Homework Assignment 3

Brendan Smith

February 23, 2016

Objective Statement: Building upon last week's assignment, we aim to analyze biomass by means of carbon stocks as a function of tree height and diameter at breast height (DBH). Although it is not 100% foolproof, a good estimate of the biomass can be extracted from these data. We are utilizing the "clean" data set that was filtered in the previous assignment. The dataset was narrowed down to five genera: Acer, Fraxinus, Populus, Quercus, and Salix. This assignment will center about linear models, box plots, regression lines and log transforms.

Methods: We begin the assignment by reacquainting ourselves with the data through the use of our basic toolbox of EDA. We do this by first importing our dataset and then by using the `str(), head(),`and `tail()` functions. Following the data refresher, we develop several linear models to quantify and visualize the relationship between DBH and height. Initial results will yield undesirable results due to outliers, and thus we remove the erroneous data via key outlier and removal tools. We then pique our curiosity regarding the effect genus and/or location plays upon the relationship between DBH and height by constructing several linear models and observing their coefficient of determination. Finally, we create a visually overwhelming master scatter plot.

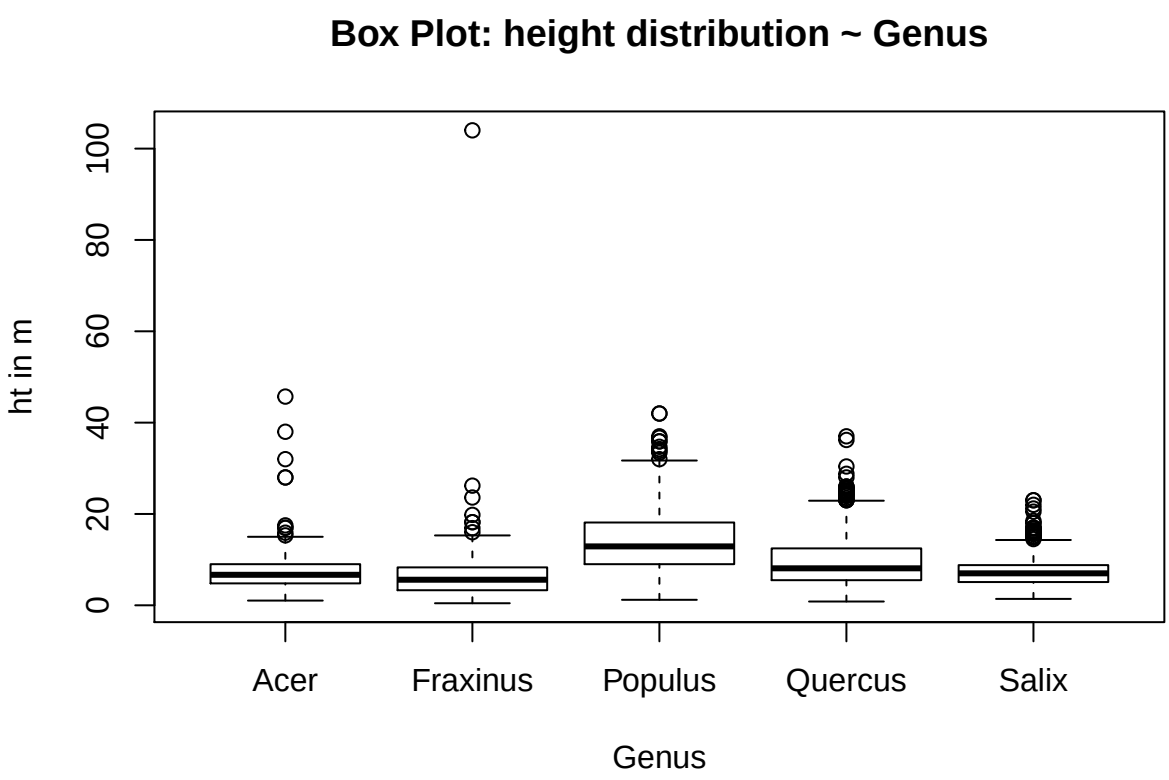
Data: The data utilized in this assignment are taken from the cleaned dataset output from the previous assignment.

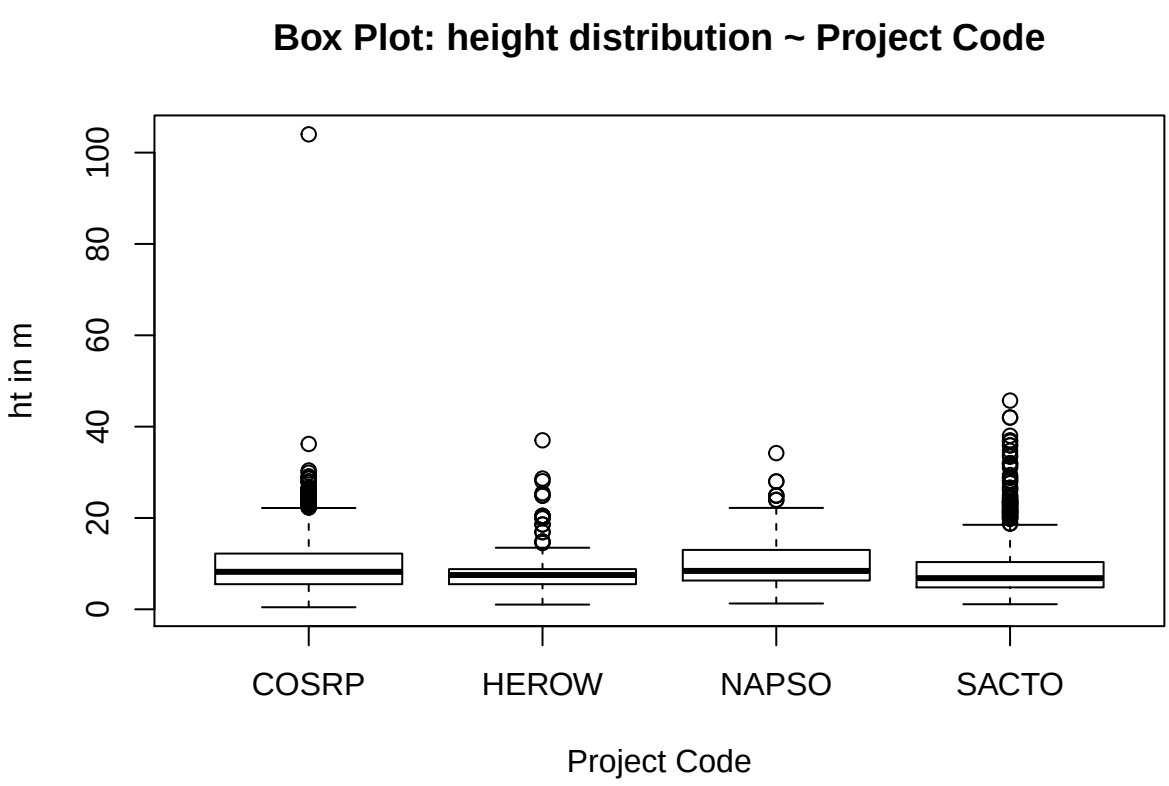
Code: In addition to code utilized from previous labs, the `outlierTest()` function from the `car` package. This function allows us to input a linear model and receive the index of the outlier points. we can then use these indeces to plot the points in a different color, or completely remove the points from the dataset.

