```r
#Copyright (c) 2018 Jouke Profijt.
#Licensed under GPLv3. See LICENSE
```

```r
BirdBones <- read.csv("../data/bird.csv",header = T, sep = ",")
#respective collums for the lenght and diameter
length <- c(2,4,6,8,10)
diameter <- c(3,5,7,9,11)
```

Introduction

Research Question

What bone or group of bones that most birds have in common, is the most significant for the function in the diffrent ecological groups?

Data

```
Data recieved from:
```

Birds' Bones and Living Habits, Kaggle dataset

```
Bone measurements were measured from a skeleton collection of
Natural History Museum of Los Angeles County,
provided by Dr. D. Liu of beijing Museaum of Natural History
```

Exploratory Data Analyses

The data contains 420 bird samples where the bone lengths and diameters have been measured. The birds are separated in 6 diffrent groups:

- Swimming Birds, SW
- Wading Birds, W
- Terrestrial Birds, T
- Raptors, R
- Scansorial Birds, P
- Singing Birds, SO

Most samples have data for:

- Length and Diameter of the Humerus
- Length and Diameter of the Ulna
- Length and Diameter of the Femur
- Length and Diameter of the Tibiotarsus
- Length and Diameter of the Taesometatarsus

I'm creating a graph which displays the bonelengths on y axis an the Id on x colorcoded by their ecological group. by evaluating this we can see if some groups have overall larger or smaller bones and we see if there are big outliers.

```r
# this omits several ggplot2 errors retaining to mising values
BirdBones.noNA <- BirdBones[complete.cases(BirdBones),]
```

```r
# Displaing the data frame structure and a small summary
str(BirdBones)
```

```
## 'data.frame':    420 obs. of  12 variables:
##  $ id   : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ huml : num  80.8 88.9 80 77.7 62.8 ...
##  $ humw : num  6.68 6.63 6.37 5.7 4.84 ...
##  $ ulnal: num  72 80.5 69.3 65.8 52.1 ...
##  $ ulnaw: num  4.88 5.59 5.28 4.77 3.73 3.47 4.5 4.55 6.13 7.05 ...
##  $ feml : num  41.8 47 43.1 40 34 ...
##  $ femw : num  3.7 4.3 3.9 3.52 2.72 4.41 3.41 3.78 5.45 7.44 ...
##  $ tibl : num  5.5 80.2 75.3 69.2 56.3 ...
##  $ tibw : num  4.03 4.51 4.04 3.4 2.96 2.73 3.56 3.81 5.58 7.31 ...
##  $ tarl : num  38.7 41.5 38.3 35.8 31.9 ...
##  $ tarw : num  3.84 4.01 3.34 3.41 3.13 2.83 3.64 3.81 4.37 6.34 ...
##  $ type : Factor w/ 6 levels "P","R","SO","SW",..: 4 4 4 4 4 4 4 4 4 4 ...
```

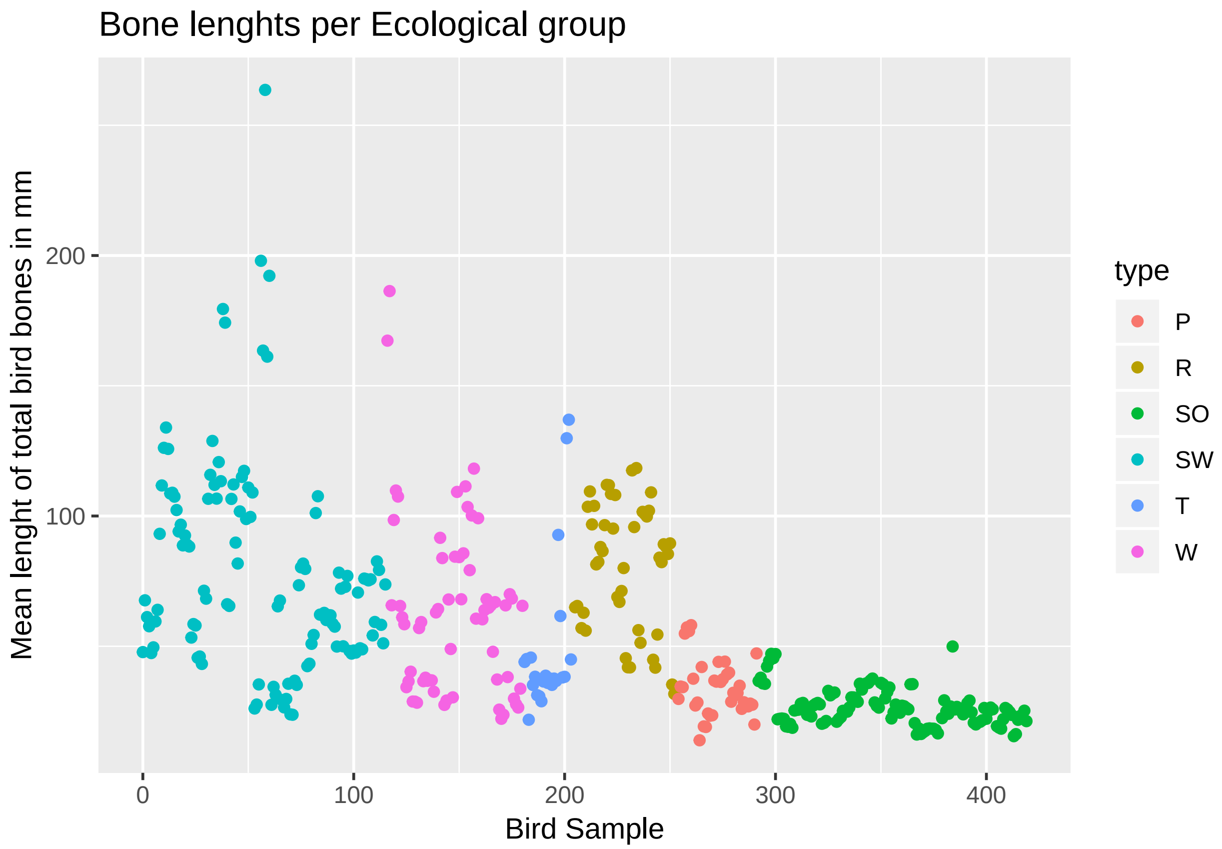
```r
summary(BirdBones)
```

```
##        id             huml             humw            ulnal       
##  Min.   :  0.0   Min.   :  9.85   Min.   : 1.140   Min.   : 14.09  
##  1st Qu.:104.8   1st Qu.: 25.17   1st Qu.: 2.190   1st Qu.: 28.05  
##  Median :209.5   Median : 44.18   Median : 3.500   Median : 43.71  
##  Mean   :209.5   Mean   : 64.65   Mean   : 4.371   Mean   : 69.12  
##  3rd Qu.:314.2   3rd Qu.: 90.31   3rd Qu.: 5.810   3rd Qu.: 97.52  
##  Max.   :419.0   Max.   :420.00   Max.   :17.840   Max.   :422.00  
##                  NA's   :1        NA's   :1        NA's   :3       
##      ulnaw             feml             femw             tibl       
##  Min.   : 1.000   Min.   : 11.83   Min.   : 0.930   Min.   :  5.50  
##  1st Qu.: 1.870   1st Qu.: 21.30   1st Qu.: 1.715   1st Qu.: 36.42  
##  Median : 2.945   Median : 31.13   Median : 2.520   Median : 52.12  
##  Mean   : 3.597   Mean   : 36.87   Mean   : 3.221   Mean   : 64.66  
##  3rd Qu.: 4.770   3rd Qu.: 47.12   3rd Qu.: 4.135   3rd Qu.: 82.87  
##  Max.   :12.000   Max.   :117.07   Max.   :11.640   Max.   :240.00  
##  NA's   :2        NA's   :2        NA's   :1        NA's   :2       
##       tibw             tarl             tarw         type   
##  Min.   : 0.870   Min.   :  7.77   Min.   : 0.660   P : 38  
##  1st Qu.: 1.565   1st Qu.: 23.04   1st Qu.: 1.425   R : 50  
##  Median : 2.490   Median : 31.74   Median : 2.230   SO:128  
##  Mean   : 3.182   Mean   : 39.23   Mean   : 2.930   SW:116  
##  3rd Qu.: 4.255   3rd Qu.: 50.25   3rd Qu.: 3.500   T : 23  
##  Max.   :11.030   Max.   :175.00   Max.   :14.090   W : 65  
##  NA's   :1        NA's   :1        NA's   :1
```

there are 420 total measurements, and by using complete cases i found that there are 413 measurements which are complete and do not contain missing values, aka > there are 7 measurements that contain missing values.

```r
library(ggplot2)
library(reshape)
source("../scripts/BoneMeans.R")
BirdBones.noNA <- BoneMeans(data = BirdBones.noNA, length = length, diameter = diameter)
ggplot(data = BirdBones.noNA, aes(id, length.mean, colour = type)) +
  ggtitle("Bone lenghts per Ecological group")+
  ylab("Mean lenght of total bird bones in mm") +
```

```r
  xlab("Bird Sample")+
  geom_point()
```

