432 Final Project Markdown

Clarisa Griffin

2025-03-30

Github Link: https://github.com/CEGriffin/432-Final-Project (https://github.com/CEGriffin/432-Final-Project)

Group members: Clarisa Griffin (20270784), Emilia Gillette (20204160), Harnaaz Sandhu (20258736), Aidan McLeod (20294254), Maddigan Kales (20259834), and Abigail Kaye (20271241)

Data Manipulation

Non-Normalized Data

```
#load libraries
library(dplyr)
library(tidyverse)
library(tidyr)
library(ggplot2)

#load data
data<-read.csv("DataSynthesis.csv") #DataSynthesis.csv is pulled from Colautti et al., 2023

#separate out tag info into different lines

#create new columns to be turned into different aspects of the plant ID
#rename family to population, because theres another column that will actually eb the family
#remove un-needed columns
data<-data%>%
  rename(population=Family)%>%
  mutate(petri_dish=Tag) %>%
  mutate(family=Tag) %>%
  mutate(common_garden=Tag)%>%
  select(c("Tag", "petri_dish", "population", "family", "common_garden", "ID", starts_with("g
h"), ends_with("Field"), starts_with("GM"), ends_with("Bolt"), starts_with("Chlor"), ends_with
("Conc"), starts_with("RGR")), -c("GM_Fecundity", ends_with("Initial"), "GM_Leaf_Number"))

#use regular expressions to replace each plant ID column  with only the aspect of the ID needed
data$petri_dish<-sub("(\\w?).+","\\1", data$petri_dish)
data$family<-sub("\\w?\\|\\w+-([A-z0-9-]+)\\|.*", "\\1", data$family)
data$common_garden<-sub("\\w?\\|.+\\|(\\w)\\|.+", "\\1", data$common_garden)

#remove maple observations (we are only interested in garlic mustard)
data<-subset(data, population!="maple")

#remove observations for which there is no common garden data
unique(data$common_garden)
```

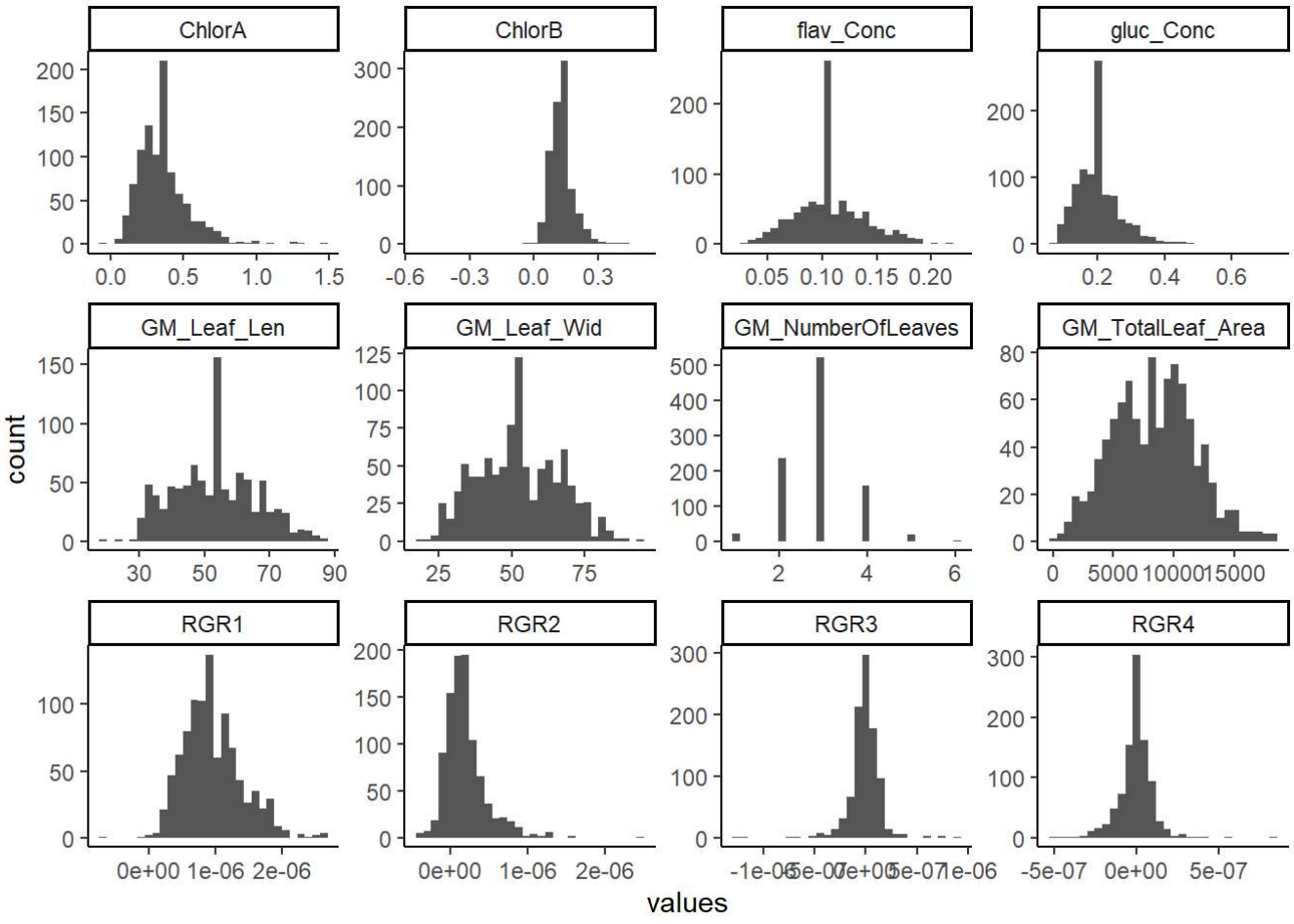
```
## [1] "Q"               "N"               "e|JBCHY1-1-50|?"  "V"
## [5] "i163_2"          "b|WSSWM3-1-0|?|?"
```

```
data<-subset(data, common_garden!="e|JBCHY1-1-50|?")
data<-subset(data, common_garden!="b|WSSWM3-1-0|?|?")
data<-subset(data, common_garden!="i163_2")

#Ensure columns are the right data class
#str(data)
data$Tag<-as.factor(data$Tag)
data$petri_dish<-as.factor(data$petri_dish)
data$population<-as.factor(data$population)
data$family<-as.factor(data$family)
data$common_garden<-as.factor(data$common_garden)
data$ID<-as.factor(data$ID)

#impute missing data by replacing NA with the mean of the column for numerical variables
#"bolt" data will be done later- needs further manipulation
data<-data%>%
  mutate(RGR1=ifelse(is.na(RGR1),mean(data$RGR1, na.rm=T),RGR1),
         RGR2=ifelse(is.na(RGR2),mean(data$RGR2, na.rm=T),RGR2),
         RGR3=ifelse(is.na(RGR3),mean(data$RGR3, na.rm=T),RGR3),
         RGR4=ifelse(is.na(RGR4),mean(data$RGR4, na.rm=T),RGR4),
         ChlorA=ifelse(is.na(ChlorA),mean(data$ChlorA, na.rm=T),ChlorA),
         ChlorB=ifelse(is.na(ChlorB),mean(data$ChlorB, na.rm=T),ChlorB),
         gluc_Conc=ifelse(is.na(gluc_Conc),mean(data$gluc_Conc, na.rm=T),gluc_Conc),
         flav_Conc=ifelse(is.na(flav_Conc),mean(data$flav_Conc, na.rm=T),flav_Conc),
         GM_Leaf_Len=ifelse(is.na(GM_Leaf_Len),mean(data$GM_Leaf_Len, na.rm=T),GM_Leaf_Len),
         GM_Leaf_Wid=ifelse(is.na(GM_Leaf_Wid),mean(data$GM_Leaf_Wid, na.rm=T),GM_Leaf_Wid),
         GM_TotalLeaf_Area =ifelse(is.na(GM_TotalLeaf_Area),mean(data$GM_TotalLeaf_Area, na.rm=
T),GM_TotalLeaf_Area),
         GM_NumberOfLeaves =ifelse(is.na(GM_NumberOfLeaves),mean(data$GM_NumberOfLeaves, na.rm=
T),GM_NumberOfLeaves)
)

#create mortality column out of "bolt" data
#most of the bolt columns are "dead", will have to do analysis based on just mortality, then aga
in with living bolt data
data<-data%>%
  mutate(mortality=Larg_Leaf_Len_Bolt)

data$mortality<-sub("\\d+","1", data$mortality)
data$mortality<-sub("Dead","0", data$mortality)

#separate out bolt data
bolt_data<-subset(data, Larg_Leaf_Len_Bolt!="Dead")

#Ensure columns are the right data class
bolt_data$Larg_Leaf_Len_Bolt<-as.numeric(bolt_data$Larg_Leaf_Len_Bolt)
bolt_data$Larg_Leaf_Wid_Bolt<-as.numeric(bolt_data$Larg_Leaf_Wid_Bolt)
bolt_data$GM_StemHeight_Bolt<-as.numeric(bolt_data$GM_StemHeight_Bolt)
bolt_data$GM_Leaf_Number_Bolt<-as.numeric(bolt_data$GM_Leaf_Number_Bolt)

#impute NAs
bolt_data<-bolt_data%>%
```

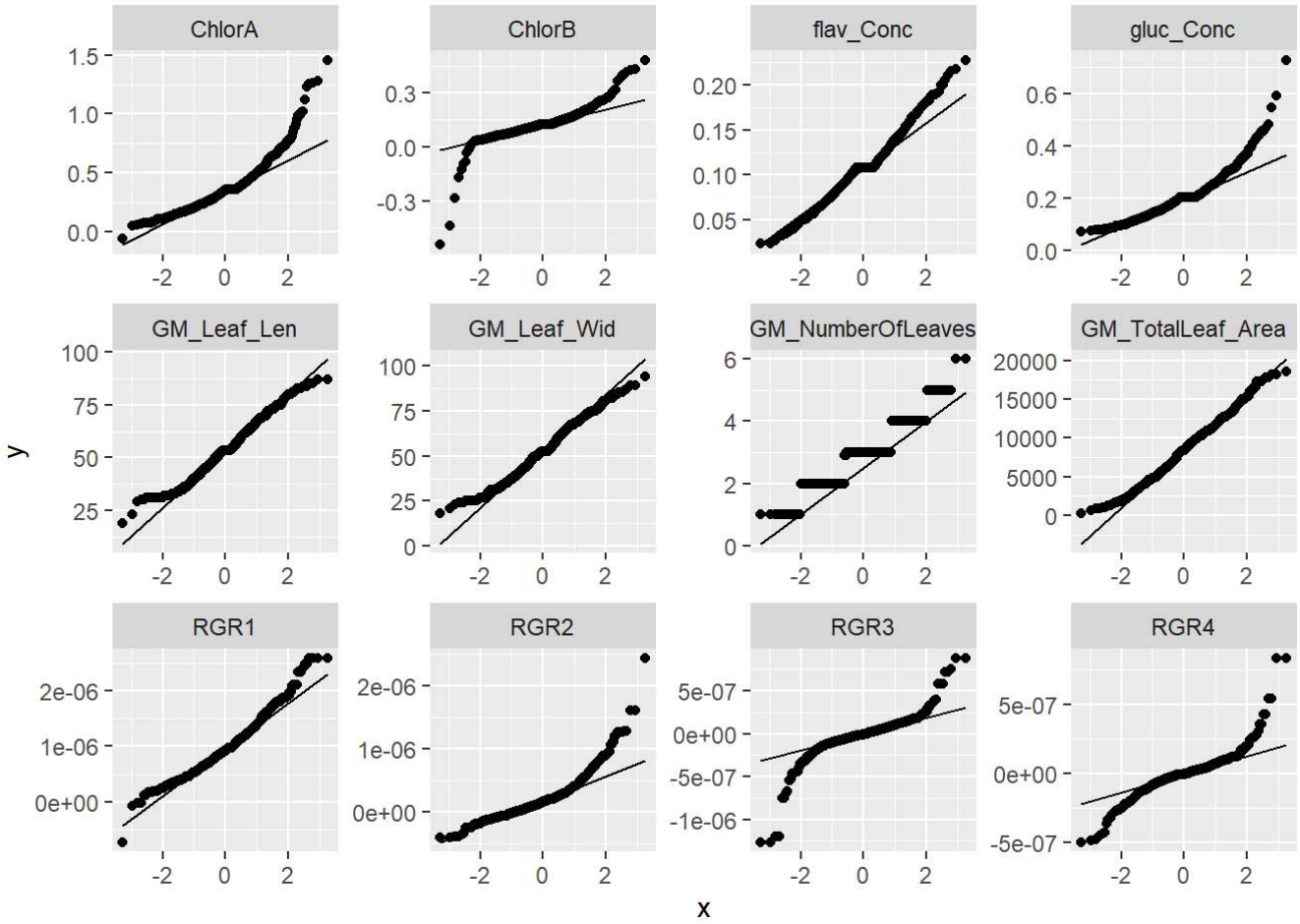
```
  mutate(Larg_Leaf_Len_Bolt=ifelse(is.na(Larg_Leaf_Len_Bolt),mean(data$Larg_Leaf_Len_Bolt, na.rm
=T),Larg_Leaf_Len_Bolt),
         Larg_Leaf_Wid_Bolt=ifelse(is.na(Larg_Leaf_Wid_Bolt),mean(data$Larg_Leaf_Wid_Bolt, na.rm
=T),Larg_Leaf_Wid_Bolt),
         GM_StemHeight_Bolt=ifelse(is.na(GM_StemHeight_Bolt),mean(data$GM_StemHeight_Bolt, na.rm
=T),GM_StemHeight_Bolt),
         GM_NumberOfLeaves=ifelse(is.na(GM_NumberOfLeaves),mean(data$GM_NumberOfLeaves, na.rm=
T),GM_NumberOfLeaves))

#write files into a csv
write.csv(data, file="./raw_data.csv", row.names=F)
write.csv(bolt_data, file="./raw_bolt_data.csv", row.names=F)
```