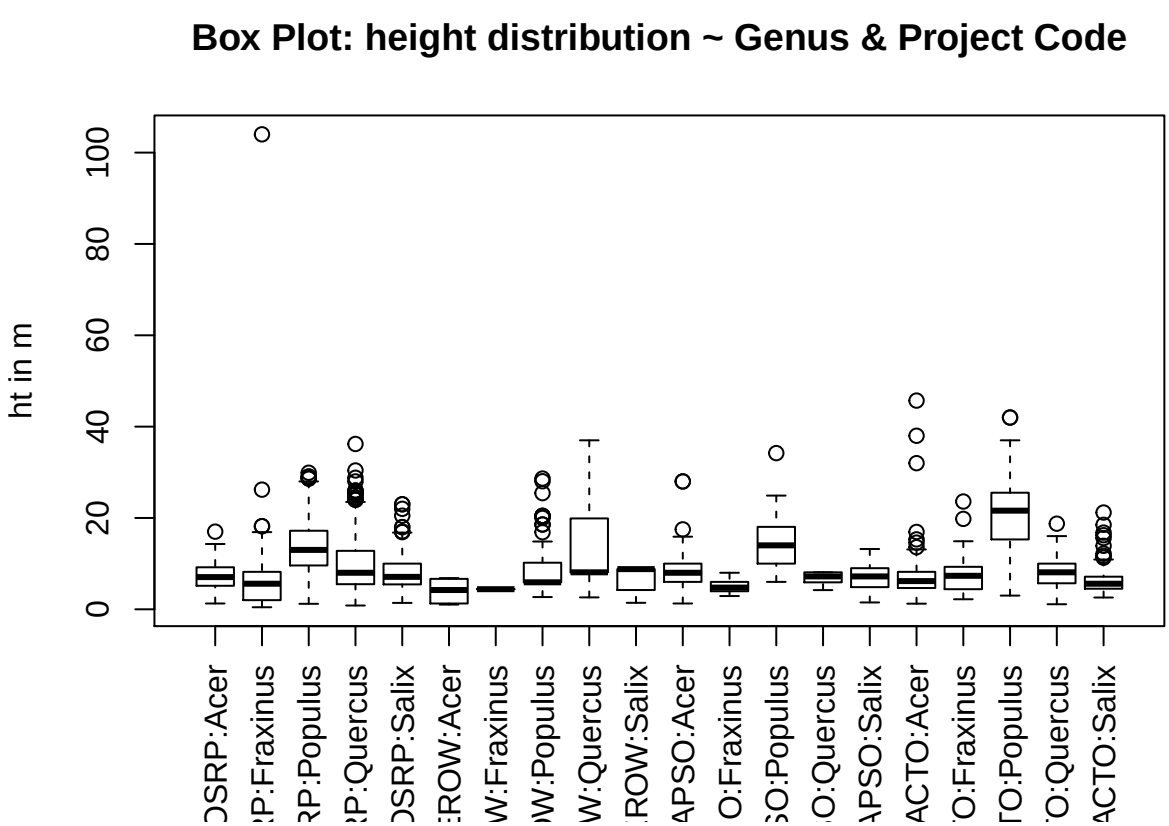
```
#Import the data filtered in last week's assignment
ripdata <- read.csv("./newripdata_survey.csv",sep = ",",header = TRUE)

#Use the basic EDA tools
#str(ripdata)
#head(ripdata)
#tail(ripdata)
#levels(ripdata$Genus)
#levels(ripdata$ProjCode)
```

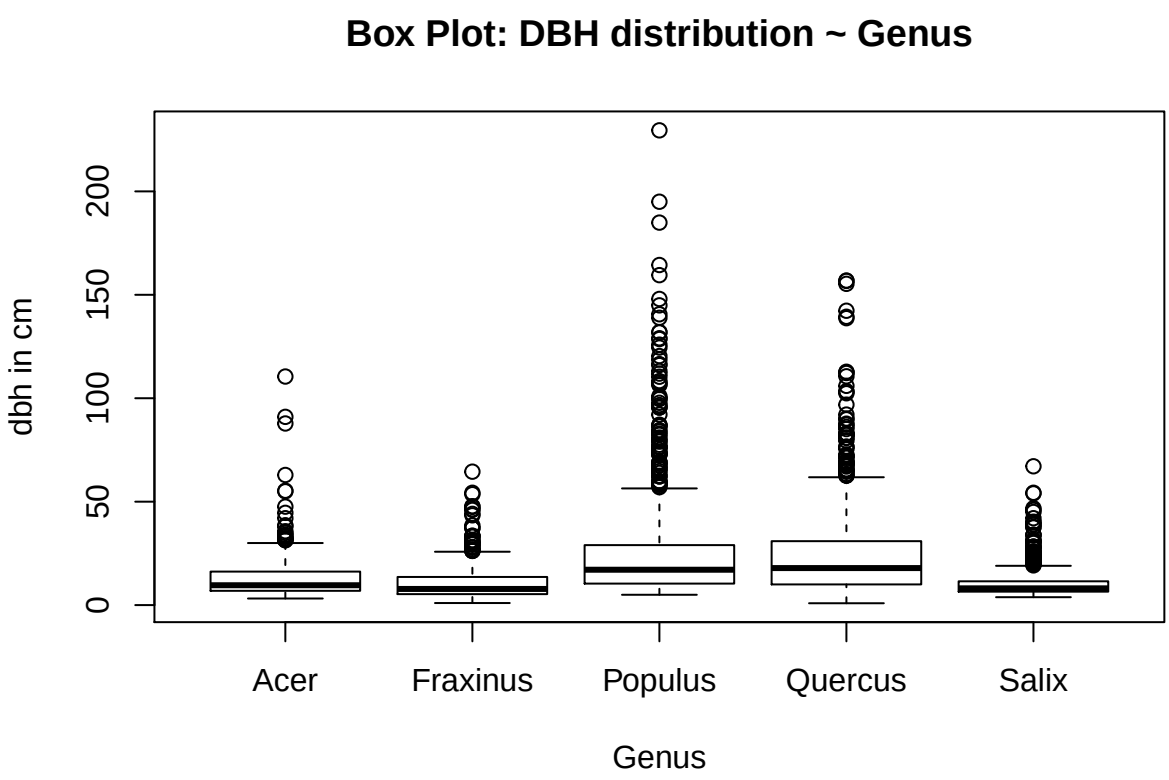
Results: Analysis begins with boxplots using the `boxplot()` function. We initially plot the height in meters for each genus, in which we can see that the genus Populus has a higher average height in comparison to the other four genera. Next, the height distribution of trees at each project location is plotted. In this case, we can see that the tree heights are pretty evenly distributed amongst the four locations. Third, we plot the interaction of project location and genus while looking at the distribution of height.

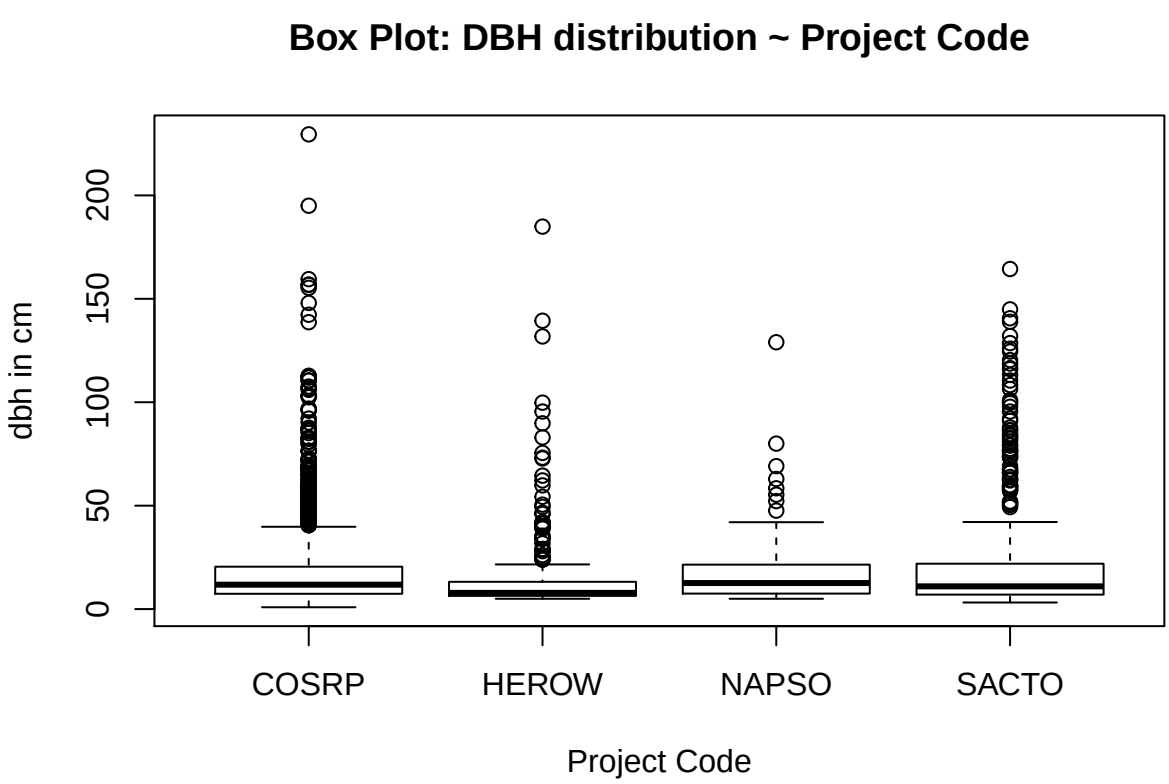

Box Plot: height distribution ~ Genus
ht in m
0
20
40
60
80
100
Acer
Fraxinus
Populus
Quercus
Salix
Genus

