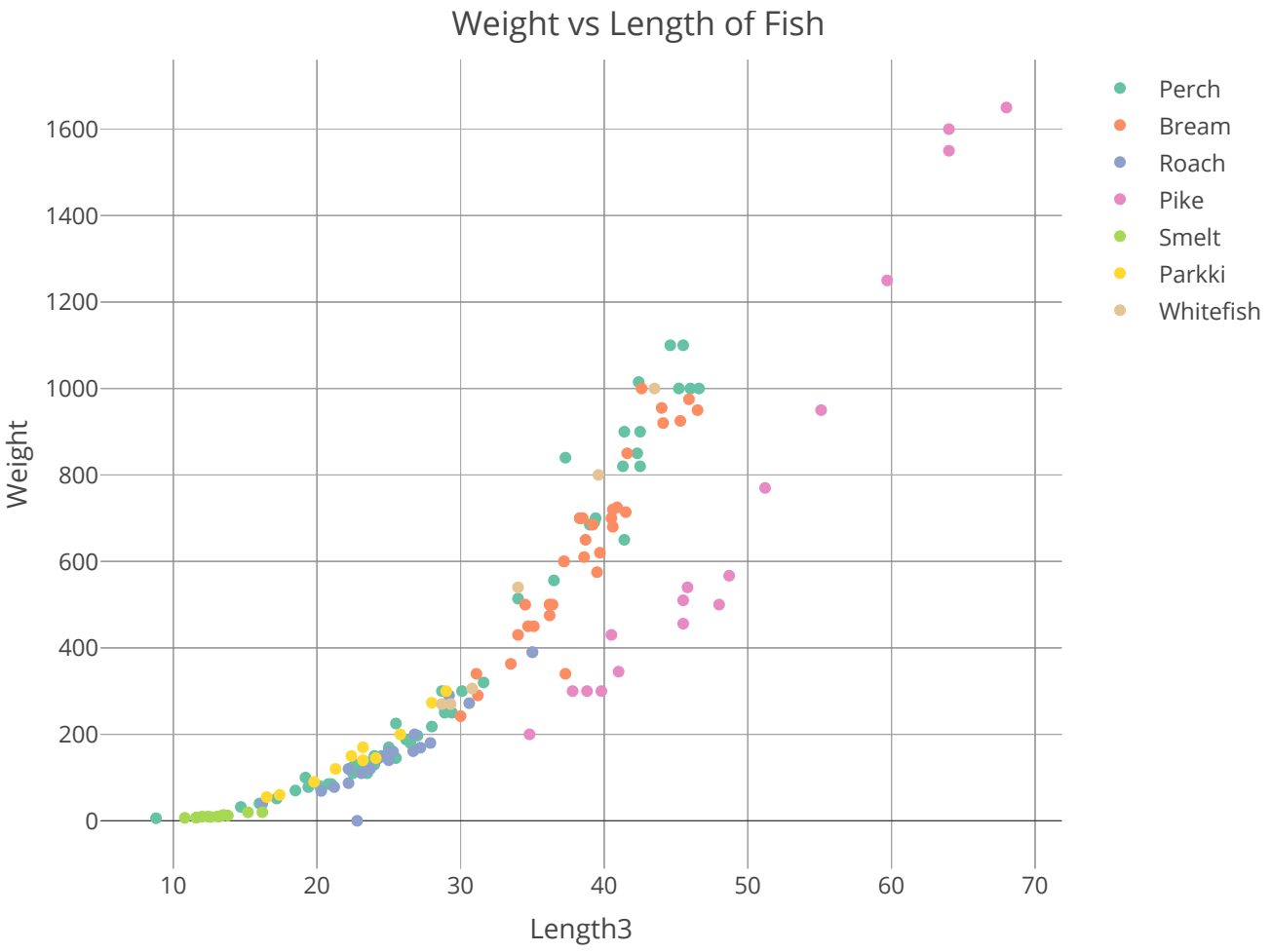


Looking at these boxplots, it looks as if there are outliers in the fish with UID's of 143, 144 and 145. I would like to see what these points look like on a graph.

I am going to create a dotplot using Length3 and Weight. My reason for Length3 is because Length3 if the length of the whole fish, resulting with less bias than using only part of the fish's length.