Normalized Data

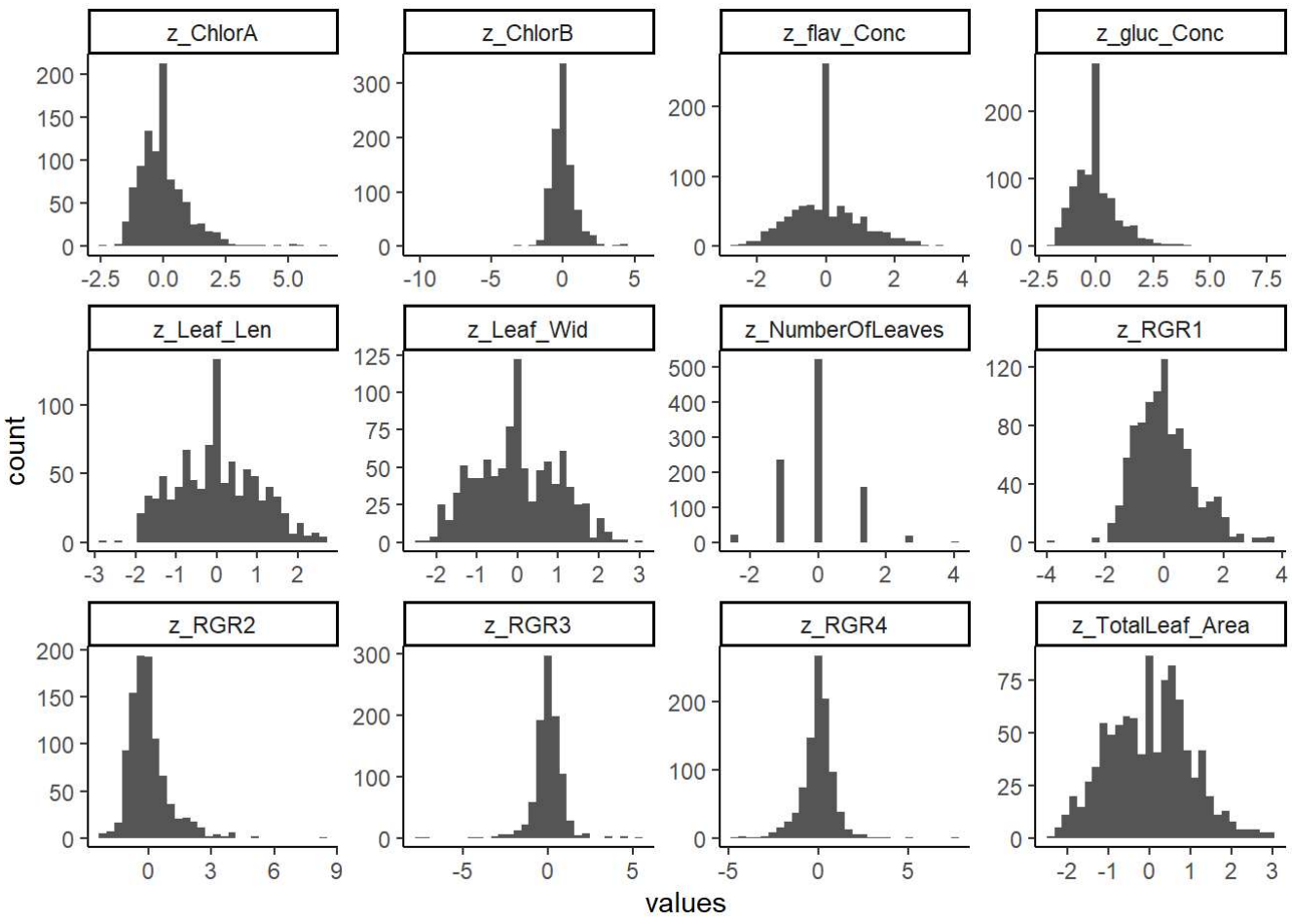
```
#Check normality of numerical variables with histogram
long_data<-pivot_longer(data, c(12:15, 20:27), names_to="metric", values_to="values")

ggplot(long_data, aes(x=values)) +
  geom_histogram() +
  theme_classic() +
  facet_wrap(~metric, scales="free")
```



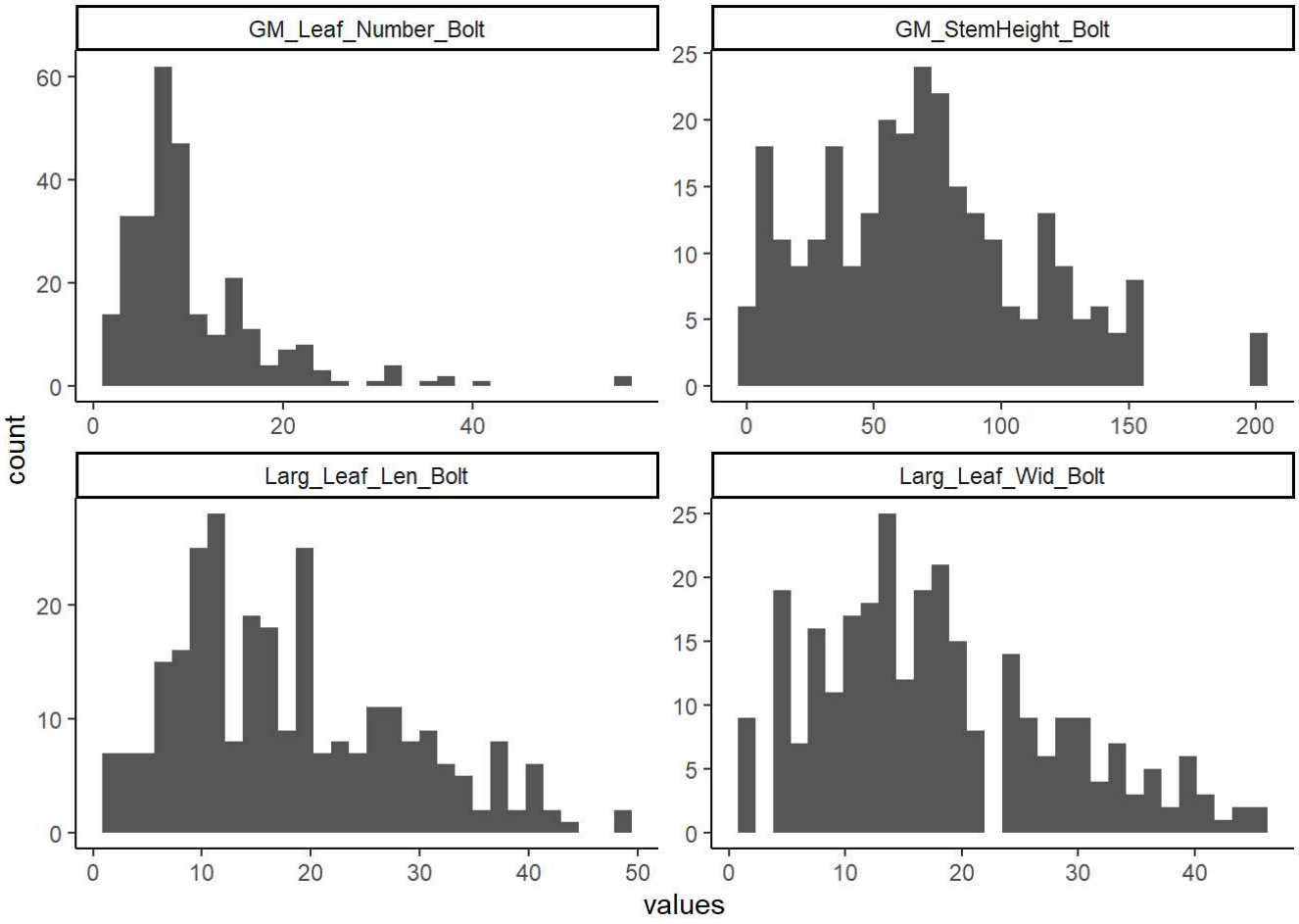
```
#check for normality with qq plots
ggplot(long_data, aes(sample=values)) +
  geom_qq() +
  stat_qq_line() +
  facet_wrap(~metric, scales="free") #nothing looks too far from normal here, probably doesnt ne
ed any kind of normalization other than conversion to Z scores
```



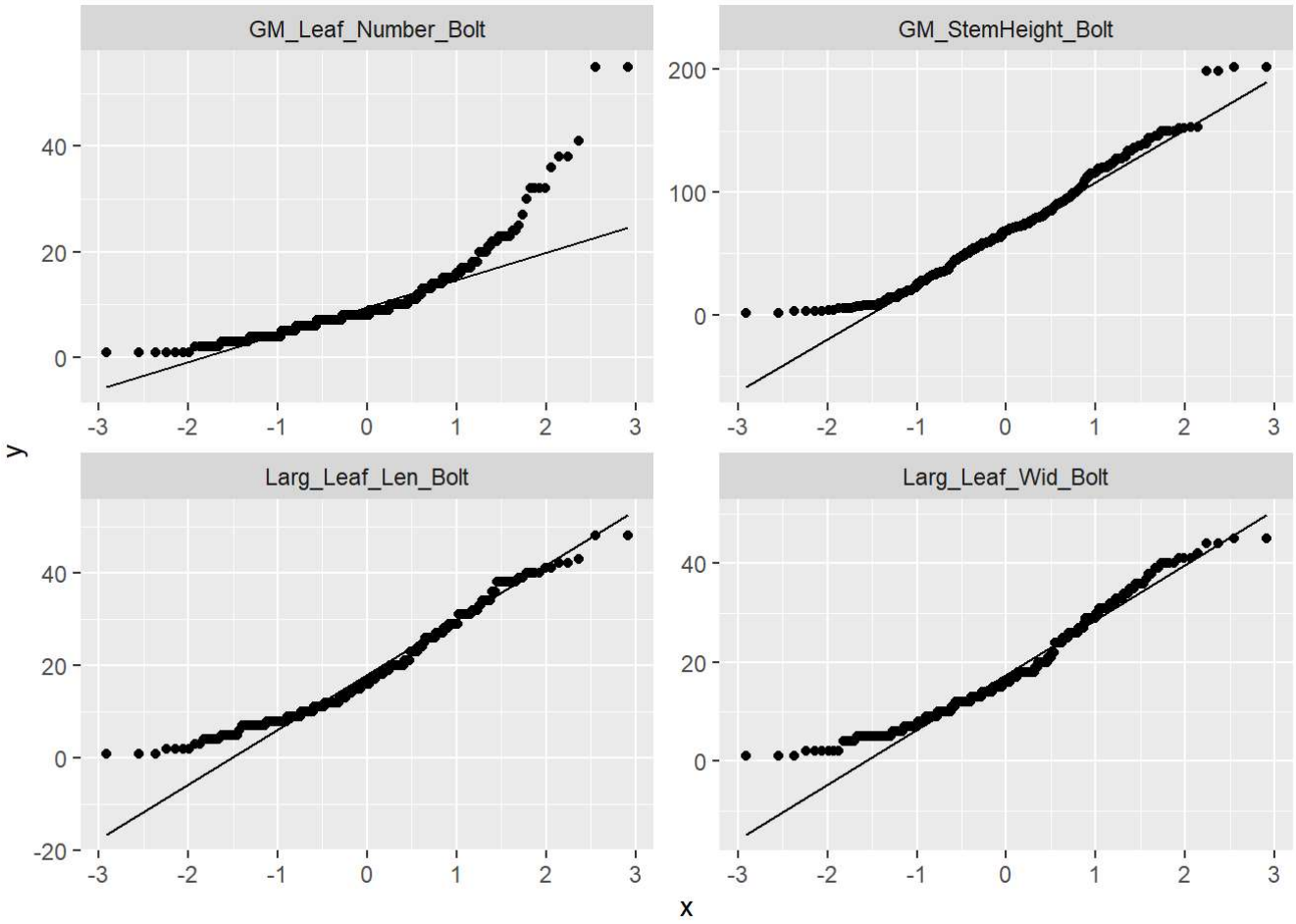
```
#transform all numerical columns to z scores
data<-data%>%
  mutate(z_RGR1=(RGR1-mean(RGR1))/sd(RGR1),
         z_RGR2=(RGR2-mean(RGR2))/sd(RGR2),
         z_RGR3=(RGR3-mean(RGR3))/sd(RGR3),
         z_RGR4=(RGR4-mean(RGR4))/sd(RGR4),
         z_ChlorA=(ChlorA-mean(ChlorA))/sd(ChlorA),
         z_ChlorB=(ChlorB-mean(ChlorB))/sd(ChlorB),
         z_gluc_Conc=(gluc_Conc-mean(gluc_Conc))/sd(gluc_Conc),
         z_flav_Conc=(flav_Conc-mean(flav_Conc))/sd(flav_Conc),
         z_Leaf_Len=(GM_Leaf_Len-mean(GM_Leaf_Len))/sd(GM_Leaf_Len),
         z_Leaf_Wid=(GM_Leaf_Wid-mean(GM_Leaf_Wid))/sd(GM_Leaf_Wid),
         z_TotalLeaf_Area=(GM_TotalLeaf_Area-mean(GM_TotalLeaf_Area))/sd(GM_TotalLeaf_Area),
         z_NumberOfLeaves=(GM_NumberOfLeaves-mean(GM_NumberOfLeaves))/sd(GM_NumberOfLeaves))%>%
   select("Tag", "petri_dish", "population", "family", "common_garden", "ID", "gh_bench", "gh_co
l", "gh_row", "Row_Field", "Col_Field", starts_with("z_"), "mortality")

#how does it look after the transformation?
long_data<-pivot_longer(data, starts_with("z_"), names_to="metric", values_to="values")

ggplot(long_data, aes(x=values)) +
  geom_histogram() +
  theme_classic() +
  facet_wrap(~metric, scales="free")
```