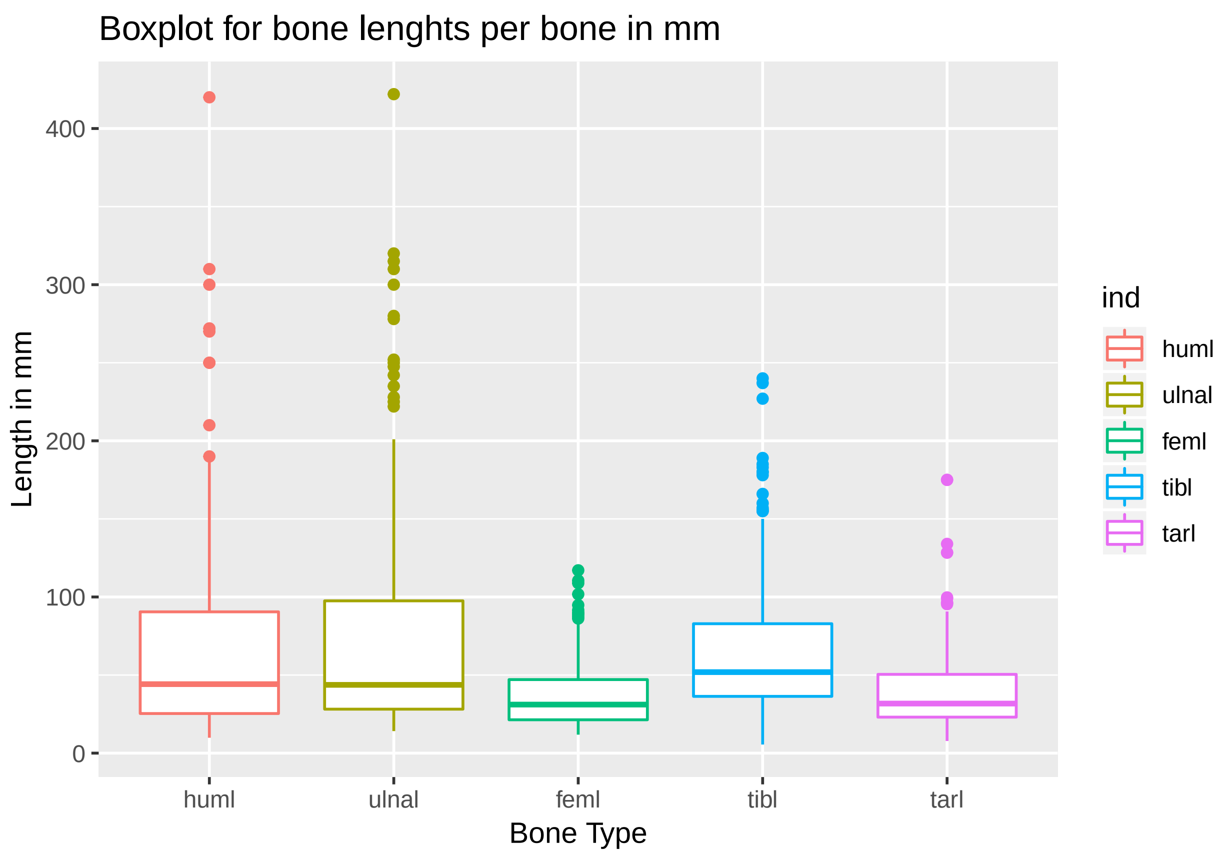


As seen above swimming birds have the biggest bones, but also shown is that there are a lot more samples in that group where there is a lot of variation. I can look into cleaning up the data and removing the biggest outliers in this group. Singing birds also have a lot of samples but there is much less variation and so more certanty.

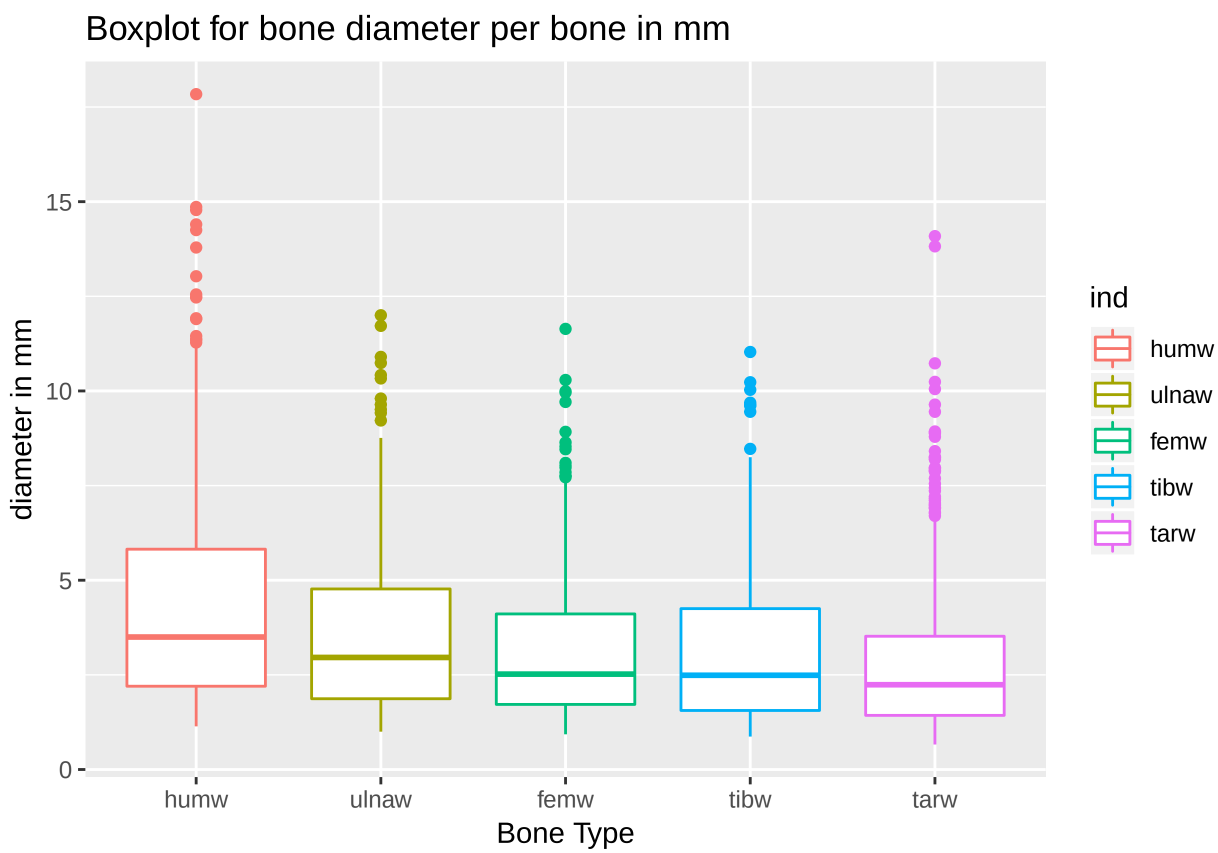
For the rest of the birds there are not a lot of sample so maby we could try and normalizing the data so there is an even amount of samples per group.

There are also 7 samples that contain missing values, we could just straight out not use these samples becouse 4 of these are part of the biggest group of samples. and the others are not part of the smallest groups.

```r
library(ggplot2)

ggplot(stack(BirdBones.noNA[length]), aes(x = ind, y = values, color = ind)) +
  geom_boxplot()+
  ggtitle("Boxplot for bone lenghts per bone in mm")+
  xlab("Bone Type")+
  ylab("Length in mm")
```




```
ggplot(stack(BirdBones.noNA[diameter]), aes(x = ind, y = values, color = ind)) +
  geom_boxplot()+
  ggtitle("Boxplot for bone diameter per bone in mm")+
  xlab("Bone Type")+
  ylab("diameter in mm")
```

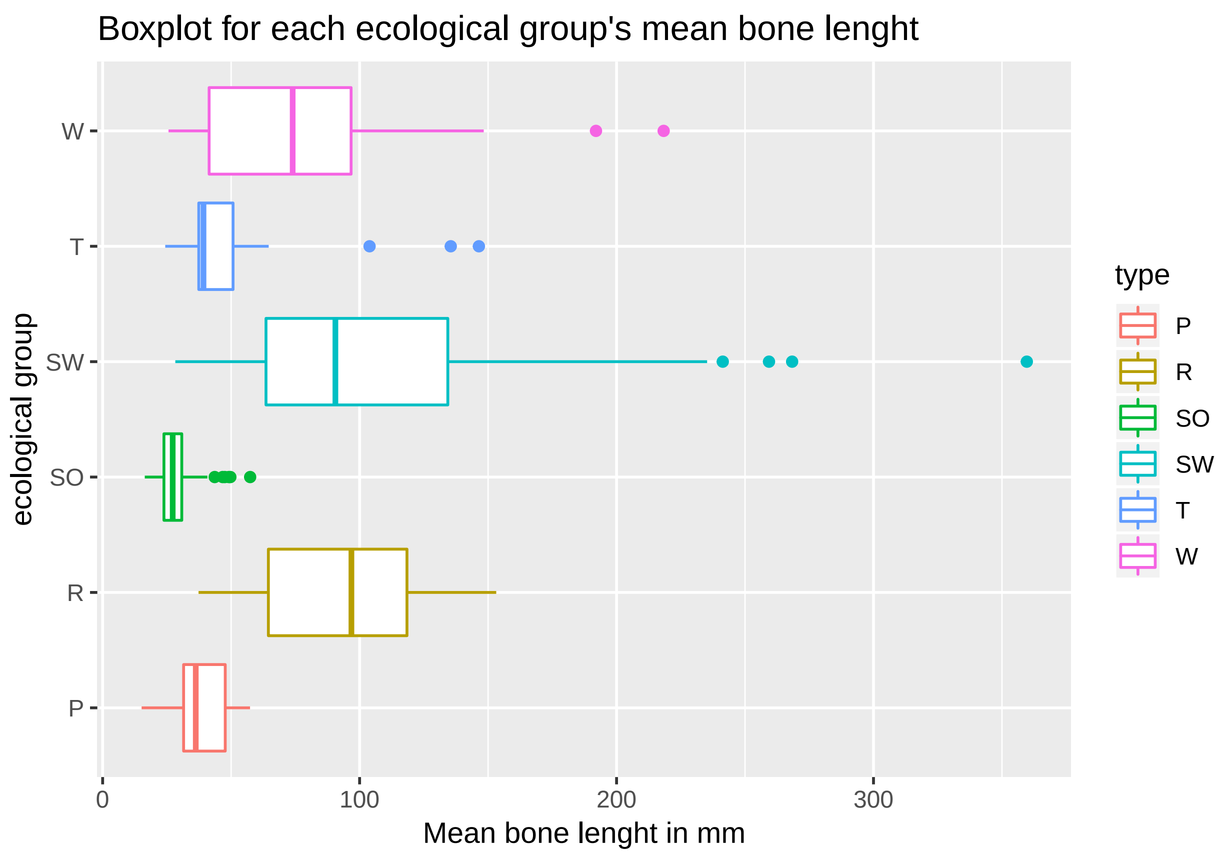


What we see above is that there are a considerable amount of outliers between the bones themselves, but this was expected as they are from diffrent groups and the diffrent groups dont have the same amount of measurements. below i will do a comparison between the group bone mean lengths which will show outliers in their respective group. using the above boxplots we can maby see which bones are not very important > see if they don't differ at all wich means we dont need them that much for classification.

