Lab 2: Multivariate data screening

Lara Katz

2024-01-26

# Set up R session

## Install and load packages.

Note that for a RMarkdown document to knit, you need to comment out these ‘install.packages’ lines of code before knitting. This is because you cannot knit when your document needs to connect to an external web server. Right now these lines of code are commented out.

# install.packages("mvnormtest")  
# install.packages("MVN")  
# install.packages("MVA")  
# install.packages("psych")  
# install.packages("Hmisc")  
# install.packages("vegan")  
# install.packages("StatMatch")  
# install.packages("MASS")  
# install.packages("raster")  
# install.packages("cluster")

library(mvnormtest)  
library(MVN)  
library(MVA)  
library(psych)  
library(Hmisc)  
library(vegan)  
library(StatMatch)  
library(MASS)  
library(raster)  
library(cluster)  
library(tidyverse)

## Importing Data

We will be using the USairpollution and the usAir\_mod.csv data sets for these examples.

The MVA US air pollution data set:

usAir <- USairpollution  
?USairpollution

## starting httpd help server ... done

# SO2: SO2 content of air in micrograms per cubic metre.  
#   
# temp: average annual temperature in Fahrenheit.  
#   
# manu: number of manufacturing enterprises employing 20 or more workers.  
#   
# popul: population size (1970 census); in thousands.  
#   
# wind# average annual wind speed in miles per hour.  
#   
# precip: average annual precipitation in inches.  
#   
# predays: average number of days with precipitation per year.

The modified USairpollution data set from your working directory is a csv file. Note you will need to modify you working directory if it is different.

usAir\_mod <- read.csv("./Data/usAir\_mod.csv", row=1, header=TRUE)

# Data screening

Your first move when conducting a multivariate analysis (or any analysis) is to screen the data. You are looking for data errors, missing data, and outliers that may influence your analysis.

## Data errors

One way to check for data errors is to examine the summary statistics for your data set.

First look at the summary statistics for usAir:

describeBy(usAir)

***Question 1: Do you see any unrealistic values? (5 pts) Note please answer all questions with points related to them.***

*The populations of Charleston, SC, and Wilmington, DE, are both below 100 (100,000), which seems low for cities on the east coast. Additionally, Providence, RI, has a high SO2 content despite not having very many large manufacturing enterprises, but this could be a result of pollution from nearby areas or prevailing wind currents.*

Now look at the summary statistics for usAir\_mod:

describeBy(usAir\_mod)

Look at the max for temperature. It will be easier to look for data errors when it is your own data.

*The maximum for temperature is now 548.5 degrees Fahrenheit, which is not a realistic value.*

## Missing Data

When you have missing entries in your data sheet, R replaces them with “NA”. You can check if you have any missing variables in *usAir\_mod*:

describe(usAir\_mod)

The *describe* function provides some of the same information as *describeBy*, but importantly shows you which variables have missing values.

We talked about two methods for dealing with missing values in lecture; **Complete Case and Imputation**. We will look at **complete case and imputation** for now.

**Complete Case** involves the removal of samples (in this case cities) with missing data:

usAir\_mod [complete.cases(usAir\_mod),]

**Imputation** involves filling in missing values with plausible data. Let’s replace NAs with the mean of the variable.

#First, let’s calculate the mean of each variable (column) with the NA removed:  
  
meanz<-colMeans(usAir\_mod,na.rm=T)  
  
#`na.rm=T`, means that you want to remove NAs  
  
#To replace your NAs with the means you just calculated you will use the following function:  
  
naFunc<-function(column) {   
 column[is.na(column)] = round(mean(column, na.rm = TRUE),2)  
 return(column)   
}  
  
#and “apply” it to the usair\_mod data set  
  
Impute <- apply(usAir\_mod,2,naFunc)

Check out the new Impute data object and make sure that the NA’s have been replaced.

describe(Impute)

## Impute   
##   
## 7 Variables 41 Observations  
## --------------------------------------------------------------------------------  
## SO2   
## n missing distinct Info Mean Gmd .05 .10   
## 41 0 27 0.998 28.05 20.68 9 10   
## .25 .50 .75 .90 .95   
## 13 26 31 56 65   
##   
## lowest : 8 9 10 11 12, highest: 56 61 65 69 94  
## --------------------------------------------------------------------------------  
## temp   
## n missing distinct Info Mean Gmd .05 .10   
## 41 0 39 1 67.81 31.76 47.1 49.0   
## .25 .50 .75 .90 .95   
## 50.6 55.0 59.4 68.4 70.3   
##   
## lowest : 43.5 45.7 47.1 47.6 49 , highest: 68.4 68.9 70.3 75.5 548.5  
## --------------------------------------------------------------------------------  
## manu   
## n missing distinct Info Mean Gmd .05 .10   
## 41 0 41 1 458.7 455.5 46.0 91.0   
## .25 .50 .75 .90 .95   
## 181.0 347.0 458.6 775.0 1064.0   
##   
## lowest : 35 44 46 80 91, highest: 775 1007 1064 1692 3344  
## --------------------------------------------------------------------------------  
## popul   
## n missing distinct Info Mean Gmd .05 .10   
## 41 0 41 1 608.6 508.6 116 158   
## .25 .50 .75 .90 .95   
## 299 515 717 905 1513   
##   
## lowest : 71 80 116 132 158, highest: 905 1233 1513 1950 3369  
## --------------------------------------------------------------------------------  
## wind   
## n missing distinct Info Mean Gmd .05 .10   
## 41 0 29 0.998 9.464 1.581 7.1 7.9   
## .25 .50 .75 .90 .95   
## 8.8 9.4 10.6 10.9 11.8   
##   
## lowest : 6 6.5 7.1 7.6 7.9 , highest: 10.9 11.2 11.8 12.4 12.7  
## --------------------------------------------------------------------------------  
## precip   
## n missing distinct Info Mean Gmd .05 .10   
## 41 0 41 1 36.72 12.85 12.95 20.66   
## .25 .50 .75 .90 .95   
## 30.96 37.01 43.11 48.52 54.47   
##   
## lowest : 7.05 7.77 12.95 15.17 20.66, highest: 48.52 49.1 54.47 56.77 59.8   
## --------------------------------------------------------------------------------  
## predays   
## n missing distinct Info Mean Gmd .05 .10   
## 41 0 34 0.999 113.9 29.14 67 82   
## .25 .50 .75 .90 .95   
## 103 115 128 147 155   
##   
## lowest : 36 58 67 78 82, highest: 147 148 155 164 166  
## --------------------------------------------------------------------------------

We will not go into this advanced function too much. However, know that *apply* allows us to perform a function on all the rows and/or columns in a data frame of matrix. As we spoke about in lecture, there are many types of imputation methods. We can explore further methods for your specific missing data.

# Multivariate Normal Distribution

Many of the analyses we will do in this course have an assumption of multivariate normality. While there are many tests of multivariate normality, they tend to be overly conservative. If we strictly followed these tests, we may never run a multivariate analysis with ecological or agricultural data. Here we will look at two multivariate tests of normality.

## Shapiro-Wilks test

Shapiro-Wilks tests if the distribution of the observed data differs from multivariate normal distribution. So, we are looking for p-values > 0.05.

mshapiro.test(t(usAir))

##   
## Shapiro-Wilk normality test  
##   
## data: Z  
## W = 0.59549, p-value = 2.025e-09

## Mardia test

Mardia’s test looks at multivariate extensions of Skewness and Kurtosis. In both cases, we are looking for p-values > 0.05 to show that our data do not deviate from the expectations of multivariate normal Skewness and Kurtosis. For the observed data to be considered multivariate normal, p-values from both the Skewness and Kurtosis statistics must be > 0.05. This function also tests for univariate normality of residuals using the Shapiro-Wilk statistic.

mvn(usAir, mvnTest = "mardia")

## $multivariateNormality  
## Test Statistic p value Result  
## 1 Mardia Skewness 226.612731693166 4.82491336101954e-15 NO  
## 2 Mardia Kurtosis 3.97754689564216 6.96298933924311e-05 NO  
## 3 MVN <NA> <NA> NO  
##   
## $univariateNormality  
## Test Variable Statistic p value Normality  
## 1 Anderson-Darling SO2 2.3841 <0.001 NO   
## 2 Anderson-Darling temp 0.9633 0.0136 NO   
## 3 Anderson-Darling manu 4.2925 <0.001 NO   
## 4 Anderson-Darling popul 3.4292 <0.001 NO   
## 5 Anderson-Darling wind 0.3784 0.3911 YES   
## 6 Anderson-Darling precip 0.8742 0.0228 NO   
## 7 Anderson-Darling predays 0.5175 0.1783 YES   
##   
## $Descriptives  
## n Mean Std.Dev Median Min Max 25th 75th Skew  
## SO2 41 30.048780 23.472272 26.00 8.00 110.0 13.00 35.00 1.584112608  
## temp 41 55.763415 7.227716 54.60 43.50 75.5 50.60 59.30 0.822975684  
## manu 41 463.097561 563.473948 347.00 35.00 3344.0 181.00 462.00 3.484603302  
## popul 41 608.609756 579.113023 515.00 71.00 3369.0 299.00 717.00 2.941257977  
## wind 41 9.443902 1.428644 9.30 6.00 12.7 8.70 10.60 0.002675131  
## precip 41 36.769024 11.771550 38.74 7.05 59.8 30.96 43.11 -0.692518149  
## predays 41 113.902439 26.506419 115.00 36.00 166.0 103.00 128.00 -0.550092270  
## Kurtosis  
## SO2 2.25541093  
## temp 0.09066032  
## manu 14.33200058  
## popul 10.57605759  
## wind 0.06015407  
## precip 0.49578021  
## predays 0.72033969

# Data transformation

The next step is preparing your data for analysis is transforming the data. Today we will look at the log, square root, and arcsine square root transformations.

## Log transformation:

Several common transformations have built-in functions in R. While you can build transformation functions on your own, we will use the ones R has developed today. First, let’s look at a histogram of our first variable, SO2, to determine if transformation is necessary:

Remember, to extract the SO2 column:

usAir$SO2   
  
#or   
  
usAir[,1]   
  
  
#Next you can simply wrap either of those commands in the histogram function:  
  
hist(usAir$SO2)   
  
#or   
  
hist(usAir[,1])

To log transform each value in our data frame:

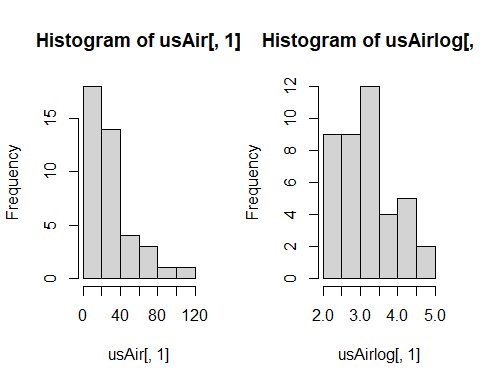
usAirlog<-log1p(usAir)  
?log1p

and the histogram:

hist(usAirlog$SO2)   
  
#or   
  
hist(usAirlog[,1])

You can compare the histograms side by side using the par function followed by hist:

par(mfrow=c(1,2))  
  
hist(usAir[,1])   
hist(usAirlog[,1])

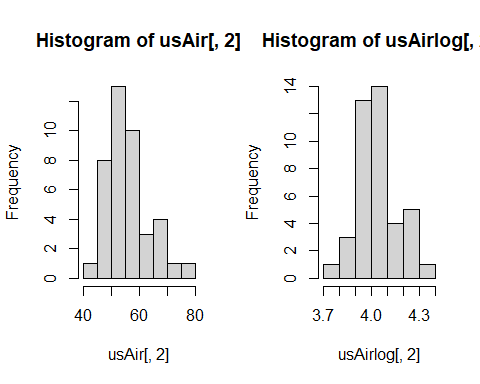


Placing 1, 2 in parentheses after the c (which stands for concatenate) in the par function indicates that you want your plots arranged in 1 row and two columns. Note this plotting is done in base R as opposed to using the ggplot functions of Tidyverse. It is helpful to know base R and Tidyverse to be able to read and trouble shoot code with a wide range of collaborators. In ggplot this code would be similar to what the *facet* function does.

Compare histograms for the raw data and the log transformed data for each variable.

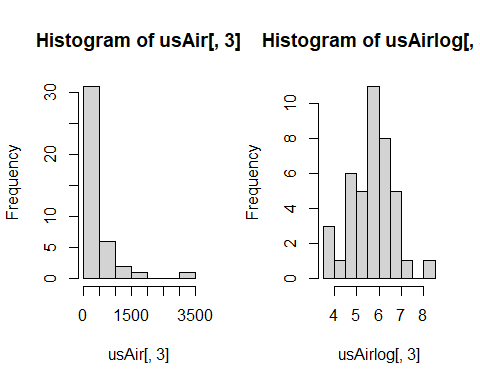
temp:

par(mfrow=c(1,2))  
  
hist(usAir[,2])   
hist(usAirlog[,2])



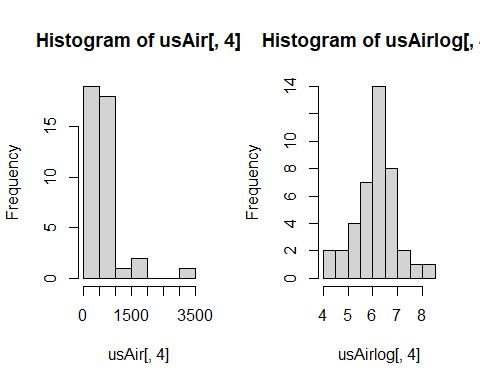
manu:

par(mfrow=c(1,2))  
  
hist(usAir[,3])   
hist(usAirlog[,3])



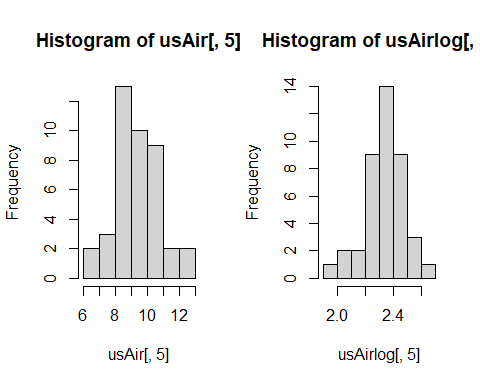
popul:

par(mfrow=c(1,2))  
  
hist(usAir[,4])   
hist(usAirlog[,4])



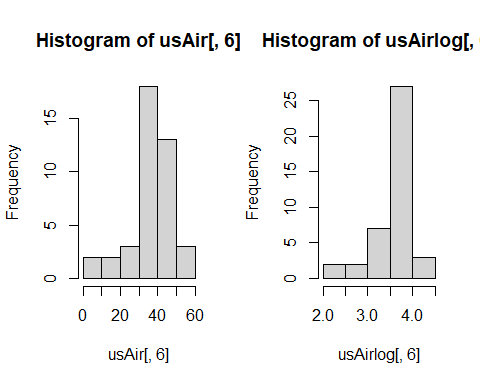
wind:

par(mfrow=c(1,2))  
  
hist(usAir[,5])   
hist(usAirlog[,5])



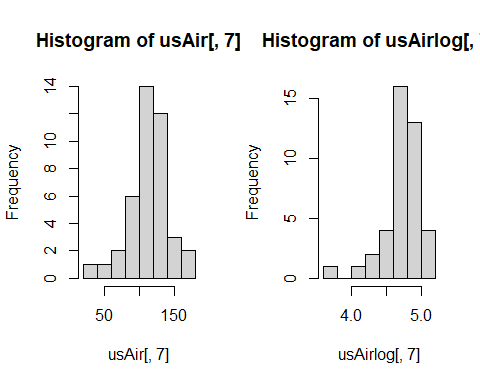
precip:

par(mfrow=c(1,2))  
  
hist(usAir[,6])   
hist(usAirlog[,6])



predays:

par(mfrow=c(1,2))  
  
hist(usAir[,7])   
hist(usAirlog[,7])



**Question 2: Which variable might not need to be log transformed? (5 pts)**

*The wind and predays variables are both normally distributed and may not need to be log-transformed.*

## Square root transformation:

To square root transform each value in our data frame:

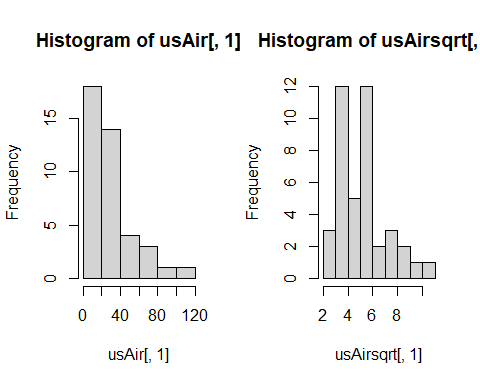
usAirsqrt<-sqrt(usAir)

and the histogram:

hist(usAirsqrt$SO2)  
  
#or   
  
hist(usAirsqrt[,1])

Compare the histograms side by side using the par function followed by hist:

par(mfrow=c(1,2))  
  
hist(usAir[,1])   
hist(usAirsqrt[,1])

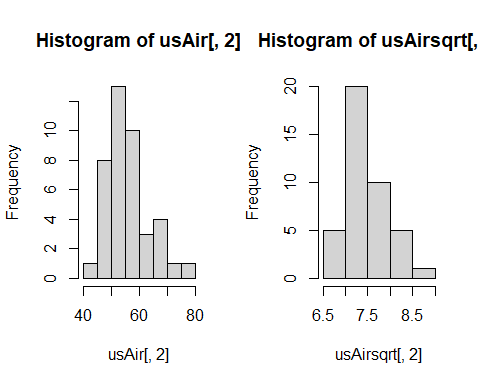


Compare histograms for the raw data and the square-root transformed data for each variable…

Remember that square root transformations are best used on count data.

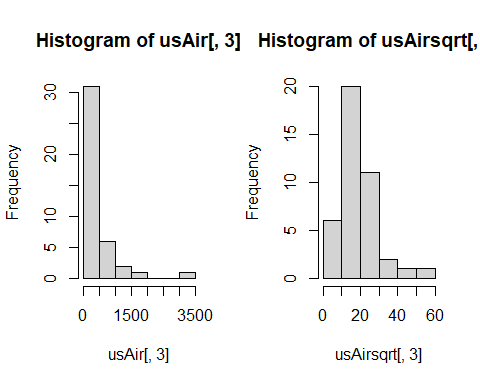
temp:

par(mfrow=c(1,2))  
  
hist(usAir[,2])   
hist(usAirsqrt[,2])



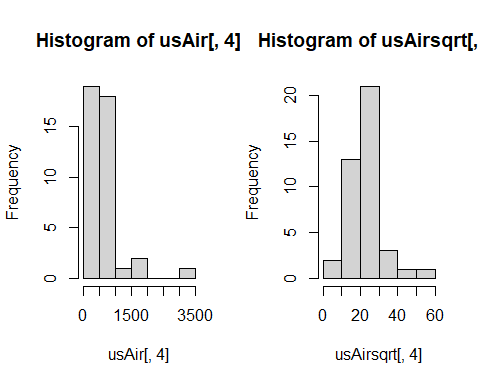
manu:

par(mfrow=c(1,2))  
  
hist(usAir[,3])   
hist(usAirsqrt[,3])



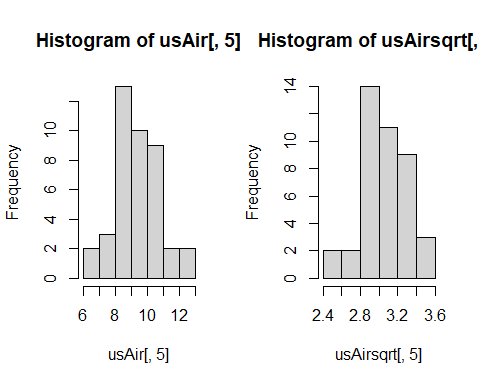
popul:

par(mfrow=c(1,2))  
  
hist(usAir[,4])   
hist(usAirsqrt[,4])



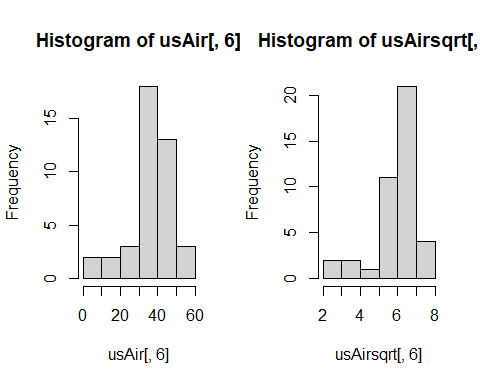
wind:

par(mfrow=c(1,2))  
  
hist(usAir[,5])   
hist(usAirsqrt[,5])



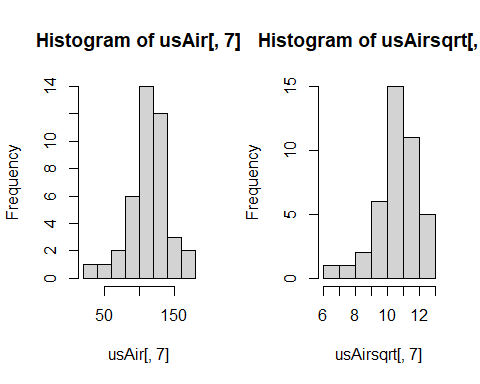
precip:

par(mfrow=c(1,2))  
  
hist(usAir[,6])   
hist(usAirsqrt[,6])



predays:

par(mfrow=c(1,2))  
  
hist(usAir[,7])   
hist(usAirsqrt[,7])



## Arcsine square root transformation: = arcsine

If you remember arcsine square root transformations are for percentage data. So, the values for your variable must range between 0 and 1. None of the variables in usAir are appropriate for this transformation. Let’s draw some random numbers between 0 and 1 so we can use the arcsine square root transformation.