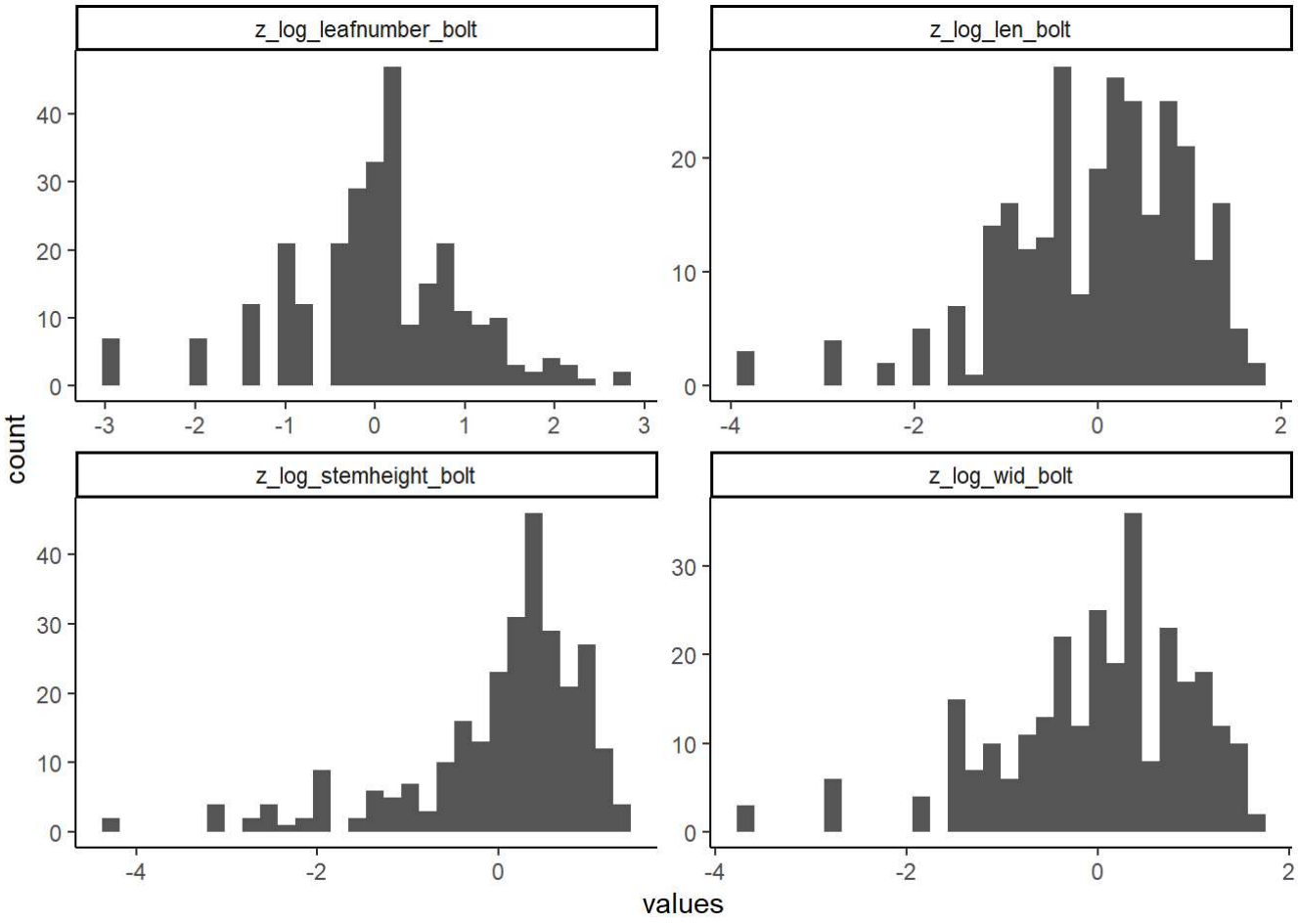
```
#bolt data check for normality - needs log transformation
long_bolt_data<-pivot_longer(bolt_data, ends_with("Bolt"), names_to="metric", values_to="value
s")

ggplot(long_bolt_data, aes(x=values)) +
  geom_histogram() +
  theme_classic() +
  facet_wrap(~metric, scales="free")
```



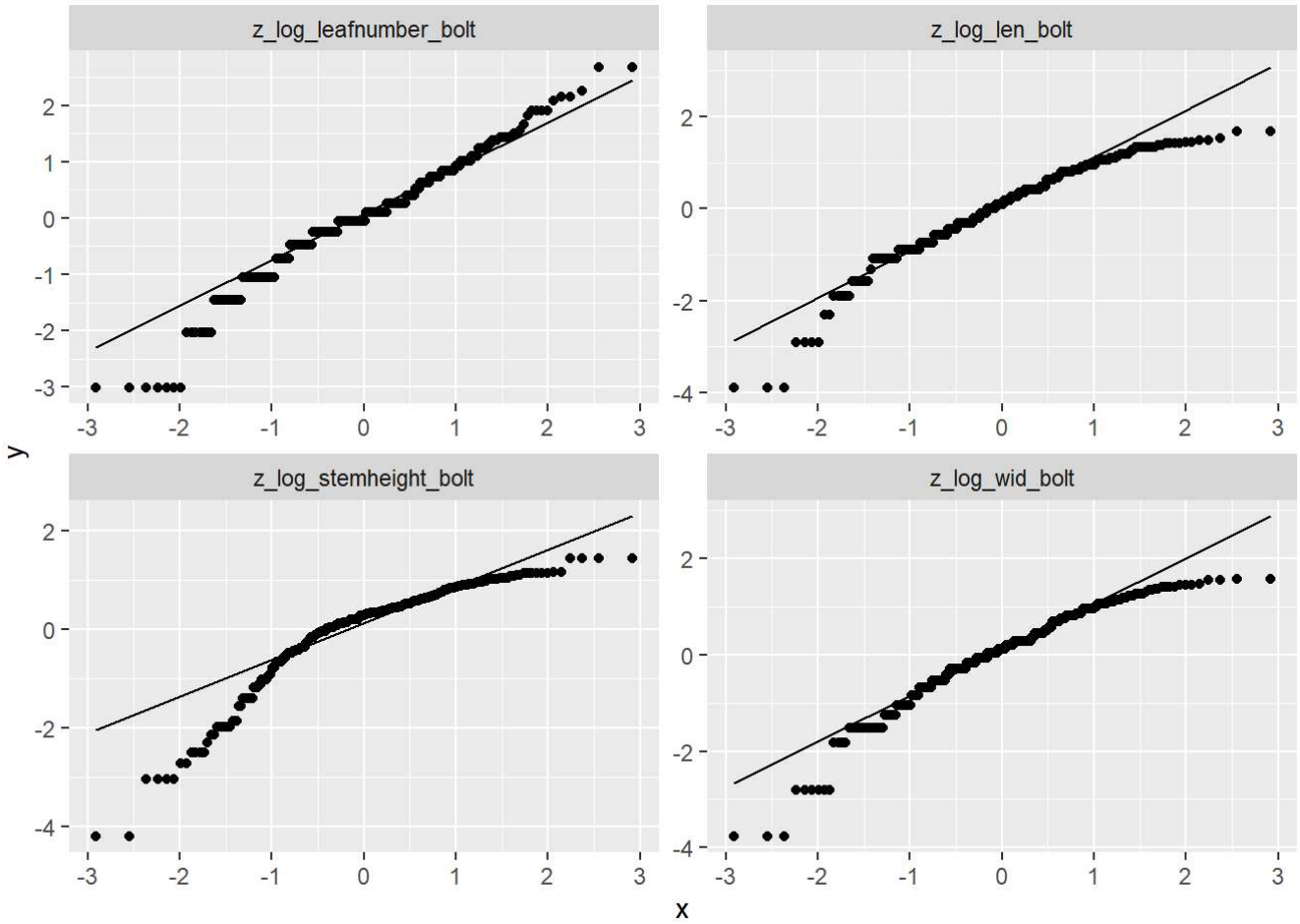

```
#visualize with qq plot
ggplot(long_bolt_data, aes(sample=values)) +
  geom_qq() +
  stat_qq_line() +
  facet_wrap(~metric, scales="free") #also doesnt look too bad
```


GM_Leaf_Number_Bolt
GM_StemHeight_Bolt
Larg_Leaf_Len_Bolt
Larg_Leaf_Wid_Bolt
y
x

```
#normalize bolt data and log transform
bolt_data<-bolt_data%>%
  mutate(Larg_Leaf_Len_Bolt=log(Larg_Leaf_Len_Bolt),
         Larg_Leaf_Wid_Bolt=log(Larg_Leaf_Wid_Bolt),
         GM_StemHeight_Bolt=log(GM_StemHeight_Bolt),
         GM_Leaf_Number_Bolt=log(GM_Leaf_Number_Bolt))%>%
  mutate(z_log_len_bolt=(Larg_Leaf_Len_Bolt-mean(Larg_Leaf_Len_Bolt))/sd(Larg_Leaf_Len_Bolt),
         z_log_wid_bolt=(Larg_Leaf_Wid_Bolt-mean(Larg_Leaf_Wid_Bolt))/sd(Larg_Leaf_Wid_Bolt),
         z_log_stemheight_bolt=(GM_StemHeight_Bolt-mean(GM_StemHeight_Bolt))/sd(GM_StemHeight_Bo
lt),
         z_log_leafnumber_bolt=(GM_Leaf_Number_Bolt-mean(GM_Leaf_Number_Bolt))/sd(GM_Leaf_Number
_Bolt),
         z_RGR1=(RGR1-mean(RGR1))/sd(RGR1),
         z_RGR2=(RGR2-mean(RGR2))/sd(RGR2),
         z_RGR3=(RGR3-mean(RGR3))/sd(RGR3),
         z_RGR4=(RGR4-mean(RGR4))/sd(RGR4),
         z_ChlorA=(ChlorA-mean(ChlorA))/sd(ChlorA),
         z_ChlorB=(ChlorB-mean(ChlorB))/sd(ChlorB),
         z_gluc_Conc=(gluc_Conc-mean(gluc_Conc))/sd(gluc_Conc),
         z_flav_Conc=(flav_Conc-mean(flav_Conc))/sd(flav_Conc),
         z_Leaf_Len=(GM_Leaf_Len-mean(GM_Leaf_Len))/sd(GM_Leaf_Len),
         z_Leaf_Wid=(GM_Leaf_Wid-mean(GM_Leaf_Wid))/sd(GM_Leaf_Wid),
         z_TotalLeaf_Area=(GM_TotalLeaf_Area-mean(GM_TotalLeaf_Area))/sd(GM_TotalLeaf_Area),
         z_NumberOfLeaves=(GM_NumberOfLeaves-mean(GM_NumberOfLeaves))/sd(GM_NumberOfLeaves)) %>%
  select("Tag", "petri_dish", "population", "family", "common_garden", "ID", "gh_bench", "gh_co
l", "gh_row", "Row_Field", "Col_Field", starts_with("z_"), "mortality")


#check to see if the log transform did anything - looks much better
long_bolt_data<-pivot_longer(bolt_data, ends_with("bolt"), names_to="metric", values_to="value
s")

ggplot(long_bolt_data, aes(x=values)) +
  geom_histogram() +
  theme_classic() +
  facet_wrap(~metric, scales="free")
```



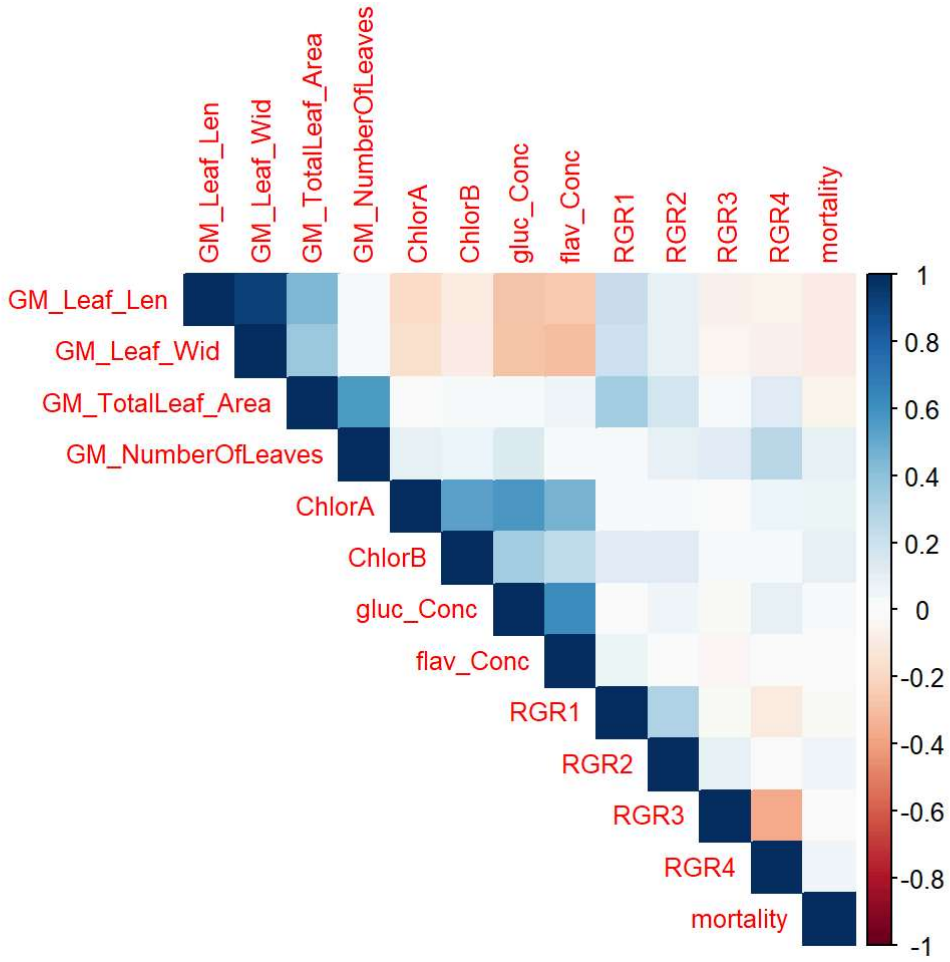

```
ggplot(long_bolt_data, aes(sample=values)) +
  geom_qq() +
  stat_qq_line() +
  facet_wrap(~metric, scales="free") #Looks better
```




```
#write files into a csv
write.csv(data, file="./normalized_data.csv", row.names=F)
write.csv(bolt_data, file="./normalized_bolt_data.csv", row.names=F)
```