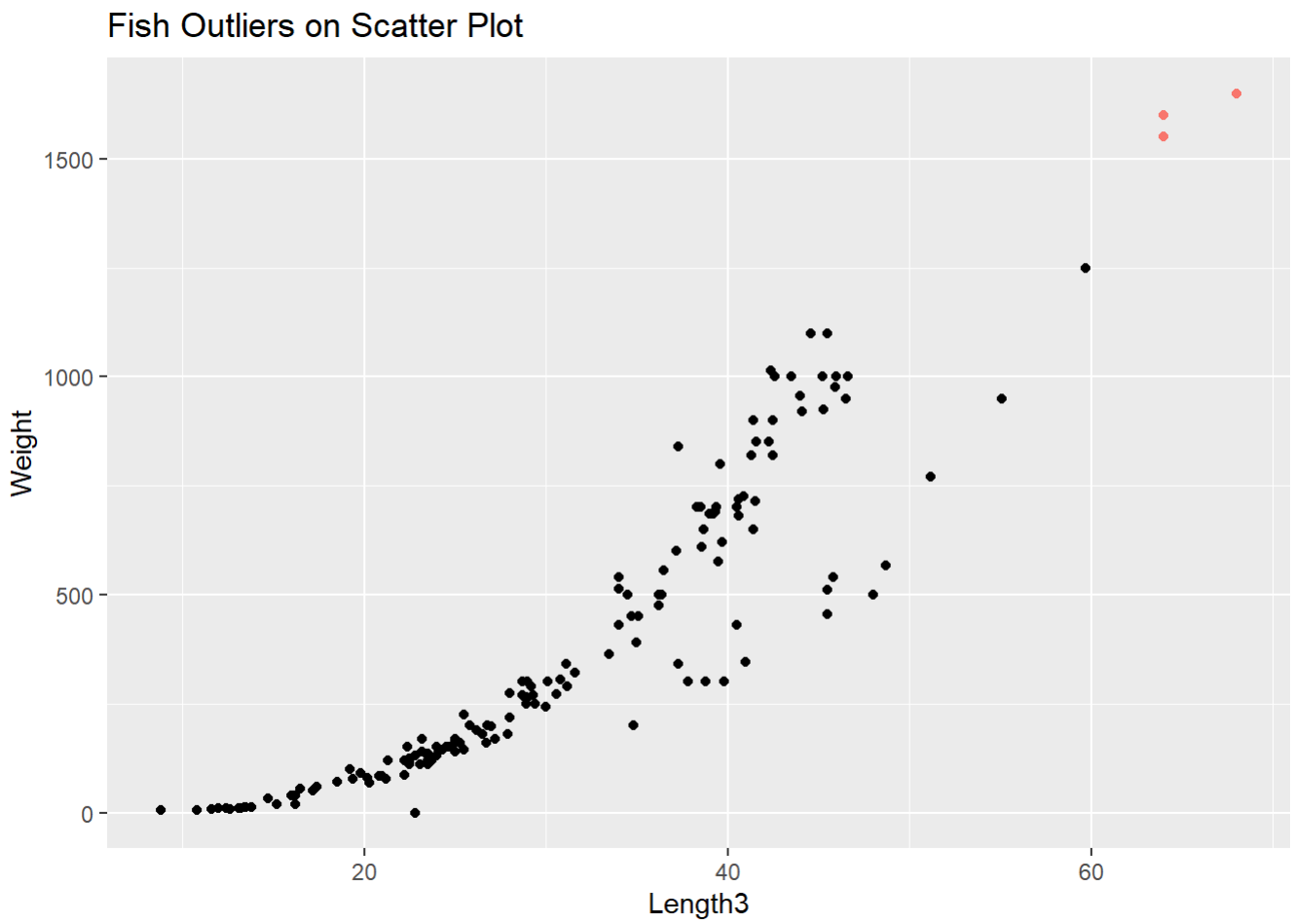
```
outliers <- ds %>%
            filter(UID > 142) %>%
            filter(UID < 146)

length3_dotplot <- plot_ly(ds,
                           y = ~Weight,
                           x = ~Length3,
                           color = ~Species,
                           text = ~paste("Species: ", Species,"<br>Weight: ", Weight, "( g)",
                                         "<br>Length: ", Length3, " (cm)",
                                         "<br>UID: ", UID)) %>%
                  layout(title = "Weight vs Length of Fish")
length3_dotplot
```




```
weight_cross_outlier <- ggplot(ds, aes(y = Weight, x = Length3)) +
                        geom_point() +
                        geom_point(data = outliers, aes(y = Weight, x = Length3, colour = "red"
)) +
                        ggtitle("Fish Outliers on Scatter Plot") +
                        theme(legend.position = "none")

weight_cross_outlier
```



As we can see There are three pike that are much larger than the rest of the fish, so we are going to remove them from the dataset. In addition When looking at the box plots there is a fish, *(uid 41)*, with zero weight, so this fish will also be removed.