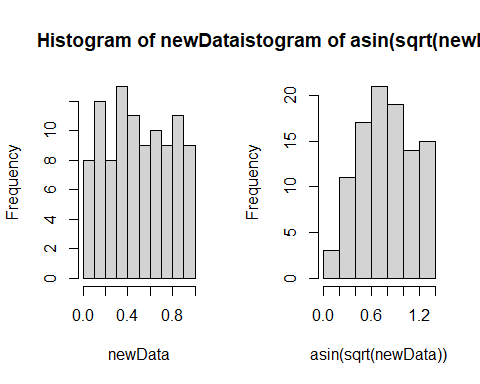
newData<- runif(100, 0, 1)

You just chose 100 random values between 0 and 1. Now let’s transform:

asin(sqrt(newData))

and compare histograms:

par(mfrow=c(1,2))  
  
hist(newData)  
hist(asin(sqrt(newData)))



# Data standardization

Column standardization adjusts for differences among variables. The focus is on the profile across a sample unit. Row standardization adjusts for differences among sample units, wherein the focus is on the profile within a sample unit. Row standardization is good when variables are measured in the same units (e.g. species). You will more often than not be using column standardization.

## Coefficient of Variation (cv)

Let’s first see if the air pollution data set needs standardization by calculating the *coefficient of variation* **(cv)** for column totals. Remember, the **cv** is the ratio of the standard deviation to the mean (σ/μ):

First calculate the column **sums**:

cSums<-colSums(usAir)

Then calculate the **standard deviation** and **mean** for the column sums:

Sdev <- sd(cSums)  
M <- mean(cSums)

Finally, calculate the **cv**:

Cv <- Sdev/M\*100

Our rule of thumb for cv is that if **cv> 50**, data standardization is necessary.

***Question 3: Is standardization necessary for the USairpollution data? (5 pts)***

*Data standardization is necessary because the CV of the dataset is 129.27.*

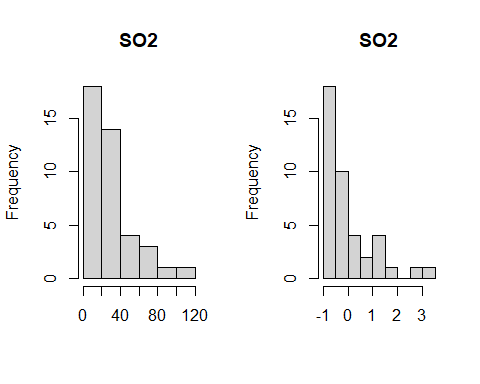
## Z- standardization = (

Your goal here is to equalize the variance for variables measured on different scales. There is a built-in function scale that will do this for you:

scaledData <- scale(usAir)

Let’s look at histograms for the scaled and unscaled data for the first variable, SO2:

par(mfrow=c(1,2))  
  
hist(usAir[,1] ,main=colnames(usAir)[1],xlab=" ")  
hist(scaledData[,1] ,main=colnames(usAir)[1],xlab=" ")



mean(scaledData[,1])

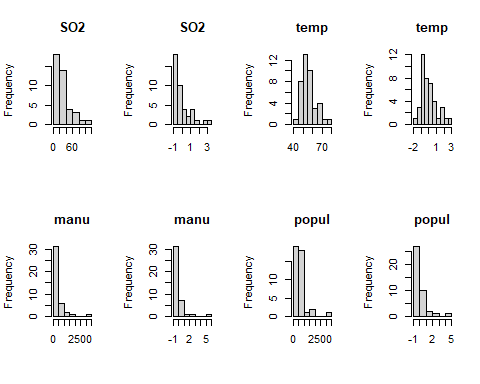
## [1] 7.948722e-17

var(scaledData[,1])

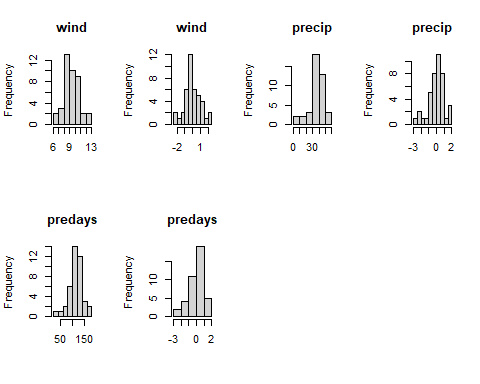
## [1] 1

Compare the raw and standardized histograms for all of the variables.

par(mfrow=c(2,4))  
hist(usAir [,1] ,main=colnames(usAir)[1],xlab=" ")  
hist(scaledData [,1] ,main=colnames(usAir)[1],xlab=" ")  
hist(usAir [,2] ,main=colnames(usAir)[2],xlab=" ")   
hist(scaledData [,2] ,main=colnames(usAir)[2],xlab=" ")   
hist(usAir [,3] ,main=colnames(usAir)[3],xlab=" ")   
hist(scaledData [,3] ,main=colnames(usAir)[3],xlab=" ")   
hist(usAir [,4] ,main=colnames(usAir)[4],xlab=" ")  
hist(scaledData [,4] ,main=colnames(usAir)[4],xlab=" ")



par(mfrow=c(2,4))  
hist(usAir [,5] ,main=colnames(usAir)[5],xlab=" ")   
hist(scaledData [,5] ,main=colnames(usAir)[5],xlab=" ")   
hist(usAir [,6] ,main=colnames(usAir)[6],xlab=" ")   
hist(scaledData [,6] ,main=colnames(usAir)[6],xlab=" ")   
hist(usAir [,7] ,main=colnames(usAir)[7],xlab=" ")  
hist(scaledData [,7] ,main=colnames(usAir)[7],xlab=" ")



***Question 4: Are you convinced that the variances are equalized? Just to check, calculate the mean and variance for each of the standardized variables. (10 pts)***

*The variances have all been equalized to 1, and the means are very close to zero but not exactly zero*:

# temperature  
mean(scaledData[,2])

## [1] -2.446595e-16

var(scaledData[,2])

## [1] 1

# manufacturing  
mean(scaledData[,3])

## [1] -2.702572e-17

var(scaledData[,3])

## [1] 1

# population  
mean(scaledData[,4])

## [1] -3.056921e-18

var(scaledData[,4])

## [1] 1

# wind  
mean(scaledData[,5])

## [1] 3.423653e-16

var(scaledData[,5])

## [1] 1

# precip  
mean(scaledData[,6])

## [1] 2.752003e-16

var(scaledData[,6])

## [1] 1

#predays  
mean(scaledData[,7])

## [1] -1.262011e-16

var(scaledData[,7])

## [1] 1

**Z standardization is very common in life sciences.**

# Detecting Outliers

Outliers are recorded values of measurements or observations that are outside the range of the bulk of the data. Outliers can inflate variance and lead to erroneous conclusions.

## Univariate outliers

One way to deal with outliers in multivariate data is to examine each variable separately. You will standardize your data into standard deviation units (z –standardization) and look for values that fall outside of three standard deviations.

First the z-standardization:

scaledData <- scale(usAir)

Next we will create histograms to look for values > than 3 sd. However, this time we will use the *par function* to look at all seven histograms at once.

par(mfrow=c(2,4))  
hist(scaledData [,1] ,main=colnames(usAir)[1],xlab=" ")  
hist(scaledData [,2] ,main=colnames(usAir)[2],xlab=" ")   
hist(scaledData [,3] ,main=colnames(usAir)[3],xlab=" ")   
hist(scaledData [,4] ,main=colnames(usAir)[4],xlab=" ")  
hist(scaledData [,5] ,main=colnames(usAir)[5],xlab=" ")   
hist(scaledData [,6] ,main=colnames(usAir)[6],xlab=" ")   
hist(scaledData [,7] ,main=colnames(usAir)[7],xlab=" ")

Finally, you can identify the outlier(s) for each variable:

scaledData [,1][scaledData [,1]>3]   
scaledData [,2][scaledData [,2]>3]   
scaledData [,3][scaledData [,3]>3]   
scaledData [,4][scaledData [,4]>3]  
scaledData [,5][scaledData [,5]>3]  
scaledData [,6][scaledData [,6]>3]   
scaledData [,7][scaledData [,7]>3]

Alternatively, you could use the apply function, less typing!

For the histogram function (hist):

par(mfrow=c(2,4))  
mapply(hist,as.data.frame(usAir),main=colnames(usAir),xlab=" ")

Here is a new function for detecting outliers called out.

out<-function(x){  
lier<-x[abs(x)>3]  
return(lier)  
}

Let’s apply that function:

apply(scaledData,2,out)

## $SO2  
## Chicago   
## 3.406199   
##   
## $temp  
## named numeric(0)  
##   
## $manu  
## Chicago   
## 5.112752   
##   
## $popul  
## Chicago   
## 4.766583   
##   
## $wind  
## named numeric(0)  
##   
## $precip  
## named numeric(0)  
##   
## $predays  
## named numeric(0)

**Question 5: Do you detect any outliers? For which variables? (5 pts)**

*I detected 1 outlier in the SO2, manu, and popul columns. All of these outliers came from the Chicago row.*

## Multivariate outliers

**we will come back to this…**

# Distance and Dissimilarity

As we know from lecture, multivariate data with *p* variables are visually represented by a collection of points forming a data cloud in *p*-dimensional space. The shape, clumping, and dispersion of the data cloud contains information we seek to describe. Several distance and dissimilarity measures are used to calculate the distance between data points.

## Euclidean Distance:

**Euclidean** distance is one of the most commonly used distance measures. It is normally preceded by column standardization (e.g. z standardization). Let’s calculate Euclidean distance for the US air pollution data set. You will use the function *vegdist* from the *vegan* (vegetation analysis) package. Look up *vegdist* to see the different indices available in this package.

?vegdist

First, z standardization:

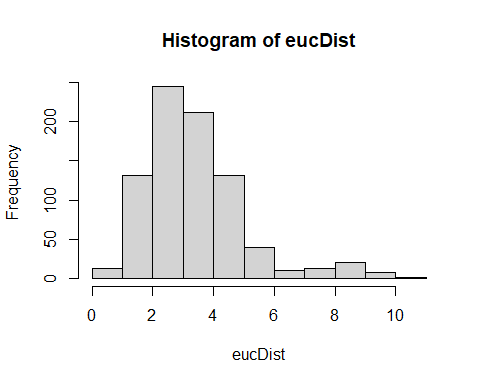
scaledData <- scale(usAir)

Then calculate distance:

eucDist <- vegdist(scaledData,"euclidean")

Let’s look at a histogram of distances:

hist(eucDist)



mean(eucDist)

## [1] 3.364608

max(eucDist)

## [1] 10.249

**Question 6: What does this frequency distribution tell you about pollution conditions across these 41 cities? (5 pts)**

*The histogram tells us that most cities are fairly close to one another in the data cloud. The majority of observations are clustered between 1 and 5 on the histogram, with a mean of 3.36. The distribution of Euclidean distances is left-skewed, and the maximum Euclidean distance is 10.25.*

Euclidean Distance can be weird. Let look at the data matrix below:

We want to determine how similar these farms are in theit production of strawberries, peaches, and raspberries.

Fruit <-rbind(c(1,0,1,1),c(2,1,0,0), c(3,0,4,4))  
colnames(Fruit)<-c("Farm","Strawberry","Peach", "Raspberry")  
Fruit

## Farm Strawberry Peach Raspberry  
## [1,] 1 0 1 1  
## [2,] 2 1 0 0  
## [3,] 3 0 4 4

Calculating Euclidean distance on these data:

eucDist<- vegdist(Fruit[,-1], "euclidean")

Gives us this distance matrix (R gives you the triangular matrix without the diagonal):

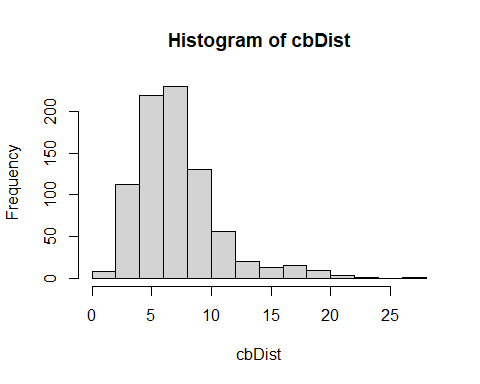
The distance between farms 1 and 2, which grow none of the same fruits:

Is **less** (i.e., these farms are more similar in their fruit production) than farms 1 and 3, which grow the same fruit:

Euclidean distance is not a jack-of-all-trades and is not appropriate for all data sets. Our next distance metric, Manhattan distance would also rank Farms 1 and 2 more similar than 1 and 3.

## City-block (Manhattan) distance

cbDist <- vegdist(scaledData,"manhattan")  
  
#Let’s look at a histogram of distances:  
  
hist(cbDist)



mean(cbDist)

## [1] 7.133932

max(cbDist)

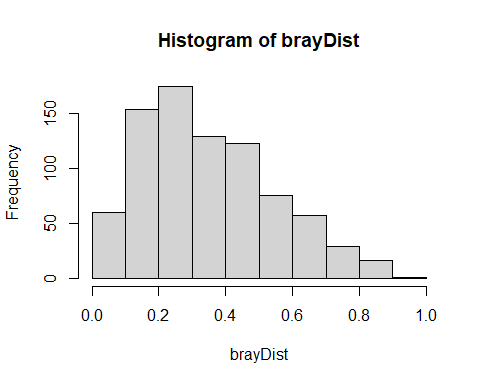
## [1] 26.00623

**Question 7: How does this distribution compare to Euclidean distance? (5 pts)**

*This histogram is also left-skewed. The mean (7.13) is higher than that of the Euclidean distance distribution, and the maximum distance is also higher (26.01).*

## Bray-Curtis dissimilarity

brayDist <- vegdist(usAir,"bray")  
  
#Histogram:  
  
hist(brayDist)



Let’s quickly look at our fruit farm data with Bray-Curtis:

brayFruit<- vegdist(Fruit[,-1], "bray")  
brayFruit

## 1 2  
## 2 1.0   
## 3 0.6 1.0

That makes more sense! Farms 1 and 2 (and 2 and 3) are at maximum dissimilarity and farms 1, 3 are more similar.

**Back to multivariate outliers!**

Your goal here is to examine deviations of the sample average distances to other samples. We will use **Bray-Curtis** distance:

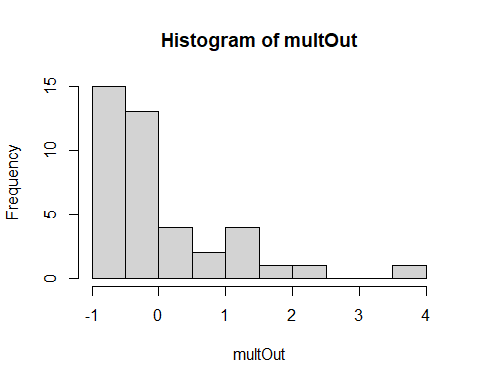
brayDist <- vegdist(usAir,"bray")

Next, calculate column means. These column means represent the average dissimilarity of each city to all other cities. You want to know if any cities are on average more than 3 standard deviation units (z scores). To achieve this, z-transform the averages:

multOut <- scale(colMeans(as.matrix(brayDist)))

Plot a histogram and look for observations >3 sd units:

hist(multOut)



You can find the cities that are outliers with:

multOut [multOut >3,]

## Chicago   
## 3.678029

Another possibility is to determine which observation are > 3 standard deviations from the mean. Using Bray-Curtis distance again:

Calculate column means:

colBray <- colMeans(as.matrix(brayDist))

Calculate the mean of the column means:

mBray <- mean(colBray)

Calculate the standard deviation:

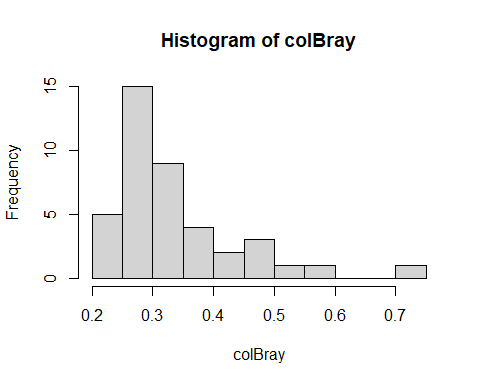
stdBray <- sd(colBray)

… 3 standard deviations

threeSD <- stdBray \* 3 + mBray

plot a histogram and look for observations >3 sd:

hist(colBray)



Find the outliers:

colBray [colBray >threeSD]

## Chicago   
## 0.7113063

# Bridle Shiner data

**Question 8: NOW, RUN THROUGH THE ABOVE EXERCISES WITH YOUR OWN DATA! (55 pts)**

## Import Data

bds <- read.csv("./Data/Katz\_BDS\_data.csv", header = TRUE, row.names = 1)  
bds <- bds[,-2]  
bds$catchment <- as.factor(bds$catchment)  
bds$lith <- as.factor(bds$lith)  
bds$marine <- as.factor(bds$marine)  
bds$ph <- bds$ph/10 # raster has pH values \* 10 so that values are integers

## Data screening

Your first move when conducting a multivariate analysis (or any analysis) is to screen the data. You are looking for data errors, missing data, and outliers that may influence your analysis.

### Data errors

One way to check for data errors is to examine the summary statistics for your data set.

First look at the summary statistics for bds:

describeBy(bds)

***Question 1: Do you see any unrealistic values? (5 pts) Note please answer all questions with points related to them.***

*While no values appear to be unrealistic, some variables are highly skewed or have very high kurtosis values. This is because some land cover types are relatively rare in the state, so these columns are zero-inflated.*

### Missing Data

When you have missing entries in your data sheet, R replaces them with “NA”. You can check if you have any missing variables in *bds*:

describe(bds)

*There are no NA values in the bds dataset because the function that generated the random points removed points with NA values.*

The *describe* function provides some of the same information as *describeBy*, but importantly shows you which variables have missing values.

We talked about two methods for dealing with missing values in lecture; **Complete Case and Imputation**. We will look at **complete case and imputation** for now.

**Complete Case** involves the removal of samples (in this case cities) with missing data:

bds [complete.cases(bds),] # all cases are complete cases

**Imputation** involves filling in missing values with plausible data. Let’s replace NAs with the mean of the variable.

# Remove columns with categorical or binary data  
bds.num <- bds %>% dplyr::select(-pab, -catchment, -lith, -marine)  
  
#First, let’s calculate the mean of each variable (column) with the NA removed:  
  
meanz <- colMeans(bds.num,na.rm=T)  
  
#`na.rm=T`, means that you want to remove NAs  
  
#To replace your NAs with the means you just calculated you will use the following function:  
  
naFunc<-function(column) {   
 column[is.na(column)] = round(mean(column, na.rm = TRUE),2)  
 return(column)   
}  
  
#and “apply” it to the usair\_mod data set  
  
Impute <- apply(bds.num,2,naFunc)

Check out the new Impute data object and make sure that the NA’s have been replaced.

describe(Impute) # same as original dataset since there were no NA values

We will not go into this advanced function too much. However, know that *apply* allows us to perform a function on all the rows and/or columns in a data frame of matrix. As we spoke about in lecture, there are many types of imputation methods. We can explore further methods for your specific missing data.

## Multivariate Normal Distribution

Many of the analyses we will do in this course have an assumption of multivariate normality. While there are many tests of multivariate normality, they tend to be overly conservative. If we strictly followed these tests, we may never run a multivariate analysis with ecological or agricultural data. Here we will look at two multivariate tests of normality.

### Shapiro-Wilks test

Shapiro-Wilks tests if the distribution of the observed data differs from multivariate normal distribution. So, we are looking for p-values > 0.05.

