HW2

Chanukya

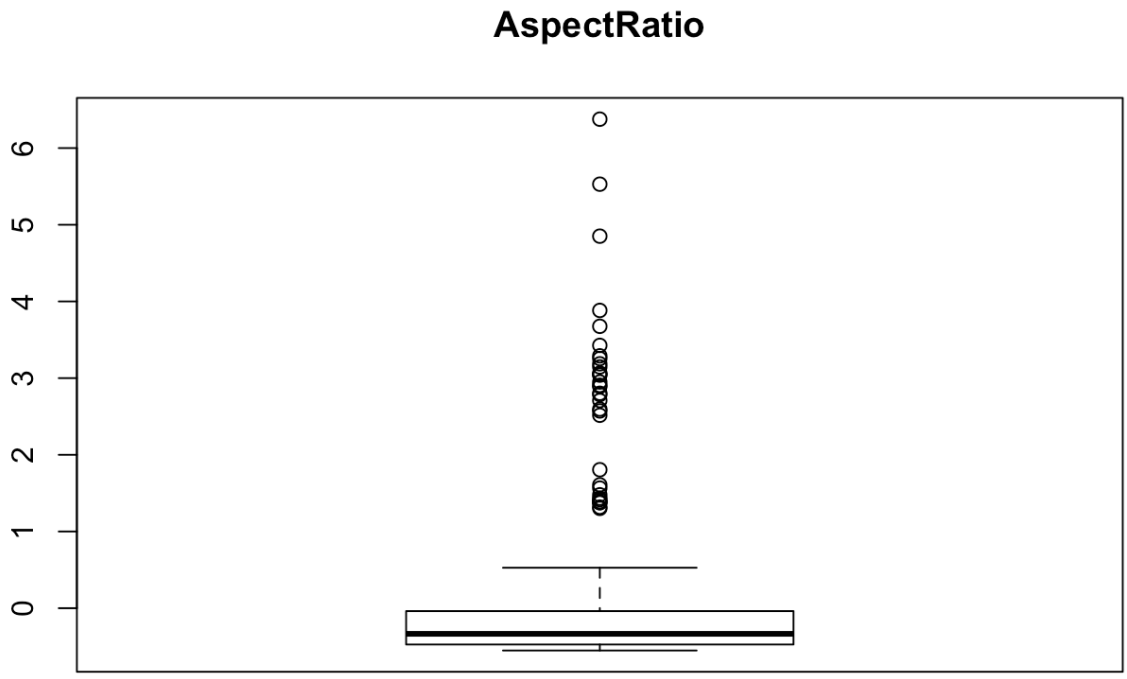
10/9/2019

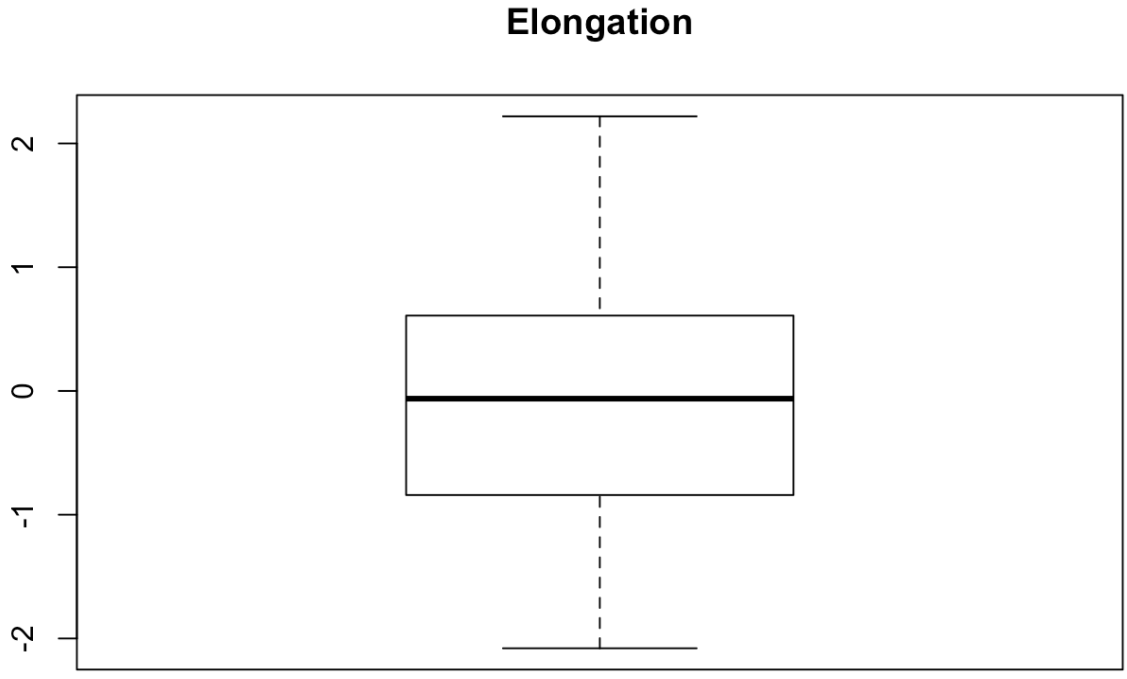
```
data  <- read.csv("/Users/chanukya/Documents/GitHub/DataMining/HW2/leaf/leaf.csv")
#View(data)
#Aspect ratio , Elongation and Solidity
#changing column names
column_names <- c("Class","SpecimenNumber","Eccentricity","AspectRatio","Elongation","Solidity","StochasticConvexity","IsoperimetricFactor","MaximalIndentationDepth","Lobedness","AvgIntensity","AvgContrast","Smoothness","ThirdMoment","Uniformity","Entropy")
for(i in 1:length(column_names)){
  names(data)[i]<- column_names[i]
}
colnames(data)
```

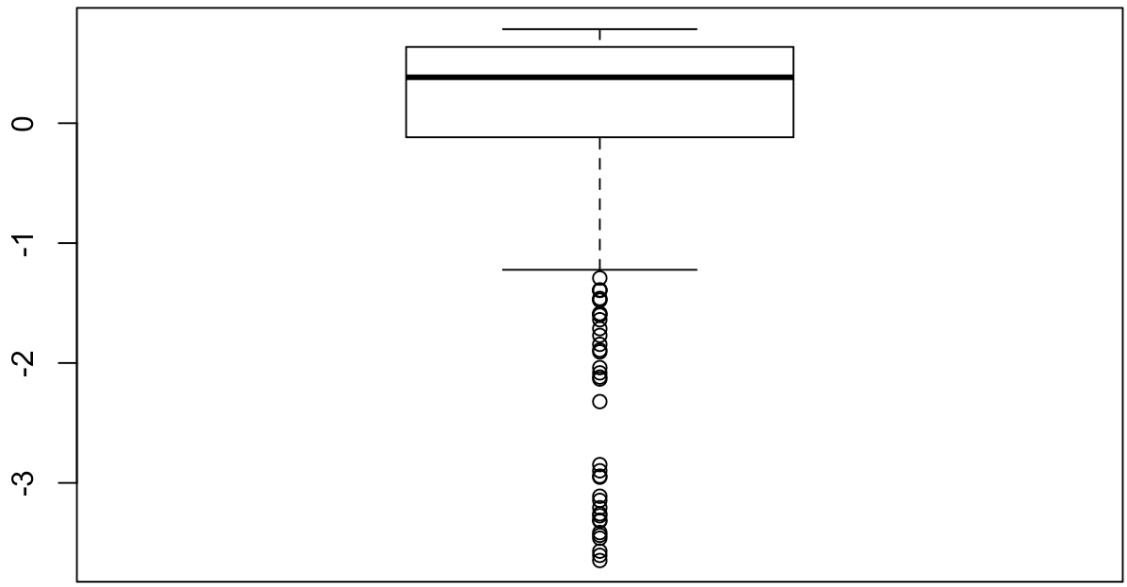
```
##  [1] "Class"                   "SpecimenNumber"
##  [3] "Eccentricity"            "AspectRatio"
##  [5] "Elongation"              "Solidity"
##  [7] "StochasticConvexity"     "IsoperimetricFactor"
##  [9] "MaximalIndentationDepth" "Lobedness"
## [11] "AvgIntensity"            "AvgContrast"
## [13] "Smoothness"              "ThirdMoment"
## [15] "Uniformity"              "Entropy"
```

```
#b)

problem_a_columns <- c("AspectRatio","Elongation","Solidity")
data <- data.frame(scale(data))
#par( mfrow = c( 1, 3 ) )
for(i in 1:length(problem_a_columns)){
  boxplot(data[problem_a_columns[i]], main =  problem_a_columns[i])
}
```