Correlation Matrix

```
#Load libraries
library(readr)
library(corrplot)
library(dplyr)

#Load in Data from GitHub
MyData <- read.csv("raw_data.csv", header=T)
View(MyData)

#select only numeric variable columns
NumericDat<-MyData%>%
  select(c(12:15, 20:28))

#Creating correlation matrix
CorMatrix <- cor(NumericDat, use = "pairwise.complete.obs")

# Plot the correlation matrix
corrplot(CorMatrix, method = "color", type = "upper", tl.cex = 0.8)
```

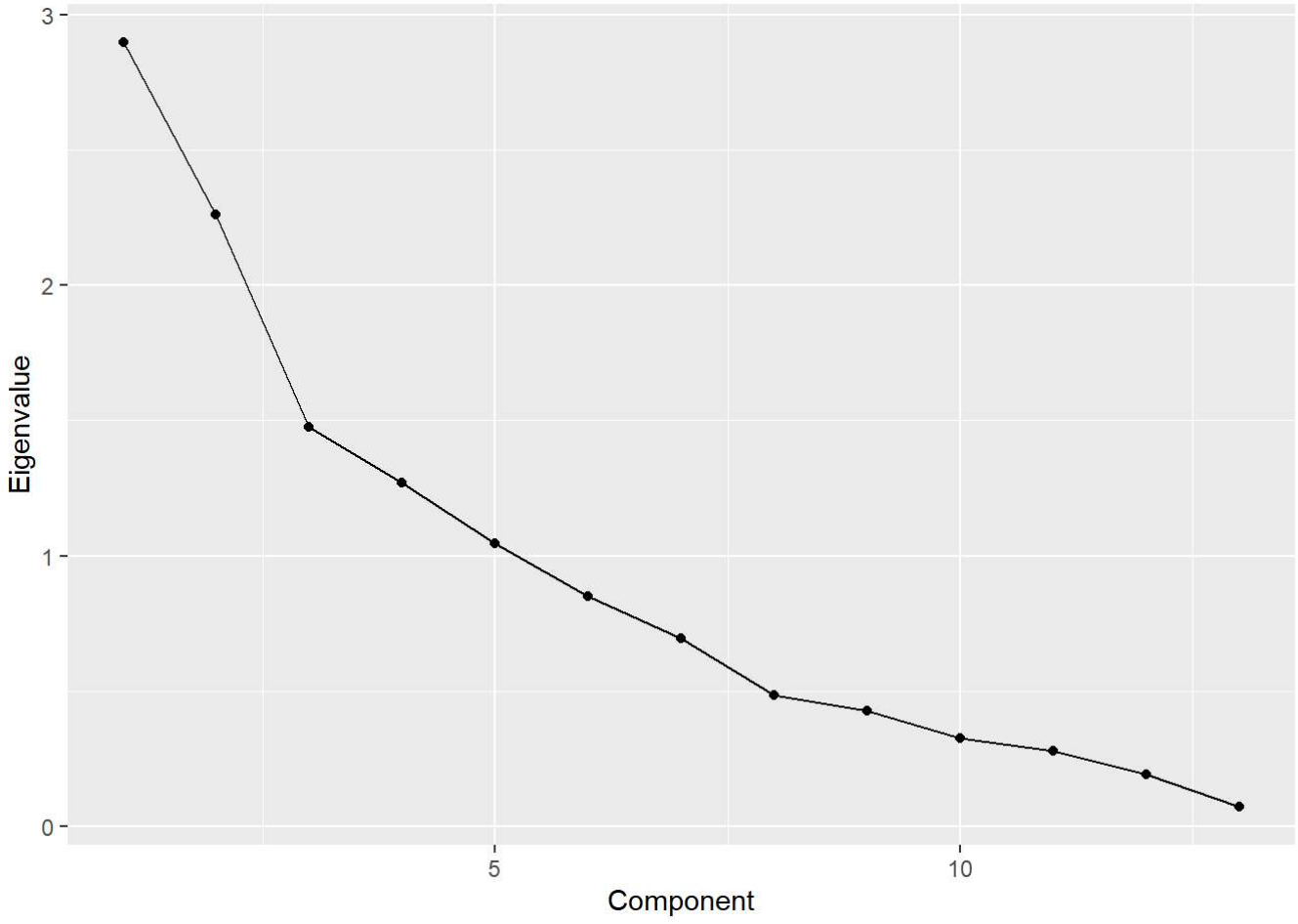



```
#Creating a table
CorTable <- as.data.frame(CorMatrix)
print(CorTable)
```

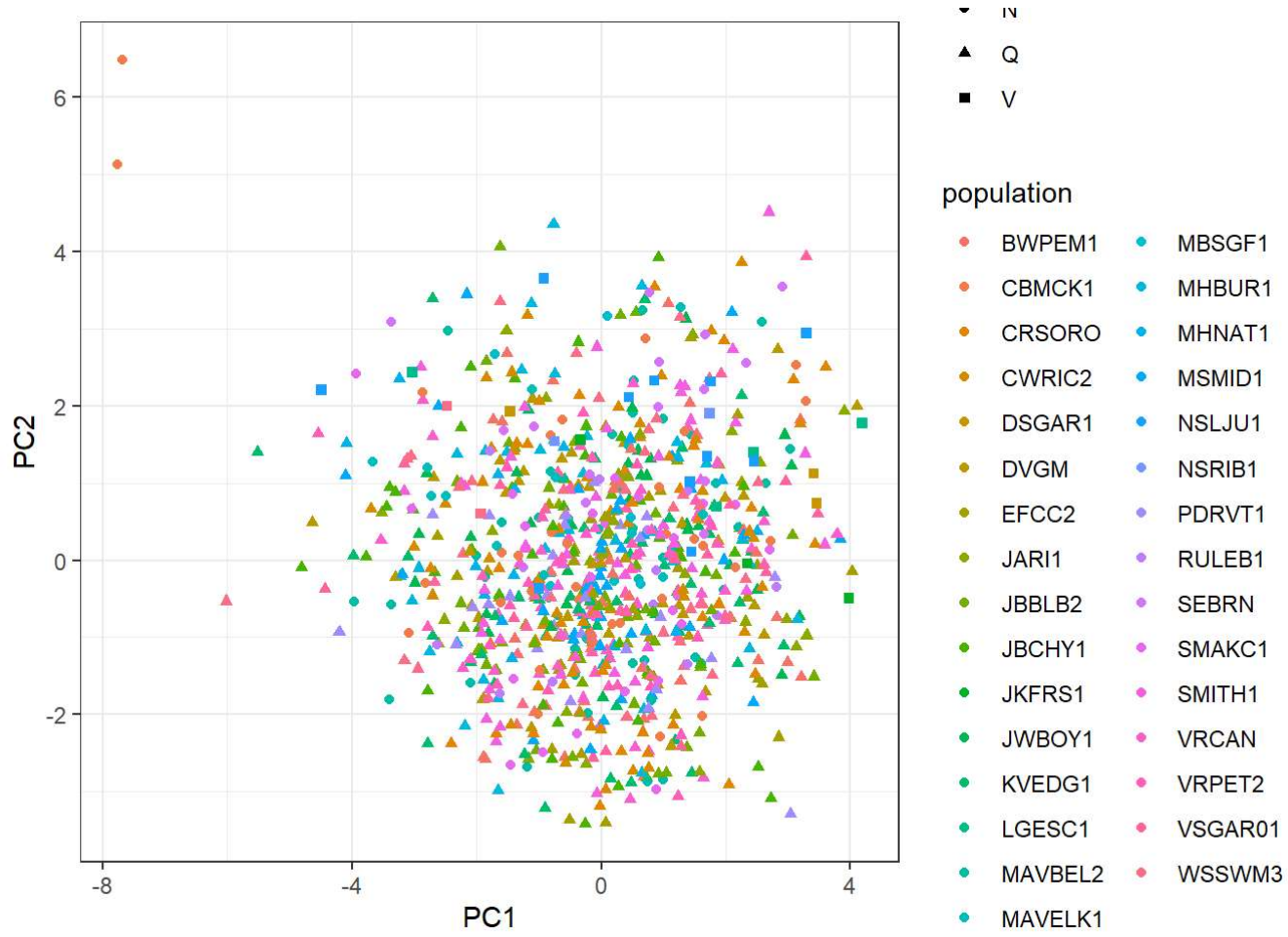
```
##                   GM_Leaf_Len GM_Leaf_Wid GM_TotalLeaf_Area GM_NumberOfLeaves
## GM_Leaf_Len        1.00000000  0.92243614       0.446761957        0.02775980
## GM_Leaf_Wid        0.92243614  1.00000000       0.368246505        0.02713755
## GM_TotalLeaf_Area  0.44676196  0.36824651       1.000000000        0.56624704
## GM_NumberOfLeaves  0.02775980  0.02713755       0.566247040        1.00000000
## ChlorA            -0.18156707 -0.16703557      -0.003909924        0.09128036
## ChlorB            -0.09772779 -0.08388664       0.033403797        0.06869063
## gluc_Conc         -0.27758920 -0.28014697       0.022591384        0.13242442
## flav_Conc         -0.25847287 -0.30414455       0.054097934        0.02630874
## RGR1               0.22233175  0.19061400       0.329982641        0.03759002
## RGR2               0.10231114  0.10780417       0.187117313        0.10844814
## RGR3              -0.06301821 -0.04303740       0.031997008        0.12028381
## RGR4              -0.05738792 -0.06052411       0.117677559        0.26444343
## mortality         -0.08476158 -0.08184683      -0.059670361        0.10732874
##                         ChlorA      ChlorB   gluc_Conc    flav_Conc        RGR1
## GM_Leaf_Len       -0.181567066 -0.09772779 -0.27758920 -0.258472868  0.22233175
## GM_Leaf_Wid       -0.167035571 -0.08388664 -0.28014697 -0.304144547  0.19061400
## GM_TotalLeaf_Area -0.003909924  0.03340380  0.02259138  0.054097934  0.32998264
## GM_NumberOfLeaves  0.091280363  0.06869063  0.13242442  0.026308744  0.03759002
## ChlorA             1.000000000  0.53602175  0.57710524  0.452253055  0.03275752
## ChlorB             0.536021752  1.00000000  0.32857776  0.248326107  0.11838743
## gluc_Conc          0.577105237  0.32857776  1.00000000  0.612528213  0.01644203
## flav_Conc          0.452253055  0.24832611  0.61252821  1.000000000  0.07014801
## RGR1               0.032757518  0.11838743  0.01644203  0.070148013  1.00000000
## RGR2               0.027259054  0.12457516  0.04789815  0.008971602  0.30293725
## RGR3               0.003926126  0.02254162 -0.02783111 -0.036904459 -0.02176874
## RGR4               0.063912223  0.03333093  0.09413978  0.014663540 -0.09174434
## mortality          0.076734885  0.10920968  0.03809176  0.002292280 -0.02704984
##                          RGR2         RGR3         RGR4    mortality
## GM_Leaf_Len       0.102311138 -0.063018210 -0.057387921 -0.084761582
## GM_Leaf_Wid       0.107804171 -0.043037401 -0.060524107 -0.081846830
## GM_TotalLeaf_Area 0.187117313  0.031997008  0.117677559 -0.059670361
## GM_NumberOfLeaves 0.108448139  0.120283810  0.264443427  0.107328737
## ChlorA            0.027259054  0.003926126  0.063912223  0.076734885
## ChlorB            0.124575163  0.022541618  0.033330928  0.109209684
## gluc_Conc         0.047898151 -0.027831109  0.094139784  0.038091765
## flav_Conc         0.008971602 -0.036904459  0.014663540  0.002292280
## RGR1              0.302937248 -0.021768740 -0.091744342 -0.027049843
## RGR2              1.000000000  0.102474019 -0.008877318  0.041433082
## RGR3              0.102474019  1.000000000 -0.364183542  0.007434289
## RGR4             -0.008877318 -0.364183542  1.000000000  0.050741678
## mortality         0.041433082  0.007434289  0.050741678  1.000000000
```