#mshapiro.test(t(bds.num)) # sample size is too large for the Shapiro-Wilks test  
bds.5000 <- sample(nrow(bds.num), 5000) # random sample of points  
mshapiro.test(t(bds.5000))

##   
## Shapiro-Wilk normality test  
##   
## data: Z  
## W = 0.95425, p-value < 2.2e-16

### Mardia test

Mardia’s test looks at multivariate extensions of Skewness and Kurtosis. In both cases, we are looking for p-values > 0.05 to show that our data do not deviate from the expectations of multivariate normal Skewness and Kurtosis. For the observed data to be considered multivariate normal, p-values from both the Skewness and Kurtosis statistics must be > 0.05. This function also tests for univariate normality of residuals using the Shapiro-Wilk statistic.

mvn(bds.num, mvnTest = "mardia")

## $multivariateNormality  
## Test Statistic p value Result  
## 1 Mardia Skewness 5815436.16861524 0 NO  
## 2 Mardia Kurtosis 5006.95216253276 0 NO  
## 3 MVN <NA> <NA> NO  
##   
## $univariateNormality  
## Test Variable Statistic p value  
## 1 Anderson-Darling clay 756.2282 <0.001   
## 2 Anderson-Darling coast 79.5410 <0.001   
## 3 Anderson-Darling elev 147.0374 <0.001   
## 4 Anderson-Darling ph 49.2399 <0.001   
## 5 Anderson-Darling X01\_water 1159.1656 <0.001   
## 6 Anderson-Darling X02\_Barren 3478.3419 <0.001   
## 7 Anderson-Darling X03\_LAc\_NHardwd 928.0348 <0.001   
## 8 Anderson-Darling X05\_NAtl\_CoastPlain\_Hardwd 2873.6516 <0.001   
## 9 Anderson-Darling X07\_LAc\_NPine.Oak 1208.7716 <0.001   
## 10 Anderson-Darling X08\_LAc\_Pine.Hemlock.Hardwd 477.6957 <0.001   
## 11 Anderson-Darling X09\_CAp\_Dry\_Oak.Pine 1227.4005 <0.001   
## 12 Anderson-Darling X10\_Ap\_Hemlock.N\_Hardwood 634.5300 <0.001   
## 13 Anderson-Darling X11\_Ac\_Low.Elev\_Spruce.Fir.Hardwood 1799.3655 <0.001   
## 14 Anderson-Darling X12\_AcAp\_Montane\_Spruce.Fir 2458.5302 <0.001   
## 15 Anderson-Darling X13\_CentAp\_Pine.Oak\_Rocky\_Wd 1293.7706 <0.001   
## 16 Anderson-Darling X14\_NAtl\_Coast\_Plain\_Maritime 3815.7504 <0.001   
## 17 Anderson-Darling X15\_NAtl\_Coast\_Plain\_Dune 2990.6041 <0.001   
## 18 Anderson-Darling X16\_CentIntAp\_FloodplainSys 2963.2122 <0.001   
## 19 Anderson-Darling X17\_CentIntAp\_RiparianSys 3345.4744 <0.001   
## 20 Anderson-Darling X18\_LAc\_FloodplainSys 1450.3219 <0.001   
## 21 Anderson-Darling X19\_Bor\_Acidic\_PeatSys 2591.2372 <0.001   
## 22 Anderson-Darling X20\_CentIntAp\_SwampSys 1722.5874 <0.001   
## 23 Anderson-Darling X21\_GulfAtl\_CoastPlain\_SwampSys 2315.0325 <0.001   
## 24 Anderson-Darling X22\_GulfAtl\_CoastPlain\_TMarshSys 3406.1730 <0.001   
## 25 Anderson-Darling X23\_LAc\_Shrub.Herb\_WetlSys 565.0197 <0.001   
## 26 Anderson-Darling X24\_NCentInt\_Wet\_Flatwd 3429.2780 <0.001   
## 27 Anderson-Darling X25\_LAc\_SwampSys 895.2823 <0.001   
## 28 Anderson-Darling X26\_NeInt\_PineBarrens 2347.4029 <0.001   
## 29 Anderson-Darling X27\_AcAp\_WdHeath.Krummholz 3307.7872 <0.001   
## 30 Anderson-Darling X28\_Bor\_JackPine.BlackSpruce 3232.8368 <0.001   
## 31 Anderson-Darling X29\_AcAp\_AlpineTundra 2872.7225 <0.001   
## 32 Anderson-Darling sand 241.3656 <0.001   
## 33 Anderson-Darling silt 121.4449 <0.001   
## 34 Anderson-Darling slope 141.2233 <0.001   
## Normality  
## 1 NO   
## 2 NO   
## 3 NO   
## 4 NO   
## 5 NO   
## 6 NO   
## 7 NO   
## 8 NO   
## 9 NO   
## 10 NO   
## 11 NO   
## 12 NO   
## 13 NO   
## 14 NO   
## 15 NO   
## 16 NO   
## 17 NO   
## 18 NO   
## 19 NO   
## 20 NO   
## 21 NO   
## 22 NO   
## 23 NO   
## 24 NO   
## 25 NO   
## 26 NO   
## 27 NO   
## 28 NO   
## 29 NO   
## 30 NO   
## 31 NO   
## 32 NO   
## 33 NO   
## 34 NO   
##   
## $Descriptives  
## n Mean Std.Dev  
## clay 10000 1.163636e+02 45.99834048  
## coast 10000 6.996792e+01 38.86045860  
## elev 10000 2.228168e+02 149.21256643  
## ph 10000 4.748460e+00 0.15796185  
## X01\_water 10000 9.549480e+00 13.03727352  
## X02\_Barren 10000 3.755941e-03 0.03509914  
## X03\_LAc\_NHardwd 10000 1.519425e+01 20.02533059  
## X05\_NAtl\_CoastPlain\_Hardwd 10000 6.935651e-01 2.85235498  
## X07\_LAc\_NPine.Oak 10000 6.279393e+00 10.02874481  
## X08\_LAc\_Pine.Hemlock.Hardwd 10000 1.533205e+01 15.54632050  
## X09\_CAp\_Dry\_Oak.Pine 10000 1.404577e+01 20.90028553  
## X10\_Ap\_Hemlock.N\_Hardwood 10000 1.994981e+01 21.71463379  
## X11\_Ac\_Low.Elev\_Spruce.Fir.Hardwood 10000 4.519731e+00 9.88455569  
## X12\_AcAp\_Montane\_Spruce.Fir 10000 2.093982e+00 6.47951319  
## X13\_CentAp\_Pine.Oak\_Rocky\_Wd 10000 1.093062e+00 1.97086658  
## X14\_NAtl\_Coast\_Plain\_Maritime 10000 1.042972e-02 0.12221188  
## X15\_NAtl\_Coast\_Plain\_Dune 10000 2.177741e-01 1.18524462  
## X16\_CentIntAp\_FloodplainSys 10000 3.177154e-01 1.24192853  
## X17\_CentIntAp\_RiparianSys 10000 4.260004e-03 0.04279314  
## X18\_LAc\_FloodplainSys 10000 5.430193e-01 0.99444794  
## X19\_Bor\_Acidic\_PeatSys 10000 1.412281e-01 0.46132339  
## X20\_CentIntAp\_SwampSys 10000 1.268840e+00 2.77330557  
## X21\_GulfAtl\_CoastPlain\_SwampSys 10000 1.626741e+00 4.80973465  
## X22\_GulfAtl\_CoastPlain\_TMarshSys 10000 2.003311e-01 1.42490819  
## X23\_LAc\_Shrub.Herb\_WetlSys 10000 1.838958e+00 2.24737690  
## X24\_NCentInt\_Wet\_Flatwd 10000 1.301390e-01 0.75768057  
## X25\_LAc\_SwampSys 10000 3.235448e+00 4.00547745  
## X26\_NeInt\_PineBarrens 10000 1.464646e+00 4.24546893  
## X27\_AcAp\_WdHeath.Krummholz 10000 2.744225e-02 0.17431419  
## X28\_Bor\_JackPine.BlackSpruce 10000 4.904319e-03 0.02299373  
## X29\_AcAp\_AlpineTundra 10000 2.132618e-01 0.86981993  
## sand 10000 5.146307e+02 70.07793104  
## silt 10000 3.690054e+02 34.87626204  
## slope 10000 5.452070e+00 2.56169935  
## Median Min Max  
## clay 98.48896027 63.32500076 351.5013123  
## coast 67.52747726 0.02999878 159.8684540  
## elev 206.62452698 10.15273380 958.8146362  
## ph 4.75848122 4.07632790 5.3803642  
## X01\_water 3.60974646 0.00000000 48.6029816  
## X02\_Barren 0.00000000 0.00000000 1.7646281  
## X03\_LAc\_NHardwd 3.37779570 0.00000000 74.8570557  
## X05\_NAtl\_CoastPlain\_Hardwd 0.00000000 0.00000000 30.4152985  
## X07\_LAc\_NPine.Oak 3.23444152 0.00000000 85.1046753  
## X08\_LAc\_Pine.Hemlock.Hardwd 9.81204796 0.00000000 74.7757034  
## X09\_CAp\_Dry\_Oak.Pine 4.53647804 0.00000000 82.1304550  
## X10\_Ap\_Hemlock.N\_Hardwood 9.86926460 0.00000000 72.0855179  
## X11\_Ac\_Low.Elev\_Spruce.Fir.Hardwood 0.05111158 0.00000000 67.5645981  
## X12\_AcAp\_Montane\_Spruce.Fir 0.04204301 0.00000000 70.0239258  
## X13\_CentAp\_Pine.Oak\_Rocky\_Wd 0.38724124 0.00000000 23.4355640  
## X14\_NAtl\_Coast\_Plain\_Maritime 0.00000000 0.00000000 2.8447349  
## X15\_NAtl\_Coast\_Plain\_Dune 0.00000000 0.00000000 31.3757744  
## X16\_CentIntAp\_FloodplainSys 0.00000000 0.00000000 9.6371632  
## X17\_CentIntAp\_RiparianSys 0.00000000 0.00000000 1.1050403  
## X18\_LAc\_FloodplainSys 0.13203755 0.00000000 6.9846883  
## X19\_Bor\_Acidic\_PeatSys 0.00000000 0.00000000 3.6116099  
## X20\_CentIntAp\_SwampSys 0.14576580 0.00000000 18.6143456  
## X21\_GulfAtl\_CoastPlain\_SwampSys 0.02536456 0.00000000 50.1728935  
## X22\_GulfAtl\_CoastPlain\_TMarshSys 0.00000000 0.00000000 17.7078152  
## X23\_LAc\_Shrub.Herb\_WetlSys 1.40760350 0.00000000 36.7693405  
## X24\_NCentInt\_Wet\_Flatwd 0.00000000 0.00000000 8.5715742  
## X25\_LAc\_SwampSys 1.58895516 0.00000000 18.0171509  
## X26\_NeInt\_PineBarrens 0.01062562 0.00000000 42.4234200  
## X27\_AcAp\_WdHeath.Krummholz 0.00000000 0.00000000 2.7061627  
## X28\_Bor\_JackPine.BlackSpruce 0.00000000 0.00000000 0.2052186  
## X29\_AcAp\_AlpineTundra 0.00000000 0.00000000 6.6756868  
## sand 529.37884521 168.58792114 647.9694214  
## silt 361.51593018 280.08151245 479.9107666  
## slope 4.85915327 1.27876532 19.1336193  
## 25th 75th Skew  
## clay 8.765207e+01 1.248974e+02 2.0543715  
## coast 3.736871e+01 1.022984e+02 0.1651141  
## elev 9.861147e+01 3.136621e+02 0.9528727  
## ph 4.657930e+00 4.843970e+00 -0.4052842  
## X01\_water 1.574614e+00 1.225102e+01 1.9544768  
## X02\_Barren 0.000000e+00 0.000000e+00 29.2499353  
## X03\_LAc\_NHardwd 0.000000e+00 2.712638e+01 1.1611271  
## X05\_NAtl\_CoastPlain\_Hardwd 0.000000e+00 8.417732e-02 6.5447826  
## X07\_LAc\_NPine.Oak 1.187279e+00 6.705354e+00 4.1429928  
## X08\_LAc\_Pine.Hemlock.Hardwd 2.998420e+00 2.447358e+01 1.1154591  
## X09\_CAp\_Dry\_Oak.Pine 7.654268e-01 1.530597e+01 1.8866450  
## X10\_Ap\_Hemlock.N\_Hardwood 1.838616e-01 3.563745e+01 0.7074348  
## X11\_Ac\_Low.Elev\_Spruce.Fir.Hardwood 0.000000e+00 4.461258e+00 3.2047992  
## X12\_AcAp\_Montane\_Spruce.Fir 0.000000e+00 5.156540e-01 4.8562377  
## X13\_CentAp\_Pine.Oak\_Rocky\_Wd 5.362386e-02 1.088998e+00 4.5099579  
## X14\_NAtl\_Coast\_Plain\_Maritime 0.000000e+00 0.000000e+00 12.4346089  
## X15\_NAtl\_Coast\_Plain\_Dune 0.000000e+00 1.162131e-02 12.4101258  
## X16\_CentIntAp\_FloodplainSys 0.000000e+00 0.000000e+00 5.0755041  
## X17\_CentIntAp\_RiparianSys 0.000000e+00 0.000000e+00 22.6615727  
## X18\_LAc\_FloodplainSys 2.211083e-02 5.049468e-01 2.9212626  
## X19\_Bor\_Acidic\_PeatSys 0.000000e+00 1.038789e-03 4.6266914  
## X20\_CentIntAp\_SwampSys 7.858016e-03 1.343050e+00 3.6515647  
## X21\_GulfAtl\_CoastPlain\_SwampSys 0.000000e+00 1.027689e+00 5.0587350  
## X22\_GulfAtl\_CoastPlain\_TMarshSys 0.000000e+00 0.000000e+00 10.0940998  
## X23\_LAc\_Shrub.Herb\_WetlSys 3.359762e-01 2.540701e+00 5.6665337  
## X24\_NCentInt\_Wet\_Flatwd 0.000000e+00 0.000000e+00 7.7308177  
## X25\_LAc\_SwampSys 6.450234e-01 3.839752e+00 1.8372535  
## X26\_NeInt\_PineBarrens 0.000000e+00 6.059866e-01 4.5754204  
## X27\_AcAp\_WdHeath.Krummholz 0.000000e+00 0.000000e+00 10.0200415  
## X28\_Bor\_JackPine.BlackSpruce 0.000000e+00 0.000000e+00 6.4227648  
## X29\_AcAp\_AlpineTundra 0.000000e+00 6.242312e-02 5.9476625  
## sand 4.779336e+02 5.667198e+02 -1.2752403  
## silt 3.408725e+02 3.959870e+02 0.3906550  
## slope 3.619393e+00 7.015670e+00 1.1460221  
## Kurtosis  
## clay 4.6291044  
## coast -1.0189930  
## elev 0.9673952  
## ph 1.3353959  
## X01\_water 2.8994487  
## X02\_Barren 1307.9921728  
## X03\_LAc\_NHardwd 0.1054214  
## X05\_NAtl\_CoastPlain\_Hardwd 50.8859458  
## X07\_LAc\_NPine.Oak 22.4769919  
## X08\_LAc\_Pine.Hemlock.Hardwd 0.3507977  
## X09\_CAp\_Dry\_Oak.Pine 2.4390227  
## X10\_Ap\_Hemlock.N\_Hardwood -0.9097099  
## X11\_Ac\_Low.Elev\_Spruce.Fir.Hardwood 12.0273276  
## X12\_AcAp\_Montane\_Spruce.Fir 29.0175764  
## X13\_CentAp\_Pine.Oak\_Rocky\_Wd 33.7332760  
## X14\_NAtl\_Coast\_Plain\_Maritime 161.0338158  
## X15\_NAtl\_Coast\_Plain\_Dune 204.9609602  
## X16\_CentIntAp\_FloodplainSys 27.2402725  
## X17\_CentIntAp\_RiparianSys 568.9728358  
## X18\_LAc\_FloodplainSys 9.8850940  
## X19\_Bor\_Acidic\_PeatSys 23.8369911  
## X20\_CentIntAp\_SwampSys 15.0955644  
## X21\_GulfAtl\_CoastPlain\_SwampSys 31.3317249  
## X22\_GulfAtl\_CoastPlain\_TMarshSys 108.6479165  
## X23\_LAc\_Shrub.Herb\_WetlSys 68.6828685  
## X24\_NCentInt\_Wet\_Flatwd 68.2168576  
## X25\_LAc\_SwampSys 2.7344943  
## X26\_NeInt\_PineBarrens 25.5675533  
## X27\_AcAp\_WdHeath.Krummholz 116.9685619  
## X28\_Bor\_JackPine.BlackSpruce 46.9934788  
## X29\_AcAp\_AlpineTundra 36.8670316  
## sand 2.0147774  
## silt -0.3706011  
## slope 2.0367486

## Data transformation

The next step is preparing your data for analysis is transforming the data. Today we will look at the log, square root, and arcsine square root transformations.

### Log transformation:

Several common transformations have built-in functions in R. While you can build transformation functions on your own, we will use the ones R has developed today. First, let’s look at a histogram of our first variable, *clay*, to determine if transformation is necessary:

hist(bds.num$clay)

To log transform each value in our data frame:

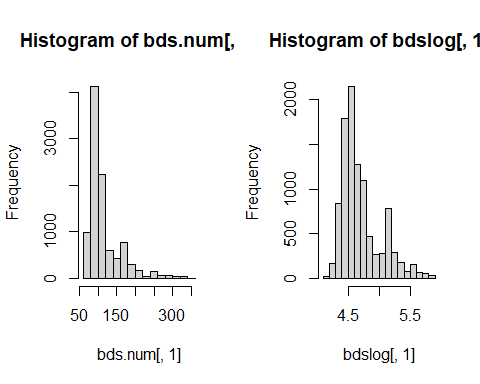
bdslog <- log1p(bds.num)

and the histogram:

hist(bdslog$clay)

You can compare the histograms side by side using the par function followed by hist:

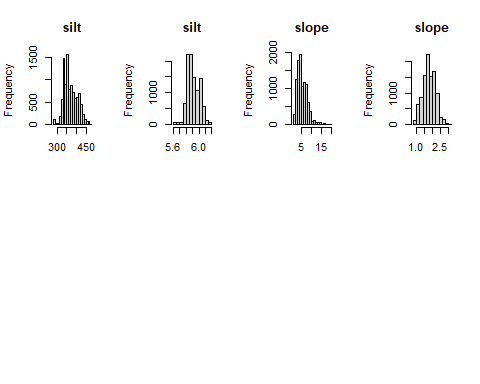
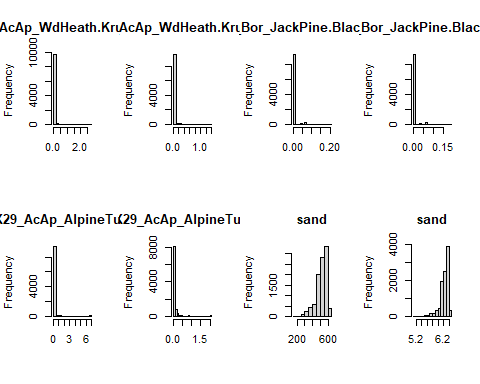
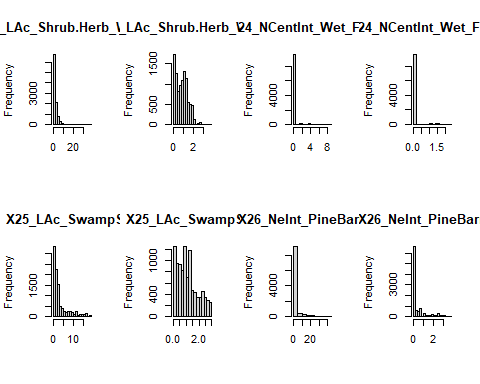
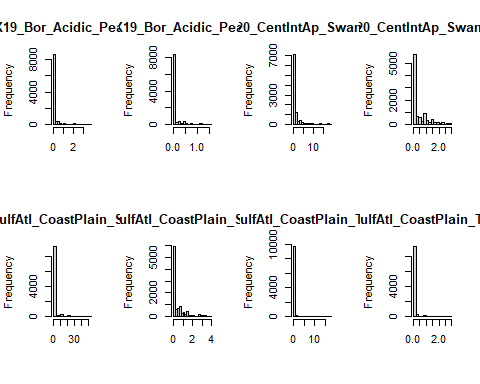
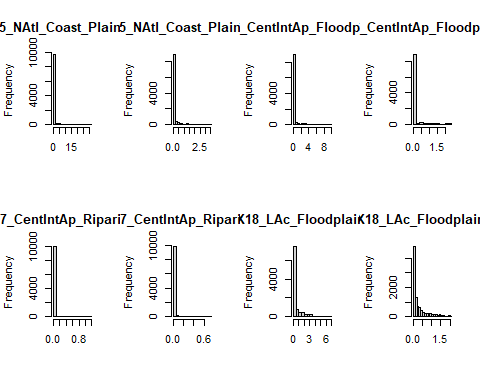
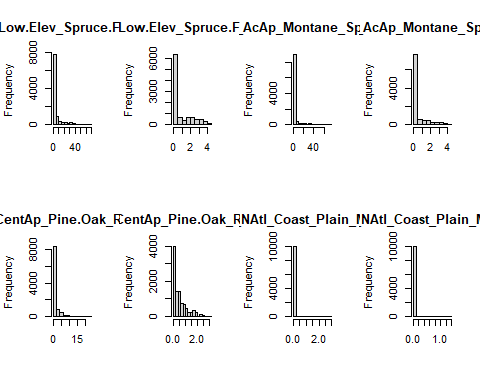
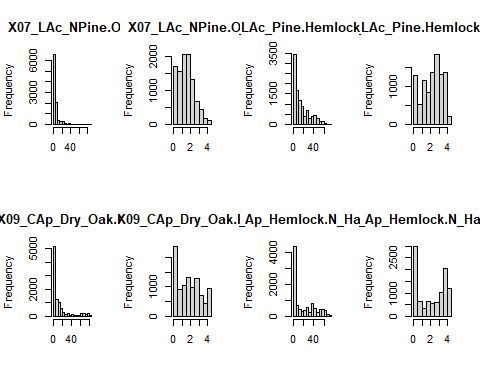
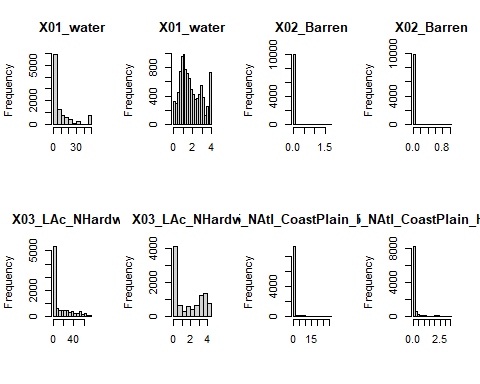
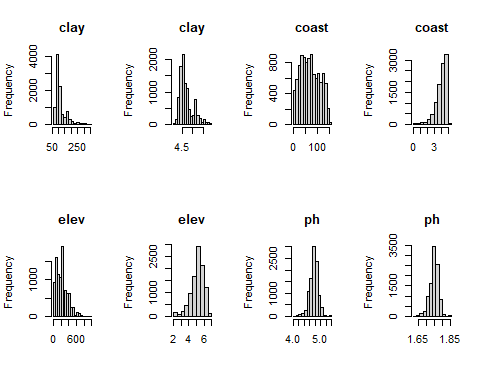
par(mfrow=c(1,2))  
  
hist(bds.num[,1])   
hist(bdslog[,1])



Placing 1, 2 in parentheses after the c (which stands for concatenate) in the par function indicates that you want your plots arranged in 1 row and two columns. Note this plotting is done in base R as opposed to using the ggplot functions of Tidyverse. It is helpful to know base R and Tidyverse to be able to read and trouble shoot code with a wide range of collaborators. In ggplot this code would be similar to what the *facet* function does.