As we can see the femur lenght and taesometatarsus length do not contain a lot of variation and maby are candidates for exclution from analysis.

```r
# diameter & lenght indexes for only the longer bones.
length.long <- c(2, 4, 8)
diameter.long <- c(3, 5, 9)
BirdBones.noNA.long <- BoneMeans(BirdBones.noNA, length.long, diameter.long)
```

```r
library(ggplot2)

ggplot(BirdBones.noNA.long, aes(x = type, y = length.mean, color = type)) +
  geom_boxplot()+
  coord_flip()+
  ggtitle("Boxplot for each ecological group's mean bone lenght")+
  ylab("Mean bone lenght in mm")+
  xlab("ecological group")
```

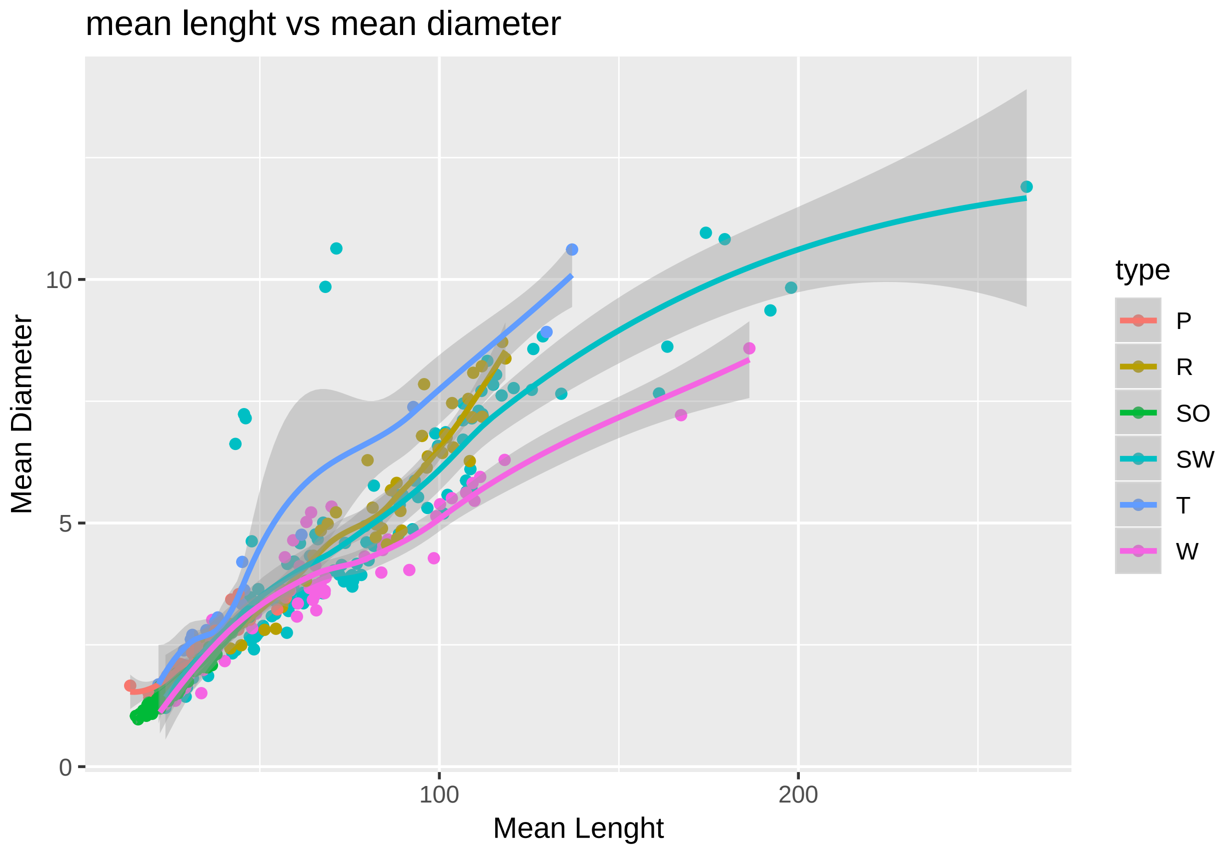


As you can see there are quite a few outliers in all groups except in group R, The raptors. but we saw in the above boxplot that there were loads of outliers between all bones, yet here that is significantly reduced. so if we are going to inspect the date we have to look at them per group and NOT by bone type.

What we can also see in these plots are which birds are most likely the largest, as seen above color cyan or SW or Swimming Birds are the biggest of them all closely followed by W or Wading Birds

```
ggplot(BirdBones.noNA,aes(x=length.mean,y=diameter.mean,color=type))+
  geom_point()+
  geom_smooth()+
  ggtitle("mean lenght vs mean diameter")+
  xlab("Mean Lenght")+
  ylab("Mean Diameter")
```

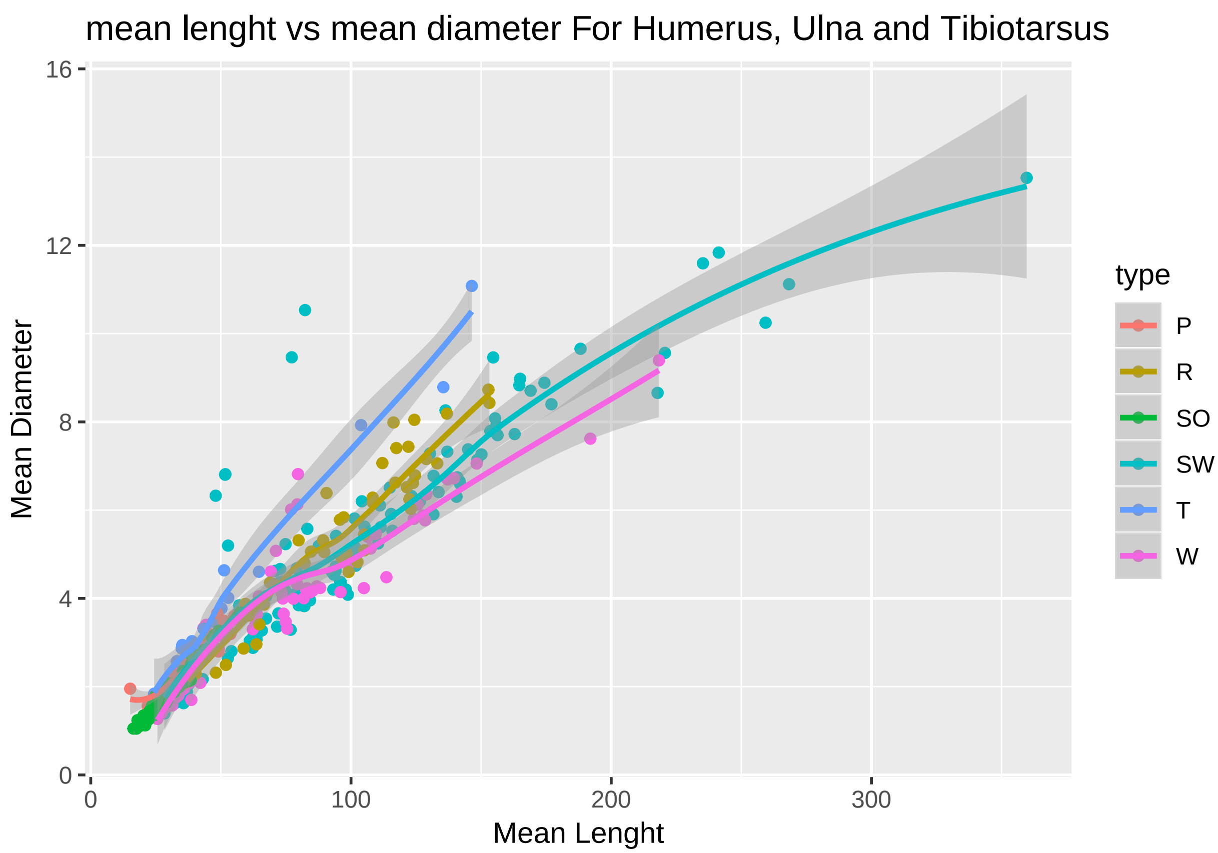
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