PCA

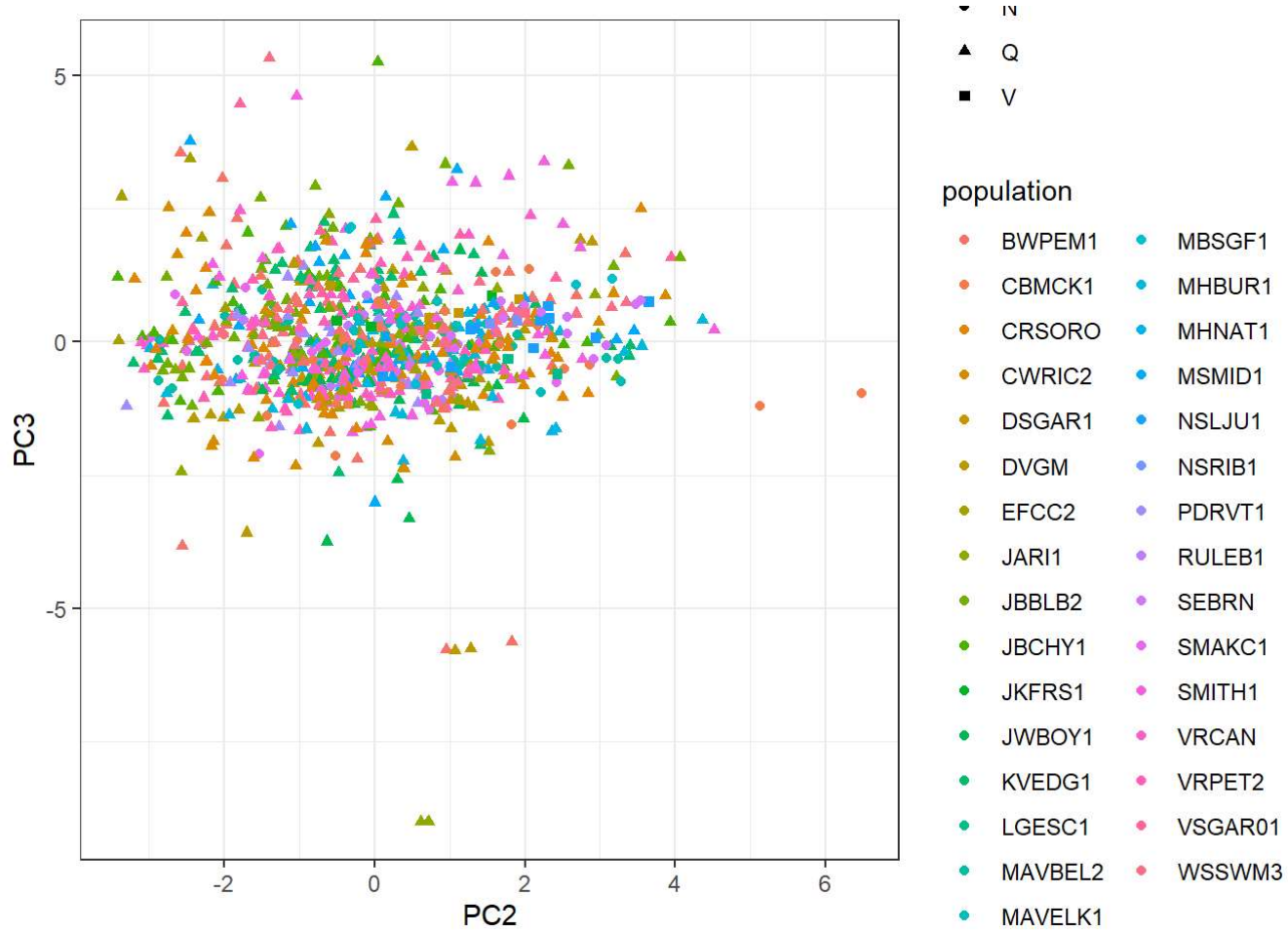
```
#Load Libraries
library(dplyr)
library(ggplot2)

#Load data
pcaData<-read.csv("normalized_data.csv", header=T)

#select out the un-needed columns, leave columns with only data
pcaData_2<-pcaData %>%
  select(-c("Tag", "petri_dish", "population", "family", "ID",
            "gh_bench", "gh_col", "gh_row", "Row_Field", "Col_Field", "common_garden"))

#run PCA
plantPCA<-princomp(na.omit(pcaData_2),cor=F)

#checking the PCA output and its structure
#str(plantPCA)
#head(plantPCA)
#names(plantPCA)

#put PCA loadings into a data frame
PCloadings<-data.frame(Component=c(1:13),Eigenvalue=plantPCA$sdev^2)

#create a scree plot - elbow is around 5.0 principal components
ggplot(aes(x=Component,y=Eigenvalue),data=PCloadings) +geom_point() + geom_line()
```



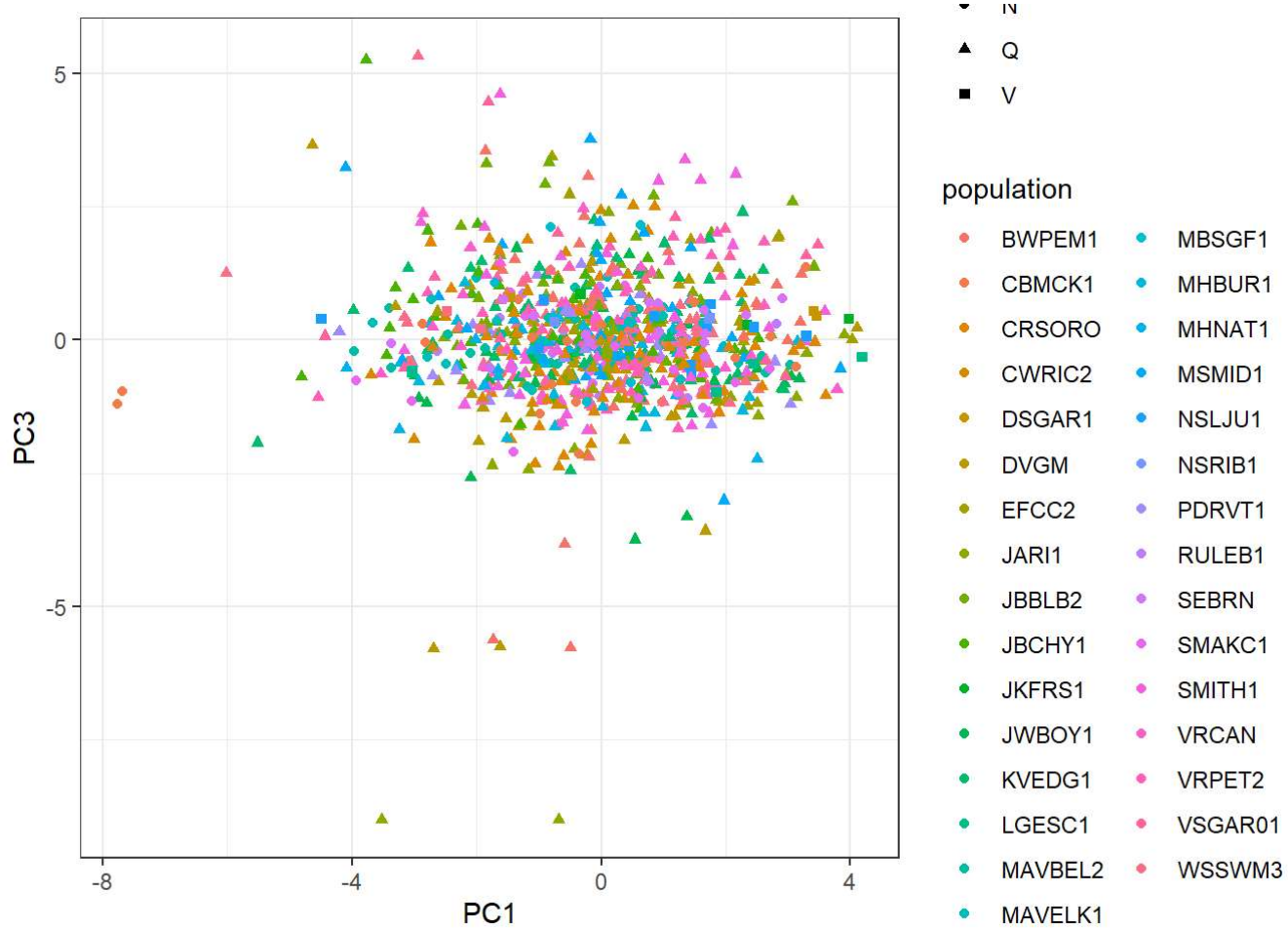
```
#what are the loadings of each variable on the components
#plantPCA$loadings

#combining the PCA with the original plant data for graphing
plantCombined<-cbind(na.omit(pcaData), plantPCA$scores)

plantCombined<-plantCombined %>%
  rename(PC1 = Comp.1, PC2 = Comp.2, PC3 = Comp.3, PC4= Comp.4, PC5 = Comp.5)

#make sure columns are the right data classes
#str(plantCombined)

plantCombined$population<-as.factor(plantCombined$population)

#creating the bivariate plots for PC1 vs PC2
ggplot(aes(x=PC1,y=PC2,colour=population, shape=common_garden),data=plantCombined)+
  geom_point() + theme_bw()
```



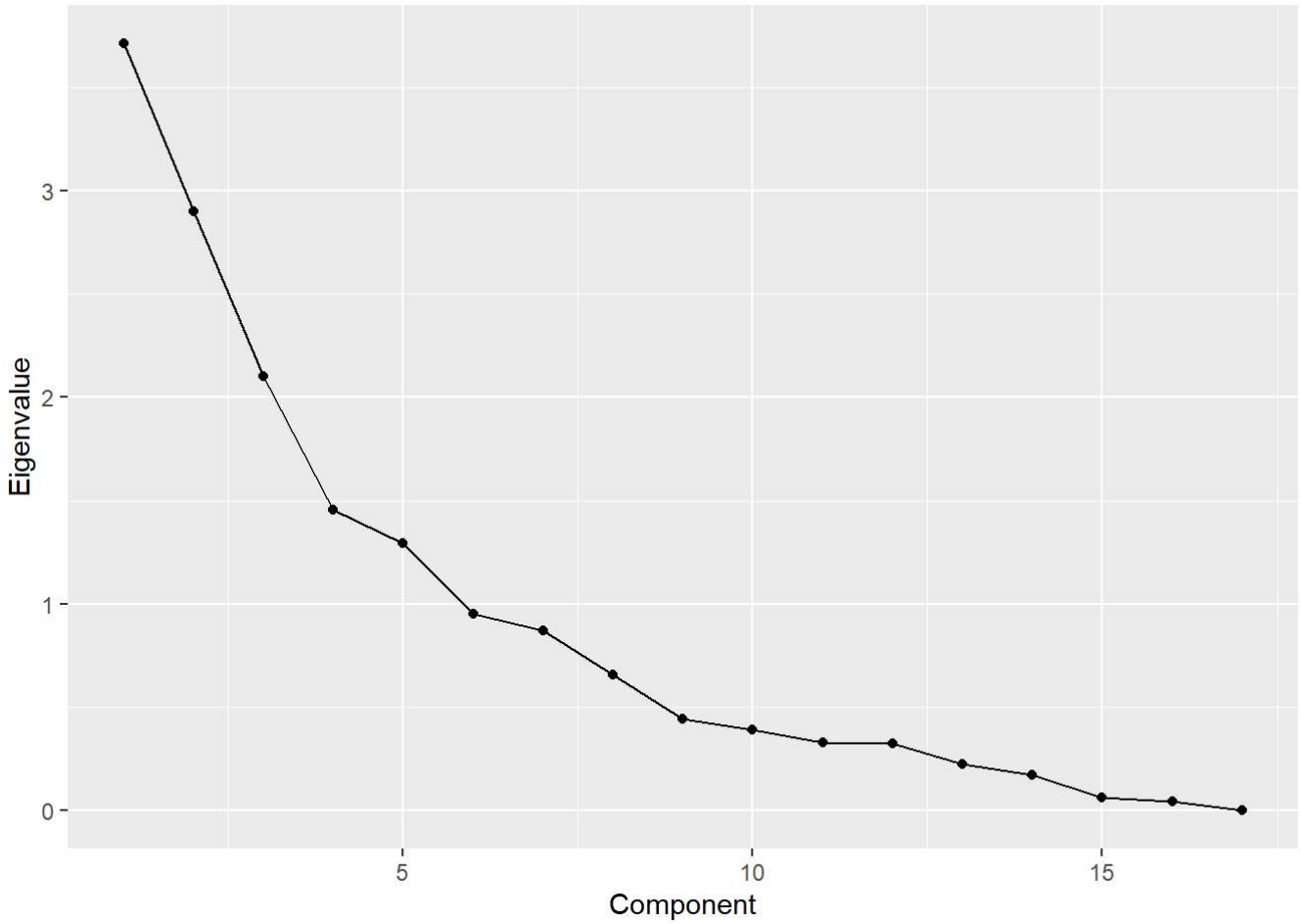

```
ggplot(aes(x=PC2,y=PC3,colour=population, shape=common_garden),data=plantCombined)+
  geom_point() + theme_bw()
```