Compare histograms for the raw data and the log transformed data for each variable.

par(mfrow=c(2,4))  
for (i in 1:ncol(bds.num)){  
 hist(bds.num[,i], main = paste(colnames(bds.num)[i]), xlab = "")  
 hist(bdslog[,i], main = paste(colnames(bdslog)[i]), xlab = "")  
}



**Question 2: Which variable might not need to be log transformed? (5 pts)**

*All of the variables need to be transformed in some way because none of them are normally distributed. The variables are also measured in different units: the soil data (e.g., clay, silt, and sand) were measured in g/kg, but the land cover types are areal proportions.*

### Square root transformation:

To square root transform each value in our data frame:

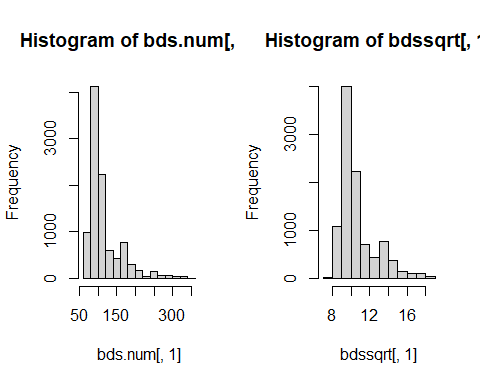
bdssqrt <- sqrt(bds.num)

and the histogram:

hist(bdssqrt$clay)

Compare the histograms side by side using the par function followed by hist:

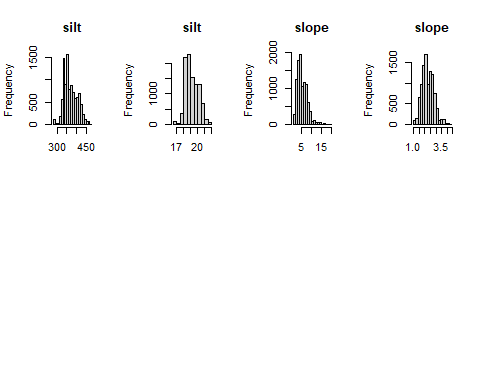
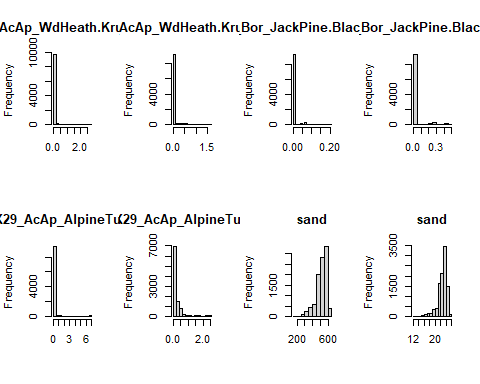
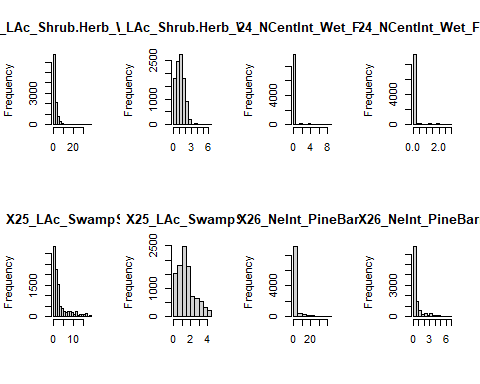
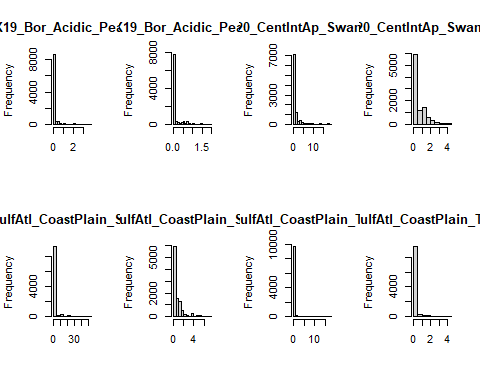
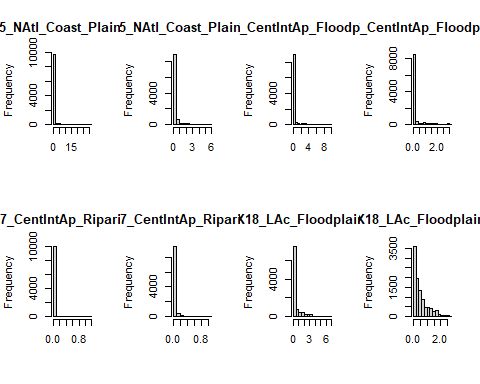
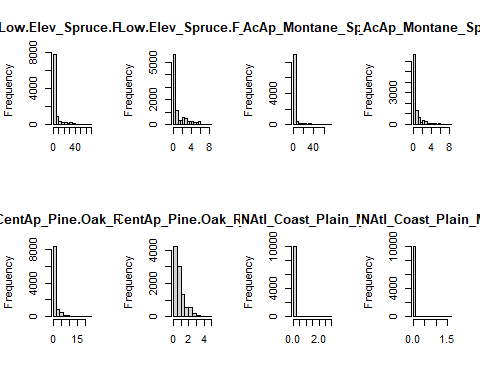
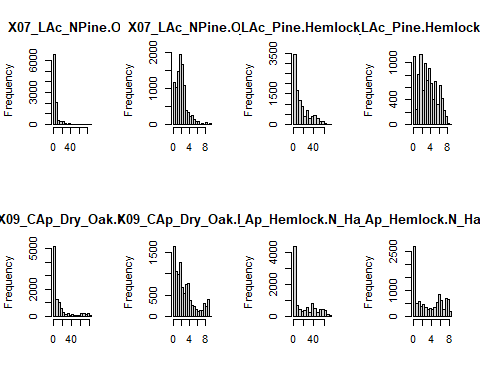
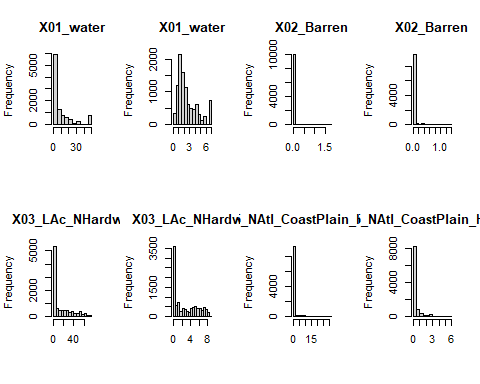
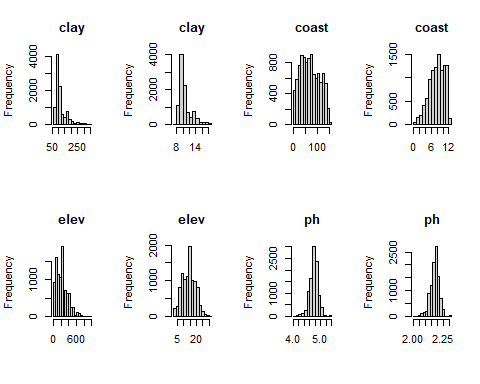
par(mfrow=c(1,2))  
  
hist(bds.num[,1])   
hist(bdssqrt[,1])



Compare histograms for the raw data and the square-root transformed data for each variable…

Remember that square root transformations are best used on count data.

par(mfrow=c(2,4))  
for (i in 1:ncol(bds.num)){  
 hist(bds.num[,i], main = paste(colnames(bds.num)[i]), xlab = "")  
 hist(bdssqrt[,i], main = paste(colnames(bdssqrt)[i]), xlab = "")  
}



### Arcsine square root transformation: = arcsine

If you remember arcsine square root transformations are for percentage data. So, the values for your variable must range between 0 and 1. All of the LANDFIRE land cover variables in bds are appropriate for this transformation.

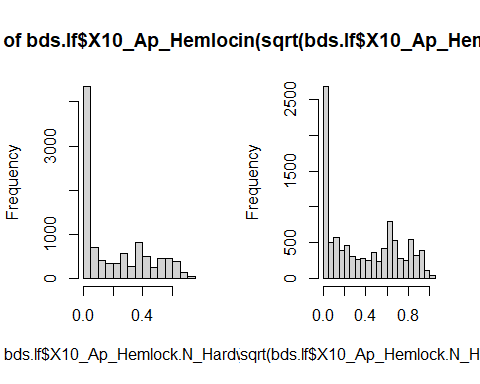
bds.lf <- bds.num[,c(5:31)]  
bds.lf <- bds.lf/100 # to percentages between 0 and 1

Now let’s transform:

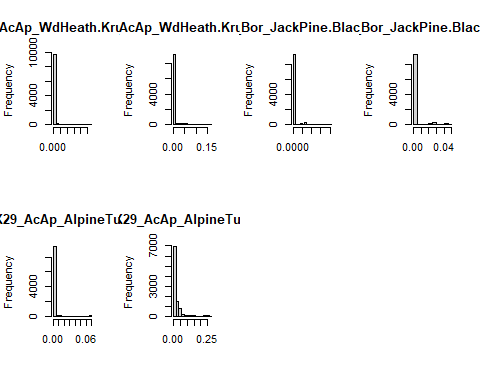
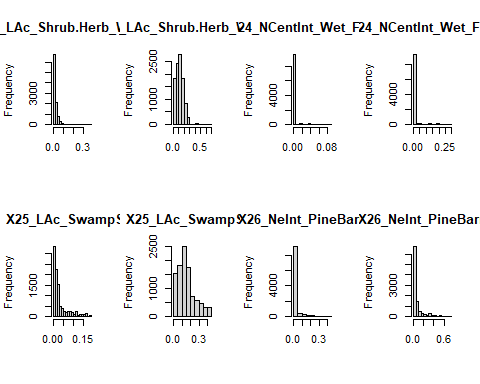
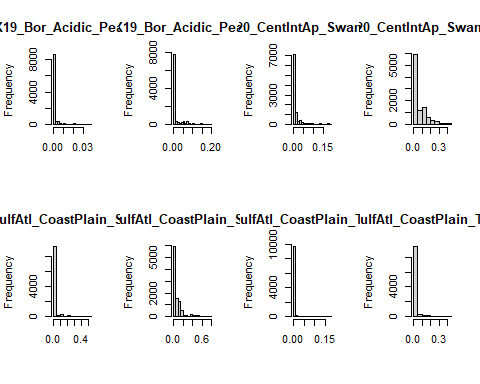
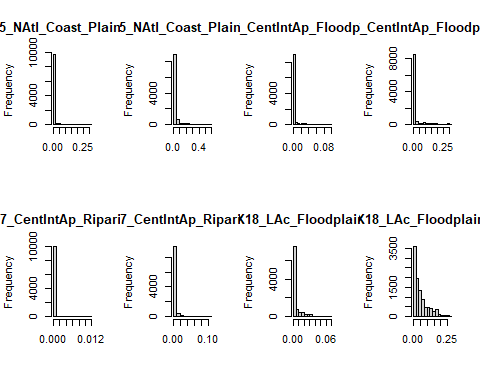
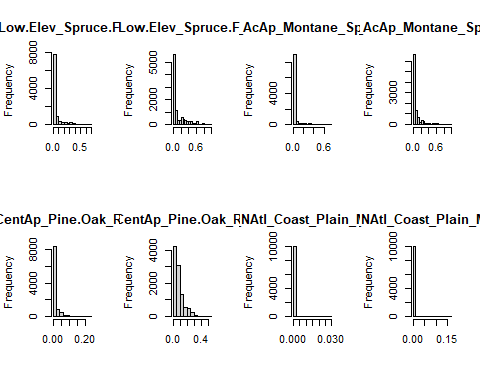
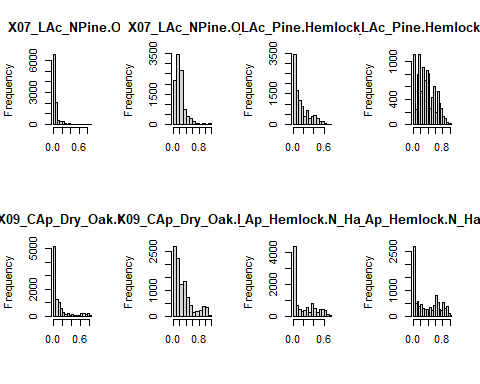
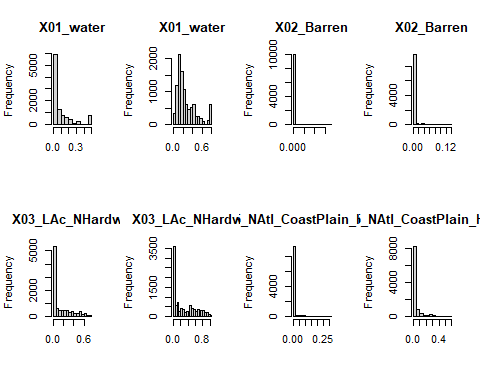
asin(sqrt(bds.lf))

and compare histograms:

par(mfrow=c(1,2))  
  
hist(bds.lf$X10\_Ap\_Hemlock.N\_Hardwood)  
hist(asin(sqrt(bds.lf$X10\_Ap\_Hemlock.N\_Hardwood)))



par(mfrow=c(2,4))  
for (i in 1:ncol(bds.lf)){  
 hist(bds.lf[,i], main = colnames(bds.lf)[i], xlab = "")  
 hist(asin(sqrt(bds.lf[,i])), main = colnames(bds.lf)[i], xlab = "")  
}



## Data standardization

Column standardization adjusts for differences among variables. The focus is on the profile across a sample unit. Row standardization adjusts for differences among sample units, wherein the focus is on the profile within a sample unit. Row standardization is good when variables are measured in the same units (e.g. species). You will more often than not be using column standardization.

### Coefficient of Variation (cv)

Let’s first see if the bridle shiner data set needs standardization by calculating the *coefficient of variation* **(cv)** for column totals. Remember, the **cv** is the ratio of the standard deviation to the mean (σ/μ):

First calculate the column **sums**:

cSums <- colSums(bds.num)

Then calculate the **standard deviation** and **mean** for the column sums:

Sdev <- sd(cSums)  
M <- mean(cSums)

Finally, calculate the **cv**:

Cv <- Sdev/M\*100

Our rule of thumb for cv is that if **cv> 50**, data standardization is necessary.

***Question 3: Is standardization necessary for the USairpollution data? (5 pts)***

*Data standardization is necessary because the CV of the dataset is 270.94*

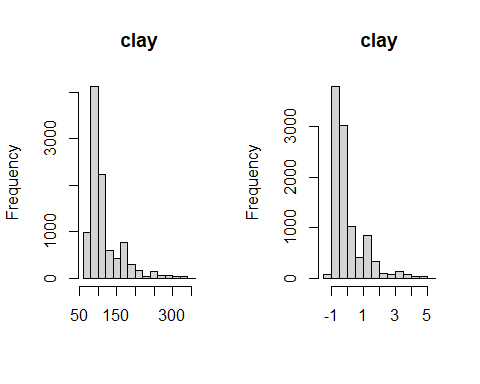
### Z- standardization = (

Your goal here is to equalize the variance for variables measured on different scales. There is a built-in function scale that will do this for you:

scaledData <- scale(bds.num)

Let’s look at histograms for the scaled and unscaled data for the first variable, clay:

par(mfrow=c(1,2))  
  
hist(bds.num[,1] ,main=colnames(bds.num)[1],xlab=" ")  
hist(scaledData[,1] ,main=colnames(bds.num)[1],xlab=" ")



mean(scaledData[,1])

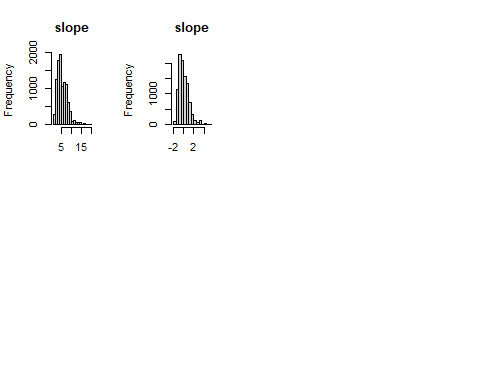
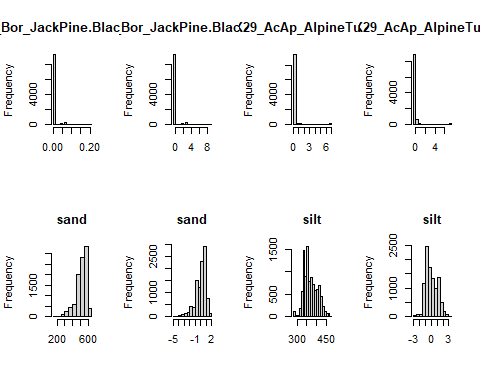
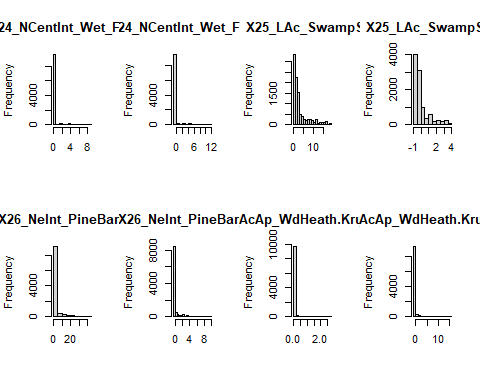
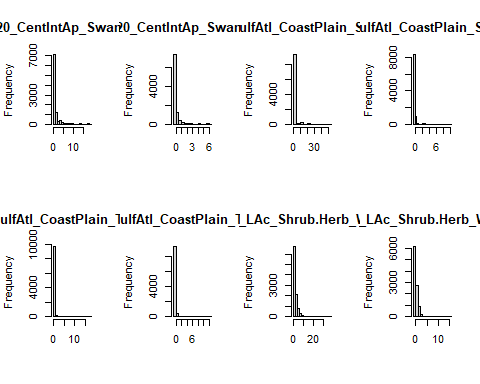
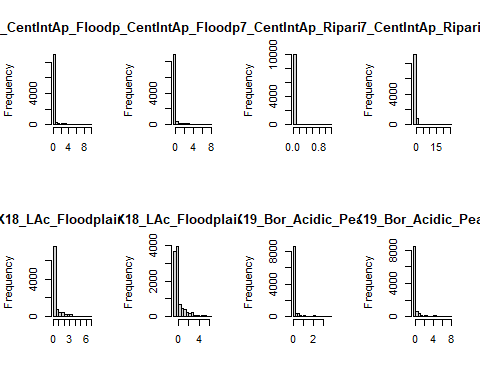
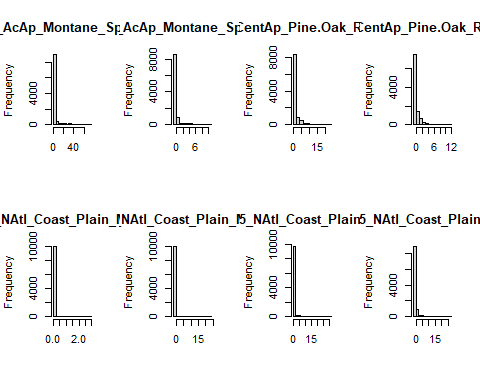
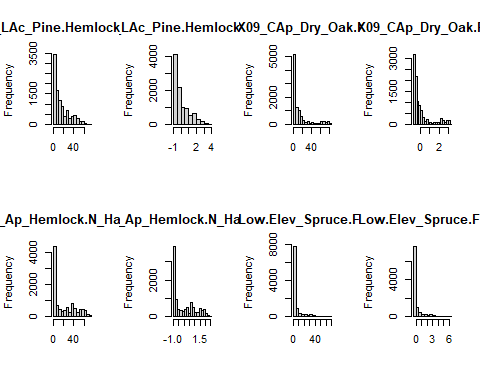
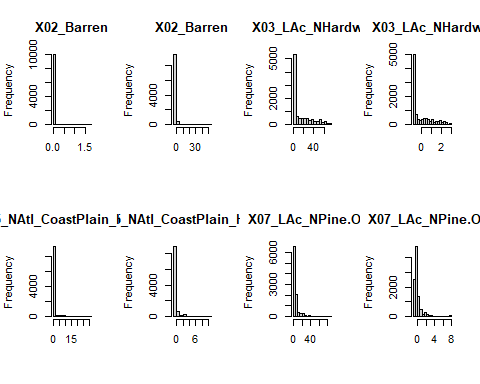
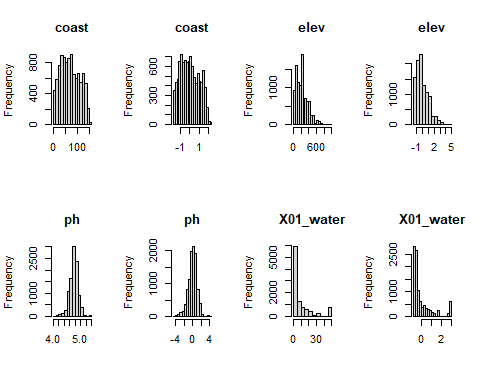
## [1] 1.473416e-17

var(scaledData[,1])

## [1] 1

Compare the raw and standardized histograms for all of the variables.

par(mfrow=c(2,4))  
for (i in 2:ncol(bds.num)){  
 hist(bds.num[,i], main = paste(colnames(bds.num)[i]), xlab=" ")  
 hist(scaledData[,i], main = paste(colnames(bds.num)[i]), xlab=" ")  
}



***Question 4: Are you convinced that the variances are equalized? Just to check, calculate the mean and variance for each of the standardized variables. (10 pts)***

*The variances have all been equalized to 1, and the means are close to zero but not exactly zero*:

q4 <- data.frame(matrix(NA, nrow=ncol(bds.num), ncol = 2))  
colnames(q4) <- c("mean", "variance")  
rownames(q4) <- colnames(bds.num)  
  
for (i in 1:ncol(bds.num)){  
 q4[i,1] <- mean(scaledData[,i])  
 q4[i,2] <- var(scaledData[,i])  
}  
min(q4$variance)

## [1] 1

max(q4$variance)

## [1] 1

min(q4$mean)

## [1] -6.663697e-17

max(q4$mean)

## [1] 5.054205e-16

**Z standardization is very common in life sciences.**

## Detecting Outliers

Outliers are recorded values of measurements or observations that are outside the range of the bulk of the data. Outliers can inflate variance and lead to erroneous conclusions.