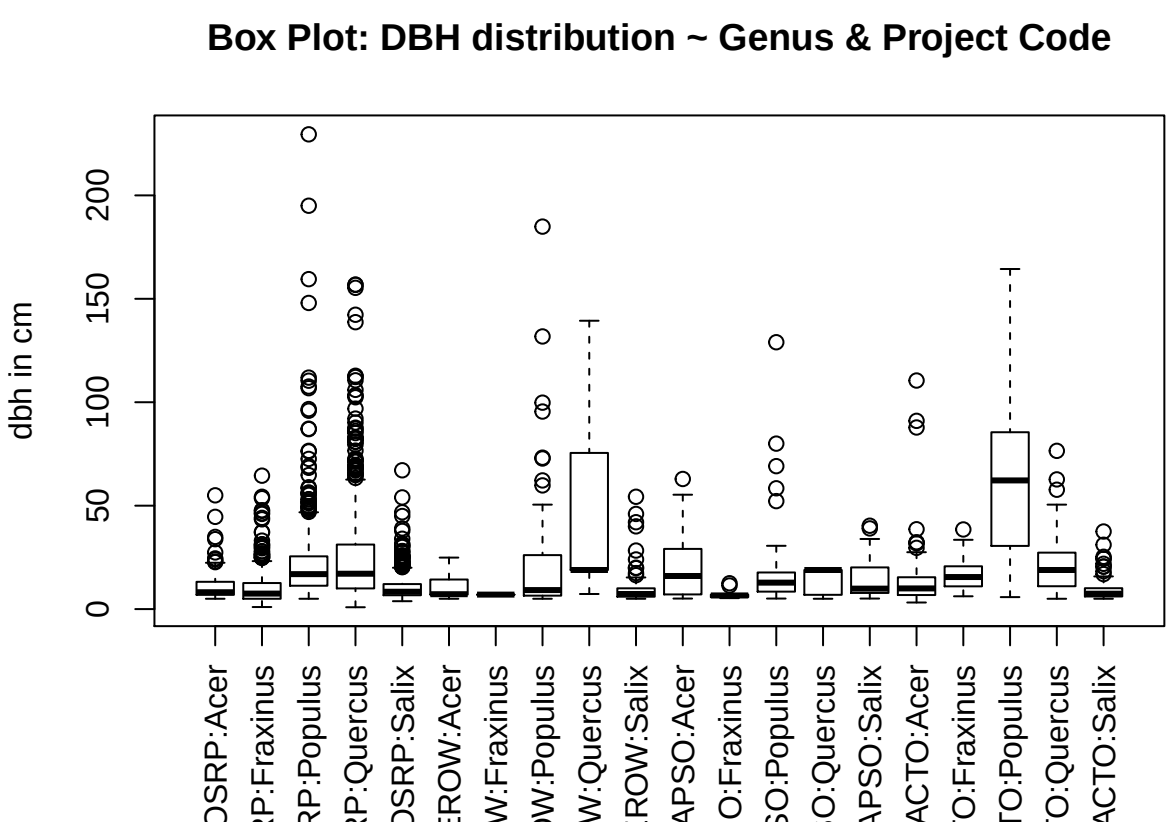

Box Plot: height distribution ~ Project Code
ht in m
0
20
40
60
80
100
COSRP
HEROW
NAPSO
SACTO
Project Code



Similarly, we plot three sets of boxplots showing the distribution of DBH in genera and project locations. The first set of boxplots indicates that Populus and Quercus are pretty even in terms of DBH, and both are larger on average than the other three genera. The second set of boxplots shows that the distribution of DBH at each project site is fairly similar, with the exception at HEROW. It should also be noted that the outliers are numerous at each site and weighted towards larger DBH. Thirdly, we plot the interaction of project location and genus while looking at the distribution of DBH.


Box Plot: DBH distribution ~ Genus
dbh in cm
0
50
100
150
200
Acer
Fraxinus
Populus
Quercus
Salix
Genus


Box Plot: DBH distribution ~ Project Code
dbh in cm
0
50
100
150
200
COSRP
HEROW
NAPSO
SACTO
Project Code



It is fairly obvious that these distributions are not normal, rather they are most likely logorithmic. This can be seen from the heavy number of outliers to one side of each of the distributions. We will transform the data in the upcoming section of the assignment, and revisit the boxplot analysis.

we follow up the preliminary EDA and boxplot analysis by assembling linear models relating tree height in centemeters to DBH in cm. So the first step is to scale the tree height data from meters to centimeters.

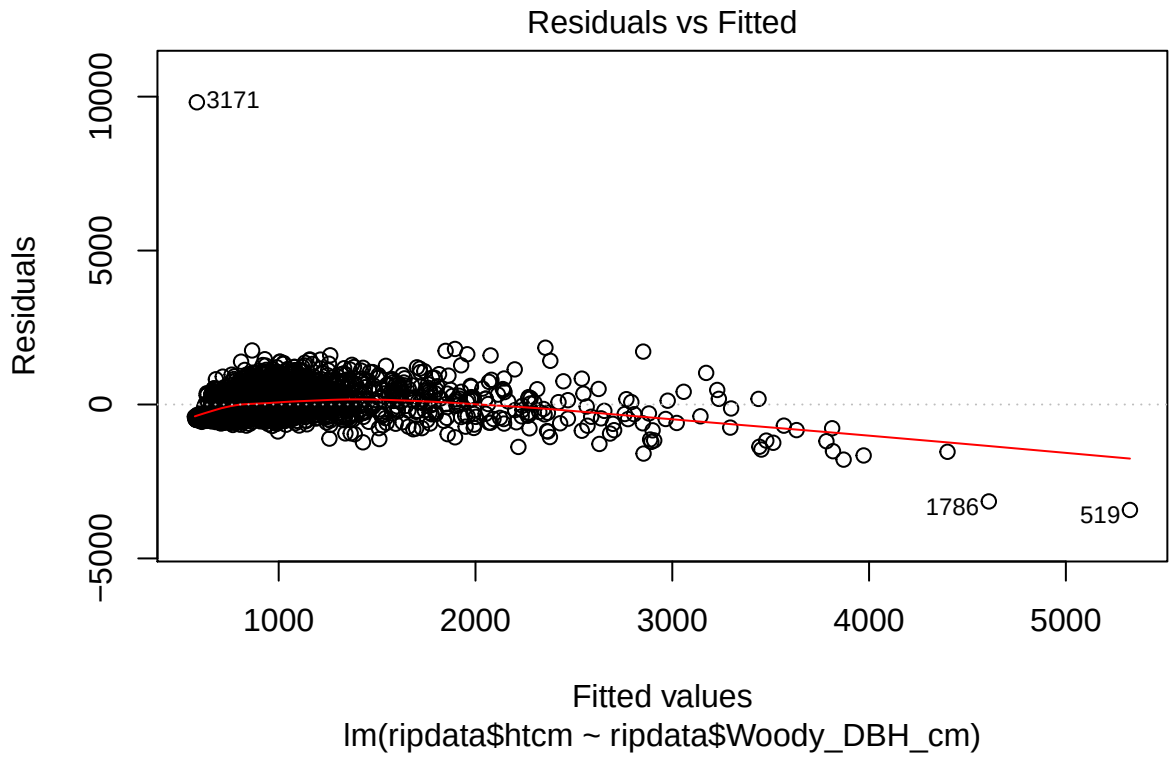
```
#Scale the tree height from meters to centimeters and place the values in the dataframe
ripdata$htcm <- ripdata$Woody_Height_m * 100
#Assemble a linear model relating tree height as a function of DBH
riplm <- lm(ripdata$htcm ~ ripdata$Woody_DBH_cm)
#View the summary of the linear model
summary(riplm)
```