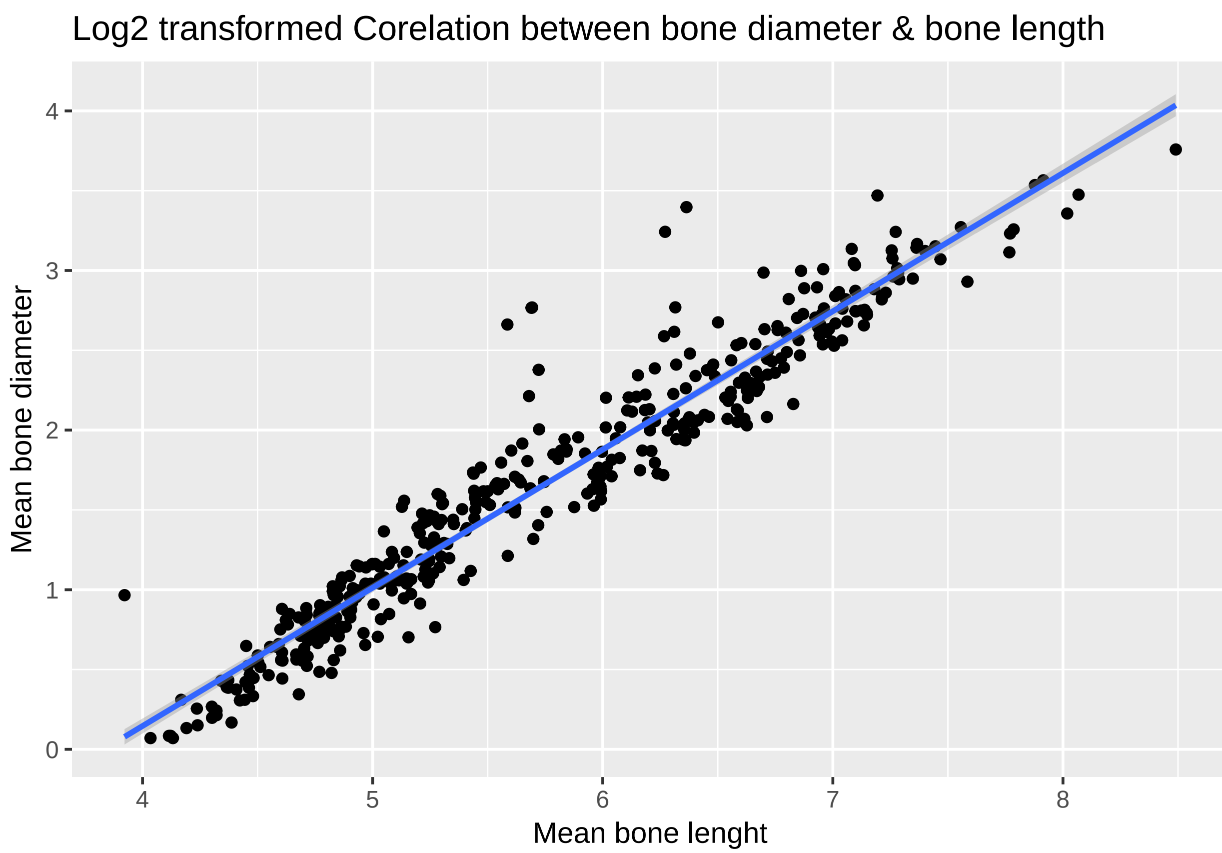
Untransformed datapoints separated by goup, again here we can see which birds are the biggest, but for smaller birds this plot is not very readable. we do see something odd, where T has a climbing line around lenght 50, other birds have a decreasing line. also Swimming Birds have some results that are very diffrent form their mean line.

```
ggplot(BirdBones.noNA.long,aes(x=length.mean,y=diameter.mean,color=type))+
  geom_point()+
  geom_smooth()+
  ggtitle("mean lenght vs mean diameter For Humerus, Ulna and Tibiotarsus")+
  xlab("Mean Lenght")+
  ylab("Mean Diameter")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



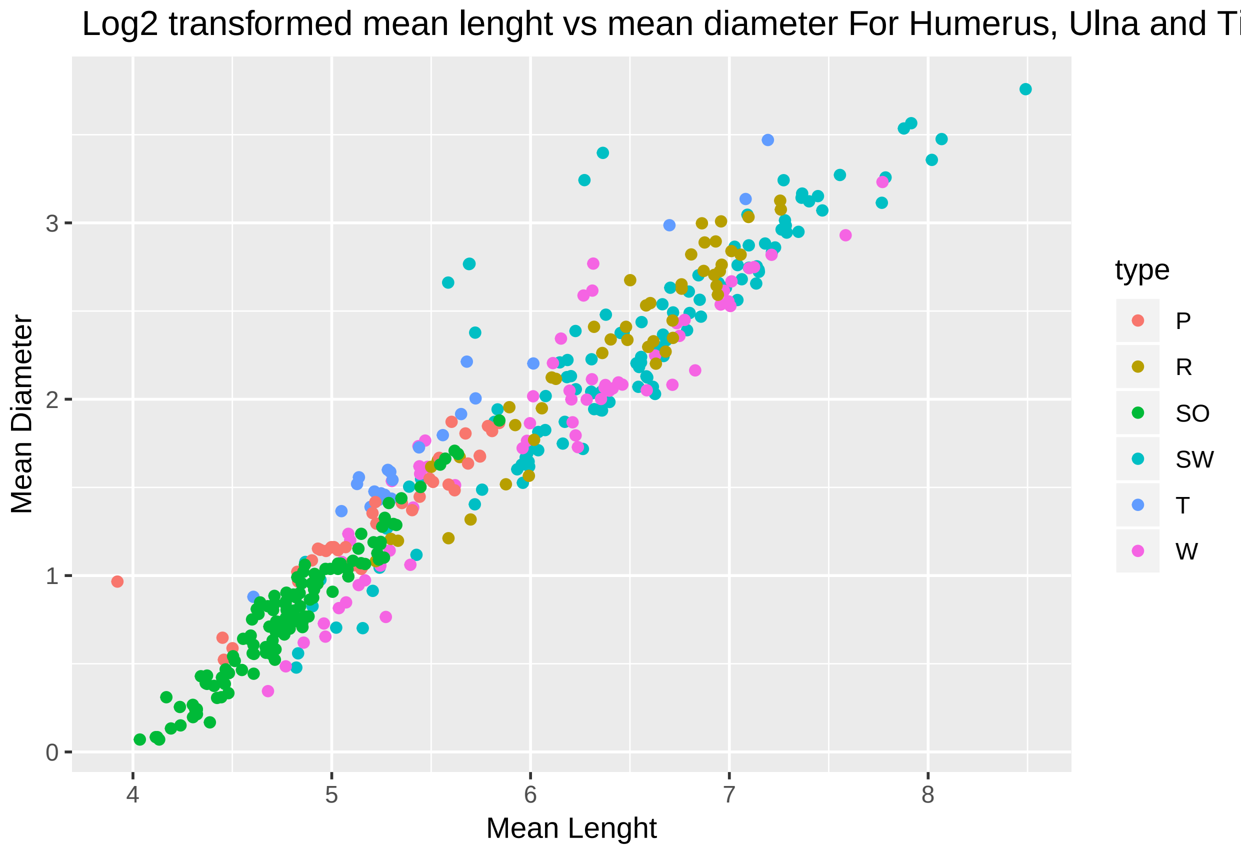

```
BirdBones.noNA.long$log2length <- log2(BirdBones.noNA.long$length.mean)
BirdBones.noNA.long$log2diameter <- log2(BirdBones.noNA.long$diameter.mean)

library(ggplot2)
ggplot(BirdBones.noNA.long, aes(x = log2length, y = log2diameter)) +
  geom_point()+
  geom_smooth(method = lm)+
  ggtitle("Log2 transformed Corelation between bone diameter & bone length")+
  xlab("Mean bone lenght")+
  ylab("Mean bone diameter")
```



As expected there is a coralation between the bone lenght and bone diameter, you can see this because the plot gives a liniar line. it does make a lot of sense if you have longer bones there you will most likely also have thicker bones(bigger diameters)

We can also see a couple of outliers in the scatter plot above. we can try and isolate these samples and take a closer look.