### Univariate outliers

One way to deal with outliers in multivariate data is to examine each variable separately. You will standardize your data into standard deviation units (z –standardization) and look for values that fall outside of three standard deviations.

First the z-standardization:

scaledData <- scale(bds.num)

Next we will create histograms to look for values > than 3 sd. However, this time we will use the *par function* to look at all seven histograms at once.

par(mfrow=c(2,4))  
for (i in 1:ncol(bds.num)){  
 hist(scaledData[,i], main = colnames(bds.num)[i], xlab = "")  
}

Finally, you can identify the outlier(s) for each variable:

for (i in 1:ncol(bds.num)){  
 print(scaledData[,i][scaledData[,i] > 3])  
}

Alternatively, you could use the apply function, less typing!

For the histogram function (hist):

par(mfrow=c(2,4))  
mapply(hist,as.data.frame(scaledData),main=colnames(bds.num),xlab = "")

Here is a new function for detecting outliers called out.

out<-function(x){  
lier<-x[abs(x)>3]  
return(lier)  
}

Let’s apply that function:

apply(scaledData,2,out)

## $clay  
## 110 120 121 126 148 154 206 244   
## 4.685073 3.854004 3.031236 4.685073 3.854004 3.263870 3.854004 3.956937   
## 285 289 294 319 343 349 390 406   
## 4.685073 3.111551 3.031236 5.111874 3.031236 5.111874 3.031236 3.756924   
## 489 525 533 544 551 602 608 615   
## 3.031236 3.817041 3.263870 3.263870 3.263870 3.756924 4.399103 3.263870   
## 627 639 655 702 709 710 720 771   
## 4.136882 3.447520 4.158541 3.031236 3.263870 3.854004 3.956937 3.111551   
## 851 942 978 986 1077 1123 1146 1261   
## 3.111551 3.263870 3.854004 3.263870 3.263870 3.447520 3.854004 3.817041   
## 1301 1304 1306 1318 1368 1371 1375 1386   
## 3.111551 3.263870 3.263870 4.136882 3.031236 3.854004 4.136882 3.822165   
## 1427 1454 1458 1466 1487 1493 1532 1595   
## 4.158541 4.288407 3.111551 4.685073 3.854004 3.751141 3.817041 3.447520   
## 1599 1635 1636 1646 1699 1765 1766 1780   
## 4.166333 4.697779 3.263870 3.263870 3.031236 3.031236 4.136882 3.956937   
## 1833 1836 1837 1862 1866 1890 1903 1924   
## 4.288407 3.817041 3.817041 3.817041 3.756924 3.263870 3.111551 4.288407   
## 1979 2093 2166 2223 2307 2310 2311 2321   
## 3.447520 3.793726 3.263870 4.399103 4.399103 3.263870 4.288407 4.136882   
## 2354 2381 2412 2416 2428 2476 2515 2586   
## 4.166333 3.031236 3.031236 4.685073 4.288407 4.685073 4.399103 3.447520   
## 2657 2734 2781 2790 2821 2853 2858 2861   
## 4.685073 3.817041 3.111551 3.263870 3.031236 4.697779 4.158541 3.447520   
## 2918 2969 2985 3032 3049 3099 3105 3118   
## 3.263870 4.158541 3.111551 4.685073 3.031236 3.817041 3.263870 3.447520   
## 3186 3196 3213 3251 3262 3320 3357 3374   
## 4.399103 3.956937 3.031236 3.263870 3.031236 3.111551 4.697779 4.697779   
## 3433 3550 3570 3621 3629 3653 3660 3680   
## 3.822165 3.111551 3.263870 4.158541 4.685073 3.956937 3.031236 4.166333   
## 3698 3765 3766 3782 3838 3870 3881 3942   
## 4.288407 3.263870 3.111551 3.447520 3.263870 3.263870 3.111551 3.956937   
## 4066 4071 4098 4111 4138 4164 4219 4223   
## 3.111551 3.263870 3.111551 3.031236 4.136882 4.685073 3.031236 3.751141   
## 4226 4263 4267 4289 4330 4397 4404 4455   
## 3.854004 3.854004 4.136882 3.111551 4.697779 3.756924 3.817041 3.263870   
## 4464 4476 4504 4507 4577 4667 4822 4844   
## 4.166333 3.263870 3.756924 3.263870 3.111551 3.854004 4.697779 4.136882   
## 4855 4889 4895 4938 5172 5178 5229 5276   
## 4.288407 3.111551 4.685073 3.111551 3.956937 3.031236 3.263870 3.111551   
## 5289 5290 5343 5380 5434 5513 5556 5655   
## 4.166333 3.263870 3.031236 3.031236 4.399103 3.447520 4.288407 3.817041   
## 5698 5771 5787 5885 5909 5916 5927 5934   
## 3.111551 3.263870 3.956937 4.697779 3.263870 4.697779 3.111551 3.817041   
## 5957 6006 6007 6027 6099 6167 6253 6313   
## 3.111551 3.031236 4.136882 3.031236 4.685073 4.697779 3.111551 3.447520   
## 6317 6432 6463 6542 6577 6586 6667 6697   
## 4.685073 3.447520 3.263870 3.263870 4.685073 3.263870 3.111551 4.685073   
## 6728 6767 6823 6859 6861 6866 6894 6905   
## 3.956937 3.817041 3.263870 3.817041 3.956937 3.756924 3.111551 3.263870   
## 6954 6962 6982 7015 7019 7028 7049 7208   
## 4.685073 4.399103 3.031236 3.817041 4.685073 3.956937 3.751141 3.031236   
## 7236 7297 7311 7349 7352 7355 7397 7407   
## 3.263870 3.751141 3.031236 3.263870 4.685073 3.263870 3.031236 3.756924   
## 7433 7445 7482 7499 7558 7655 7664 7682   
## 3.111551 3.263870 3.031236 3.854004 3.817041 3.263870 3.756924 3.031236   
## 7685 7782 7833 7853 7906 7938 7954 7974   
## 3.756924 3.854004 3.031236 3.817041 4.166333 4.697779 3.956937 4.288407   
## 7989 8078 8093 8111 8123 8128 8205 8207   
## 3.263870 4.685073 3.956937 3.111551 3.263870 3.111551 3.111551 3.956937   
## 8212 8316 8323 8388 8541 8583 8643 8664   
## 3.111551 3.111551 3.854004 4.685073 3.111551 3.854004 4.685073 3.956937   
## 8670 8681 8697 8731 8757 8766 8767 8769   
## 3.263870 3.756924 3.756924 3.031236 5.111874 4.136882 3.111551 3.263870   
## 8840 8851 8873 8887 8913 8963 8964 8967   
## 3.263870 3.822165 3.263870 5.111874 3.111551 3.263870 3.447520 3.031236   
## 8969 8989 9025 9035 9047 9083 9150 9160   
## 3.111551 4.685073 4.399103 4.697779 3.263870 3.111551 3.031236 3.111551   
## 9174 9261 9330 9357 9359 9370 9402 9437   
## 3.447520 4.158541 3.793726 3.111551 3.031236 3.793726 3.031236 3.031236   
## 9466 9482 9486 9538 9578 9606 9633 9734   
## 4.399103 4.166333 3.263870 3.111551 3.111551 5.111874 4.288407 3.111551   
## 9808 9868 9879 9894 9956   
## 3.447520 3.447520 3.031236 4.166333 3.822165   
##   
## $coast  
## named numeric(0)  
##   
## $elev  
## 180 252 447 513 626 640 885 988   
## 4.152904 3.184484 3.500920 3.843741 3.184484 3.126059 3.131508 4.676631   
## 1048 1060 1117 1151 1154 1776 1782 1873   
## 3.027771 3.500920 3.051423 4.932546 3.234392 3.051423 4.152904 3.053811   
## 1933 2027 2251 2262 2351 2367 2569 2588   
## 4.521125 3.500920 3.843741 3.053811 3.843741 4.932546 4.676631 3.253447   
## 2633 2731 2850 2958 2960 3506 3667 3704   
## 3.051423 3.843741 3.395877 3.500920 3.253447 4.152904 3.051423 3.051423   
## 3727 3761 3797 3809 3828 3843 3877 3954   
## 3.149159 3.051423 4.932546 3.131508 4.152904 4.676631 3.234392 3.859409   
## 3980 4029 4030 4214 4388 4474 4586 4663   
## 3.859409 4.152904 4.152904 3.859409 3.184484 4.152904 4.521125 3.253447   
## 4713 4726 4799 4851 4948 5102 5103 5115   
## 3.234392 3.253447 3.051423 3.184484 3.126059 3.126059 3.253447 3.859409   
## 5149 5255 5271 5363 5494 5509 5543 5653   
## 3.843741 3.051423 3.253447 4.521125 3.053811 3.053811 3.184484 3.126059   
## 5729 5851 6012 6053 6176 6290 6296 6447   
## 3.843741 3.051423 4.152904 3.253447 3.843741 3.184484 3.395877 3.051423   
## 6558 6567 6787 6832 7046 7125 7159 7258   
## 3.051423 3.126059 4.521125 3.053811 3.131508 4.676631 3.051423 3.184484   
## 7275 7510 7631 7644 7948 8047 8076 8268   
## 3.051423 3.184484 3.843741 3.051423 3.053811 3.149159 3.053811 3.053811   
## 8756 8803 8810 8827 8866 9177 9236 9426   
## 3.149159 3.053811 3.149159 3.131508 3.149159 3.184484 3.051423 3.843741   
## 9711 9803 9805 9845 9864   
## 3.131508 3.859409 3.184484 3.500920 4.521125   
##   
## $ph  
## 183 277 640 988 1014 1117 1229 1281   
## 3.589503 -3.173630 -3.448802 -3.741204 -3.173630 -3.576069 3.589503 -3.145752   
## 1369 1558 1612 1689 1776 1933 2172 2222   
## 3.589503 4.000361 -3.215213 -3.173630 -3.576069 -4.255027 -3.161626 -3.215213   
## 2384 2395 2506 2569 2633 2850 2870 2967   
## -3.215213 -3.173630 -3.072000 -3.741204 -3.576069 -3.677618 -3.161626 -3.145752   
## 3036 3098 3156 3439 3523 3561 3667 3668   
## 4.000361 4.000361 3.589503 4.000361 -3.173630 -3.215213 -3.576069 4.000361   
## 3704 3727 3761 3797 3843 3883 3954 3980   
## -3.576069 -3.309123 -3.576069 -3.309123 -3.741204 3.589503 -3.927203 -3.927203   
## 4004 4076 4121 4190 4196 4214 4274 4309   
## -3.173630 -3.173630 -3.072000 3.589503 3.589503 -3.927203 3.589503 4.000361   
## 4315 4392 4586 4772 4799 4901 4948 5034   
## -3.161626 -3.072000 -4.255027 -3.161626 -3.576069 -3.215213 -3.448802 -3.215213   
## 5115 5138 5255 5275 5278 5317 5345 5363   
## -3.927203 4.000361 -3.576069 -3.161626 -3.487108 -3.173630 -3.072000 -4.255027   
## 5653 5697 5772 5851 5933 5972 5973 6086   
## -3.448802 -3.145752 3.589503 -3.576069 3.589503 4.000361 3.589503 3.589503   
## 6091 6115 6159 6175 6277 6296 6447 6452   
## -3.072000 3.589503 4.000361 3.589503 -3.161626 -3.677618 -3.576069 -3.161626   
## 6474 6556 6558 6567 6787 6856 7093 7125   
## -3.072000 3.589503 -3.576069 -3.448802 -3.927203 3.589503 -3.161626 -3.741204   
## 7159 7209 7217 7243 7275 7296 7345 7359   
## -3.576069 -3.173630 3.589503 -3.215213 -3.576069 -3.215213 4.000361 4.000361   
## 7512 7526 7623 7644 7794 7935 8047 8170   
## -3.173630 -3.072000 3.589503 -3.576069 -3.215213 3.589503 -3.309123 -3.072000   
## 8247 8321 8422 8496 8508 8584 8719 8756   
## -3.161626 -3.215213 3.589503 4.000361 3.589503 -3.173630 4.000361 -3.309123   
## 8810 8813 8850 8866 9013 9110 9236 9256   
## -3.309123 3.589503 -3.215213 -3.309123 -3.215213 -3.173630 -3.576069 -3.215213   
## 9505 9631 9651 9697 9746 9803 9864   
## 4.000361 -3.072000 4.000361 -3.145752 -3.215213 -3.927203 -4.255027   
##   
## $X01\_water  
## named numeric(0)  
##   
## $X02\_Barren  
## 163 180 383 446 447 468 513 525   
## 3.032123 7.618335 6.198083 3.032123 8.185764 3.231654 4.960657 7.475360   
## 688 741 963 1060 1069 1222 1261 1278   
## 3.032123 6.198083 6.198083 8.185764 6.198083 6.198083 7.475360 6.198083   
## 1447 1532 1782 1825 1836 1837 1862 2027   
## 50.168524 7.475360 7.618335 3.032123 7.475360 7.475360 7.475360 8.185764   
## 2089 2093 2201 2251 2351 2382 2597 2731   
## 5.813196 17.368476 3.032123 4.960657 4.960657 3.032123 3.032123 4.960657   
## 2734 2911 2958 3099 3122 3133 3194 3283   
## 7.475360 6.198083 8.185764 7.475360 3.032123 5.813196 6.198083 3.032123   
## 3506 3552 3636 3828 3925 3993 4019 4029   
## 7.618335 6.198083 5.813196 7.618335 3.231654 6.198083 3.231654 7.618335   
## 4030 4139 4246 4404 4472 4474 4629 4654   
## 7.618335 3.231654 50.168524 7.475360 3.032123 7.618335 6.198083 3.032123   
## 4776 5061 5149 5195 5480 5507 5655 5729   
## 3.231654 6.198083 4.960657 6.198083 3.032123 6.198083 7.475360 4.960657   
## 5730 5934 6012 6056 6071 6176 6286 6736   
## 5.813196 7.475360 7.618335 6.198083 3.032123 4.960657 3.032123 6.198083   
## 6745 6767 6859 6977 7015 7037 7262 7370   
## 3.032123 7.475360 7.475360 6.198083 7.475360 6.198083 6.198083 3.231654   
## 7446 7459 7558 7591 7631 7707 7853 8328   
## 3.231654 3.032123 7.475360 3.032123 4.960657 6.198083 7.475360 6.198083   
## 8503 9155 9197 9330 9370 9426 9713 9795   
## 6.198083 6.198083 6.198083 17.368476 17.368476 4.960657 6.198083 11.201237   
## 9845 9984   
## 8.185764 3.032123   
##   
## $X03\_LAc\_NHardwd  
## named numeric(0)  
##   
## $X05\_NAtl\_CoastPlain\_Hardwd  
## 120 139 140 148 182 206 319 349   
## 8.774050 3.796301 3.796301 8.774050 3.593645 8.774050 10.420068 10.420068   
## 383 467 525 542 608 710 741 880   
## 5.397427 3.796301 4.200372 8.029734 10.138043 8.774050 5.397427 4.076675   
## 937 963 978 1056 1069 1081 1102 1146   
## 9.934868 5.397427 8.774050 9.934868 5.397427 3.796301 4.076675 8.774050   
## 1222 1236 1261 1278 1371 1454 1487 1532   
## 5.397427 4.076675 4.200372 5.397427 8.774050 5.655847 8.774050 4.200372   
## 1635 1638 1666 1833 1836 1837 1862 1924   
## 3.195751 3.796301 8.029734 5.655847 4.200372 4.200372 4.200372 5.655847   
## 1929 1993 2193 2223 2244 2307 2311 2428   
## 9.934868 3.796301 9.934868 10.138043 8.029734 10.138043 5.655847 5.655847   
## 2461 2469 2515 2734 2746 2750 2828 2853   
## 4.076675 3.796301 10.138043 4.200372 8.029734 9.934868 8.029734 3.195751   
## 2911 2927 3052 3099 3179 3186 3194 3278   
## 5.397427 4.076675 3.593645 4.200372 3.593645 10.138043 5.397427 4.076675   
## 3357 3374 3376 3430 3552 3574 3698 3993   
## 3.195751 3.195751 4.076675 9.934868 5.397427 3.796301 5.655847 5.397427   
## 4011 4032 4200 4207 4226 4263 4330 4348   
## 3.796301 9.934868 8.029734 3.593645 8.774050 8.774050 3.195751 3.796301   
## 4404 4450 4454 4457 4477 4581 4597 4629   
## 4.200372 8.029734 9.934868 3.796301 9.934868 8.029734 8.029734 5.397427   
## 4667 4766 4822 4855 4873 4964 5036 5061   
## 8.774050 3.593645 3.195751 5.655847 8.029734 9.934868 3.796301 5.397427   
## 5064 5098 5195 5390 5429 5434 5507 5556   
## 8.029734 4.076675 5.397427 3.593645 3.796301 10.138043 5.397427 5.655847   
## 5655 5830 5885 5916 5934 6056 6116 6167   
## 4.200372 3.796301 3.195751 3.195751 4.200372 5.397427 4.076675 3.195751   
## 6169 6232 6332 6436 6488 6574 6736 6767   
## 8.029734 3.593645 4.076675 8.029734 3.796301 3.593645 5.397427 4.200372   
## 6859 6962 6977 7015 7037 7057 7113 7154   
## 4.200372 10.138043 5.397427 4.200372 5.397427 3.796301 3.796301 3.796301   
## 7225 7240 7262 7452 7499 7530 7558 7589   
## 3.593645 3.796301 5.397427 4.076675 8.774050 4.076675 4.200372 3.796301   
## 7595 7707 7715 7758 7782 7814 7837 7853   
## 9.934868 5.397427 9.934868 3.796301 8.774050 4.076675 3.597291 4.200372   
## 7856 7933 7938 7974 8177 8244 8323 8328   
## 8.029734 8.029734 3.195751 5.655847 9.934868 3.593645 8.774050 5.397427   
## 8377 8446 8470 8480 8503 8583 8691 8757   
## 3.593645 3.796301 3.593645 4.076675 5.397427 8.774050 9.934868 10.420068   
## 8869 8870 8887 8928 9025 9035 9140 9155   
## 4.076675 3.796301 10.420068 8.029734 10.138043 3.195751 9.934868 5.397427   
## 9162 9192 9197 9369 9466 9561 9573 9577   
## 3.796301 8.029734 5.397427 3.593645 10.138043 3.796301 4.076675 4.076675   
## 9606 9633 9713 9896 9974 9992   
## 10.420068 5.655847 5.397427 9.934868 8.029734 8.029734   
##   
## $X07\_LAc\_NPine.Oak  
## 313 339 344 384 386 395 406 598   
## 7.597612 3.427035 3.068568 3.448197 4.422096 7.597612 7.859935 4.371312   
## 602 608 617 639 707 740 788 790   
## 7.859935 4.828173 3.011632 6.070829 7.597612 4.422096 3.068568 3.448197   
## 834 875 893 940 944 1009 1045 1123   
## 5.898257 5.898257 3.011632 4.422096 5.898257 3.448197 3.068568 6.070829   
## 1167 1217 1246 1252 1298 1386 1426 1493   
## 4.371312 7.597612 3.448197 3.011632 3.068568 7.000154 5.898257 7.799414   
## 1501 1526 1595 1758 1845 1866 1885 1888   
## 3.068568 7.597612 6.070829 3.448197 3.068568 7.859935 5.898257 3.427035   
## 1979 1994 2129 2158 2223 2245 2307 2368   
## 6.070829 4.422096 4.422096 7.597612 4.828173 3.448197 4.828173 3.068568   
## 2490 2515 2538 2586 2605 2612 2660 2670   
## 7.597612 4.828173 5.898257 6.070829 3.448197 3.448197 4.422096 3.448197   
## 2861 2879 2891 2895 2902 3063 3118 3144   
## 6.070829 3.068568 3.011632 3.068568 3.448197 4.422096 6.070829 3.756650   
## 3177 3184 3186 3321 3352 3354 3355 3433   
## 5.898257 7.597612 4.828173 5.898257 4.422096 4.422096 4.371312 7.000154   
## 3513 3682 3741 3771 3782 3866 3927 3948   
## 5.898257 3.011632 3.068568 3.011632 6.070829 5.898257 5.898257 3.011632   
## 3969 3984 4018 4022 4096 4120 4130 4151   
## 3.011632 3.011632 5.898257 3.068568 4.422096 3.011632 3.011632 3.011632   
## 4223 4249 4297 4311 4370 4397 4418 4504   
## 7.799414 3.011632 3.068568 7.597612 4.422096 7.859935 7.597612 7.859935   
## 4605 4619 4633 4732 4738 4755 4756 4769   
## 7.597612 3.756650 3.448197 7.597612 5.898257 5.898257 7.597612 5.898257   
## 4793 5032 5048 5071 5126 5143 5227 5315   
## 3.011632 3.011632 7.597612 4.422096 3.011632 3.068568 7.597612 4.422096   
## 5369 5408 5409 5434 5472 5473 5474 5513   
## 3.011632 5.898257 5.898257 4.828173 