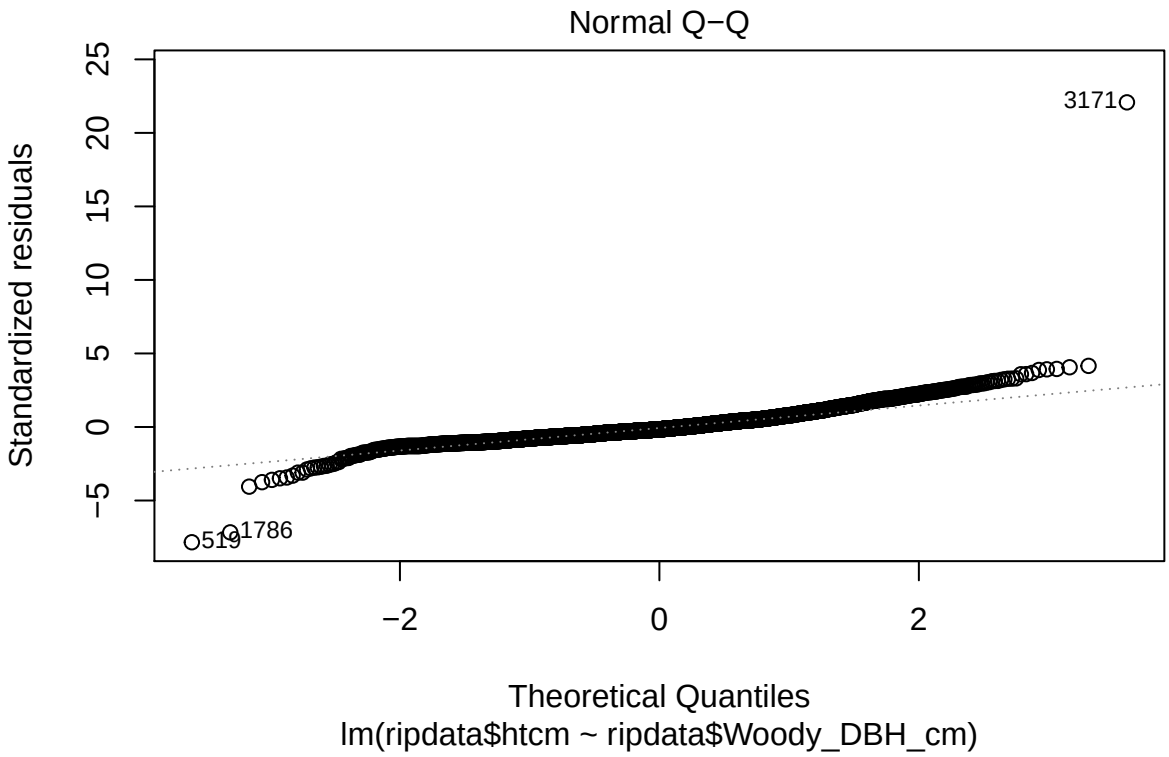
```
##
## Call:
## lm(formula = ripdata$htcm ~ ripdata$Woody_DBH_cm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3425.7  -260.8   -71.8   198.0  9815.7
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            555.214     10.546   52.64   <2e-16 ***
## ripdata$Woody_DBH_cm    20.786      0.387   53.71   <2e-16 ***
```

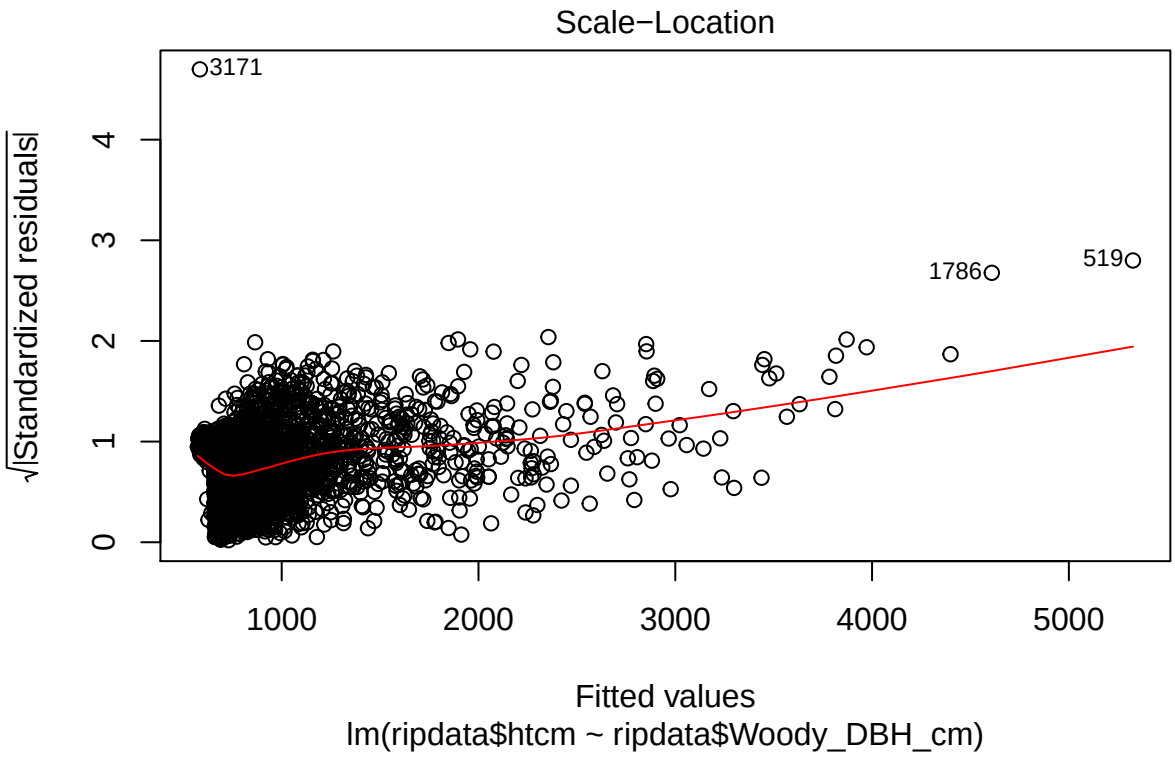
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 444.7 on 3190 degrees of freedom
## Multiple R-squared:  0.4749, Adjusted R-squared:  0.4747
## F-statistic:  2884 on 1 and 3190 DF,  p-value: < 2.2e-16
```

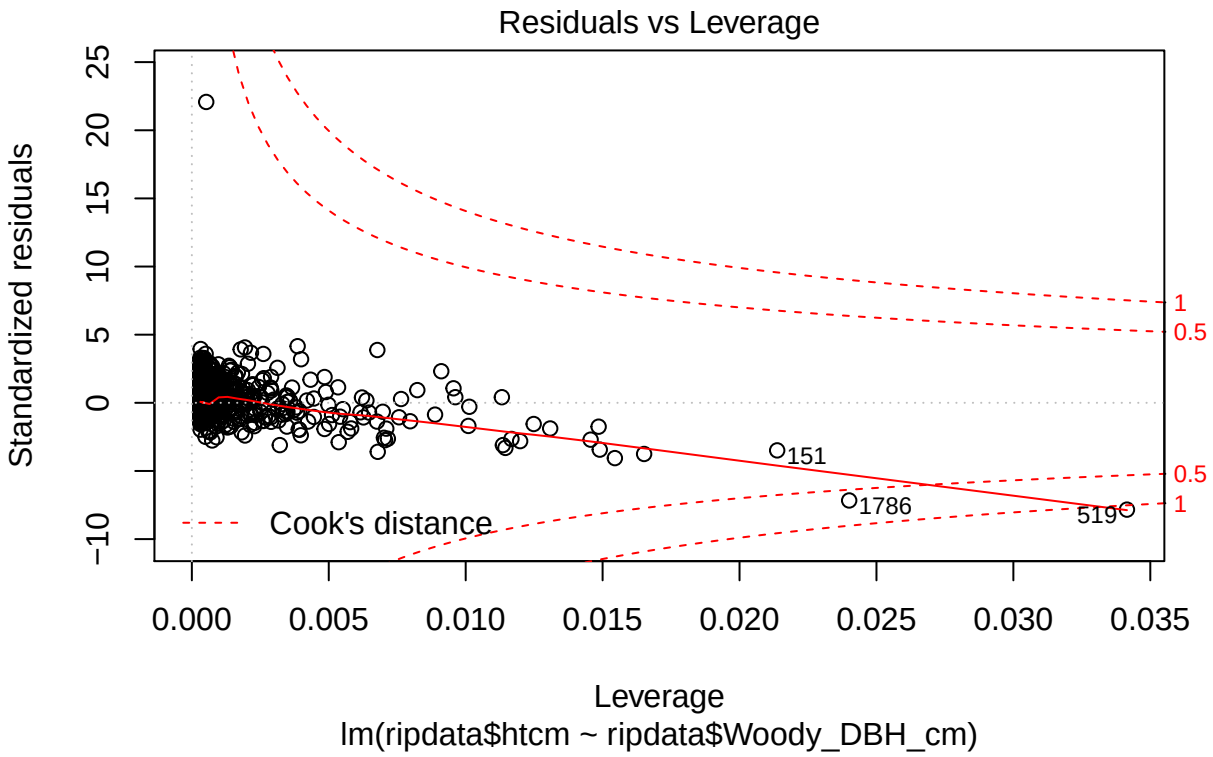
From the summary, we can see that the coefficient of determination value is actually quite low (`0.4749`), indicating that the linear model regression is not a decent fit. We now plot the linear model, which yields four separate regression plots. Followed by the scattergram plot of DBH versus height, which we overlay with the linear regression line.