```
ggplot(BirdBones.noNA.long,aes(x=log2length,y=log2diameter,color=type))+
  geom_point()+
  ggtitle(" Log2 transformed mean lenght vs mean diameter For Humerus, Ulna and Tibiotarsus")+
  xlab("Mean Lenght")+
  ylab("Mean Diameter")
```

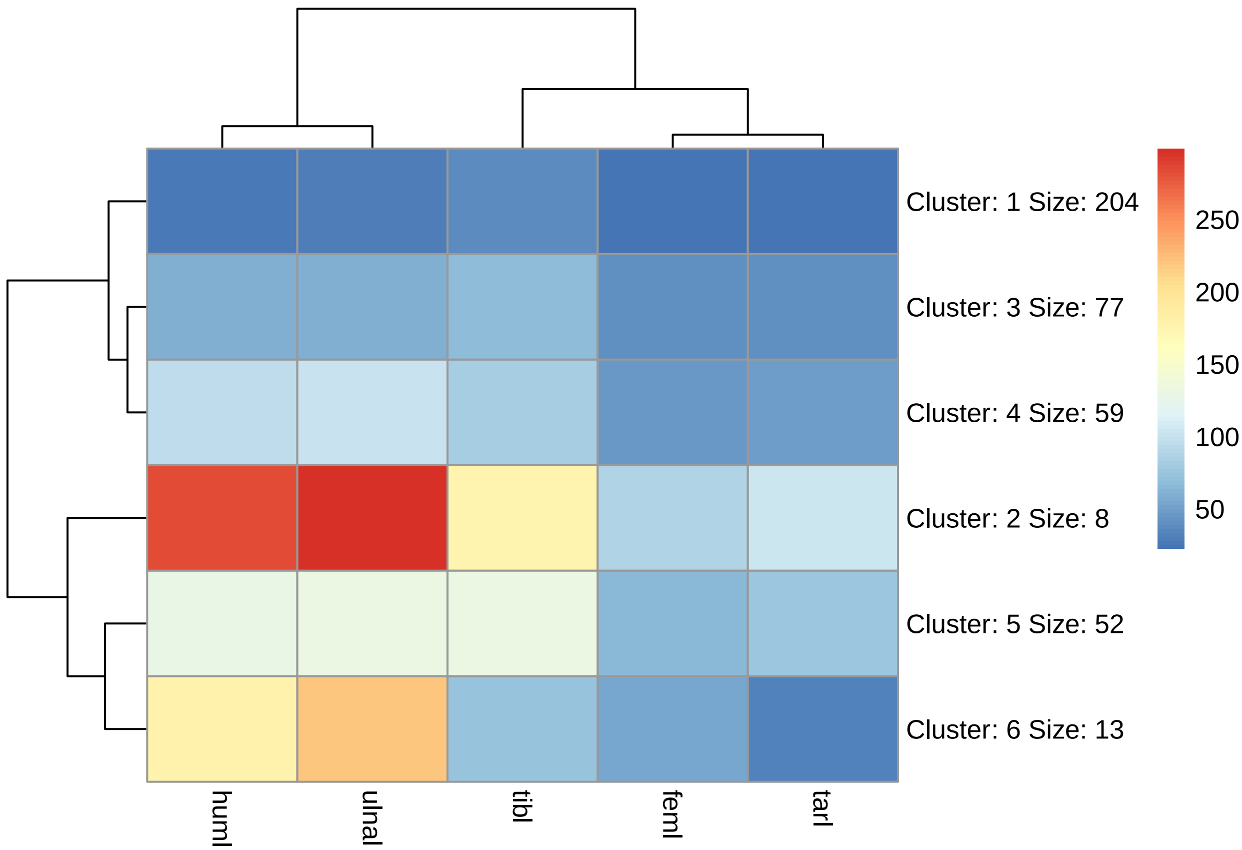


Same plot as above but colorcoded so we can see to which group the outliers belong.

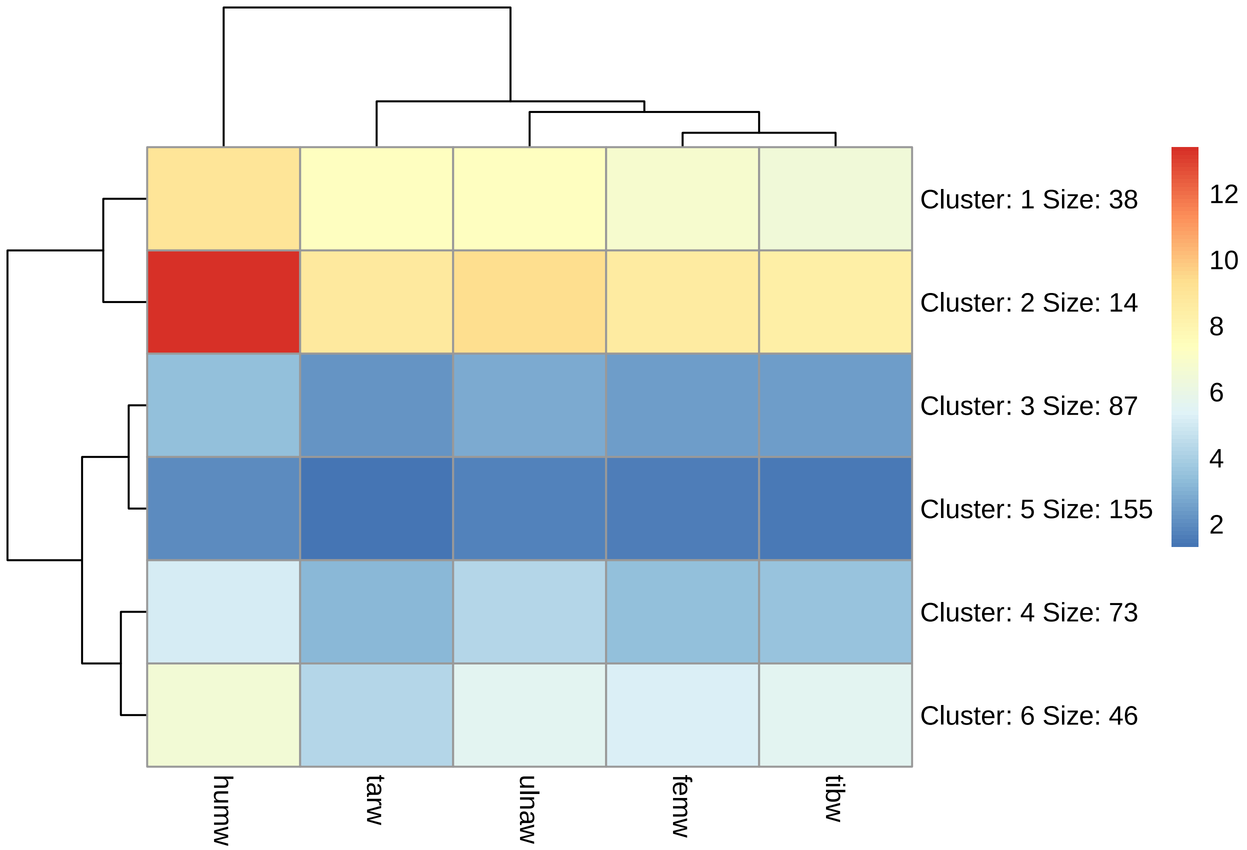
```
# m <- as.matrix(BirdBones.noNA$length.mean, ncol=2)
# 6 groups so 6 clusters is assumed
# cl <- kmeans(m, 6)
#
# ```
# ```{r}
# BirdBones.noNA$cluster <- factor(cl$cluster)
# centers <- as.data.frame((cl$centers))
# ```
# ```{r}
# library(ggplot2)
#
#
# ggplot(data=BirdBones.noNA, aes(x=length.me43an, y=id, color=type )) +
#  geom_point() +
#  geom_point(data=centers, aes(x=V1,y=V2, color='Center')) +
#  geom_point(data=centers, aes(x=V1,y=V2, color='Center'), size=50, alpha=.4, legend=FALSE)
```

```
library(ggplot2)
library(pheatmap)
df.hum <- data.frame(log2(BirdBones.noNA$huml), log2(BirdBones.noNA$humw))
kmeans.hum <- kmeans((df.hum), 6)

dm.len <- data.matrix(BirdBones.noNA[length])
dm.dia <- data.matrix(BirdBones.noNA[diameter])
```

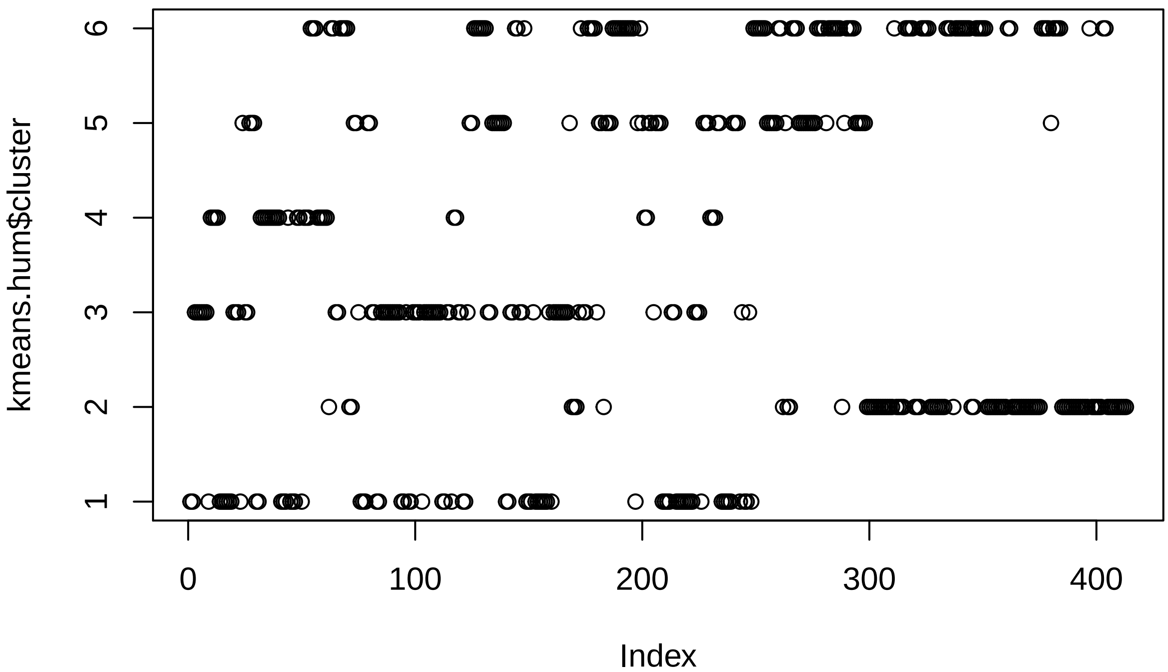
```r
pheatmap(dm.len, kmeans_k = 6)
```



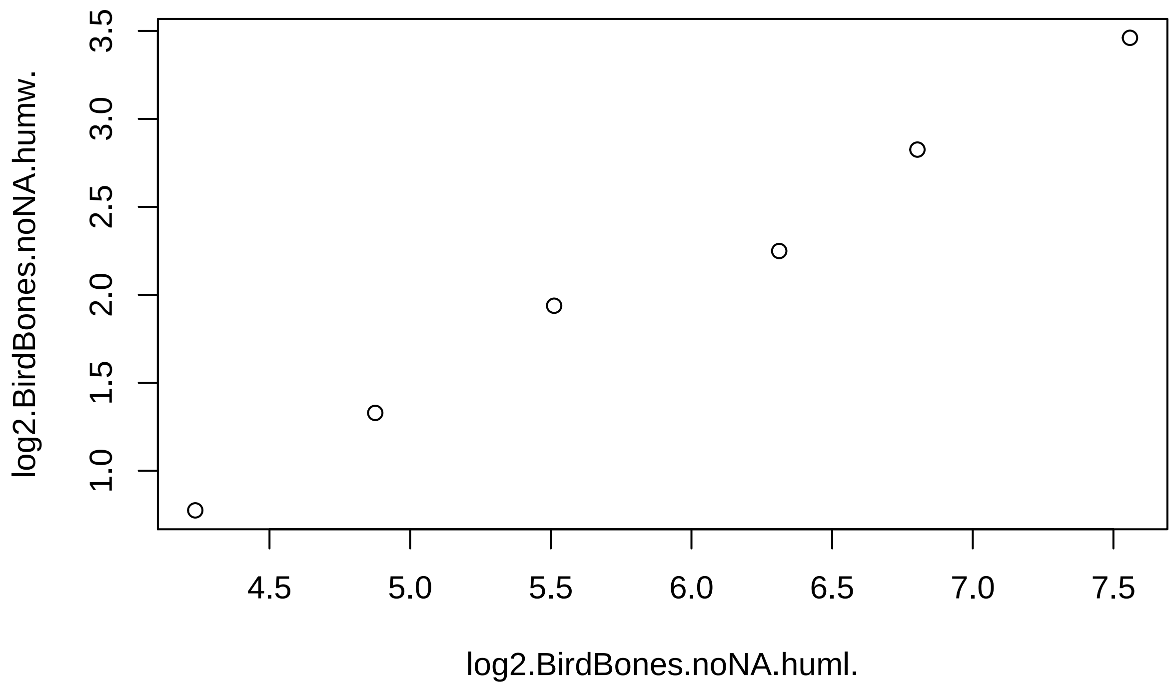

```r
pheatmap(dm.dia, kmeans_k = 6)
```




```r
plot(kmeans.hum$cluster)
```




```
plot(kmeans.hum$centers)
```



For data cleaning we already have a dataset without NA's(BirdBones.NoNA). now we need to remove the found outliers and discard the unneeeded bones.