Step 4: Data Transformation.

```
ds_trimm <- ds %>%
  filter(UID != 143 & UID != 144 & UID != 145 & UID != 41)

# Write trimmed dataframe to csv
# write_csv(ds_trimm, "Fish_trimmed.csv")
```

Step 5: Correlation

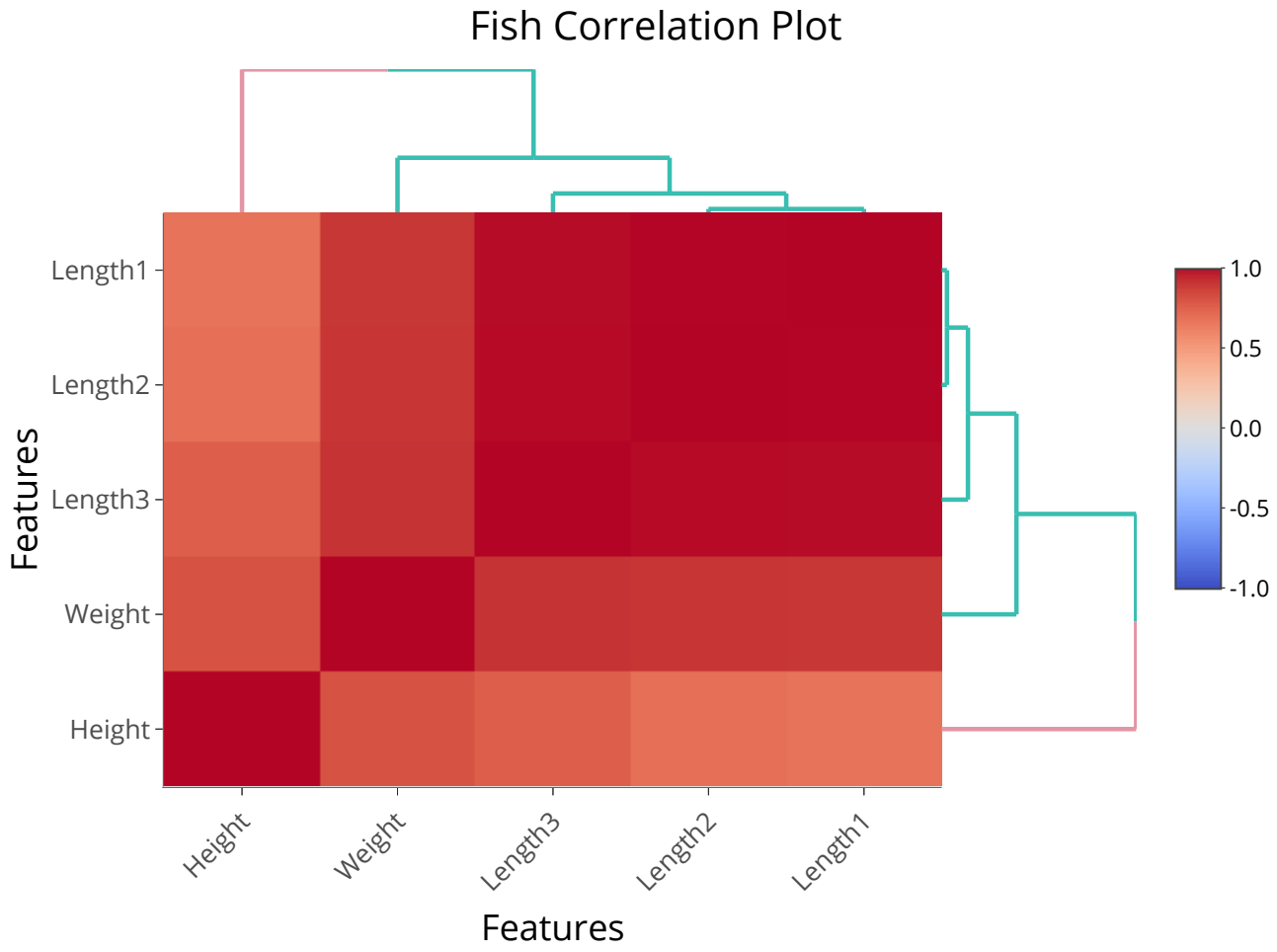
```
# Correlation of non-categorical/UID variables
corr_data <- cor(ds_trimm[,6:2], use = "everything")
head(corr_data)
```

```
##            Height   Length3   Length2   Length1    Weight
## Height  1.0000000 0.7554163 0.6911658 0.6754999 0.8019538
## Length3 0.7554163 1.0000000 0.9930258 0.9905800 0.9073728
## Length2 0.6911658 0.9930258 1.0000000 0.9994169 0.8997339
## Length1 0.6754999 0.9905800 0.9994169 1.0000000 0.8957401
## Weight  0.8019538 0.9073728 0.8997339 0.8957401 1.0000000
```

```
# Correlation to Weight
print(corr_data[,5])
```

```
##    Height   Length3   Length2   Length1    Weight
## 0.8019538 0.9073728 0.8997339 0.8957401 1.0000000
```

```
heatmaply_cor(corr_data, xlab = "Features", ylab = "Features", k_col = 2, k_row = 2, main = "Fis
h Correlation Plot")
```



Here we have the correlation data, and since we are predicting weight those are the numbers we are looking at. Based on this data it can help determine which variables we want to use depending on the model and model types.