```
plot(riplm)
```

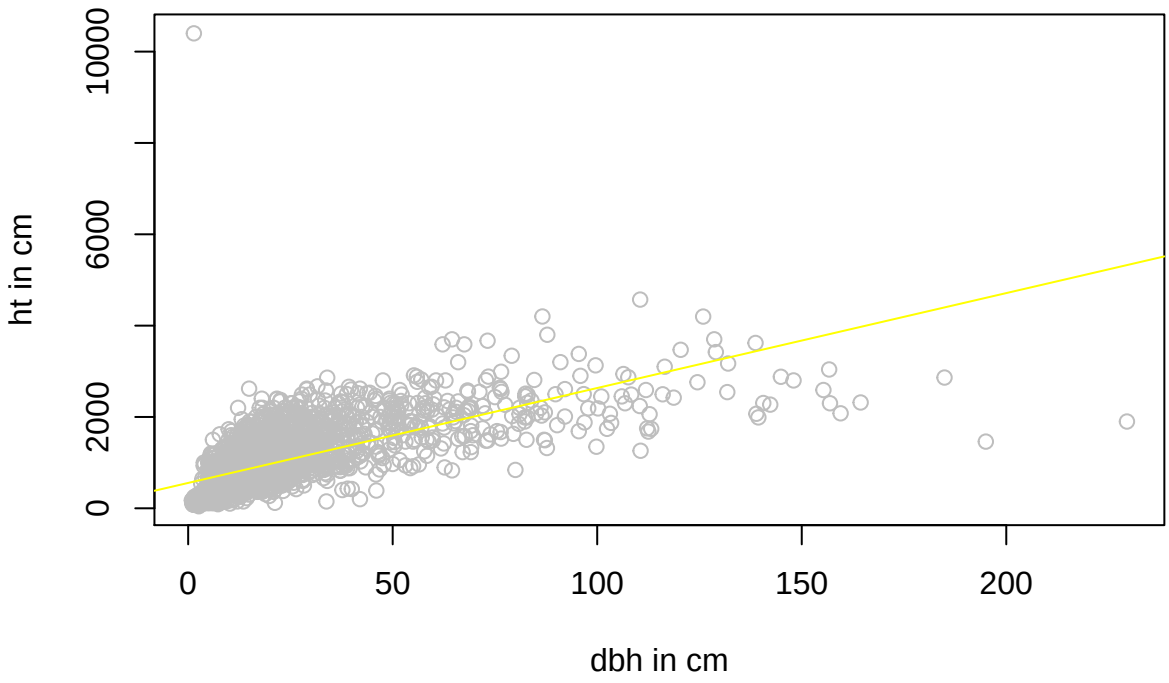



Normal Q−Q
Standardized residuals
25
20
15
10
5
0
−5
3171
1786
519
−2
0
2
Theoretical Quantiles
lm(ripdata$htcm ~ ripdata$Woody_DBH_cm)


Scale−Location
3171
1786
519
√|Standardized residuals|
0
1
2
3
4
1000
2000
3000
4000
5000
Fitted values
lm(ripdata$htcm ~ ripdata$Woody_DBH_cm)




```
plot(ripdata$Woody_DBH_cm,ripdata$htcm,col = "grey",xlab="dbh in cm",ylab="ht in cm")
abline(riplm,col = "yellow")
```



Here, we can see there are a few outliers skewing our regression model, and we still have no transformed our data, thus our scattergram is somewhat uninformative. We solve one of these issues by indicating and removing the outliers.

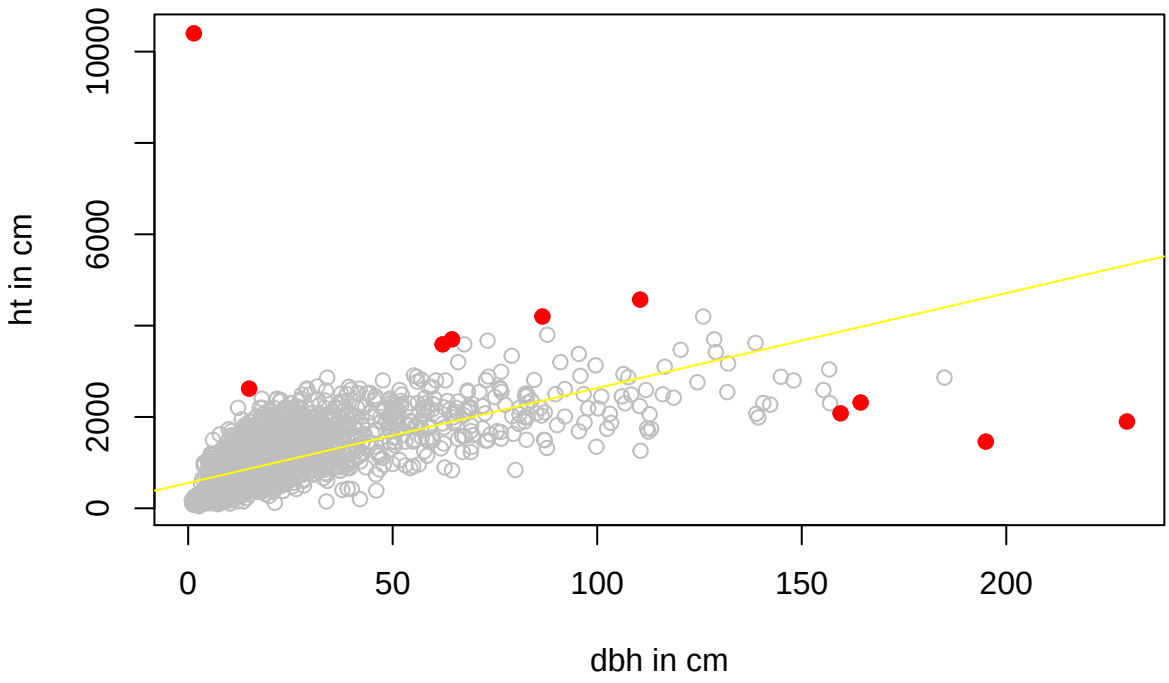
The `outlierTest()` function parses through the linear model and conveniently indicates the outlying datum points. This allows us to preliminarily highlight the indexed outliers in a plot, and finally remove these data from the data frame.

```
#Load the "car" library so we can use "outlierTest()"
library(car)

plot(ripdata$Woody_DBH_cm,ripdata$htcm,col = "grey",xlab="dbh in cm",ylab="ht in cm")
abline(riplm,col = "yellow")

#Perform the "outlierTest()" function on the riparian data linear model assembled earlier
ripolt <- outlierTest(riplm,cutoff = 2) #We can use the "cutoff" parameter to identify observations wit
ripolt.ids <- as.integer(names(ripolt$rstudent)) #typecast the indeces to outlier data poitns as intege

#Create the for loop for us to iterate through the outlierTest results and plot these points on our sca
for (i in 1:length(ripolt.ids)){
  r<-ripolt.ids[i]
  points(ripdata$Woody_DBH_cm[r], ripdata$htcm[r], col="red",pch=19) #plot each outlier point individua
}
```



These data points do look like outliers. An alternate method could be to look at the summary information for the DBH and height, comparing the median to the mean. The median and mean should be in the same "ballpark", if not then the maximum or minimum values can be removed in order to adjust these values. Of course, this will only work on the maximum and minimum extremes of the data, and does not take into account the data trend. With that said, this method seems to be elegant and straightforward.