3.448197 7.597612 4.371312 6.070829   
## 5526 5559 5576 5610 5617 5640 5664 5727   
## 7.597612 5.898257 3.068568 4.422096 3.011632 3.011632 4.422096 4.371312   
## 5746 5757 5778 5809 5854 5918 5974 5994   
## 7.597612 3.011632 4.422096 3.756650 7.597612 3.068568 3.448197 4.371312   
## 6040 6089 6093 6201 6219 6313 6390 6432   
## 3.756650 7.597612 4.422096 3.068568 7.597612 6.070829 7.597612 6.070829   
## 6524 6541 6668 6709 6721 6789 6866 6889   
## 3.011632 3.011632 3.011632 3.068568 5.898257 5.898257 7.859935 3.011632   
## 6920 6962 7049 7297 7363 7375 7376 7391   
## 3.068568 4.828173 7.799414 7.799414 5.898257 3.448197 7.597612 3.427035   
## 7407 7412 7501 7664 7669 7684 7685 7696   
## 7.859935 3.448197 3.756650 7.859935 4.371312 3.448197 7.859935 4.422096   
## 7728 7731 7761 7815 7825 7826 7854 7905   
## 5.898257 7.597612 7.597612 5.898257 3.011632 5.898257 4.422096 5.898257   
## 8014 8016 8070 8090 8125 8138 8182 8198   
## 3.011632 3.448197 3.011632 3.011632 3.427035 3.427035 5.898257 7.597612   
## 8367 8568 8681 8697 8753 8851 8879 8896   
## 5.898257 3.011632 7.859935 7.859935 3.756650 7.000154 5.898257 3.448197   
## 8910 8961 8964 9000 9009 9011 9025 9127   
## 5.898257 3.448197 6.070829 4.422096 5.898257 4.422096 4.828173 3.427035   
## 9174 9189 9310 9466 9492 9494 9569 9623   
## 6.070829 5.898257 3.011632 4.828173 5.898257 7.597612 5.898257 3.011632   
## 9703 9756 9808 9863 9868 9926 9931 9956   
## 3.448197 3.448197 6.070829 3.448197 6.070829 5.898257 5.898257 7.000154   
##   
## $X08\_LAc\_Pine.Hemlock.Hardwd  
## 271 377 858 959 1074 1294 1533 1552   
## 3.032554 3.301183 3.341907 3.201645 3.341907 3.301183 3.823647 3.823647   
## 1654 2111 2140 2291 2763 2874 2883 3111   
## 3.032554 3.023890 3.622254 3.023890 3.301183 3.032554 3.301183 3.023890   
## 3269 3356 3599 3700 3836 3847 4001 4038   
## 3.023890 3.032554 3.023890 3.731223 3.023890 3.201645 3.622254 3.032554   
## 4362 4369 4396 4607 4628 4699 4727 5210   
## 3.032554 3.471571 3.023890 3.823647 3.023890 3.201645 3.731223 3.301183   
## 5248 5312 5634 5788 5879 6421 6533 6737   
## 3.301183 3.032554 3.032554 3.201645 3.023890 3.032554 3.301183 3.023890   
## 6874 6933 6940 7025 7132 7165 7418 7747   
## 3.023890 3.622254 3.301183 3.023890 3.032554 3.023890 3.471571 3.341907   
## 7944 8506 8548 8565 8649 8880 8882 8965   
## 3.023890 3.023890 3.622254 3.341907 3.023890 3.301183 3.023890 3.023890   
## 9503 9516 9616 9635 9725   
## 3.731223 3.032554 3.471571 3.023890 3.201645   
##   
## $X09\_CAp\_Dry\_Oak.Pine  
## 143 161 168 195 292 322 351 372   
## 3.122052 3.178831 3.257596 3.120521 3.107452 3.199835 3.120521 3.178831   
## 379 411 422 620 731 745 908 922   
## 3.122052 3.199835 3.127289 3.199835 3.127289 3.008198 3.011369 3.008198   
## 1005 1017 1039 1094 1274 1303 1360 1370   
## 3.178831 3.008198 3.008198 3.178831 3.011369 3.199835 3.178831 3.127289   
## 1434 1476 1547 1557 1616 1735 1752 1797   
## 3.107452 3.011369 3.120521 3.011369 3.120521 3.107452 3.178831 3.122052   
## 1821 1858 1967 2025 2108 2139 2165 2297   
## 3.257596 3.011369 3.178831 3.122052 3.011369 3.178831 3.107452 3.008198   
## 2456 2525 2551 2584 2663 2709 2736 2762   
## 3.257596 3.011369 3.008198 3.127289 3.107452 3.008198 3.107452 3.011369   
## 2771 2796 2943 2979 3014 3091 3112 3121   
## 3.127289 3.107452 3.199835 3.011369 3.122052 3.011369 3.107452 3.257596   
## 3132 3232 3266 3313 3326 3359 3481 3519   
## 3.122052 3.199835 3.122052 3.122052 3.120521 3.127289 3.178831 3.008198   
## 3536 3594 3596 3632 3719 3759 3787 3807   
## 3.008198 3.122052 3.127289 3.127289 3.008198 3.048695 3.008198 3.120521   
## 3861 3926 3944 4072 4176 4343 4364 4367   
## 3.127289 3.122052 3.107452 3.178831 3.048695 3.178831 3.178831 3.008198   
## 4377 4383 4401 4422 4490 4508 4544 4559   
## 3.127289 3.107452 3.107452 3.107452 3.178831 3.011369 3.257596 3.257596   
## 4645 4675 4680 4747 4814 4825 4865 4886   
## 3.127289 3.107452 3.011369 3.178831 3.122052 3.107452 3.048695 3.048695   
## 4890 4910 4983 4999 5021 5104 5298 5356   
## 3.107452 3.122052 3.122052 3.011369 3.107452 3.199835 3.107452 3.107452   
## 5393 5458 5475 5483 5500 5589 5590 5633   
## 3.011369 3.199835 3.199835 3.048695 3.107452 3.199835 3.107452 3.178831   
## 5671 5693 5895 5930 5961 6009 6011 6069   
## 3.122052 3.107452 3.122052 3.120521 3.122052 3.048695 3.199835 3.199835   
## 6102 6110 6166 6226 6269 6278 6381 6396   
## 3.178831 3.120521 3.107452 3.008198 3.011369 3.120521 3.008198 3.011369   
## 6472 6535 6575 6581 6692 6717 6835 6956   
## 3.122052 3.257596 3.011369 3.011369 3.011369 3.257596 3.011369 3.107452   
## 7013 7034 7170 7194 7292 7302 7319 7420   
## 3.107452 3.199835 3.199835 3.107452 3.178831 3.120521 3.048695 3.178831   
## 7531 7564 7607 7666 7679 7705 7749 7832   
## 3.120521 3.127289 3.008198 3.122052 3.127289 3.107452 3.048695 3.257596   
## 7875 7936 8018 8048 8079 8084 8100 8133   
## 3.199835 3.127289 3.048695 3.008198 3.120521 3.011369 3.008198 3.107452   
## 8137 8140 8154 8162 8171 8193 8259 8405   
## 3.178831 3.199835 3.011369 3.107452 3.011369 3.127289 3.011369 3.122052   
## 8488 8562 8594 8624 8660 8675 8718 8786   
## 3.008198 3.257596 3.178831 3.107452 3.199835 3.257596 3.257596 3.178831   
## 8806 8854 8856 8912 8922 8954 9087 9307   
## 3.127289 3.107452 3.008198 3.178831 3.178831 3.011369 3.127289 3.127289   
## 9316 9403 9476 9533 9543 9663 9677 9736   
## 3.107452 3.178831 3.127289 3.011369 3.107452 3.011369 3.011369 3.008198   
## 9749 9767 9791 9843   
## 3.120521 3.011369 3.011369 3.120521   
##   
## $X10\_Ap\_Hemlock.N\_Hardwood  
## named numeric(0)  
##   
## $X11\_Ac\_Low.Elev\_Spruce.Fir.Hardwood  
## 201 243 253 255 275 280 392 434   
## 4.409938 6.362367 3.161637 3.125936 4.409938 3.086578 4.443154 4.520291   
## 457 540 600 614 676 730 756 827   
## 4.520291 3.246447 3.125936 4.380608 4.443154 3.125936 3.246447 4.520291   
## 838 901 925 1076 1104 1122 1209 1241   
## 3.246447 4.409938 4.380608 3.125936 3.182673 3.277762 6.362367 5.387356   
## 1344 1390 1404 1497 1565 1576 1601 1611   
## 3.161637 5.887201 4.443154 4.409938 4.615162 3.125936 6.362367 6.362367   
## 1664 1679 1692 1720 1814 1830 1857 1895   
## 3.161637 3.161637 5.733946 3.125936 4.615162 6.362367 4.443154 3.256189   
## 1956 2107 2143 2226 2235 2286 2454 2474   
## 5.887201 3.161637 4.380608 6.362367 4.409938 6.362367 4.409938 4.380608   
## 2485 2511 2542 2580 2604 2655 2678 2699   
## 4.615162 5.733946 4.409938 3.636445 5.387356 4.886416 4.615162 4.409938   
## 2776 2842 2929 2935 2956 2992 2996 2998   
## 3.256189 6.362367 3.636445 3.125936 4.886416 4.443154 3.161637 3.256189   
## 3025 3054 3074 3114 3134 3173 3189 3239   
## 5.387356 6.362367 5.733946 3.277762 3.161637 4.409938 3.246447 3.636445   
## 3257 3267 3275 3284 3310 3314 3327 3400   
## 3.246447 6.362367 6.362367 4.409938 6.362367 3.277762 5.887201 3.125936   
## 3418 3426 3450 3499 3507 3566 3568 3585   
## 6.362367 3.572165 3.161637 5.887201 4.615162 5.387356 5.887201 3.636445   
## 3638 3699 3751 3762 3814 3848 3909 3916   
## 4.443154 6.362367 3.636445 6.362367 3.125936 3.256189 6.362367 4.615162   
## 3930 3962 3991 4014 4090 4093 4135 4211   
## 3.086578 3.636445 3.636445 4.443154 3.246447 3.086578 3.182673 3.125936   
## 4260 4389 4403 4502 4556 4582 4592 4595   
## 4.443154 4.409938 3.256189 4.886416 3.125936 4.886416 3.086578 3.161637   
## 4704 4718 4719 4753 4849 4902 5080 5089   
## 4.615162 3.086578 4.409938 6.362367 6.362367 5.887201 3.246447 3.161637   
## 5161 5233 5307 5334 5446 5943 5954 6048   
## 4.520291 4.443154 6.362367 3.246447 6.362367 6.362367 3.256189 3.277762   
## 6080 6088 6195 6204 6205 6212 6217 6342   
## 3.086578 6.362367 5.387356 3.636445 4.886416 3.256189 5.887201 6.362367   
## 6399 6427 6433 6526 6671 6757 6809 6812   
## 4.380608 5.887201 4.409938 4.886416 4.409938 4.615162 4.520291 5.387356   
## 6814 6885 6888 6927 6983 7060 7167 7183   
## 3.182673 6.362367 3.161637 5.887201 3.125936 4.380608 4.409938 5.733946   
## 7281 7398 7562 7573 7676 7880 7897 7918   
## 3.228968 5.733946 5.733946 6.362367 3.246447 3.125936 5.387356 5.887201   
## 7991 8005 8009 8012 8053 8103 8124 8176   
## 6.362367 3.256189 3.246447 5.387356 4.615162 5.733946 3.161637 3.636445   
## 8180 8191 8235 8266 8278 8295 8296 8305   
## 4.998502 3.161637 4.409938 3.246447 6.362367 4.443154 4.443154 3.256189   
## 8315 8327 8365 8401 8402 8444 8455 8502   
## 4.409938 4.886416 3.636445 3.125936 6.378118 4.409938 6.362367 3.277762   
## 8519 8547 8556 8619 8620 8629 8636 8647   
## 3.086578 5.887201 3.256189 3.161637 4.443154 3.182673 4.409938 3.256189   
## 8650 8741 8805 8820 8852 8909 8915 8917   
## 5.887201 6.362367 4.409938 4.443154 4.409938 6.362367 3.636445 6.362367   
## 9042 9050 9228 9284 9297 9349 9453 9465   
## 5.733946 3.125936 4.380608 3.246447 4.443154 3.572165 3.256189 3.182673   
## 9474 9509 9521 9536 9592 9661 9682 9773   
## 6.362367 4.886416 6.362367 4.615162 3.572165 6.362367 5.887201 5.387356   
## 9883 9889 9890 9957 9999   
## 3.161637 4.443154 5.887201 5.733946 3.636445   
##   
## $X12\_AcAp\_Montane\_Spruce.Fir  
## 180 252 277 326 389 425 447 513   
## 9.096943 4.464112 3.705689 4.721123 4.721123 4.721123 5.510156 6.055513   
## 626 640 650 667 694 716 734 755   
## 4.464112 4.375264 4.721123 4.721123 3.223112 4.706077 3.336293 4.721123   
## 793 820 855 885 896 960 988 1014   
## 5.430103 3.264524 4.721123 4.510356 4.258508 4.721123 9.790792 3.705689   
## 1046 1048 1052 1060 1061 1117 1151 1154   
## 3.223112 4.443414 3.378187 5.510156 4.721123 4.106218 9.776566 4.200697   
## 1192 1243 1249 1281 1334 1335 1385 1411   
## 4.721123 3.378187 4.927001 4.505765 5.430103 3.378187 3.521810 3.378187   
## 1430 1443 1462 1507 1548 1612 1653 1689   
## 3.378187 4.721123 3.378187 3.378187 4.721123 4.891771 3.111576 3.705689   
## 1704 1737 1746 1776 1781 1782 1800 1820   
## 4.721123 4.721123 3.223112 4.106218 4.721123 9.096943 3.264524 3.247377   
## 1873 1933 1941 1998 2010 2027 2059 2089   
## 3.637166 10.483804 4.510356 3.264524 3.111576 5.510156 3.378187 3.023947   
## 2142 2169 2172 2194 2222 2251 2262 2277   
## 3.521810 4.721123 3.403369 4.706077 4.891771 6.055513 3.637166 4.721123   
## 2341 2351 2367 2384 2395 2413 2465 2494   
## 3.378187 6.055513 9.776566 4.891771 3.705689 3.247377 4.706077 4.721123   
## 2532 2563 2569 2588 2633 2728 2731 2817   
## 3.378187 3.378187 9.790792 4.469807 4.106218 4.706077 6.055513 3.378187   
## 2834 2850 2870 2881 2916 2958 2960 2967   
## 3.378187 5.073454 3.403369 4.721123 3.378187 5.510156 4.469807 4.505765   
## 2991 3006 3085 3133 3171 3219 3225 3258   
## 3.378187 3.378187 3.938296 3.023947 3.247377 4.706077 4.721123 4.706077   
## 3288 3330 3363 3391 3396 3405 3456 3486   
## 3.223112 3.378187 4.721123 4.721123 4.721123 4.721123 3.938296 4.706077   
## 3506 3523 3561 3571 3636 3667 3671 3704   
## 9.096943 3.705689 4.891771 4.721123 3.023947 4.106218 3.223112 4.106218   
## 3727 3761 3797 3809 3828 3843 3877 3951   
## 5.423042 4.106218 9.776566 4.510356 9.096943 9.790792 4.200697 3.111576   
## 3954 3980 4004 4029 4030 4034 4076 4102   
## 7.642663 7.642663 3.705689 9.096943 9.096943 3.378187 3.705689 4.721123   
## 4110 4129 4147 4214 4228 4290 4315 4374   
## 4.706077 4.721123 3.521810 7.642663 4.721123 4.721123 3.403369 4.721123   
## 4388 4399 4447 4468 4474 4488 4538 4548   
## 4.464112 3.223112 4.721123 3.336293 9.096943 3.111576 3.298249 4.721123   
## 4552 4586 4646 4653 4655 4663 4668 4698   
## 3.378187 10.483804 4.706077 4.721123 4.721123 4.469807 4.721123 4.721123   
## 4713 4726 4730 4772 4799 4848 4851 4901   
## 4.200697 4.469807 3.223112 3.403369 4.106218 3.378187 4.464112 4.891771   
## 4948 5004 5034 5086 5095 5103 5115 5120   
## 4.375264 3.223112 4.891771 3.298249 4.258508 4.469807 7.642663 4.721123   
## 5139 5149 5163 5221 5255 5271 5275 5278   
## 4.721123 6.055513 5.430103 5.430103 4.106218 4.469807 3.403369 5.513627   
## 5317 5363 5453 5494 5509 5543 5653 5661   
## 3.705689 10.483804 3.378187 3.637166 3.637166 4.464112 4.375264 3.223112   
## 5663 5697 5729 5730 5813 5816 5817 5822   
## 3.247377 4.505765 6.055513 3.023947 3.938296 4.706077 4.721123 3.378187   
## 5840 5851 5888 5962 5970 5991 6012 6029   
## 3.938296 4.106218 4.721123 4.721123 4.721123 3.298249 9.096943 4.721123   
## 6053 6176 6182 6258 6277 6290 6292 6296   
## 4.469807 6.055513 3.298249 4.927001 3.403369 4.464112 5.430103 5.073454   
## 6326 6376 6398 6413 6441 6447 6452 6508   
## 3.378187 3.298249 4.721123 3.264524 4.721123 4.106218 3.403369 4.721123   
## 6558 6560 6567 6603 6732 6746 6775 6787   
## 4.106218 4.706077 4.375264 3.378187 4.706077 4.721123 3.247377 10.483804   
## 6832 6844 6850 7004 7005 7033 7040 7042   
## 3.637166 3.521810 4.721123 4.706077 4.721123 4.721123 3.264524 4.721123   
## 7046 7066 7084 7093 7125 7143 7159 7198   
## 4.510356 3.378187 4.721123 3.403369 9.790792 5.430103 4.106218 3.378187   
## 7209 7213 7243 7245 7258 7271 7275 7296   
## 3.705689 3.521810 4.891771 4.721123 4.464112 4.706077 4.106218 4.891771   
## 7393 7400 7422 7443 7510 7512 7540 7631   
## 4.721123 4.721123 3.298249 3.378187 4.464112 3.705689 3.378187 6.055513   
## 7644 7648 7760 7789 7794 7932 7948 7960   
## 4.106218 3.247377 3.378187 3.521810 4.891771 3.378187 3.637166 4.721123   
## 7963 8047 8076 8082 8095 8144 8161 8190   
## 4.721123 5.423042 3.637166 3.378187 4.721123 3.378187 3.938296 4.721123   
## 8199 8247 8268 8321 8584 8609 8645 8743   
## 4.721123 3.403369 3.637166 4.891771 3.705689 3.378187 3.247377 3.264524   
## 8756 8776 8791 8803 8810 8827 8842 8850   
## 5.423042 4.706077 3.223112 3.637166 5.423042 4.510356 4.706077 4.891771   
## 8863 8866 8885 8923 8937 8939 8958 8971   
## 4.721123 5.423042 4.721123 3.336293 3.378187 4.721123 3.521810 4.721123   
## 9013 9094 9110 9114 9168 9177 9210 9211   
## 4.891771 3.378187 3.705689 3.378187 3.111576 4.464112 3.264524 4.721123   
## 9230 9236 9256 9303 9338 9415 9426 9450   
## 4.721123 4.106218 4.891771 4.721123 3.938296 3.336293 6.055513 4.721123   
## 9478 9508 9544 9549 9697 9711 9726 9746   
## 4.927001 3.264524 3.247377 3.378187 4.505765 4.510356 3.378187 4.891771   
## 9770 9787 9803 9805 9845 9864 9911 9929   
## 3.247377 4.721123 7.642663 4.464112 5.510156 10.483804 5.430103 3.223112   
## 9994   
## 4.258508   
##   
## $X13\_CentAp\_Pine.Oak\_Rocky\_Wd  
## 76 80 94 95 96 98 100 102   
## 3.485489 3.485489 3.856261 5.077236 3.554708 3.216669 4.272988 6.939121   
## 104 105 108 124 125 137 151 152   
## 3.554708 3.745126 5.077236 3.856261 3.856261 3.485489 5.077236 5.077236   
## 155 177 190 194 221 347 358 397   
## 3.856261 3.485489 5.077236 3.856261 3.554708 3.554708 4.639478 3.554708   
## 410 438 534 539 587 633 645 679   
## 3.745126 5.077236 3.856261 6.939121 3.554708 3.745126 3.554708 3.216669   
## 743 770 782 839 993 1096 1098 1195   
## 3.216669 3.856261 6.939121 11.336385 3.745126 3.554708 3.485489 5.077236   
## 1299 1469 1531 1578 1640 1677 1823 1832   
## 3.485489 3.485489 5.077236 3.656602 3.656602 3.554708 5.077236 11.336385   
## 1896 1931 2021 2024 2082 2164 2175 2265   
## 3.856261 3.856261 6.939121 5.077236 5.077236 11.336385 3.554708 3.745126   
## 2295 2298 2386 2445 2500 2528 2566 2751   
## 3.745126 4.639478 3.745126 11.336385 3.554708 5.077236 3.554708 3.656602   
## 2767 2794 2814 2845 2849 2859 2889 2937   
## 5.077236 3.554708 3.856261 3.216669 3.856261 4.639478 4.639478 3.554708   
## 2986 3071 3117 3125 3137 3139 3140 3228   
## 3.856261 5.077236 3.745126 6.939121 3.656602 4.639478 3.485489 4.639478   
## 3240 3285 3294 3342 3385 3399 3497 