AspectRatio
0
1
2
3
4
5
6

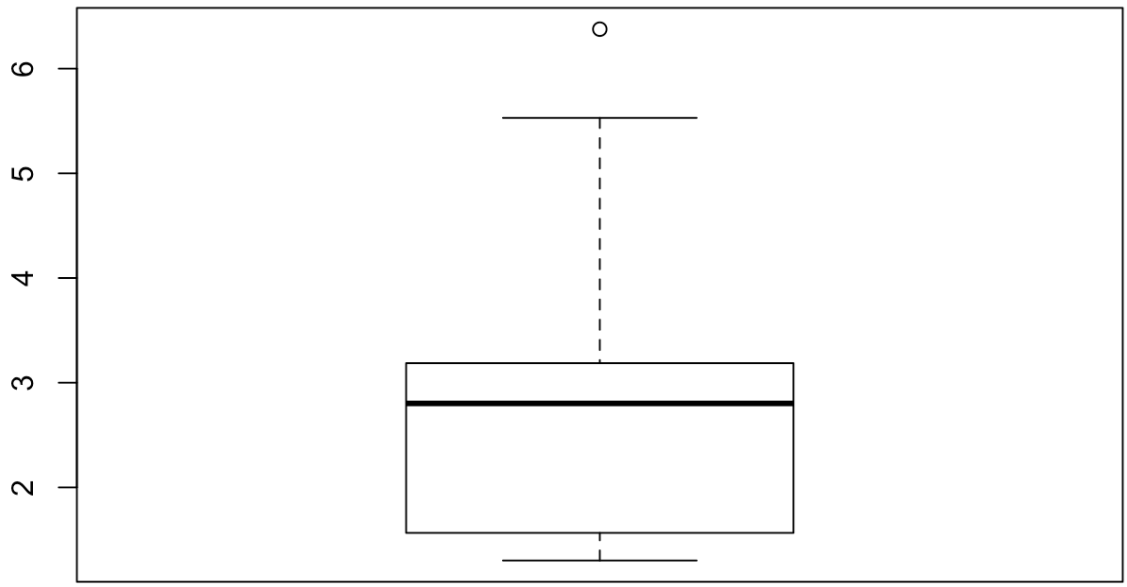

Elongation
-2
-1
0
1
2




```
# c)
data1 <- data
outliers_aspect <- boxplot(data1$AspectRatio, plot = FALSE)$out
#here are the outliers
outliers_aspect
```

```
##  [1] 1.389720 1.375617 1.608137 1.301799 1.377154 1.478409 1.565599
##  [8] 1.317593 1.415735 1.434564 1.805651 2.708373 3.290461 3.675882
## [15] 2.802095 3.883002 3.254724 2.785341 2.516162 2.572111 2.950000
## [22] 2.595897 3.039150 2.890554 3.426876 5.529205 4.851357 3.048757
## [29] 3.186709 3.065665 3.145977 2.906578 6.376899
```