```
huml.3rd.q <- 90.31
huml.1st.q <-25.17

out <- huml.1st.q - 1.5*(huml.3rd.q - huml.1st.q)
out.large <- huml.3rd.q + 1.5*(huml.3rd.q - huml.1st.q)
outliers <- subset(BirdBones.noNA, huml > out.large | huml < out)
Birdbones.Clean <- BirdBones.noNA[! BirdBones.noNA$id %in% outliers$id, ]
```

```
summary(Birdbones.Clean)
```

```
##        id             huml             humw            ulnal
##  Min.   :  0.0   Min.   :  9.85   Min.   : 1.140   Min.   : 14.09
##  1st Qu.:108.8   1st Qu.: 25.04   1st Qu.: 2.188   1st Qu.: 28.00
##  Median :213.5   Median : 42.49   Median : 3.440   Median : 42.74
##  Mean   :211.3   Mean   : 60.20   Mean   : 4.177   Mean   : 64.13
##  3rd Qu.:314.2   3rd Qu.: 88.93   3rd Qu.: 5.702   3rd Qu.: 95.17
##  Max.   :419.0   Max.   :188.00   Max.   :14.780   Max.   :280.00
##      ulnaw             feml             femw             tibl
##  Min.   : 1.000   Min.   : 11.83   Min.   : 0.930   Min.   :  5.50
##  1st Qu.: 1.867   1st Qu.: 21.23   1st Qu.: 1.690   1st Qu.: 36.05
##  Median : 2.910   Median : 30.43   Median : 2.475   Median : 51.06
##  Mean   : 3.473   Mean   : 35.76   Mean   : 3.106   Mean   : 62.38
##  3rd Qu.: 4.615   3rd Qu.: 45.40   3rd Qu.: 4.050   3rd Qu.: 80.27
##  Max.   :12.000   Max.   :117.07   Max.   :11.640   Max.   :227.00
```

```
##       tibw            tarl             tarw         type
##  Min.   : 0.870   Min.   :  7.77   Min.   : 0.660   P : 38
##  1st Qu.: 1.540   1st Qu.: 23.01   1st Qu.: 1.417   R : 48
##  Median : 2.440   Median : 31.43   Median : 2.210   SO:124
##  Mean   : 3.059   Mean   : 38.09   Mean   : 2.836   SW:108
##  3rd Qu.: 4.122   3rd Qu.: 48.28   3rd Qu.: 3.353   T : 23
##  Max.   :10.030   Max.   :175.00   Max.   :14.090   W : 63
##   length.mean     diameter.mean
##  Min.   : 13.90   Min.   : 0.972
##  1st Qu.: 27.26   1st Qu.: 1.753
##  Median : 39.53   Median : 2.709
##  Mean   : 52.11   Mean   : 3.330
##  3rd Qu.: 69.18   3rd Qu.: 4.359
##  Max.   :167.32   Max.   :10.636
```

```r
long.bones <- c(1, 2,3, 4,5,8,9, 12)
Birdbones.Clean <- Birdbones.Clean[,long.bones ]
```

```r
write.csv(Birdbones.Clean, "../data/CleanData.csv")
```

After creating the csv file im going to use weka to create an arff, i know there is a write.arff function but i can't get that to install properly.

Weka Analysis

For this classification we want the accuracy to be as high as possible, as wrongly classified fossils dont have that big of an impact as if someones health is on the line.

Using ZeroR We get 30% guessed correctly. It looks for the values with the largest sample count which is SO and guesses that its most likely that any bird is that bird. Zero R : Zero Rules.

ZeroR predicts class value: SO

- Correctly Classified Instances 124 30.6931 %
- Incorrectly Classified Instances 280 69.3069 %

```
 conf.matrix <-
   rbind(
    c(0,0,0,0,0,108),
    c(0,0,0,0,0,63),
    c(0,0,0,0,0,23),
    c(0,0,0,0,0,48),
    c(0,0,0,0,0,38),
    c(0,0,0,0,0,124))
colnames(conf.matrix) <- c("SW", "W", "T", "R", "P", "SO")

row.names(conf.matrix) <- c("SW", "W", "T", "R", "P", "SO")

conf.matrix
```

```
##    SW W T R P  SO
## SW  0 0 0 0 0 108
## W   0 0 0 0 0  63
## T   0 0 0 0 0  23
## R   0 0 0 0 0  48
## P   0 0 0 0 0  38
## SO  0 0 0 0 0 124
```

Using One R without any changes gives a model that is overfitted. what i would want from one R is 6 diffrent classifiers each for 1. With a default bucket size of 6 we get 15 diffrent classiefiers.

with bucket size 12 we get 7 classiefiers, and 15 we get 3.

```
 conf.matrix <-
   rbind(
    c(70,10,0,14,8,6),
    c(24,13,1,10,11,4),
    c(1,7,5,2,5,3),
    c(23,8,0,11,6,0),
    c(0,5,4,1,14,14),
    c(0,1,3,0,12,108))
colnames(conf.matrix) <- c("SW", "W", "T", "R", "P", "SO")

row.names(conf.matrix) <- c("SW", "W", "T", "R", "P", "SO")

conf.matrix
```

```
##    SW  W T  R  P  SO
## SW 70 10 0 14  8   6
## W  24 13 1 10 11   4
## T   1  7 5  2  5   3
```

```
## R   23   8 0 11   6    0
## P    0   5 4  1 14   14
## SO   0   1 3  0 12  108
```

One R with bucket size 11 seems to give us 6 diffrent classiefiers which is what i want. But the accuracy of the One R model is not very high.

One R Classiefier model with bucket size 11

huml:

```
< 29.71   -> SO
< 34.31   -> T
< 45.64   -> P
< 108.105 -> SW
< 126.94  -> R
>= 126.94 -> SW
```

Correctly Classified Instances 208 51.4851 %
Incorrectly Classified Instances 196 48.5149 %

```r
 conf.matrix <-
   rbind(
    c(69,11,1,10,8,9),
    c(35,2,4,4,10,8),
    c(4,1,4,2,8,4),
    c(29,5,1,6,6,1),
    c(5,0,7,0,16,10),
    c(2,0,3,0,10,109))
colnames(conf.matrix) <- c("SW", "W", "T", "R", "P", "SO")

row.names(conf.matrix) <- c("SW", "W", "T", "R", "P", "SO")

conf.matrix
```

```
##    SW  W T  R  P  SO
## SW 69 11 1 10  8   9
## W  35  2 4  4 10   8
## T   4  1 4  2  8   4
## R  29  5 1  6  6   1
## P   5  0 7  0 16  10
## SO  2  0 3  0 10 109
```

We get a lower accuracy but from the first run we were sure that the rule was overfitted

Next i tried Naive Bayes but it has almost the same result and not a lot of options to change:

Correctly Classified Instances 209 51.7327 % Incorrectly Classified Instances 195 48.2673 %

=== Confusion Matrix ===

a b c d e f <– classified as 35 33 1 23 8 8 | a = SW 10 20 1 8 11 13 | b = W 2 4 0 1 15 1 | c = T 7 7 0 25 8 1 | d = R 0 4 0 0 18 16 | e = P 0 1 0 0 12 111 | f = SO

```r
 conf.matrix <-
   rbind(
    c(35,33,1,23,8,8),
    c(10,20,1,8,11,13),
    c(2,4,0,1,15,1),
    c(7,7,0,25,8,1),
```

```
    c(0,4,0,0,18,16),
    c(0,1,0,0,12,111))
colnames(conf.matrix) <- c("SW", "W", "T", "R", "P", "SO")

row.names(conf.matrix) <- c("SW", "W", "T", "R", "P", "SO")

conf.matrix
```

```
##     SW  W T  R  P  SO
## SW 35 33 1 23  8   8
## W  10 20 1  8 11  13
## T   2  4 0  1 15   1
## R   7  7 0 25  8   1
## P   0  4 0  0 18  16
## SO  0  1 0  0 12 111
```

With using Random.Forest i have done 3 diffrent runs, Becouse this accuracy is already much higher than the one & zero R performance. one with 10 max depht, 15 max depht and 20 max depht.

Simple logistic also gives promising results with default settings:

Correctly Classified Instances 304 75.2475 % Incorrectly Classified Instances 100 24.7525 %

=== Confusion Matrix ===

a b c d e f <– classified as 78 17 2 5 0 6 | a = SW 24 33 0 0 0 6 | b = W 0 0 11 3 8 1 | c = T 4 3 0 35 5 1 | d = R 0 6 1 1 25 5 | e = P 0 0 1 1 0 122 | f = SO

SMO with default settings gives very close results to One R:

Correctly Classified Instances 217 53.7129 % Incorrectly Classified Instances 187 46.2871 %

=== Confusion Matrix ===

a b c d e f <– classified as 94 0 0 0 0 14 | a = SW 41 0 0 0 0 22 | b = W 7 0 0 0 0 16 | c = T 41 0 0 0 0 7 | d = R 9 0 0 0 0 29 | e = P 1 0 0 0 0 123 | f = SO

Nearest neighbour IBk gives very promising results and we might look into the future:

Correctly Classified Instances 336 83.1683 % Incorrectly Classified Instances 68 16.8317 %

=== Confusion Matrix ===

a b c d e f <– classified as 91 8 0 4 0 5 | a = SW 12 44 1 1 3 2 | b = W 0 1 16 2 2 2 | c = T 2 1 0 43 0 2 | d = R 0 0 1 1 28 8 | e = P 6 1 0 1 2 114 | f = SO

j48 with default settings gives us medioker results and mighnt not be very interesting to use in the future:

Correctly Classified Instances 271 67.0792 % Incorrectly Classified Instances 133 32.9208 %

=== Confusion Matrix ===

a b c d e f <– classified as 68 23 2 11 1 3 | a = SW 18 30 2 4 4 5 | b = W 3 3 11 1 1 4 | c = T 8 5 0 31 3 1 | d = R 1 2 4 2 23 6 | e = P 5 4 5 1 1 108 | f = SO

Tester: weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matrix "weka.experiment.ResultMatrixPlainT -mean-prec 2 -stddev-prec 2 -col-name-width 0 -row-name-width 25 -mean-width 2 -stddev-width 2 -sig-width 1 -count-width 5 -print-col-names -print-row-names -enum-col-names" Analysing: Percent_correct Datasets: 1 Resultsets: 8 Confidence: 0.05 (two tailed) Sorted by: - Date: 10/6/18, 7:59 PM

Dataset (1) rules.Ze | (2) rules (3) trees (4) trees (5) funct (6) funct (7) bayes (8) lazy.

CleanData (100) 30.69 | 53.39 v 66.84 v 79.14 v 74.54 v 53.52 v 50.67 v 83.32 v