3631   
## 3.856261 4.639478 3.745126 3.485489 3.856261 6.939121 6.939121 3.656602   
## 3691 3693 3757 3841 3858 3931 3950 4099   
## 3.656602 4.272988 3.656602 3.216669 3.656602 11.336385 11.336385 3.216669   
## 4231 4258 4279 4375 4430 4432 4453 4506   
## 4.272988 6.939121 5.077236 6.939121 5.077236 11.336385 3.216669 5.077236   
## 4691 4884 4960 4997 5001 5012 5018 5096   
## 4.272988 4.272988 3.554708 5.077236 3.656602 3.216669 3.216669 3.554708   
## 5181 5217 5273 5322 5329 5352 5461 5516   
## 11.336385 3.554708 5.077236 4.272988 3.745126 5.077236 4.639478 4.272988   
## 5562 5682 5731 5751 5841 5855 5866 5959   
## 3.745126 3.216669 3.656602 3.485489 3.656602 11.336385 6.939121 3.554708   
## 5964 6015 6054 6083 6117 6160 6178 6192   
## 3.745126 4.639478 3.554708 3.856261 3.745126 3.216669 5.077236 11.336385   
## 6222 6229 6255 6334 6358 6371 6408 6449   
## 3.554708 3.856261 3.554708 5.077236 3.554708 3.745126 11.336385 5.077236   
## 6495 6579 6666 6713 6762 6781 6797 6886   
## 11.336385 3.554708 3.856261 3.856261 3.745126 3.856261 11.336385 3.216669   
## 6906 6909 6961 6963 6991 7036 7041 7078   
## 5.077236 3.745126 3.485489 3.856261 4.272988 3.656602 4.639478 3.745126   
## 7351 7360 7384 7395 7405 7491 7628 7828   
## 3.656602 3.216669 3.856261 3.745126 3.554708 11.336385 3.745126 3.216669   
## 7915 8021 8044 8064 8105 8106 8226 8260   
## 3.485489 3.216669 3.216669 3.554708 3.656602 11.336385 3.485489 3.745126   
## 8338 8376 8651 8655 8657 8750 8831 8918   
## 4.639478 3.554708 6.939121 3.745126 3.216669 3.216669 3.656602 3.554708   
## 8930 8977 9028 9063 9064 9153 9367 9427   
## 3.554708 5.077236 5.077236 3.856261 4.639478 3.745126 4.639478 4.272988   
## 9446 9654 9666 9669 9704 9785 9877 9995   
## 3.856261 3.745126 5.077236 3.856261 11.336385 4.272988 3.554708 11.336385   
## 9996   
## 3.554708   
##   
## $X14\_NAtl\_Coast\_Plain\_Maritime  
## 239 262 265 336 732 749 997 1050   
## 12.142958 3.513742 3.513742 12.142958 12.142958 12.142958 12.142958 12.142958   
## 1051 1267 1314 1316 1450 1538 1604 1791   
## 3.513742 12.142958 12.142958 12.142958 12.142958 12.142958 3.513742 12.142958   
## 2019 2077 2242 2332 2339 2434 2449 2458   
## 12.142958 12.142958 12.142958 12.142958 12.142958 12.142958 12.142958 3.513742   
## 2591 2666 2737 2744 2755 2765 2917 2959   
## 12.142958 12.142958 3.513742 12.142958 12.142958 12.142958 12.142958 12.142958   
## 3344 3468 3527 3533 3664 3744 3778 3960   
## 3.513742 12.142958 12.142958 12.142958 12.142958 3.513742 12.142958 3.513742   
## 3966 4122 4312 4333 4444 4478 4594 4742   
## 12.142958 12.142958 12.142958 3.513742 12.142958 12.142958 12.142958 12.142958   
## 4759 4952 5169 5262 5328 5432 5659 5844   
## 12.142958 12.142958 12.142958 12.142958 12.142958 3.513742 3.513742 12.142958   
## 6121 6139 6327 6695 6904 7149 7440 7464   
## 3.513742 12.142958 3.513742 12.142958 3.513742 12.142958 12.142958 12.142958   
## 7475 7810 7912 8271 8285 8303 8437 8504   
## 12.142958 12.142958 12.142958 3.513742 12.142958 12.142958 12.142958 12.142958   
## 8555 8644 8942 9008 9472 9547 9642 9728   
## 3.513742 12.142958 12.142958 12.142958 3.513742 12.142958 3.513742 3.513742   
## 9762 9795 9886   
## 12.142958 23.191733 12.142958   
##   
## $X15\_NAtl\_Coast\_Plain\_Dune  
## 163 259 446 507 530 535 546 688   
## 3.828210 4.756375 3.828210 3.665748 4.756375 4.756375 4.756375 3.828210   
## 706 735 769 818 879 880 891 1064   
## 4.756375 3.724601 4.756375 3.665748 16.840177 5.488299 4.756375 4.756375   
## 1102 1236 1496 1551 1609 1825 1877 1913   
## 5.488299 5.488299 4.756375 16.840177 4.756375 3.828210 3.724601 4.971929   
## 1938 2031 2067 2074 2093 2201 2231 2313   
## 16.840177 4.756375 3.665748 16.840177 8.202613 3.828210 4.756375 3.724601   
## 2378 2382 2461 2502 2597 2927 2964 3020   
## 3.665748 3.828210 5.488299 4.971929 3.828210 5.488299 3.724601 3.665748   
## 3122 3159 3278 3282 3283 3376 3466 3475   
## 3.828210 4.756375 5.488299 4.756375 3.828210 5.488299 4.971929 3.665748   
## 3553 3798 3864 3871 3936 4016 4023 4039   
## 4.971929 4.756375 4.971929 4.756375 4.756375 3.665748 4.756375 16.840177   
## 4146 4150 4237 4286 4304 4334 4354 4379   
## 16.840177 4.971929 4.971929 4.971929 4.971929 4.971929 4.756375 3.665748   
## 4472 4533 4551 4557 4654 4744 4771 4783   
## 3.828210 4.756375 3.724601 3.665748 3.828210 3.724601 3.665748 3.665748   
## 4807 4853 4986 5098 5159 5167 5226 5423   
## 4.756375 4.756375 4.756375 5.488299 4.756375 16.840177 16.840177 4.756375   
## 5480 5626 5677 5686 5725 5929 5963 6020   
## 3.828210 16.840177 3.724601 4.756375 3.724601 4.756375 3.724601 4.756375   
## 6071 6101 6116 6138 6286 6332 6387 6501   
## 3.828210 3.665748 5.488299 16.840177 3.828210 5.488299 4.971929 16.840177   
## 6745 6867 6992 7126 7371 7452 7459 7530   
## 3.828210 4.756375 16.840177 4.971929 3.724601 5.488299 3.828210 5.488299   
## 7538 7545 7549 7591 7629 7635 7718 7814   
## 3.665748 4.971929 16.840177 3.828210 4.756375 3.724601 16.840177 5.488299   
## 7837 7917 7939 7976 8037 8228 8349 8369   
## 26.288244 4.971929 16.840177 4.756375 4.756375 4.756375 3.665748 4.756375   
## 8374 8480 8522 8633 8730 8812 8822 8853   
## 16.840177 5.488299 4.756375 3.665748 3.665748 16.840177 4.971929 4.756375   
## 8869 9065 9090 9294 9327 9330 9370 9462   
## 5.488299 4.971929 4.756375 3.724601 4.971929 8.202613 8.202613 16.840177   
## 9469 9484 9501 9573 9577 9602 9738 9821   
## 3.665748 4.756375 4.971929 5.488299 5.488299 4.971929 16.840177 4.971929   
## 9881 9925 9984   
## 4.756375 4.756375 3.828210   
##   
## $X16\_CentIntAp\_FloodplainSys  
## 1 2 3 146 147 440 477 485   
## 4.464251 6.241242 6.241242 5.130176 5.130176 5.340199 4.939850 4.024069   
## 502 521 651 678 690 744 805 813   
## 6.402402 6.841088 5.130176 4.024069 5.658195 6.241242 6.241242 5.658195   
## 831 853 861 872 939 992 1088 1163   
## 6.841088 6.241242 5.130176 6.241242 4.024069 7.504013 4.939850 6.402402   
## 1196 1269 1287 1376 1379 1402 1416 1417   
## 6.241242 6.241242 6.841088 6.241242 4.464251 6.841088 5.130176 4.464251   
## 1504 1556 1684 1762 1778 1802 1808 1818   
## 7.504013 5.130176 6.241242 5.658195 5.130176 6.402402 6.841088 6.841088   
## 1834 1865 1889 1904 1975 2005 2042 2049   
## 4.024069 6.241242 7.504013 6.241242 5.130176 4.464251 6.241242 6.402402   
## 2062 2176 2220 2243 2249 2263 2433 2438   
## 5.658195 4.024069 6.841088 6.241242 6.241242 5.130176 6.241242 6.241242   
## 2498 2549 2556 2575 2596 2649 2661 2706   
## 6.841088 4.939850 6.241242 5.130176 5.130176 7.504013 4.024069 6.841088   
## 2726 2885 2901 2970 2981 2997 3043 3092   
## 4.024069 4.939850 5.130176 6.241242 6.402402 6.402402 4.024069 5.340199   
## 3113 3222 3230 3259 3298 3324 3373 3387   
## 5.130176 6.241242 4.024069 4.464251 6.402402 6.402402 6.241242 6.241242   
## 3532 3563 3606 3608 3642 3717 3854 3923   
## 7.504013 5.658195 6.241242 5.130176 6.841088 6.241242 5.130176 6.241242   
## 4044 4172 4181 4201 4227 4235 4291 4294   
## 4.024069 7.504013 4.024069 4.464251 6.241242 6.841088 5.130176 6.841088   
## 4340 4405 4419 4436 4641 4673 4701 4717   
## 6.841088 4.024069 6.841088 6.402402 4.024069 4.464251 6.241242 5.130176   
## 4774 4811 4877 4928 5063 5150 5157 5175   
## 4.024069 4.464251 6.241242 6.841088 5.658195 6.241242 6.241242 6.841088   
## 5218 5332 5413 5441 5450 5515 5536 5592   
## 5.658195 6.241242 7.504013 6.241242 7.504013 7.504013 5.658195 6.241242   
## 5599 5606 5639 5669 5870 5905 5985 6026   
## 6.241242 6.402402 6.241242 5.340199 6.241242 5.658195 6.402402 4.464251   
## 6058 6062 6070 6095 6185 6235 6291 6297   
## 6.241242 6.241242 7.504013 5.658195 4.939850 6.241242 6.241242 4.939850   
## 6340 6345 6347 6349 6354 6457 6487 6620   
## 5.340199 5.340199 7.504013 6.241242 5.658195 5.340199 5.658195 6.241242   
## 6628 6629 6644 6684 6689 6749 6783 6945   
## 4.024069 5.130176 4.024069 5.340199 4.464251 4.024069 6.241242 5.130176   
## 6973 7003 7053 7085 7134 7175 7215 7256   
## 4.024069 4.939850 5.658195 7.504013 6.241242 6.241242 5.130176 5.658195   
## 7291 7379 7428 7431 7436 7441 7487 7533   
## 6.241242 4.024069 5.130176 5.658195 5.130176 5.658195 5.130176 7.504013   
## 7583 7602 7614 7641 7672 7686 7712 7784   
## 5.658195 7.504013 6.402402 5.130176 7.504013 6.841088 4.939850 6.841088   
## 7822 7907 8027 8113 8134 8152 8155 8230   
## 6.841088 4.024069 4.024069 7.504013 5.130176 6.241242 6.241242 5.130176   
## 8257 8276 8311 8345 8355 8415 8419 8436   
## 5.658195 6.241242 5.130176 4.939850 6.241242 4.024069 6.241242 6.241242   
## 8462 8464 8500 8557 8580 8640 8658 8676   
## 4.464251 4.024069 6.241242 4.939850 6.402402 5.658195 7.504013 5.340199   
## 8710 8778 8783 8790 8793 8848 8986 9093   
## 6.241242 6.841088 6.241242 4.024069 5.658195 6.241242 4.024069 4.464251   
## 9166 9200 9232 9374 9404 9406 9464 9548   
## 5.658195 5.340199 6.039909 6.241242 5.658195 5.130176 5.130176 4.939850   
## 9553 9554 9571 9604 9639 9766 9771 9781   
## 5.658195 5.130176 6.241242 4.024069 6.841088 6.841088 4.939850 6.841088   
## 9793 9814 9825 9867 9870 9907 9960   
## 6.841088 4.024069 4.464251 4.024069 4.024069 6.241242 7.504013   
##   
## $X17\_CentIntAp\_RiparianSys  
## 322 358 411 620 745 922 1017 1039   
## 3.028706 25.723287 3.028706 3.028706 5.977657 5.977657 5.977657 5.977657   
## 1303 2297 2298 2551 2709 2859 2889 2943   
## 3.028706 5.977657 25.723287 5.977657 5.977657 25.723287 25.723287 3.028706   
## 3139 3228 3232 3285 3519 3536 3719 3787   
## 25.723287 25.723287 3.028706 25.723287 5.977657 5.977657 5.977657 5.977657   
## 4367 5104 5458 5461 5475 5589 6011 6015   
## 5.977657 3.028706 3.028706 25.723287 3.028706 3.028706 3.028706 25.723287   
## 6069 6226 6381 7034 7041 7170 7607 7875   
## 3.028706 5.977657 5.977657 3.028706 25.723287 3.028706 5.977657 3.028706   
## 8048 8100 8140 8338 8488 8660 8856 9064   
## 5.977657 5.977657 3.028706 25.723287 5.977657 3.028706 5.977657 25.723287   
## 9367 9736   
## 25.723287 5.977657   
##   
## $X18\_LAc\_FloodplainSys  
## 324 329 473 578 583 608 658 778   
## 5.828811 5.326766 5.828811 3.000004 3.918816 3.839153 3.000004 3.821872   
## 826 928 929 949 1012 1087 1106 1171   
## 4.553656 3.821872 5.326766 4.801501 3.000004 3.821872 5.326766 3.821872   
## 1326 1339 1356 1389 1407 1455 1514 1528   
## 5.326766 5.326766 4.553656 5.828811 6.349637 3.000004 3.918816 3.821872   
## 1607 1700 1718 1774 1796 1807 1813 1970   
## 5.326766 5.326766 5.326766 5.326766 5.326766 4.310461 3.821872 5.326766   
## 2015 2020 2038 2061 2148 2173 2185 2191   
## 3.000004 4.310461 5.326766 5.326766 3.571272 4.553656 3.000004 5.326766   
## 2223 2271 2299 2307 2363 2398 2436 2447   
## 3.839153 5.326766 3.571272 3.839153 3.918816 3.918816 4.658721 3.000004   
## 2460 2515 2573 2593 2614 2620 2640 2644   
## 3.437088 3.839153 3.821872 4.801501 4.658721 3.437088 3.918816 3.000004   
## 2669 2681 2704 2721 2724 2772 2786 2825   
## 6.477633 5.326766 5.828811 5.326766 3.000004 5.828811 4.658721 3.821872   
## 2836 2838 2855 2866 2915 2953 2975 2978   
## 5.326766 3.821872 5.326766 3.821872 5.828811 5.326766 5.326766 3.821872   
## 3047 3051 3072 3129 3186 3261 3318 3334   
## 4.553656 3.000004 3.437088 5.326766 3.839153 5.828811 5.828811 3.000004   
## 3384 3455 3469 3540 3600 3657 3678 3868   
## 5.828811 3.821872 3.069899 5.828811 5.326766 3.821872 5.828811 3.000004   
## 3933 3986 4047 4065 4075 4089 4194 4316   
## 3.437088 5.326766 3.000004 3.821872 4.553656 5.326766 4.801501 4.801501   
## 4639 4703 4712 4723 4794 4820 4821 4860   
## 3.000004 3.821872 3.000004 5.828811 3.821872 5.326766 4.553656 3.918816   
## 4915 4966 5006 5128 5206 5250 5267 5310   
## 5.326766 3.821872 4.658721 3.069899 3.918816 5.326766 3.069899 4.658721   
## 5381 5396 5406 5434 5485 5553 5593 5673   
## 4.553656 3.918816 3.437088 3.839153 5.326766 4.310461 4.801501 5.326766   
## 5812 5977 6082 6146 6161 6233 6251 6302   
## 3.918816 3.437088 3.437088 4.310461 5.828811 4.801501 6.349637 5.326766   
## 6337 6366 6406 6407 6525 6568 6609 6637   
## 4.553656 3.000004 5.326766 3.918816 4.310461 3.069899 5.326766 5.326766   
## 6642 6687 6710 6715 6731 6743 6760 6763   
## 5.326766 3.437088 3.069899 3.821872 3.821872 5.326766 4.658721 3.000004   
## 6784 6838 6870 6960 6962 6987 6988 7063   
## 4.553656 5.828811 3.821872 3.069899 3.839153 3.069899 3.918816 5.828811   
## 7162 7203 7207 7219 7426 7430 7434 7442   
## 4.801501 4.658721 4.801501 3.918816 3.918816 4.553656 5.326766 4.658721   
## 7532 7543 7567 7593 7654 7663 7687 7695   
## 4.553656 4.553656 4.801501 3.821872 4.310461 5.326766 3.000004 3.821872   
## 7743 7783 7830 7883 7929 7930 8023 8102   
## 3.000004 3.000004 4.553656 3.918816 3.821872 3.821872 6.349637 4.553656   
## 8129 8139 8194 8241 8284 8333 8383 8468   
## 4.553656 4.553656 6.349637 3.918816 6.349637 4.553656 3.000004 5.326766   
## 8535 8566 8567 8625 8705 8830 8929 8931   
## 5.828811 5.828811 5.828811 5.828811 3.821872 3.069899 3.000004 3.821872   
## 9025 9071 9107 9130 9184 9188 9202 9221   
## 3.839153 5.828811 4.801501 3.069899 5.326766 5.828811 3.571272 3.821872   
## 9223 9252 9267 9345 9418 9436 9466 9514   
## 5.326766 3.821872 5.326766 3.069899 3.437088 5.326766 3.839153 5.326766   
## 9589 9648 9684 9720 9826 9880 9915 9923   
## 5.326766 6.477633 3.918816 3.821872 3.918816 5.828811 4.310461 5.828811   
## 9968   
## 3.000004   
##   
## $X19\_Bor\_Acidic\_PeatSys  
## 204 254 259 280 297 329 362 380   
## 4.237924 3.695042 7.522666 3.693565 5.431512 4.780857 4.294198 3.695042   
## 400 435 478 494 530 535 546 591   
## 3.695042 4.237924 4.237924 4.294198 7.522666 7.522666 7.522666 3.695042   
## 603 630 706 716 769 891 903 929   
## 4.237924 3.274409 7.522666 3.416228 7.522666 7.522666 4.237924 4.780857   
## 941 1011 1064 1106 1110 1121 1201 1213   
## 4.237924 4.237924 7.522666 4.780857 4.237924 4.237924 4.237924 4.237924   
## 1313 1326 1339 1496 1553 1565 1607 1609   
## 3.695042 4.780857 4.780857 7.522666 3.695042 7.240958 4.780857 7.522666   
## 1652 1681 1686 1687 1700 1701 1718 1733   
## 4.237924 4.294198 5.431512 4.237924 4.780857 3.695042 4.780857 5.431512   
## 1774 1783 1796 1807 1814 1824 1842 1892   
## 4.780857 4.237924 4.780857 3.228003 7.240958 4.237924 3.274409 4.237924   
## 1906 1913 1969 1970 2020 2031 2038 2061   
## 4.294198 5.760728 4.294198 4.780857 3.228003 7.522666 4.780857 4.780857   
## 2187 2191 2194 2198 2211 2231 2261 2271   
## 4.237924 4.780857 3.416228 4.294198 4.237924 7.522666 4.237924 4.780857   
## 2338 2410 2465 2475 2485 2502 2627 2641   
## 4.237924 3.695042 3.416228 4.237924 7.240958 5.760728 3.695042 5.431512   
## 2671 2678 2681 2683 2685 2721 2728 2730   
## 4.237924 7.240958 4.780857 4.237924 4.237924 4.780857 3.416228 5.431512   
## 2732 2826 2836 2855 2878 2953 2975 3011   
## 4.237924 4.294198 4.780857 4.780857 4.294198 4.780857 4.780857 4.237924   
## 3107 3129 3159 3219 3258 3270 3279 3282   
## 3.695042 4.780857 7.522666 3.416228 3.416228 3.274409 3.695042 7.522666   
## 3322 3466 3486 3491 3507 3510 3553 3600   
## 4.237924 5.760728 3.416228 4.237924 7.240958 4.294198 5.760728 4.780857   
## 3672 3687 3785 3798 3851 3864 3871 3916   
## 4.294198 3.695042 3.695042 7.522666 4.237924 5.760728 7.522666 7.240958   
## 3930 3936 3986 4023 4084 4089 4093 4110   
## 3.693565 7.522666 4.780857 7.522666 4.237924 4.780857 3.693565 3.416228   
## 4143 4150 4189 4199 4237 4262 4286 4304   
## 4.237924 5.760728 3.695042 4.237924 5.760728 4.294198 5.760728 5.760728   
## 4334 4354 4376 4462 4515 4525 4533 4567   
## 5.760728 7.522666 4.237924 4.237924 4.237924 3.695042 7.522666 