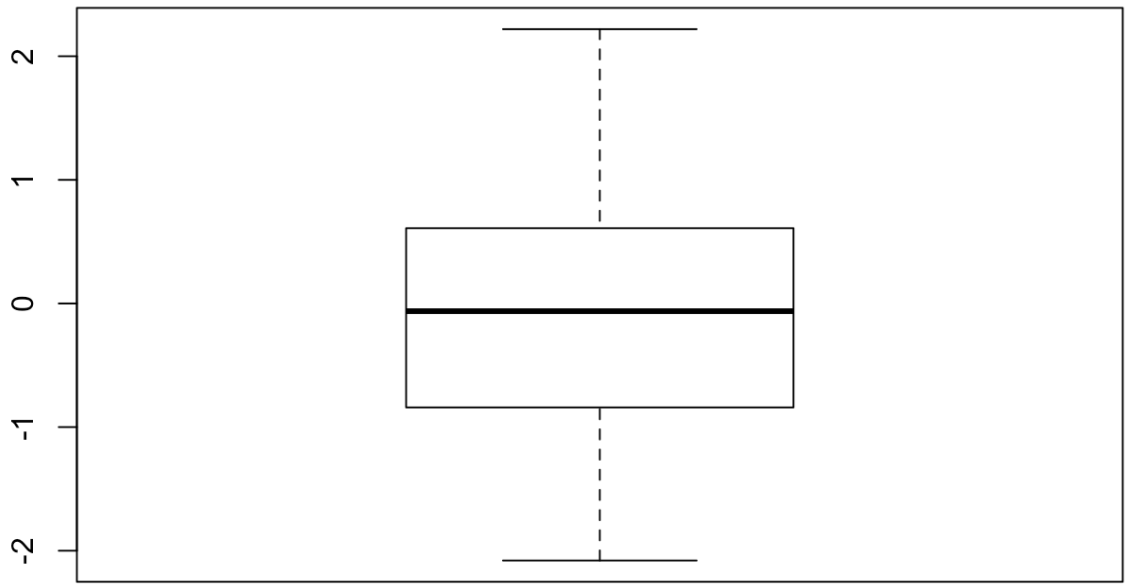
```
data1 <- data1[which(data1$AspectRatio %in% outliers_aspect),]
boxplot(data1$AspectRatio)
```




```
data1 <- data
outliers_elongation <- boxplot(data1$Elongation, plot = FALSE)$out
outliers_elongation
```

```
## numeric(0)
```

```
# no outliers
data1 <- data1[which(data1$Elongation %in% outliers_elongation),]
boxplot(data$Elongation)
```