```
#Remove the suspect data
ripss <-ripdata[-ripolt.ids,]

#Create a new linear model with subset data
ripsslm <- lm(ripss$htcm ~ ripss$Woody_DBH_cm)
#Perform EDA on linear model and touched up data
summary(riplm)
```

```
##
## Call:
## lm(formula = ripdata$htcm ~ ripdata$Woody_DBH_cm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3425.7  -260.8   -71.8   198.0  9815.7
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            555.214     10.546   52.64   <2e-16 ***
## ripdata$Woody_DBH_cm    20.786      0.387   53.71   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 444.7 on 3190 degrees of freedom
## Multiple R-squared:  0.4749, Adjusted R-squared:  0.4747
## F-statistic:  2884 on 1 and 3190 DF,  p-value: < 2.2e-16
```

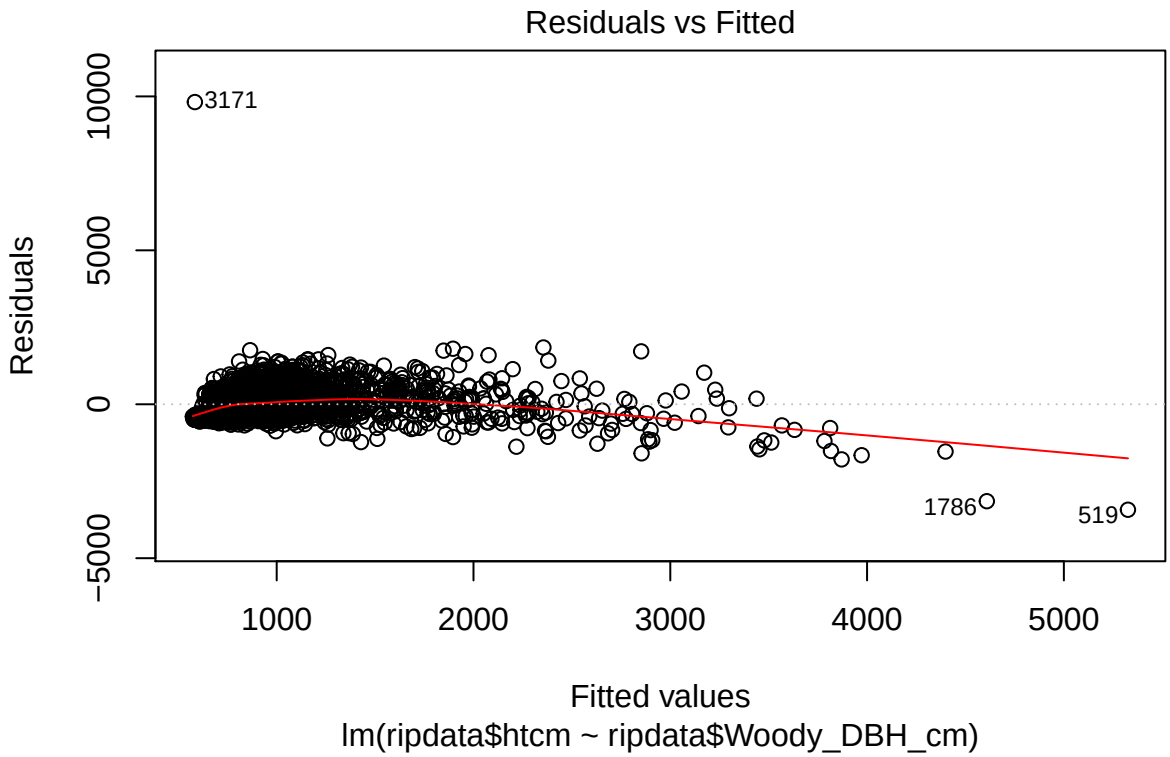
```
summary(ripss$Woody_DBH_cm)
```

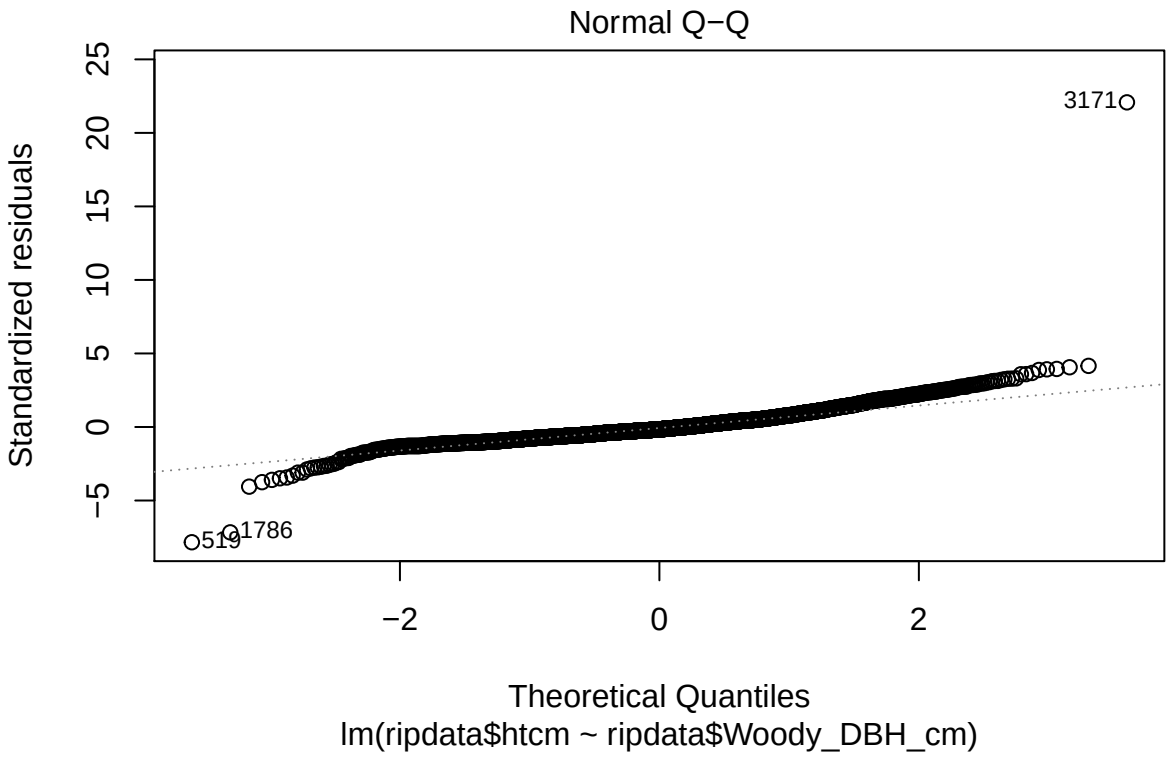
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.90    7.10   11.10   17.85   20.20  184.90
```