```
(v/ /*) |   (1/0/0)   (1/0/0)   (1/0/0)   (1/0/0)   (1/0/0)   (1/0/0)   (1/0,
```

Key: (1) rules.ZeroR '' 48055541465867954 (2) rules.OneR '-B 11' -3459427003147861443 (3) trees.J48 '-C 0.25 -M 2' -217733168393644444 (4) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1 -depth 15' 1116839470751428698 (5) functions.SimpleLogistic '-I 0 -M 500 -H 50 -W 0.0' 7397710626304705059 (6) functions.SMO '-C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4"' -6585883636378691736 (7) bayes.NaiveBayes '' 5995231201785697655 (8) lazy.IBk '-K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""' -3080186098777067172

So from our experimentation with diffrent classification algorithems we conclude that Random.Forest and IBk ran the best of them all.

Random.Forest

First i have changed the max depth value and tested 10, 15, 20. in this testing using the experimenter i concluded that a max depth of 15 gives the best results.

Trees.Ra(Max 10) : 78.85%

Trees(Max 15) : 79.14%

Trees(Max 20) : 79.09%

```r
 conf.matrix <-
   rbind(
    c(87,11,0,4,0,6),
    c(17,34,0,2,4,6),
    c(2,0,12,2,4,3),
    c(7,1,0,37,3,0),
    c(0,0,1,1,30,6),
    c(0,0,1,1,1,121))
colnames(conf.matrix) <- c("SW", "W", "T", "R", "P", "SO")

row.names(conf.matrix) <- c("SW", "W", "T", "R", "P", "SO")

conf.matrix
```

```
##    SW  W  T  R  P  SO
## SW 87 11  0  4  0   6
## W  17 34  0  2  4   6
## T   2  0 12  2  4   3
## R   7  1  0 37  3   0
## P   0  0  1  1 30   6
## SO  0  0  1  1  1 121
```

Changing other settings only gives worse results.

IBk, Nearest Neigbour

It seems that it doesn't really matter what settings are used for this algoreithem as it gives them all the same accuracy. and if accuracy is the only metric we really want to maximize we can use default settings.

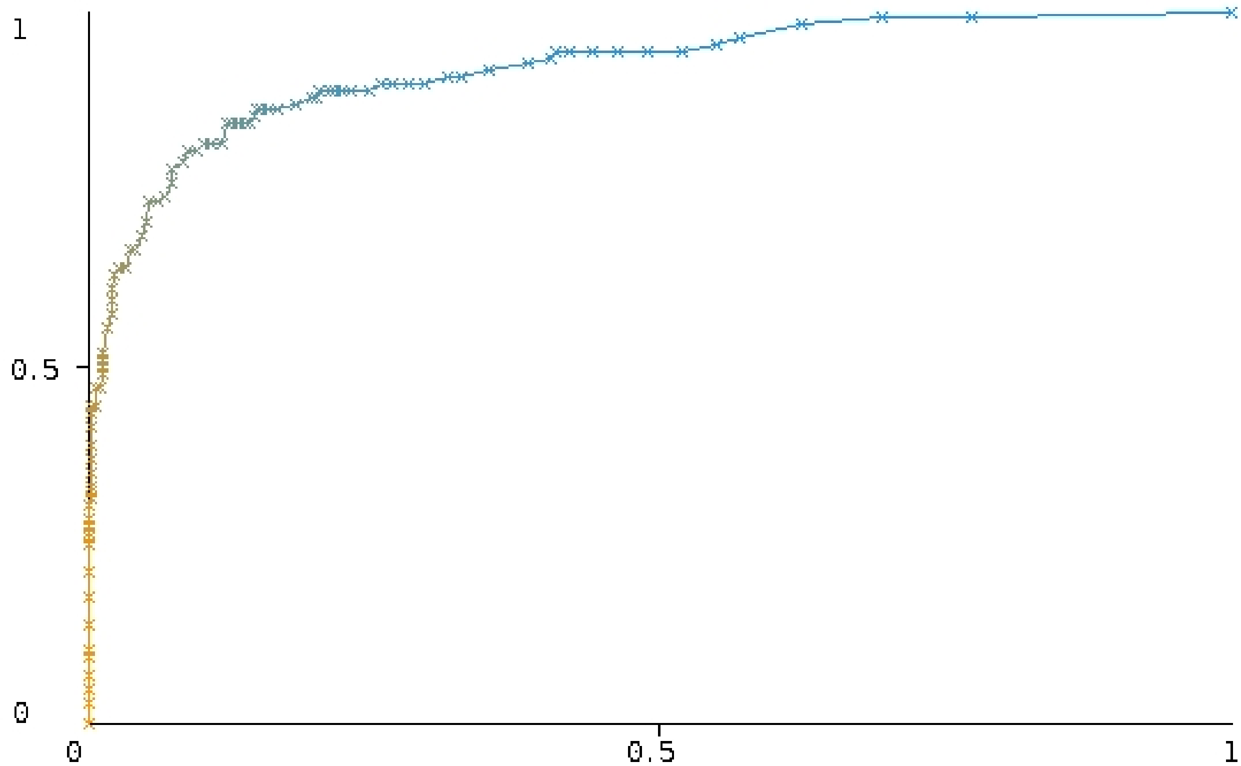


Figure 1: ROC class: SW Alg: Random.Forest

Learning Curve

To analyse the learning curve for these algorithems i am gradually removing more data and trying to classify the data with the algorithem. This can be done by using the weka experimenter, using *InstancesResultListener -O weka_experiment.arff* as a destination, *CrossValidationResultProducer* with *splitEvaluator: ClassifierSplitEvaluator* with classifier: *FilteredClassifier, filter: RemovePercentage* and classifier set to *random.forest maxdepth 15* or *IBk*

Results For Random.Forest

Dataset (1) meta.Fil | (2) meta. (3) meta. (4) meta. (5) meta. (6) meta. (7) meta. (8) meta. (9) meta.

CleanData (100) 22.07 | 23.27 25.17 26.82 v 29.17 v 32.59 v 35.48 v 39.24 v 47.05 v

```
(v/ /*) |   (0/1/0)   (0/1/0)   (1/0/0)   (1/0/0)   (1/0/0)   (1/0/0)   (1/0,
```

Results For IBk

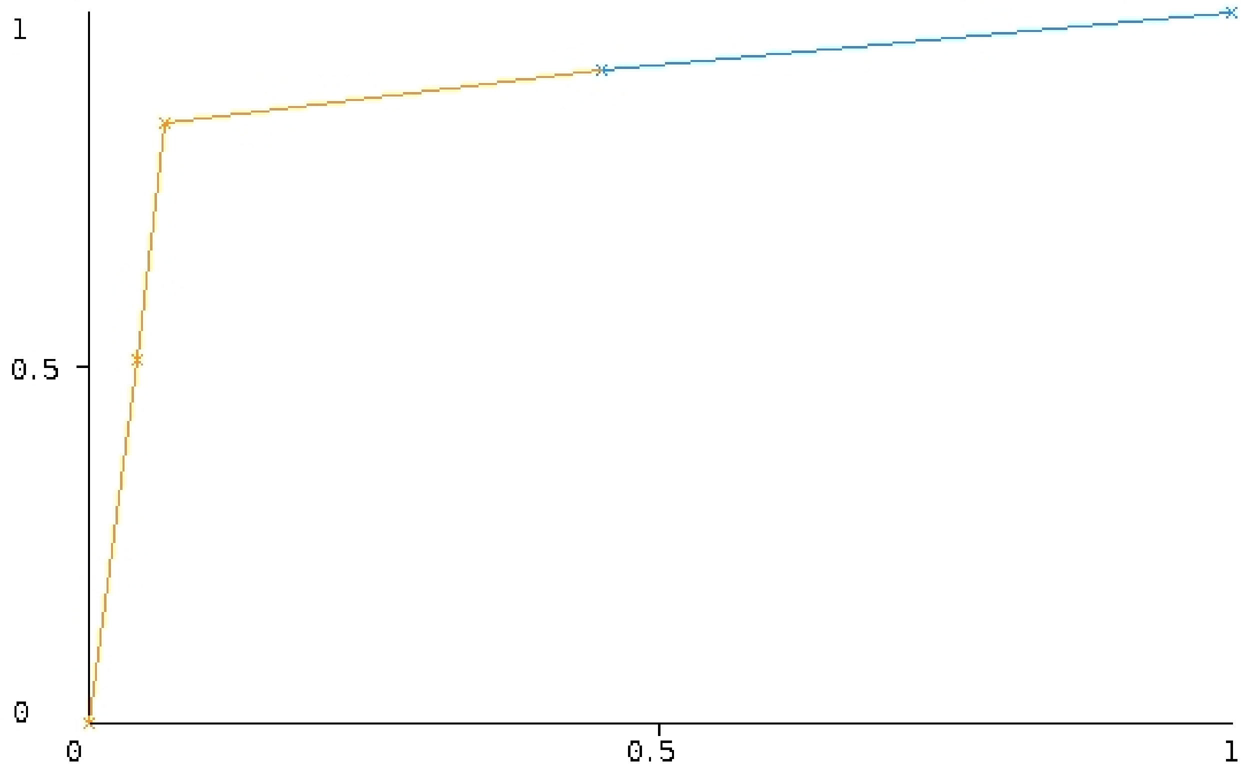


Figure 2: ROC class: SW Alg: IBk

Dataset (1) meta.Fil | (2) meta. (3) meta. (4) meta. (5) meta. (6) meta. (7) meta. (8) meta. (9) meta.

CleanData (100) 18.26 | 19.08 20.63 23.01 v 25.26 v 28.52 v 31.58 v 35.75 v 45.99 v

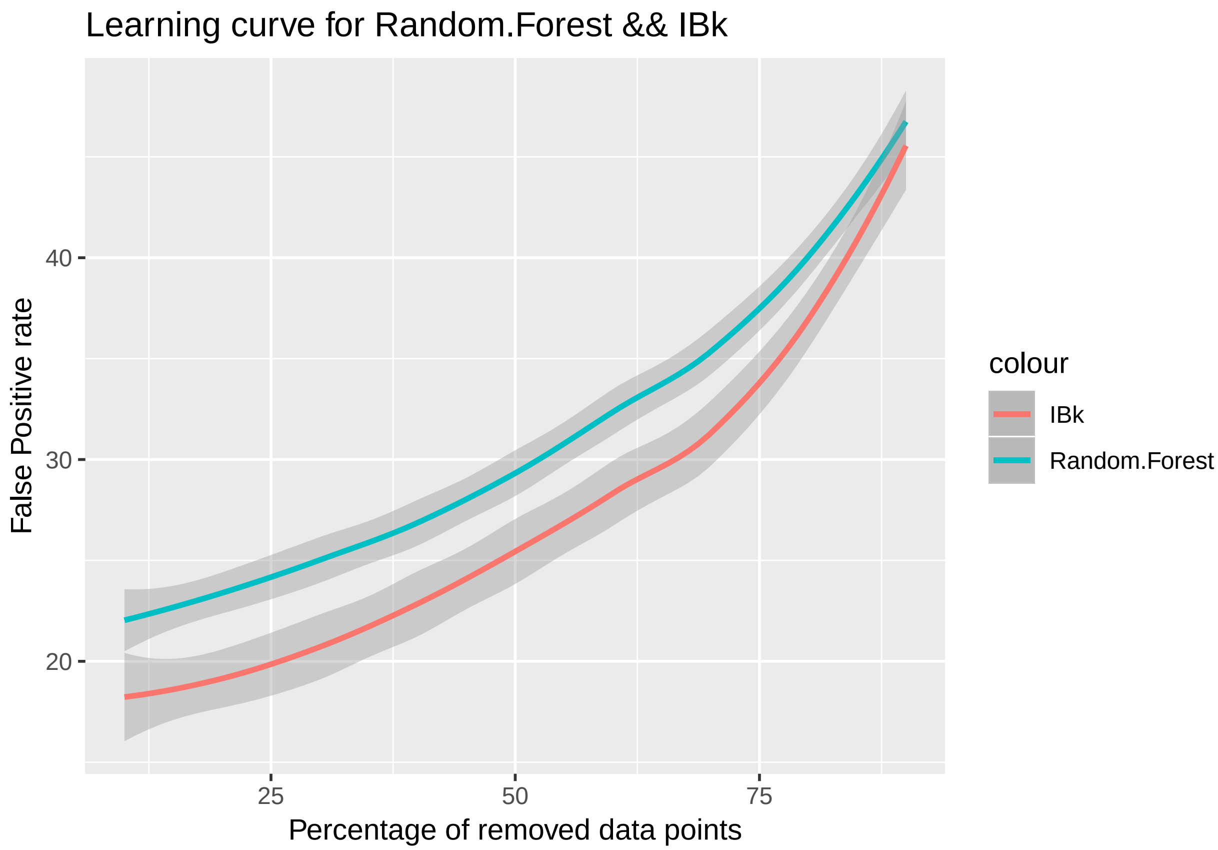
```
                         (v/ /*) |   (0/1/0)    (0/1/0)    (1/0/0)    (1/0/0)    (1/0/0)    (1/0/0)    (1/0,
```