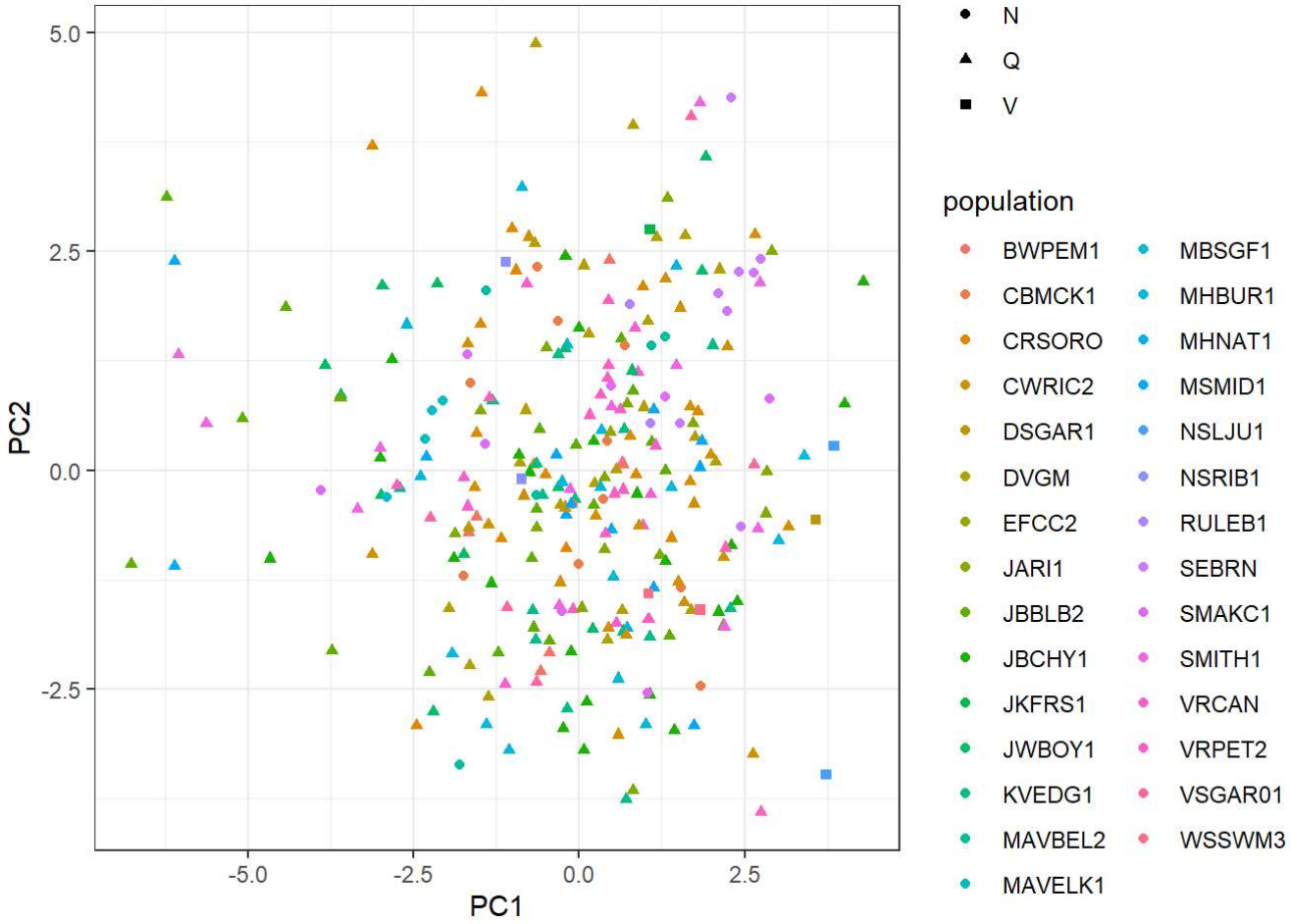

```
ggplot(aes(x=PC1,y=PC3,colour=population, shape=common_garden),data=plantCombined)+
  geom_point() + theme_bw()
```



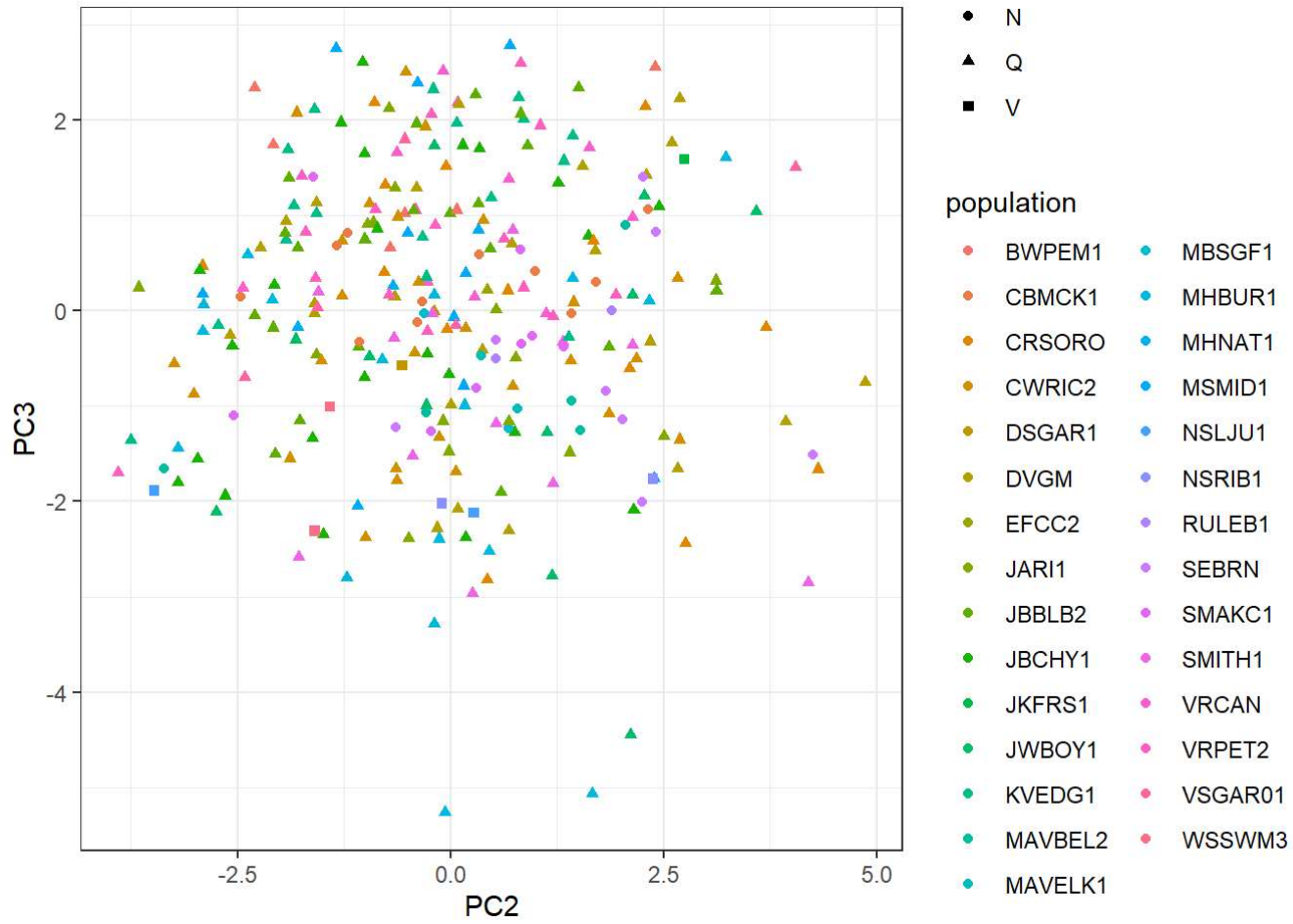

```
#checking if the data exhibits the same trend when including growth data when plants are bolts

#Load data
boltData<-read.csv("normalized_bolt_data.csv", header=T)

#select out un-needed columns, keep only numerical variables
pcaData2<-boltData %>%
  select(-c("Tag", "petri_dish", "population", "family", "ID",
            "gh_bench", "gh_col", "gh_row", "Row_Field", "Col_Field",
            "common_garden"))

#make PCA
boltPCA<-princomp(pcaData2,cor=F)

#checking the PCA output and its structure
#str(boltPCA)
#head(boltPCA)
#names(boltPCA)

#put PCA loadings into a data frame
PCloadings2<-data.frame(Component=c(1:17),Eigenvalue=boltPCA$sdev^2)

#make a scree plot - elbow is around 5.0 principal components
ggplot(aes(x=Component,y=Eigenvalue),data=PCloadings2) +geom_point() + geom_line()
```



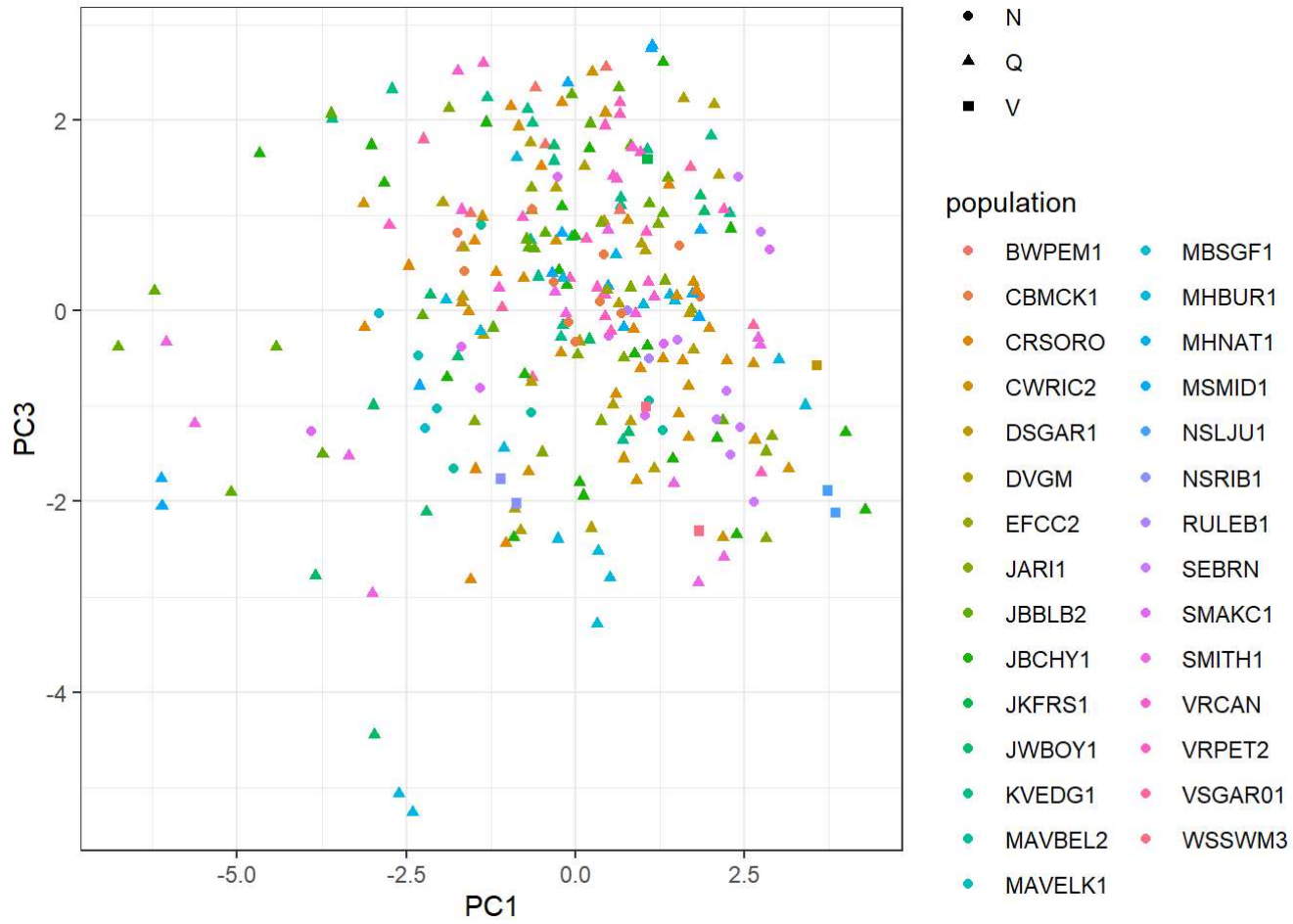

```
#what are the loadings of each variable on the components
#boltPCA$loadings

#combining the PCA with the original plant data for graphing
boltCombined<-cbind(boltData, boltPCA$scores)

boltCombined<-boltCombined %>%
  rename(PC1 = Comp.1, PC2 = Comp.2, PC3 = Comp.3, PC4= Comp.4, PC5 = Comp.5,
         PC6=Comp.6)

#str(boltCombined)

boltCombined$population<-as.factor(boltCombined$population)

#creating the bivariate plots for PC1 vs PC2, 2 vs 3, 1 vs 3
ggplot(aes(x=PC1,y=PC2,colour=population, shape=common_garden),data=boltCombined)+
  geom_point()+ theme_bw()
```




```
ggplot(aes(x=PC2,y=PC3,colour=population, shape=common_garden),data=boltCombined)+
  geom_point()+ theme_bw()
```



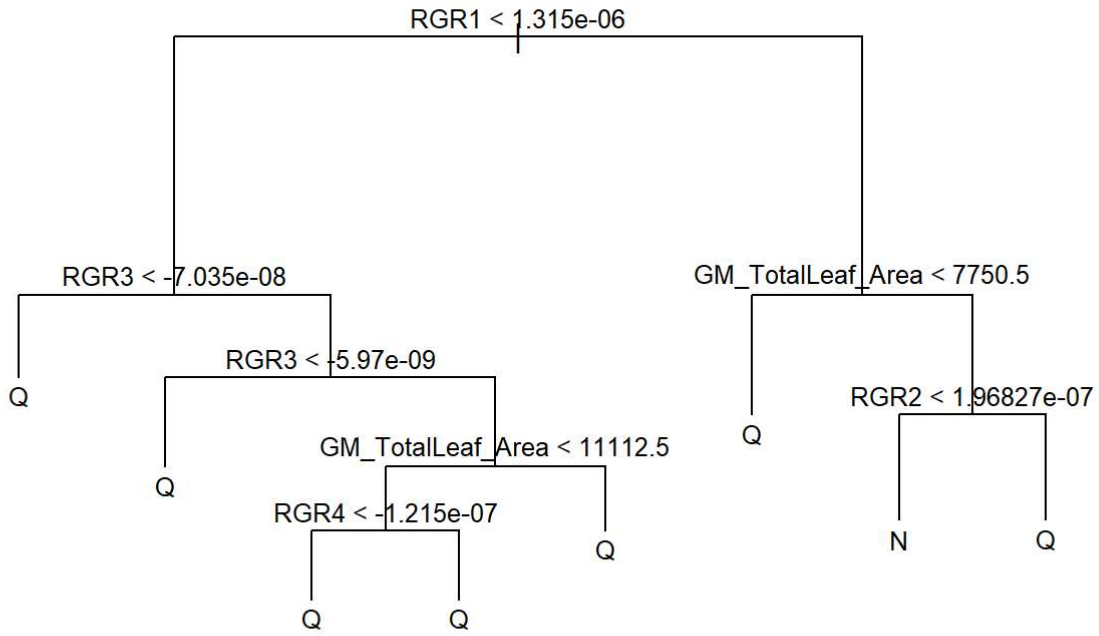

```
ggplot(aes(x=PC1,y=PC3,colour=population, shape=common_garden),data=boltCombined)+
  geom_point()+ theme_bw()
```