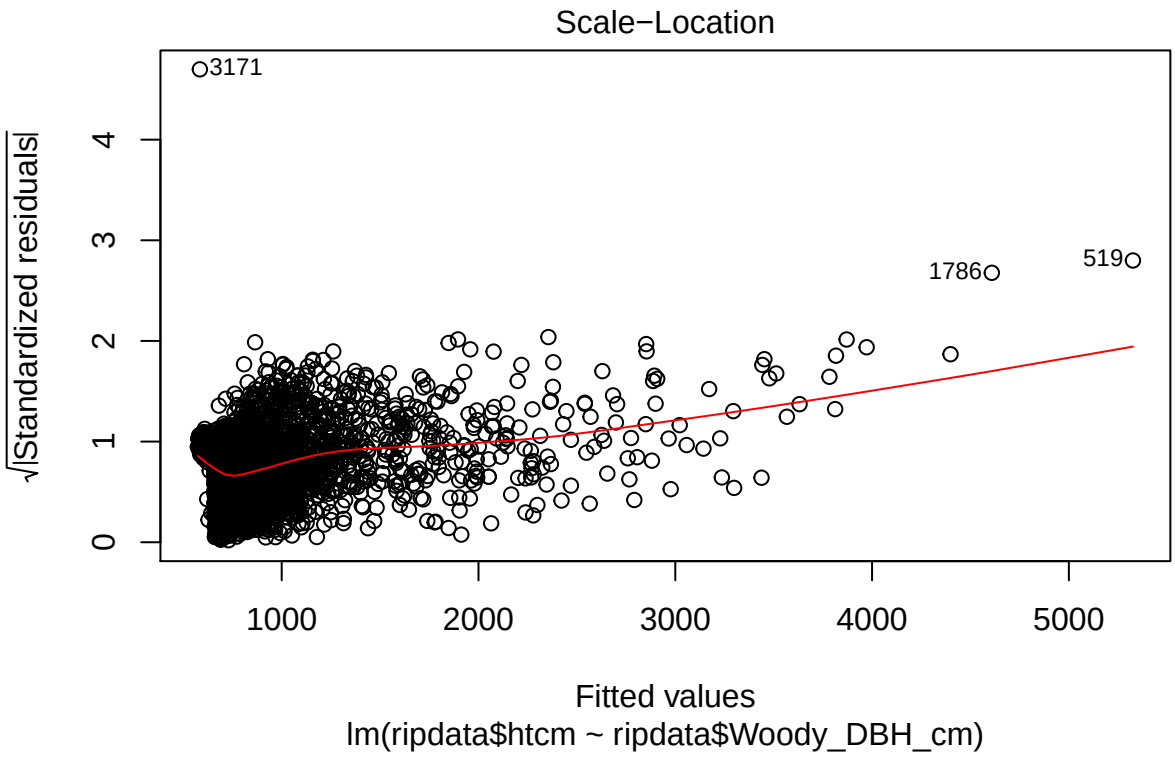
```
summary(ripss$htcm)
```

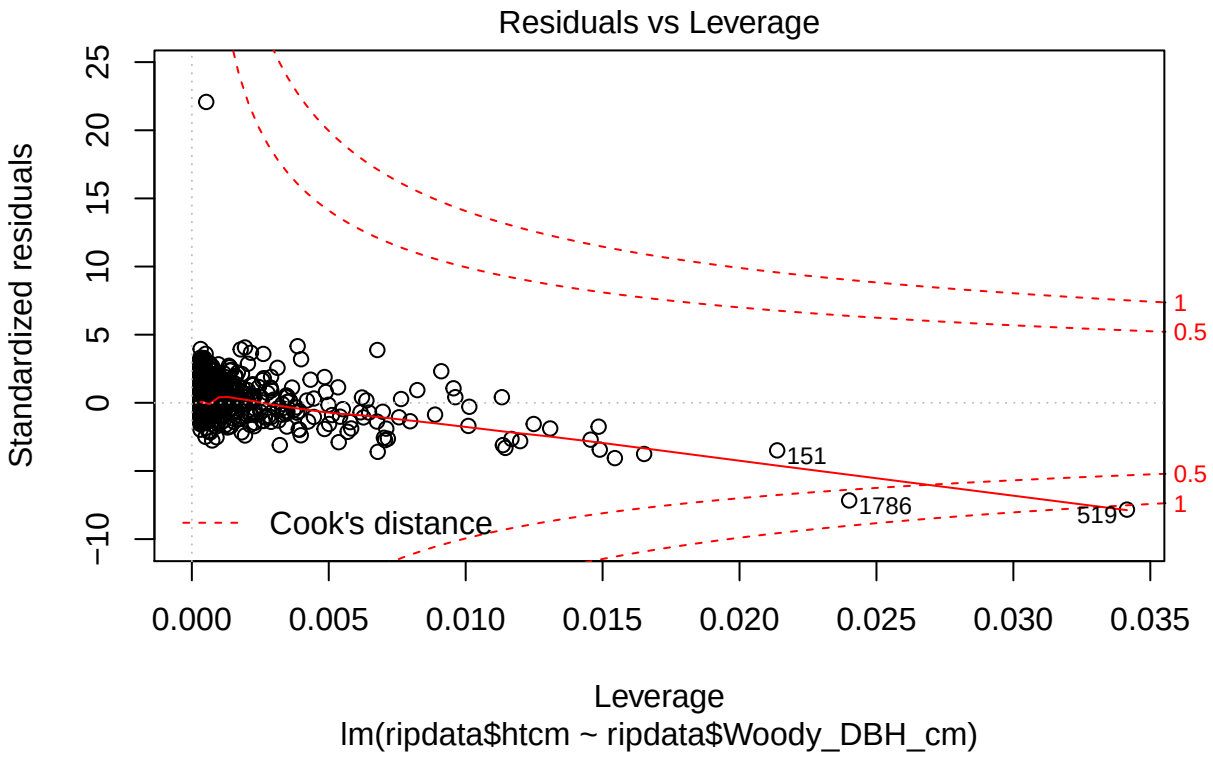
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    45.0   537.2   800.0   923.5  1160.0  4198.0
```

```
#plot the linear model and scattergram for comparison
plot(riplm)
```

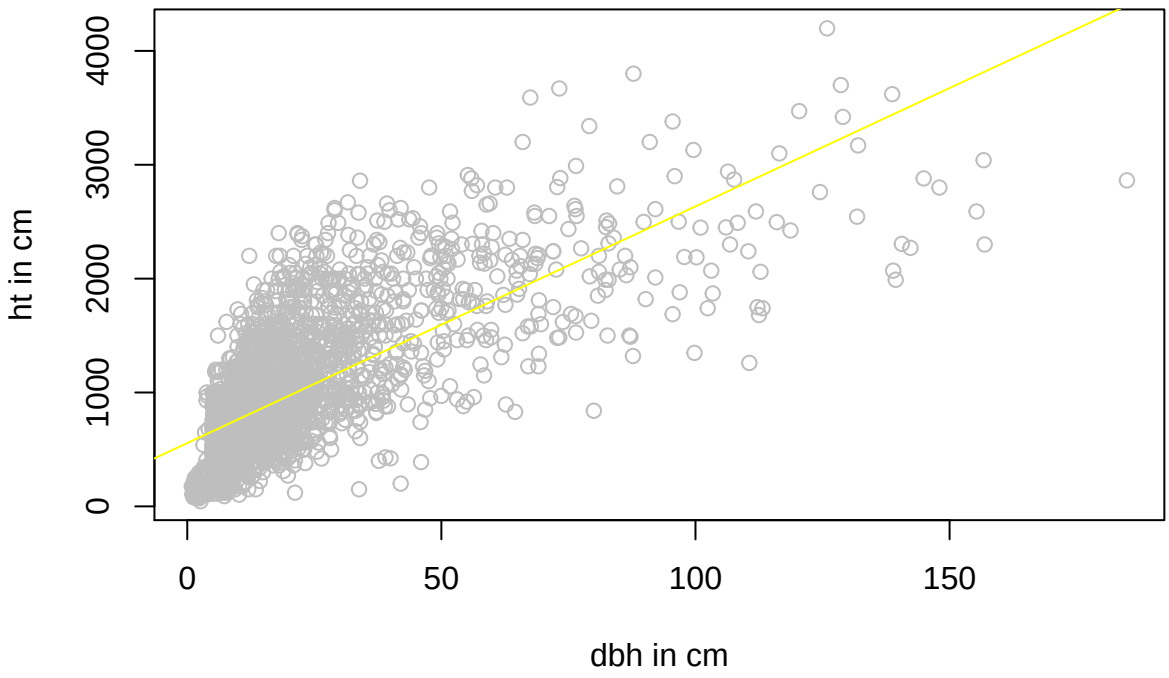



Normal Q−Q
Standardized residuals
25
20
15
10
5
0
−5
3171
1786
519
−2
0
2
Theoretical Quantiles
lm(ripdata$htcm ~ ripdata$Woody_DBH_cm)


Scale−Location
3171
1786
519
√|Standardized residuals|
0
1
2
3
4
1000
2000
3000
4000
5000
Fitted values
lm(ripdata$htcm ~ ripdata$Woody_DBH_cm)




```
plot(ripss$Woody_DBH_cm,ripss$htcm,col = "grey",xlab="dbh in cm",ylab="ht in cm")
abline(riplm,col = "yellow")
```



Once the outlier data were removed, we can see an improvement in the dataset. The median and mean of the new subset are much closer; however, it is even more apparent that we must perform a logrithmic transformation of the data in order to improve our r-squared value.

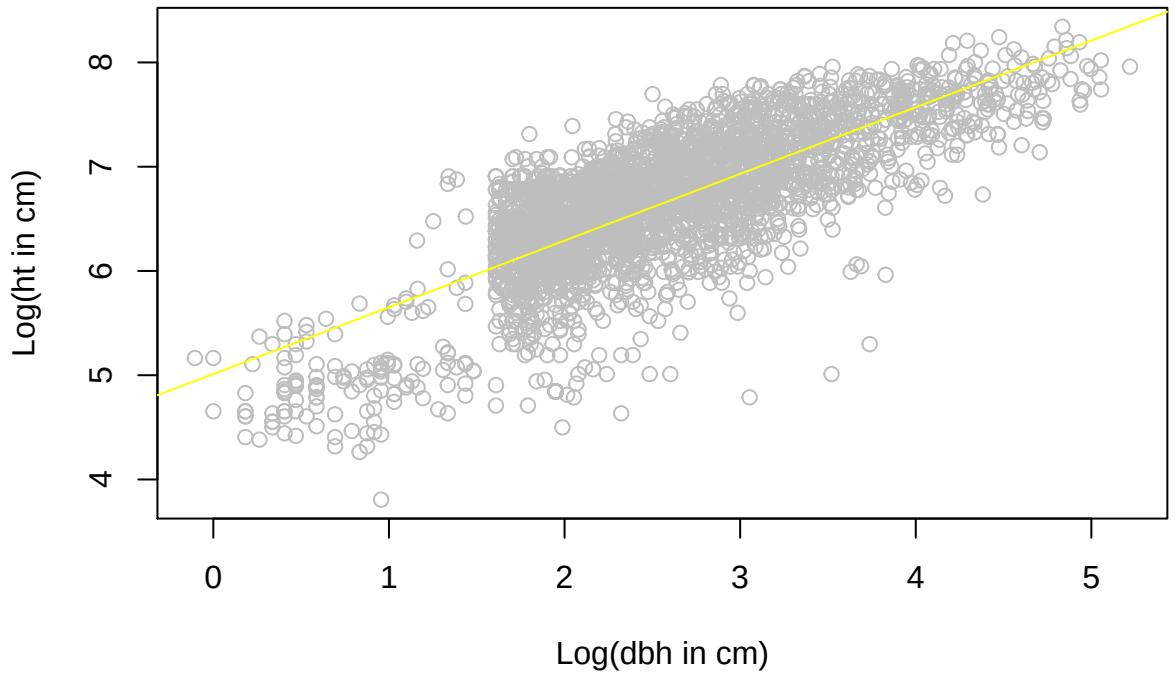
We begin transforming our data by creating a new dataframe to store our log corrected values, then inserting the log transformed height and DBH. Next, a linear model is assembled and summary printed. Finally, a scatterplot and regression line are printed.