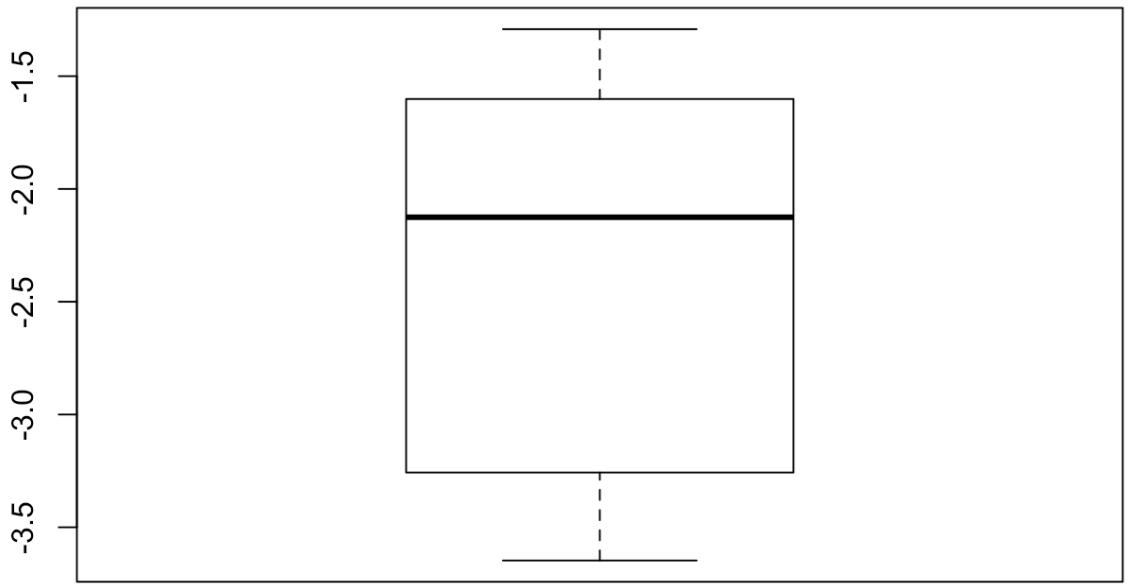



```
# no errors in elongation

#POPULATION PARAMETER CALCULATIONS
data1 <- data
outliers_solidity <- boxplot(data1$Solidity, plot = FALSE)$out
# outliers in solidity
outliers_solidity
```

```
##  [1] -2.321918 -1.461500 -1.587891 -1.291613 -2.847530 -2.943500 -3.311777
##  [8] -3.414197 -3.145899 -2.952129 -3.437035 -3.273859 -3.257211 -3.317704
## [15] -3.111033 -3.604306 -3.461703 -3.647279 -3.569091 -2.898958 -1.639232
## [22] -1.394382 -1.390983 -2.082385 -2.038540 -1.770068 -1.891927 -1.844247
## [29] -1.601227 -2.133551 -1.392813 -1.476405 -1.472745 -1.907268 -1.589896
## [36] -1.713236 -2.117338 -3.207177
```

```
data1 <- data1[which(data1$Solidity %in% outliers_solidity),]
boxplot(data1$Solidity)
```




```
#f)
Accu1<-0.6410256
Accu2<-0.4871795
n<-length(data1)
d<-Accu1-Accu2
sigma1= Accu1*(1-Accu1)/n
sigma2<-Accu2*(1-Accu2)/n
sigmat<-sqrt(sigma1+sigma2)

Z<-2.33 #confidence level 98%
dt1<-d+(Z*sigmat)
dt2<-d-(Z*sigmat)
dt<-c(dt2,dt1)
#confidence Interval
print(dt)
```

```
## [1] -0.2496996  0.5573918
```

```
# since 0 is in the confidence intervals we can say models are statistically i
nsignificant.
```

```
#g)
data1 <- data
in_train <- sample(nrow(data1))
data1 <- data1[in_train,]
train_data <- data1[1:300,]
test_data  <- data1[301:nrow(data1),]
library(C50)

train_data$Class <- as.factor(train_data$Class)
vars = c("SpecimenNumber","Eccentricity","AspectRatio","Elongation","Solidit
y","StochasticConvexity","IsoperimetricFactor","MaximalIndentationDepth","Lobe
dness","AvgIntensity","AvgContrast","Smoothness","ThirdMoment","Uniformity","E
ntropy")
tree_mod <- C5.0(x = train_data[, vars], y = train_data$Class)
test1 <- function(a,b){
  predicted <- predict(a,b)
  return(predicted)
}

#input<- data.frame(19,0.451,2.700,0.5169,0.6800,0.8421,0.609,0.02530,0.1100,
0.007984,0.042948,0.0050183,0.00071981,2.7289e-05,0.42553)
input <- data1[1, vars]
predicted <- test1(tree_mod,input)
predicted
```

```
## [1] 0.485631845219829
## 30 Levels: -1.58121940869884 -1.49135631070238 ... 1.5639890211774
```

```
input <- data1[5, vars]
predicted <- test1(tree_mod,input)
predicted
```

```
## [1] 1.20453662919154
## 30 Levels: -1.58121940869884 -1.49135631070238 ... 1.5639890211774
```

```
input <- data1[7, vars]
predicted <- test1(tree_mod,input)
predicted
```

```
## [1] -1.49135631070238
## 30 Levels: -1.58121940869884 -1.49135631070238 ... 1.5639890211774
```