Decision tree

```
#Load libraries
library(dplyr)
library(tree)

plantData<-read.csv("raw_data.csv", header=T)

## Not Down Sampled Decision Tree without Bolt Data

#Select out un-needed columns, use only columns with numerical data
plantDecide_raw<-plantData %>%
  select(-c("population", "ID", "gh_bench", "gh_col", "gh_row",
            "Row_Field", "Col_Field", "petri_dish", "family", "Tag"))

#keeps only rows where common_garden is Q, N, or V
plantDecideQ<-subset(plantDecide_raw, common_garden == "Q")
plantDecideN<-subset(plantDecide_raw, common_garden == "N")
plantDecideV<-subset(plantDecide_raw, common_garden == "V")

plantDecide_raw<- rbind(plantDecideQ, plantDecideN, plantDecideV)

#select out bolt data
plantDecide<-plantDecide_raw %>%
  select(-c("Larg_Leaf_Len_Bolt", "Larg_Leaf_Wid_Bolt", "GM_StemHeight_Bolt", "GM_Leaf_Number_Bo
lt"))

#make sure columns ARE THE RIGHT DATA CLASSES
plantDecide$common_garden<-as.factor(plantDecide$common_garden)

#select out unneeded columns
plantTemp<- plantDecide %>%
  select(-c("common_garden"))

#create a matrix
plant_matrix<-as.matrix(plantTemp)

#checking covariance
#cov(plantTemp)

#separating into testing and training datasets
train<-c(1:nrow(plantDecide)) %% 2
plantTrain<-plantDecide[train == 1,]

test<-1-train
plantTest<-plantDecide[test == 1,]

#make a tree
plantTree<- tree(common_garden ~ ., data=plantTrain)

#plot the tree with text - these 2 following lines need to be run together
#plot(plantTree)
#text(plantTree, cex =0.8)
```

```
#prune nodes
pruneTree<-cv.tree(plantTree, best=8, FUN=prune.tree)
plot(pruneTree) #run this and the next line together
text(pruneTree, cex=0.8)
```




```
#setting up confusion matrix
plantConfuse<-data.frame(Obs=plantTest$common_garden,Pred=predict(pruneTree, plantTest, type="cl
ass"))
table(plantConfuse)
```

```
##    Pred
## Obs   N   Q   V
##   N   7  70   4
##   Q   7 376   3
##   V   5   4   4
```

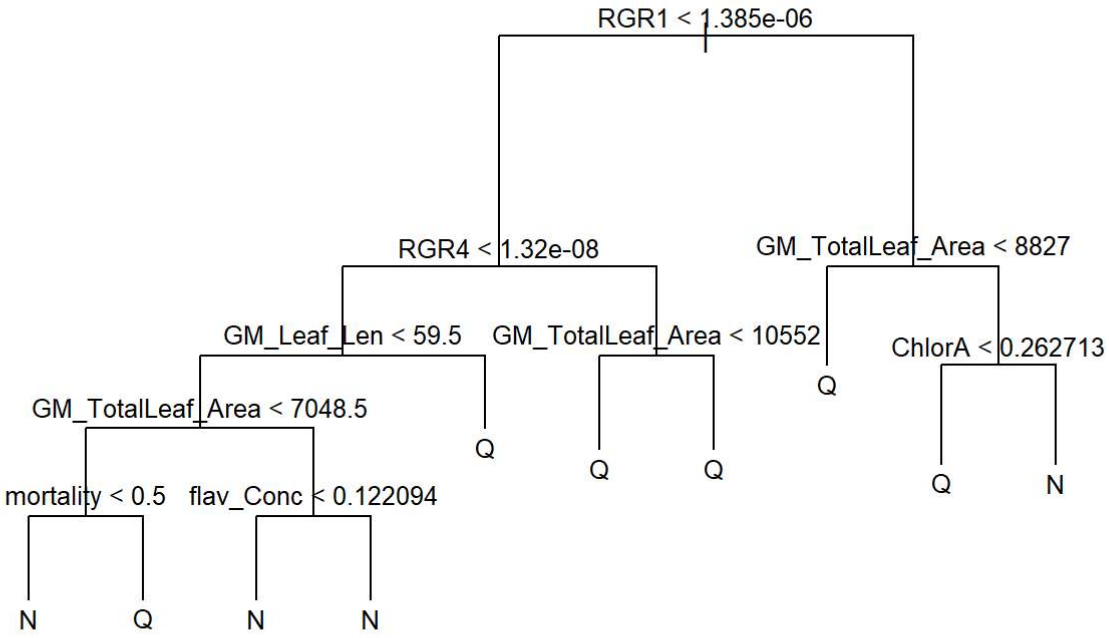
```
#calculating the misclassification rate
MisClass<-plantConfuse %>%
  filter(Obs!=Pred)
nrow(MisClass)/nrow(plantConfuse)
```

```
## [1] 0.19375
```

```
#checking that misclass and correct class = 1
correct<-plantConfuse %>%
  filter(Obs==Pred)
nrow(correct)/nrow(plantConfuse)
```

```
## [1] 0.80625
```

```
#sum does in fact equal 1

## Making a Decision tree with down sampling to account for fewer Vancouver samples
set.seed(100)
plantDecideQ2<-plantDecideQ[sample(nrow(plantDecideQ), 163),]

plantDecide2<-rbind(plantDecideQ2, plantDecideN, plantDecideV)

plantDecide2<-plantDecide2 %>%
  select(-c("Larg_Leaf_Len_Bolt", "Larg_Leaf_Wid_Bolt", "GM_StemHeight_Bolt", "GM_Leaf_Number_Bo
lt"))

plantDecide2$common_garden<-as.factor(plantDecide2$common_garden)

train2<-c(1:nrow(plantDecide2)) %% 2
plantTrain2<-plantDecide2[train2 == 1,]

test2<-1-train2
plantTest2<-plantDecide2[test2 == 1,]

plantTree2<-tree(common_garden ~ ., data=plantTrain2)

#plot the tree with text - run these 2 lines of code together
#plot(plantTree2)
#text(plantTree2, cex =0.8)

#prune nodes
pruneTree2<-cv.tree(plantTree2, best=8, FUN=prune.tree)
plot(pruneTree2) #run this and the next line of code together
text(pruneTree2, cex=0.8)
```




```
#setting up confusion matrix
plantConfuse2<-data.frame(Obs=plantTest2$common_garden,Pred=predict(pruneTree2, plantTest2, type
="class"))
table(plantConfuse2)
```

```
##    Pred
## Obs  N  Q  V
##   N 47 33  2
##   Q 26 55  0
##   V  8  4  0
```

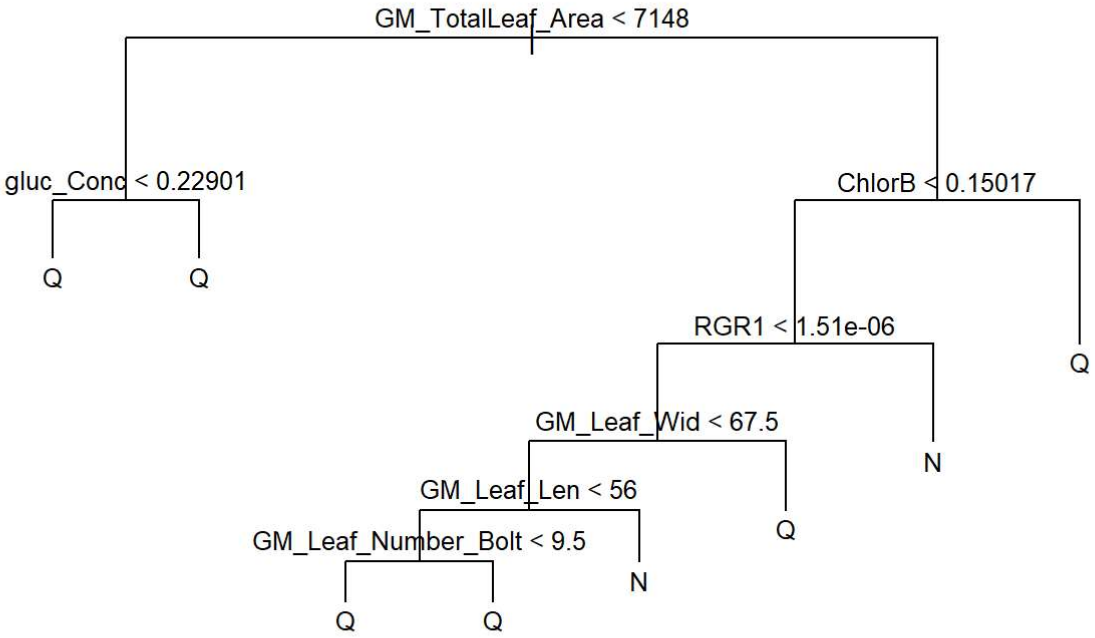
```
#calculating the misclassification rate
MisClass2<-plantConfuse2 %>%
  filter(Obs!=Pred)
nrow(MisClass2)/nrow(plantConfuse2)
```

```
## [1] 0.4171429
```

```
#checking that misclass and correct class = 1
correct2<-plantConfuse2 %>%
  filter(Obs==Pred)
nrow(correct2)/nrow(plantConfuse2)
```

```
## [1] 0.5828571
```

```
## Making a Decision Tree with only plants that survived (mortality = 1) without down sampling
plantDecide3<-subset(plantDecide_raw, mortality == 1)

#ensures common garden is only Q, N or V
plantDecideQ_3<-subset(plantDecide3, common_garden == "Q")
plantDecideN_3<-subset(plantDecide3, common_garden == "N")
plantDecideV_3<-subset(plantDecide3, common_garden == "V")

plantDecide3<-rbind(plantDecideQ_3, plantDecideN_3, plantDecideV_3)

#make sure columns are the right characters classes
plantDecide3$common_garden<-as.factor(plantDecide3$common_garden)

#select out un needed columns
plantTemp2<- plantDecide3 %>%
  select(-c("common_garden", "mortality"))

#make sure columns are the right data classes
plantTemp2$Larg_Leaf_Len_Bolt <- as.numeric(plantTemp2$Larg_Leaf_Len_Bolt)
plantTemp2$Larg_Leaf_Wid_Bolt <- as.numeric(plantTemp2$Larg_Leaf_Wid_Bolt)
plantTemp2$GM_StemHeight_Bolt <- as.numeric(plantTemp2$GM_StemHeight_Bolt)
plantTemp2$GM_Leaf_Number_Bolt <- as.numeric(plantTemp2$GM_Leaf_Number_Bolt)

#str(plantTemp2)

#make a matrix
plant_matrix2<-as.matrix(plantTemp2)

plantDecide3<- plantDecide3 %>%
  select(-mortality)

#split training and testing data
train3<-c(1:nrow(plantDecide3)) %% 2
plantTrain3<-plantDecide3[train3 == 1,]

test3<-1-train3
plantTest3<-plantDecide3[test3 == 1,]

plantTree3<- tree(common_garden ~ ., data=plantTrain3)

#plot the tree with text
#plot(plantTree3)
#text(plantTree3, cex =0.8)

#prune nodes
pruneTree3<-cv.tree(plantTree3, best=8, FUN=prune.tree)
plot(pruneTree3)
text(pruneTree3, cex=0.8)
```




```
#setting up confusion matrix
plantConfuse3<-data.frame(Obs=plantTest3$common_garden,Pred=predict(pruneTree3, plantTest3, type
="class"))
table(plantConfuse3)
```

```
##    Pred
## Obs   N   Q   V
##   N   8  11   0
##   Q  11 105   0
##   V   0   4   0
```

```
#calculating the misclassification rate
MisClass3<-plantConfuse3 %>%
  filter(Obs!=Pred)
nrow(MisClass3)/nrow(plantConfuse3)
```

```
## [1] 0.1870504
```

```
#checking that misclass and correct class = 1
correct3<-plantConfuse3 %>%
  filter(Obs==Pred)
nrow(correct3)/nrow(plantConfuse3)
```

```
## [1] 0.8129496
```

```
## Making a Decision tree with down sampling to account for mortality == 1
set.seed(100)
plantDecideQ_4<-plantDecideQ_3[sample(nrow(plantDecideQ_3), 38),]

plantDecide4<-rbind(plantDecideQ_4, plantDecideN_3, plantDecideV_3)

plantDecide4$common_garden<-as.factor(plantDecide4$common_garden)

plantDecide4<- plantDecide4 %>%
  select(-mortality)

train4<-c(1:nrow(plantDecide4)) %% 2
plantTrain4<-plantDecide4[train4 == 1,]

test4<-1-train4
plantTest4<-plantDecide4[test4 == 1,]

plantTree4<- tree(common_garden ~ ., data=plantTrain4)

#plot the tree with text
#plot(plantTree4)
#text(plantTree4, cex =0.8)

#setting up confusion matrix
plantConfuse4<-data.frame(Obs=plantTest4$common_garden,Pred=predict(plantTree4, plantTest4, type
="class"))
table(plantConfuse4)
```

```
##    Pred
## Obs  N  Q  V
##   N  6 13  0
##   Q  7 12  0
##   V  0  4  0
```

```
#calculating the misclassification rate
MisClass4<-plantConfuse4 %>%
  filter(Obs!=Pred)
nrow(MisClass4)/nrow(plantConfuse4)
```

```
## [1] 0.5714286
```