```
#Create a new data frame for which we will insert our log corrected data
ripsslog <-ripss
ripsslog$htcmlog <- log(ripsslog$htcm)
ripsslog$dbhlog <- log(ripsslog$Woody_DBH_cm)
#Create a new linear model
rsloglm <- lm(ripsslog$htcmlog~ripsslog$dbhlog)
summary(rsloglm)
```

```
##
## Call:
## lm(formula = ripsslog$htcmlog ~ ripsslog$dbhlog)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.25280 -0.24353  0.03849  0.29499  1.15026
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)     5.011542   0.025388  197.40   <2e-16 ***
## ripsslog$dbhlog 0.639659   0.009572   66.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4323 on 3180 degrees of freedom
## Multiple R-squared:  0.5841, Adjusted R-squared:  0.584
## F-statistic:  4466 on 1 and 3180 DF,  p-value: < 2.2e-16
```

```r
#Plot scattergram and regression line
plot(ripsslog$dbhlog,ripsslog$htcmlog,col = "grey", xlab = "Log(dbh in cm)", ylab = "Log(ht in cm)")
abline(rsloglm,col = "yellow")
```



This global model that is natural log transformed produces a favorable looking scatterplot. the transformation yields a much more evenly distributed plot, allowing for the analyst to easily see that there is a logorithmic correlation between height and DBH. The r-squared value has risen to `0.5939`, which is not great but within the realm of determing that the regression line is a good fit.

```r
Acerdf <- ripsslog[ripsslog$Genus=="Acer",]
Acerlm <- lm(Acerdf$htcm ~ Acerdf$dbhlog)
Acerr2 <- summary(Acerlm)$r.squared

Fraxdf <- ripsslog[ripsslog$Genus=="Fraxinus",]
Fraxlm <- lm(Fraxdf$htcm ~ Fraxdf$dbhlog)
Fraxr2 <- summary(Fraxlm)$r.squared
```

```
Popdf <- ripsslog[ripsslog$Genus=="Populus",]
Poplm <- lm(Popdf$htcm ~ Popdf$dbhlog)
Popr2 <- summary(Poplm)$r.squared

Quedf <- ripsslog[ripsslog$Genus=="Quercus",]
Quelm <- lm(Quedf$htcm ~ Quedf$dbhlog)
Quer2 <- summary(Quelm)$r.squared

Saldf <- ripsslog[ripsslog$Genus=="Salix",]
Sallm <- lm(Saldf$htcm ~ Saldf$dbhlog)
Salr2 <- summary(Sallm)$r.squared
#CORSPdf <- ripsslog[ripsslog$ProjCode=="CORSP",] # There are no values at this location
#CORSPlm <- lm(CORSPdf$htcm ~ CORSPdf$dbhlog) #
#summary(CORSPlm)

HEROWdf <- ripsslog[ripsslog$ProjCode=="HEROW",]
HEROWlm <- lm(HEROWdf$htcm ~ HEROWdf$dbhlog)
HEROWr2 <- summary(HEROWlm)$r.squared

NAPSOdf <- ripsslog[ripsslog$ProjCode=="NAPSO",]
NAPSOlm <- lm(NAPSOdf$htcm ~ NAPSOdf$dbhlog)
NAPSOr2 <- summary(NAPSOlm)$r.squared

SACTOdf <- ripsslog[ripsslog$ProjCode=="SACTO",]
SACTOlm <- lm(SACTOdf$htcm ~ SACTOdf$dbhlog)
SACTOr2 <- summary(SACTOlm)$r.squared
```

By examining the coefficient of variation (`R^2`) for each linear model, we can detemine that the "best" model is the one for project site SACTO with `R^2`=0.6990893. This proves that the linear model for the SACTO location is the best representation of all other linear models at locations. This model should be used for the evaluation and projection of carbon stocks.

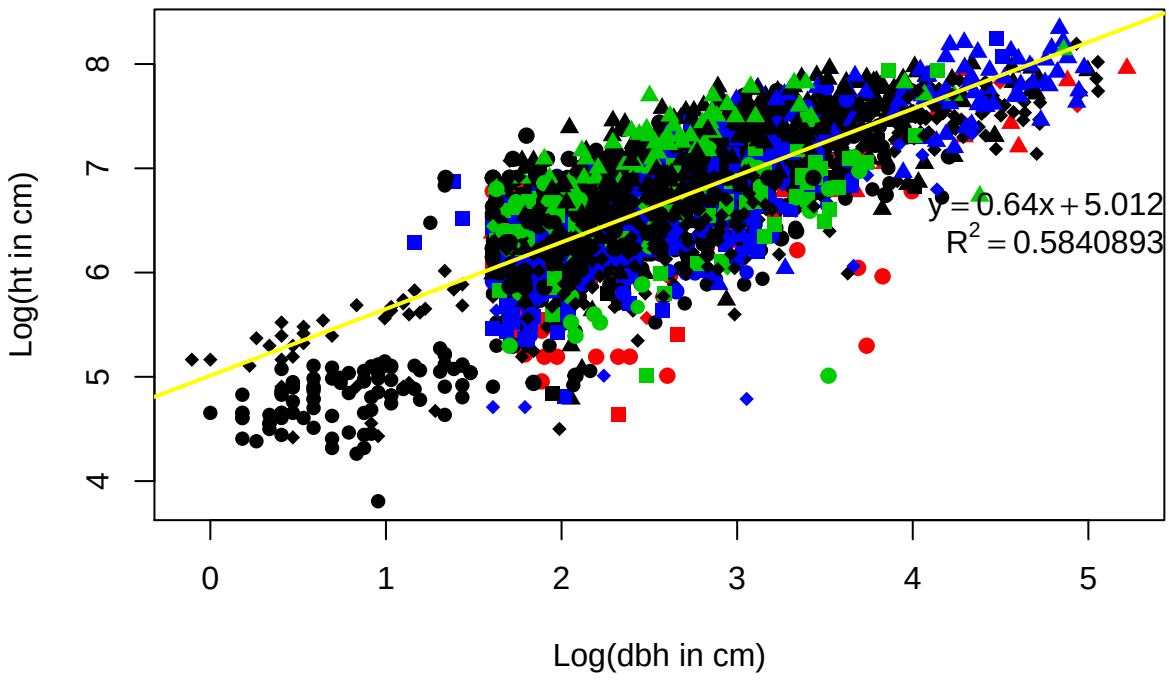
Finally, we create a single scatteplot with all log corrected data, but each Project Code is represented by a different color and each Genus is represented by a different symbol.

```
#Plot the Master Scatter Plot, we use the ProjCode to determine the color of the points, and the Genus
plot(ripsslog$dbhlog,ripsslog$htcmlog,pch=c(15,16,17,18,19)[as.numeric(ripsslog$Genus)],col=ripsslog$Pr

abline(rsloglm,col = "yellow",lwd = "2")

ripsslogrs <- summary(rsloglm)

lm_coef <- round(coef(rsloglm), 3) # extract coefficients
mtext(bquote(y == .(lm_coef[2])*x + .(lm_coef[1])), adj=1, padj=7.5) # display equation
mtext(bquote(R^2 == .(ripsslogrs$r.squared)), adj=1, padj=8)
```



Discussion: Examining the results, it is obvious to see that the initial data yielded unfavorable fitting results. Firstly, there were several outliers that needed extraction; secondly, the data needed logrithmic transformation. This resulted in a favorable coefficient of determination. This lab additionally allowed us to explore a powerful analysis tool: `outlierTest()`.

Limitations: We have removed outlier data based on the Bonferroni test, which is a post processing solution. It would be much better if these datum points (if by human error) can be caught at the source. Furthermore, we have restricted our data transformation to log-log, but there are surely better fitting tranformation and fitting options.