```
RemovedPercentage <- c(10, 20, 30, 40, 50, 60, 70, 80, 90)
Random.Forest <- c(22.07, 23.27, 25.17, 26.82, 29.17, 32.59, 35.48, 39.24, 47.05)
IBk <- c(18.26, 19.08, 20.63, 23.01, 25.26, 28.52, 31.58, 35.75, 45.99)

dataf <- data.frame(RemovedPercentage, Random.Forest, IBk)
dataf
```

```
##   RemovedPercentage Random.Forest   IBk
## 1                10         22.07 18.26
## 2                20         23.27 19.08
## 3                30         25.17 20.63
## 4                40         26.82 23.01
## 5                50         29.17 25.26
## 6                60         32.59 28.52
## 7                70         35.48 31.58
## 8                80         39.24 35.75
## 9                90         47.05 45.99
```

```r
library(ggplot2)
ggplot(dataf)+
  geom_smooth(aes(x = RemovedPercentage, y = Random.Forest, color="Random.Forest"))+
  geom_smooth(aes(x = RemovedPercentage, y = IBk, color="IBk"))+
  ggtitle("Learning curve for Random.Forest && IBk")+
  xlab("Percentage of removed data points")+
  ylab("False Positive rate")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Becouse our learning data is not very large (404), it is not really easy to determine the minimum amount of data needed for classification and we can assume that 400 samples is nessesary.

```r
experimenter_data <- read.csv("../data/AlgorithemPreformance.csv", header = T, sep = ",")
```

From my analyses im going to choose Random.Forest as classifier algorithem. altough IBk gives a higher accuracy because we haven't normalised the data using nearest neighbour isn't reliable and thats why we shouldn't use it.

To give an answer to the research question we need to run a few more weka experiments with Random forest. We want to see what bone in the birds have te most inpact on classification accuracy.

```r
# New Clean dataset with all bones.
Birdbones.Clean.All <- BirdBones.noNA[! BirdBones.noNA$id %in% outliers$id, ]

write.csv(Birdbones.Clean.All, "../data/CleanDataAll.csv")
```

Base Results with all bones:

Accuracy : 84.6535 %

Removed humurus: huml & humw collums:

Accuracy: 82.1782 %

Diffrence: -2.4753 %

Removed ulnal & ulnaw collums:

Accuracy: 83.1683 %

Diffrence: -1.4852 %

Removed feml & femw collums:

Accuracy: 80.4455 %

Diffrence: -4.208 %

Removed tibl & tibw collums:

Accuracy: 84.4059 %

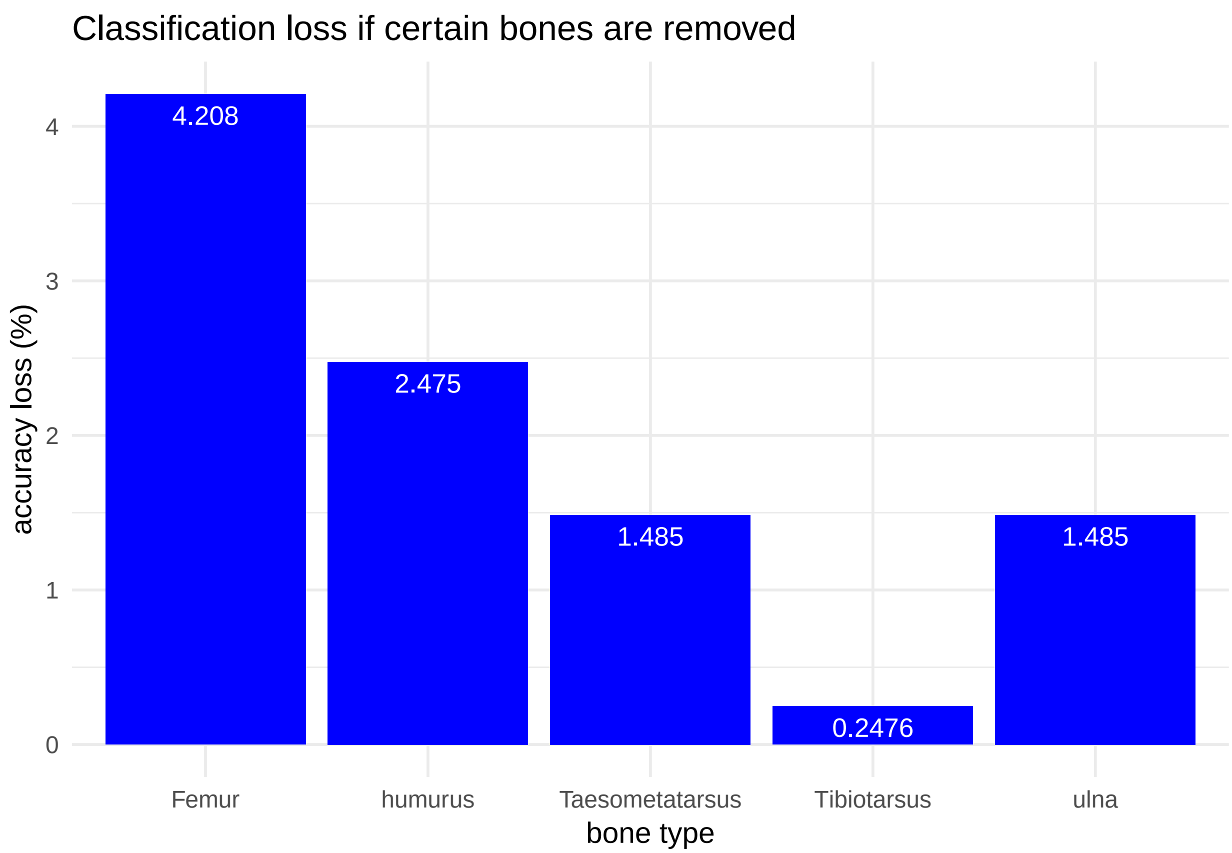
Diffrence: -0.2476 %

Removed tarl & tarw collums:

Accuracy: 83.1683 %

Diffrence: -1.4852 %

```
humurus <- signif(84.6535 - 82.1782, 4)
ulna <-  signif(84.6535 -83.1683, 4)
Femur <-  signif(84.6535 -80.4455, 4)
Tibiotarsus <-  signif(84.6535 - 84.4059, 4)
Taesometatarsus <-  signif(84.6535-83.1683, 4)

loss <- data.frame(Bone=c("humurus", "ulna", "Femur", "Tibiotarsus", "Taesometatarsus"),
                   Classify_loss=c(humurus, ulna, Femur, Tibiotarsus, Taesometatarsus))

library(ggplot2)
ggplot(data=loss, aes(x=Bone, y=Classify_loss)) +
  geom_bar(stat="identity", fill = "blue")+
  geom_text(aes(label=Classify_loss), vjust=1.6, color="white", size=3.5)+
  theme_minimal()+
  labs(x = "bone type",
       y = "accuracy loss (%)",
       title = "Classification loss if certain bones are removed"
       )
```



above we see the diffrence it makes per bone what the classification accuracy is. The Femur seems to be the most important bone for the functioning in bird specicies.