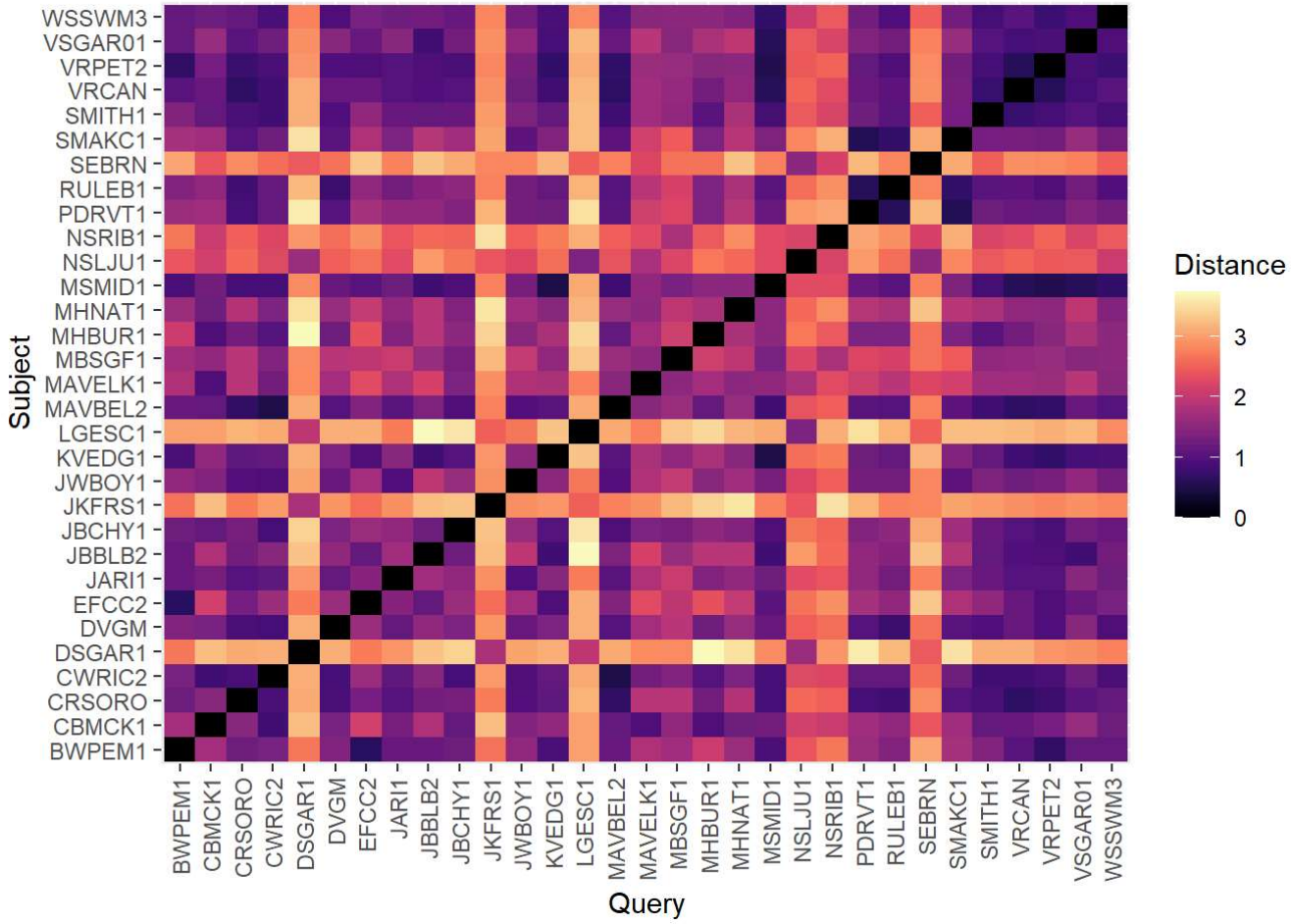
```
#checking that misclass and correct class = 1
correct4<-plantConfuse4 %>%
  filter(Obs==Pred)
nrow(correct4)/nrow(plantConfuse4)
```

```
## [1] 0.4285714
```

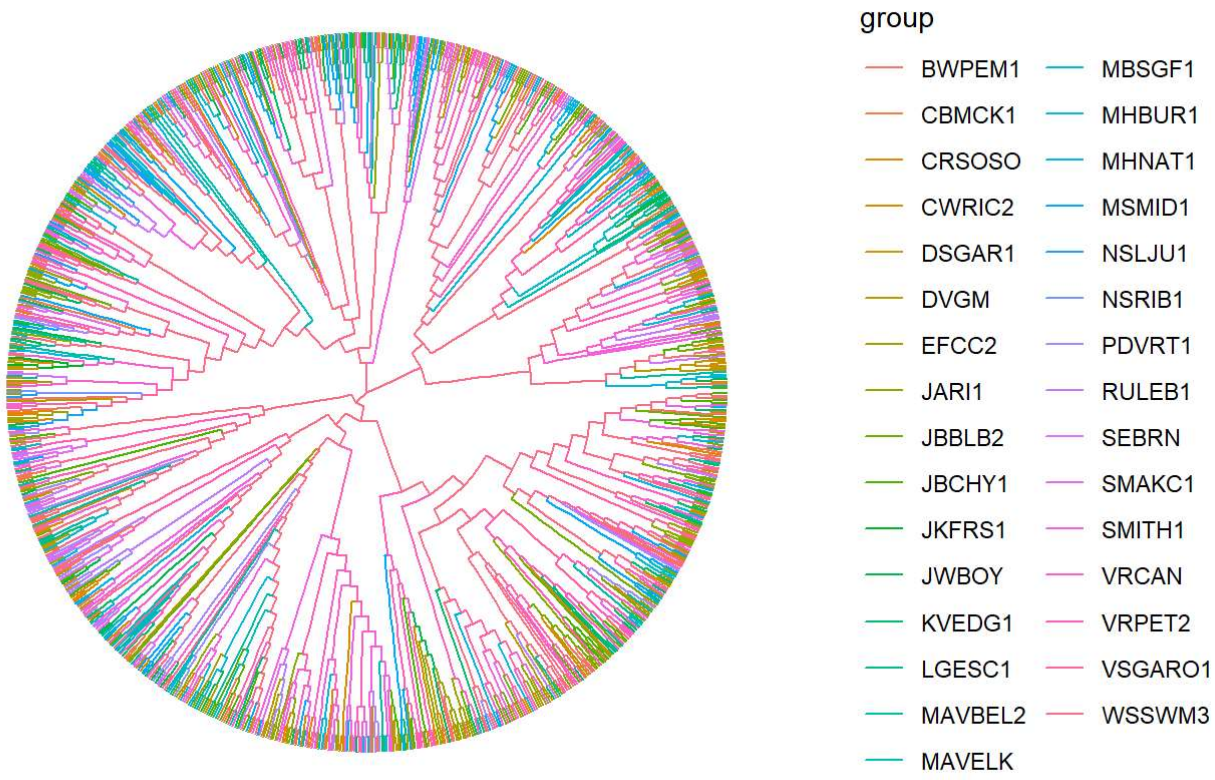
Cluster Tree

```
#Create cluster tree to asses whether plants
#group by common garden or population

#load libraries
library(ggplot2)
library(ape)
library(reshape2)
library(viridis)
library(dplyr)
library(ggtree)

#load data
data<-read.csv("./normalized_data.csv", row.names=NULL)

#give data row names
uniq_name<-make.names(data$Tag, unique=T)
row.names(data)<-uniq_name

#Taking the mean values for each population for each measurement so we can discriminate at the p
opulation level
p<-data%>%
  group_by(population)%>%
  summarize(z_RGR1=mean(z_RGR1),
            z_RGR2=mean(z_RGR2),
            z_RGR3=mean(z_RGR3),
            z_RGR4=mean(z_RGR4),
            z_ChlorA=mean(z_ChlorA),
            z_ChlorB=mean(z_ChlorB),
            z_gluc_Conc=mean(z_gluc_Conc),
            z_flav_Conc=mean(z_flav_Conc),
            z_Leaf_Len=mean(z_Leaf_Len),
            z_Leaf_Wid=mean(z_Leaf_Wid),
            z_TotalLeaf_Area=mean(z_TotalLeaf_Area),
            z_NumberOfLeaves=mean(z_NumberOfLeaves))

#select out population
p2<-p %>%
  select(-c("population"))

#make p2 a data frame
p2<-data.frame(p2)

#give p2 row names
row.names(p2)<-p$population

#make a dataframe with only the important columns
data2<-data%>%
  select(starts_with("z_"))
#head(data2)

#make a distance matrix - this is a linear matrix by sample
distance<-dist(data2, method = "euclidean")
```

```
a<-as.matrix(distance)

distMrx<-melt(a)
names(distMrx)<-c("Query","Subject","Distance")

#distance matrix by population

distp<-dist(p2, method = "euclidean")
a2<-as.matrix(distp)

distMrx2<-melt(a2)
names(distMrx2)<-c("Query","Subject","Distance")

#visualize distances for population-level traits

ggplot(data = distMrx2, aes(x=Query, y=Subject, fill=Distance)) +
  geom_tile() +theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))+
  scale_fill_viridis(option="magma")
```



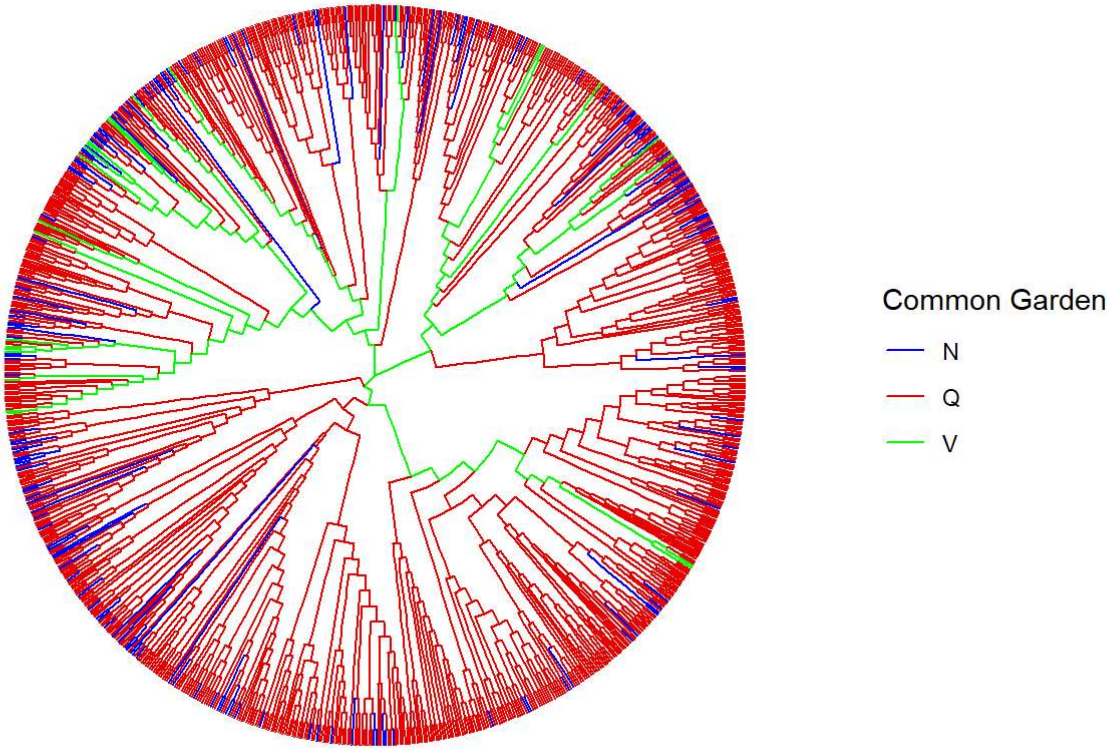
```
#tree building for sample-level, colour by population
tree1<-nj(distance)
#str(tree1)
#ggtree(tree1, layout="circular")

#extract population and common garden from the tags
pop<-gsub("^[a-z]\\.","",tree1$tip.label)
pop2<-sub("\\..*","",pop)
popGroups<-split(tree1$tip.label, pop2)
popCol<-groupOTU(tree1,popGroups)

ggtree(popCol,layout="circular",aes(colour=group), branch.length="none")
```



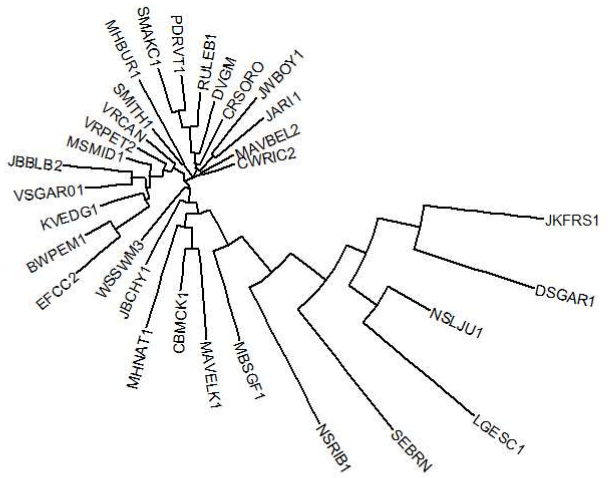

```
#ggtree(popCol,layout="rectangular",aes(colour=group))

#tree building for sample-level, colour by common garden
garden<-gsub(".*([QNV]).*","\\1",tree1$tip.label)
grep("b.WSSWM3.1.0....|e.JBCHY1.1.50..|e.JBCHY1.1.50...1", garden)
```

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

```
garden<-garden[-c(812,813,960)]

gardenGroups<-split(tree1$tip.label, garden)
gardenCol<-groupOTU(tree1,gardenGroups)

ggtree(gardenCol,layout="circular",aes(colour=as.factor(group)),branch.length = "none") +guides
(color = guide_legend(title = "Common Garden")) +
  scale_color_manual(values=c("blue", "red2", "green"))
```




```
#tree building for population-level trait
tree2<-nj(distp)
#str(tree2)
ggtree(tree2, layout="circular") + geom_tiplab(size=2,aes(angle=angle))
```


SMAKC1
PDRVT1
RULEB1
DVGM
CRSORO
WBOY1
JARI1
MAVBEL2
CWRIC2
MHBUR1
SMITH1
VRCAN
VRPET2
MSMID1
JBBLB2
VSGAR01
KVEDG1
BWPEM1
EFCC2
WSSWM3
JBCHY1
MHNAT1
CBMCK1
MAVELK1
MBSGF1
NSRIB1
SEBRN
LGESC1
NSLJU1
DSGAR1
JKFRS1