4.237924   
## 4592 4636 4646 4652 4704 4715 4718 4740   
## 3.693565 4.294198 3.416228 3.695042 7.240958 4.237924 3.693565 5.431512   
## 4748 4760 4795 4804 4807 4820 4840 4846   
## 4.237924 4.294198 4.294198 4.294198 7.522666 4.780857 5.431512 4.237924   
## 4853 4871 4915 4923 4986 5008 5055 5131   
## 7.522666 4.294198 4.780857 4.294198 7.522666 4.237924 3.274409 4.237924   
## 5145 5156 5159 5177 5224 5250 5291 5349   
## 4.237924 4.237924 7.522666 4.294198 4.237924 4.780857 4.237924 4.237924   
## 5412 5422 5423 5427 5476 5485 5502 5553   
## 4.294198 4.237924 7.522666 3.695042 4.294198 4.780857 4.237924 3.228003   
## 5582 5657 5673 5686 5709 5713 5759 5808   
## 3.695042 4.237924 4.780857 7.522666 4.294198 3.695042 4.237924 4.237924   
## 5816 5819 5834 5929 5981 6020 6080 6146   
## 3.416228 4.237924 3.695042 7.522666 3.695042 7.522666 3.693565 3.228003   
## 6163 6302 6305 6360 6387 6406 6454 6496   
## 3.695042 4.780857 4.294198 4.237924 5.760728 4.780857 4.294198 4.294198   
## 6525 6560 6609 6613 6637 6642 6693 6732   
## 3.228003 3.416228 4.780857 4.237924 4.780857 4.780857 5.431512 3.416228   
## 6738 6743 6757 6863 6865 6867 6946 6950   
## 4.237924 4.780857 7.240958 4.237924 4.237924 7.522666 5.431512 3.695042   
## 6994 7004 7007 7126 7255 7257 7263 7271   
## 4.294198 3.416228 4.294198 5.760728 5.431512 5.431512 4.294198 3.416228   
## 7273 7336 7411 7434 7545 7578 7625 7629   
## 4.237924 4.237924 4.237924 4.780857 5.760728 4.237924 3.695042 7.522666   
## 7654 7663 7701 7720 7748 7835 7917 7976   
## 3.228003 4.780857 4.237924 3.695042 4.237924 5.431512 5.760728 7.522666   
## 7983 8037 8053 8087 8180 8214 8228 8233   
## 3.695042 7.522666 7.240958 5.431512 4.799332 4.237924 7.522666 3.695042   
## 8254 8351 8369 8402 8468 8519 8522 8600   
## 4.294198 4.237924 7.522666 4.771026 4.780857 3.693565 7.522666 3.695042   
## 8704 8776 8822 8842 8853 8898 8927 8943   
## 4.237924 3.416228 5.760728 3.416228 7.522666 4.237924 4.237924 3.695042   
## 8994 9034 9065 9074 9076 9090 9108 9184   
## 4.237924 3.695042 5.760728 4.237924 3.695042 7.522666 4.237924 4.780857   
## 9185 9223 9231 9267 9286 9322 9326 9327   
## 4.237924 4.780857 3.695042 4.780857 4.294198 4.237924 3.695042 5.760728   
## 9358 9394 9436 9456 9484 9501 9514 9515   
## 4.237924 4.237924 4.780857 3.695042 7.522666 5.760728 4.780857 4.237924   
## 9526 9536 9583 9589 9602 9612 9688 9707   
## 4.237924 7.240958 4.237924 4.780857 5.760728 3.695042 3.695042 4.294198   
## 9710 9742 9778 9790 9813 9821 9866 9881   
## 3.695042 4.294198 4.237924 4.237924 4.237924 5.760728 4.294198 7.522666   
## 9903 9915 9925 9939   
## 4.237924 3.228003 7.522666 3.695042   
##   
## $X20\_CentIntAp\_SwampSys  
## 2 3 4 5 6 7 8 40   
## 4.402183 4.402183 6.254452 6.254452 4.551060 4.551060 3.633516 4.551060   
## 41 42 187 205 239 262 265 287   
## 4.551060 4.551060 3.633516 3.633516 5.887710 4.420257 4.420257 3.633516   
## 336 452 486 497 526 636 670 732   
## 5.887710 4.117437 6.254452 3.633516 6.254452 3.633516 3.633516 5.887710   
## 744 745 749 775 796 805 853 872   
## 4.402183 3.001856 5.887710 3.633516 4.551060 4.402183 4.402183 4.402183   
## 922 965 994 997 1017 1039 1050 1051   
## 3.001856 6.254452 4.551060 5.887710 3.001856 3.001856 5.887710 4.420257   
## 1079 1196 1267 1269 1314 1316 1376 1442   
## 4.551060 4.402183 5.887710 4.402183 5.887710 5.887710 4.402183 4.117437   
## 1450 1538 1604 1672 1678 1684 1768 1791   
## 5.887710 5.887710 4.420257 3.633516 4.551060 4.402183 4.117437 5.887710   
## 1865 1904 2019 2042 2077 2113 2124 2127   
## 4.402183 4.402183 5.887710 4.402183 5.887710 3.633516 6.254452 4.117437   
## 2230 2242 2243 2249 2297 2318 2332 2334   
## 3.633516 5.887710 4.402183 4.402183 3.001856 4.551060 5.887710 4.117437   
## 2339 2433 2434 2438 2449 2458 2551 2553   
## 5.887710 4.402183 5.887710 4.402183 5.887710 4.420257 3.001856 4.551060   
## 2556 2583 2591 2666 2709 2737 2744 2745   
## 4.402183 3.633516 5.887710 5.887710 3.001856 4.420257 5.887710 4.551060   
## 2755 2765 2770 2807 2917 2950 2959 2966   
## 5.887710 5.887710 4.551060 3.633516 5.887710 6.254452 5.887710 4.117437   
## 2970 3168 3216 3222 3300 3344 3373 3387   
## 4.402183 4.551060 3.633516 4.402183 4.117437 4.420257 4.402183 4.402183   
## 3457 3468 3516 3519 3527 3529 3533 3536   
## 6.254452 5.887710 4.551060 3.001856 5.887710 4.551060 5.887710 3.001856   
## 3569 3606 3607 3664 3677 3713 3717 3719   
## 3.633516 4.402183 3.633516 5.887710 3.633516 6.254452 4.402183 3.001856   
## 3744 3778 3779 3787 3872 3923 3939 3960   
## 4.420257 5.887710 6.254452 3.001856 6.254452 4.402183 4.551060 4.420257   
## 3966 4026 4122 4188 4204 4227 4312 4327   
## 5.887710 6.254452 5.887710 6.254452 4.551060 4.402183 5.887710 4.551060   
## 4333 4346 4367 4402 4444 4478 4498 4542   
## 4.420257 6.254452 3.001856 4.551060 5.887710 5.887710 6.254452 3.633516   
## 4594 4701 4742 4759 4792 4877 4952 5150   
## 5.887710 4.402183 5.887710 5.887710 4.551060 4.402183 5.887710 4.402183   
## 5157 5169 5262 5284 5325 5328 5332 5432   
## 4.402183 5.887710 5.887710 4.117437 6.254452 5.887710 4.402183 4.420257   
## 5441 5592 5599 5639 5659 5676 5844 5870   
## 4.402183 4.402183 4.402183 4.402183 4.420257 3.633516 5.887710 4.402183   
## 5897 5904 5938 6033 6058 6062 6121 6137   
## 4.551060 3.633516 3.633516 4.117437 4.402183 4.402183 4.420257 6.254452   
## 6139 6226 6235 6254 6291 6327 6349 6381   
## 5.887710 3.001856 4.402183 3.633516 4.402183 4.420257 4.402183 3.001856   
## 6503 6537 6620 6621 6638 6647 6695 6783   
## 4.551060 6.254452 4.402183 6.254452 3.633516 6.254452 5.887710 4.402183   
## 6840 6904 6998 7061 7134 7149 7161 7175   
## 6.254452 4.420257 4.551060 4.117437 4.402183 5.887710 3.633516 4.402183   
## 7233 7278 7290 7291 7303 7320 7350 7364   
## 3.633516 4.551060 6.254452 4.402183 4.117437 4.551060 3.633516 3.633516   
## 7377 7429 7440 7464 7475 7486 7550 7607   
## 3.633516 6.254452 5.887710 5.887710 5.887710 3.633516 6.254452 3.001856   
## 7810 7834 7912 7966 8048 8100 8152 8155   
## 5.887710 6.254452 5.887710 6.254452 3.001856 3.001856 4.402183 4.402183   
## 8239 8271 8276 8285 8303 8355 8394 8419   
## 4.551060 4.420257 4.402183 5.887710 5.887710 4.402183 3.633516 4.402183   
## 8428 8436 8437 8488 8500 8504 8555 8644   
## 3.633516 4.402183 5.887710 3.001856 4.402183 5.887710 4.420257 5.887710   
## 8680 8701 8710 8783 8848 8856 8942 9008   
## 4.117437 4.551060 4.402183 4.402183 4.402183 3.001856 5.887710 5.887710   
## 9350 9374 9380 9472 9547 9571 9642 9698   
## 3.633516 4.402183 4.551060 4.420257 5.887710 4.402183 4.420257 3.633516   
## 9728 9736 9762 9795 9865 9886 9907 9924   
## 4.420257 3.001856 5.887710 4.562072 4.551060 5.887710 4.402183 4.551060   
##   
## $X21\_GulfAtl\_CoastPlain\_SwampSys  
## 121 170 224 294 319 330 343 349   
## 4.194793 4.688219 3.984337 4.194793 7.390859 4.860756 4.194793 7.390859   
## 383 390 394 466 489 492 525 627   
## 5.895175 4.194793 3.251783 4.688219 4.194793 4.860756 10.093312 5.186166   
## 653 655 702 703 741 795 883 924   
## 4.688219 3.437377 4.194793 6.772727 5.895175 4.860756 3.984337 3.216223   
## 963 985 996 1013 1037 1069 1152 1210   
## 5.895175 3.984337 6.772727 3.984337 4.688219 5.895175 3.984337 4.688219   
## 1222 1235 1255 1261 1278 1318 1368 1375   
## 5.895175 4.688219 3.251783 10.093312 5.895175 5.186166 4.194793 5.186166   
## 1427 1441 1513 1532 1543 1579 1597 1599   
## 3.437377 3.984337 5.406429 10.093312 4.688219 4.688219 4.860756 3.556360   
## 1694 1699 1765 1766 1810 1816 1836 1837   
## 7.409597 4.194793 4.194793 5.186166 3.984337 5.406429 10.093312 10.093312   
## 1862 1881 1947 1996 2093 2103 2110 2145   
## 10.093312 5.406429 4.860756 3.216223 5.142700 3.984337 3.984337 3.984337   
## 2147 2241 2321 2340 2354 2381 2412 2499   
## 7.409597 6.772727 5.186166 4.688219 3.556360 4.194793 4.194793 3.984337   
## 2558 2662 2672 2734 2785 2821 2847 2858   
## 4.860756 4.688219 4.688219 10.093312 3.984337 4.194793 3.216223 3.437377   
## 2911 2963 2969 3049 3053 3065 3099 3154   
## 5.895175 5.406429 3.437377 4.194793 5.406429 4.688219 10.093312 4.860756   
## 3194 3213 3248 3262 3487 3546 3552 3621   
## 5.895175 4.194793 3.984337 4.194793 7.409597 7.409597 5.895175 3.437377   
## 3660 3680 3718 3928 3993 4091 4111 4138   
## 4.194793 3.556360 3.984337 4.688219 5.895175 5.406429 4.194793 5.186166   
## 4177 4215 4219 4267 4285 4307 4373 4404   
## 3.251783 4.688219 4.194793 5.186166 7.409597 3.984337 4.688219 10.093312   
## 4464 4465 4560 4574 4629 4692 4808 4844   
## 3.556360 4.860756 5.406429 4.860756 5.895175 3.984337 3.251783 5.186166   
## 4891 4892 4897 4903 4911 4932 5061 5072   
## 4.860756 3.216223 3.984337 3.251783 7.409597 4.688219 5.895175 4.860756   
## 5107 5178 5195 5200 5230 5289 5305 5343   
## 3.984337 4.194793 5.895175 4.860756 3.216223 3.556360 4.860756 4.194793   
## 5380 5398 5465 5507 5547 5655 5716 5796   
## 4.194793 5.406429 4.688219 5.895175 6.772727 10.093312 5.406429 5.406429   
## 5850 5934 6002 6006 6007 6027 6056 6128   
## 4.688219 10.093312 3.984337 4.194793 5.186166 4.194793 5.895175 3.984337   
## 6209 6238 6264 6491 6674 6736 6753 6764   
## 5.406429 3.216223 4.688219 6.772727 3.984337 5.895175 7.409597 5.406429   
## 6767 6782 6848 6859 6977 6982 7015 7028   
## 10.093312 4.688219 7.409597 10.093312 5.895175 4.194793 10.093312 3.556360   
## 7037 7064 7138 7208 7212 7251 7262 7311   
## 5.895175 3.984337 3.216223 4.194793 3.216223 6.772727 5.895175 4.194793   
## 7356 7382 7397 7425 7474 7482 7558 7570   
## 3.984337 3.984337 4.194793 4.860756 5.406429 4.194793 10.093312 3.984337   
## 7633 7682 7707 7739 7740 7744 7771 7833   
## 4.688219 4.194793 5.895175 6.772727 5.406429 7.409597 3.216223 4.194793   
## 7853 7858 7881 7882 7906 7910 7914 8032   
## 10.093312 4.860756 3.216223 4.688219 3.556360 5.406429 3.984337 3.251783   
## 8148 8298 8302 8328 8332 8392 8406 8483   
## 3.984337 3.216223 3.984337 5.895175 3.984337 6.772727 3.216223 5.406429   
## 8485 8503 8536 8599 8607 8679 8731 8757   
## 3.251783 5.895175 7.409597 4.860756 3.984337 4.688219 4.194793 7.390859   
## 8760 8766 8844 8874 8887 8903 8911 8967   
## 3.216223 5.186166 4.860756 4.688219 7.390859 3.984337 3.984337 4.194793   
## 8973 9045 9073 9121 9132 9150 9155 9197   
## 5.406429 4.688219 4.688219 3.984337 3.984337 4.194793 5.895175 5.895175   
## 9261 9305 9321 9330 9355 9359 9370 9393   
## 3.437377 4.688219 3.251783 5.142700 3.216223 4.194793 5.142700 3.251783   
## 9402 9437 9482 9606 9644 9702 9713 9733   
## 4.194793 4.194793 3.556360 7.390859 3.216223 3.984337 5.895175 5.406429   
## 9751 9879 9894 9909 9912   
## 4.688219 4.194793 3.556360 3.984337 3.984337   
##   
## $X22\_GulfAtl\_CoastPlain\_TMarshSys  
## 237 239 319 336 349 487 525 627   
## 3.228672 11.397097 5.449559 11.397097 5.449559 3.228672 3.398224 5.887189   
## 655 732 749 997 1050 1161 1261 1267   
## 4.679889 11.397097 11.397097 11.397097 11.397097 3.228672 3.398224 11.397097   
## 1314 1316 1318 1375 1427 1447 1450 1532   
## 11.397097 11.397097 5.887189 5.887189 4.679889 12.286745 11.397097 3.398224   
## 1538 1766 1791 1836 1837 1862 2019 2077   
## 11.397097 5.887189 11.397097 3.398224 3.398224 3.398224 11.397097 11.397097   
## 2242 2321 2332 2339 2434 2449 2591 2666   
## 11.397097 5.887189 11.397097 11.397097 11.397097 11.397097 11.397097 11.397097   
## 2734 2744 2755 2765 2858 2917 2959 2969   
## 3.398224 11.397097 11.397097 11.397097 4.679889 11.397097 11.397097 4.679889   
## 3099 3130 3319 3468 3527 3533 3621 3664   
## 3.398224 3.228672 3.228672 11.397097 11.397097 11.397097 4.679889 11.397097   
## 3778 3940 3966 4122 4138 4246 4267 4312   
## 11.397097 3.228672 11.397097 11.397097 5.887189 12.286745 5.887189 11.397097   
## 4404 4444 4478 4594 4742 4759 4818 4844   
## 3.398224 11.397097 11.397097 11.397097 11.397097 11.397097 3.228672 5.887189   
## 4952 5169 5262 5328 5655 5844 5934 6007   
## 11.397097 11.397097 11.397097 11.397097 3.398224 11.397097 3.398224 5.887189   
## 6139 6343 6393 6512 6626 6695 6767 6859   
## 11.397097 3.228672 3.228672 3.228672 3.228672 11.397097 3.398224 3.398224   
## 7015 7149 7440 7464 7475 7558 7810 7853   
## 3.398224 11.397097 11.397097 11.397097 11.397097 3.398224 11.397097 3.398224   
## 7912 8204 8285 8303 8437 8504 8644 8757   
## 11.397097 3.228672 11.397097 11.397097 11.397097 11.397097 11.397097 5.449559   
## 8766 8887 8942 9008 9229 9261 9300 9547   
## 5.887189 5.449559 11.397097 11.397097 3.228672 4.679889 3.228672 11.397097   
## 9606 9762 9795 9886   
## 5.449559 11.397097 3.858446 11.397097   
##   
## $X23\_LAc\_Shrub.Herb\_WetlSys  
## 116 368 409 527 747 762 1136 1162   
## 15.542734 6.018059 6.018059 15.542734 6.018059 4.504001 6.018059 4.504001   
## 1288 1512 1660 1801 1847 1976 2648 2676   
## 6.018059 6.018059 15.542734 15.542734 15.542734 4.504001 6.018059 6.018059   
## 2933 3060 3149 3208 3242 3265 3496 3556   
## 6.018059 6.018059 15.542734 4.504001 6.018059 6.018059 6.018059 6.018059   
## 3618 3663 3907 3934 3978 4067 4302 4328   
## 4.504001 6.018059 6.018059 6.018059 6.018059 6.018059 15.542734 4.504001   
## 4424 4587 4714 4757 4784 4847 4927 5405   
## 6.018059 6.018059 6.018059 6.018059 4.504001 6.018059 6.018059 6.018059   
## 5703 6221 6271 6593 6633 6711 6785 6893   
## 6.018059 6.018059 4.504001 6.018059 4.504001 4.504001 6.018059 6.018059   
## 6937 7023 7079 7160 7247 8561 8692 8713   
## 15.542734 4.504001 6.018059 4.504001 6.018059 4.504001 6.018059 15.542734   
## 8816 8829 8934 8955 9031 9154 9222 9233   
## 6.018059 15.542734 4.504001 6.018059 6.018059 6.018059 4.504001 6.018059   
## 9567 9687 9796 9928 9938   
## 6.018059 6.018059 6.018059 4.504001 15.542734   
##   
## $X24\_NCentInt\_Wet\_Flatwd  
## 1 4 5 146 147 239 262 265   
## 3.552649 10.301528 10.301528 4.976982 4.976982 4.708174 11.141153 11.141153   
## 336 452 477 486 526 651 690 732   
## 4.708174 7.486824 5.226254 10.301528 10.301528 4.976982 3.916004 4.708174   
## 749 813 861 965 997 1050 1051 1088   
## 4.708174 3.916004 4.976982 10.301528 4.708174 4.708174 11.141153 5.226254   
## 1267 1314 1316 1379 1416 1417 1442 1450   
## 4.708174 4.708174 4.708174 3.552649 4.976982 3.552649 7.486824 4.708174   
## 1538 1556 1567 1604 1762 1768 1778 1791   
## 4.708174 4.976982 4.817485 11.141153 3.916004 7.486824 4.976982 4.708174   
## 1975 2005 2019 2062 2077 2124 2127 2242   
## 4.976982 3.552649 4.708174 3.916004 4.708174 10.301528 7.486824 4.708174   
## 2263 2332 2334 2339 2434 2449 2458 2549   
## 4.976982 4.708174 7.486824 4.708174 4.708174 4.708174 11.141153 5.226254   
## 2575 2591 2596 2666 2737 2744 2755 2765   
## 4.976982 4.708174 4.976982 4.708174 11.141153 4.708174 4.708174 4.708174   
## 2885 2901 2913 2917 2950 2959 2966 3113   
## 5.226254 4.976982 4.817485 4.708174 10.301528 4.708174 7.486824 4.976982   
## 3259 3300 3344 3457 3468 3527 3533 3563   
## 3.552649 7.486824 11.141153 10.301528 4.708174 4.708174 4.708174 3.916004   
## 3573 3608 3664 3713 3744 3778 3779 3854   
## 4.817485 4.976982 4.708174 10.301528 11.141153 4.708174 10.301528 4.976982   
## 3872 3960 3966 4026 4122 4188 4201 4291   
## 10.301528 11.141153 4.708174 10.301528 4.708174 10.301528 3.552649 4.976982   
## 4312 4333 4346 4444 4478 4498 4594 4673   
## 4.708174 11.141153 10.301528 4.708174 4.708174 10.301528 4.708174 3.552649   
## 4717 4742 4759 4811 4841 4952 5063 5169   
## 4.976982 4.708174 4.708174 3.552649 4.817485 4.708174 3.916004 4.708174   
## 5218 5262 5284 5325 5328 5432 5536 5659   
## 3.916004 4.708174 7.486824 10.301528 4.708174 11.141153 3.916004 11.141153   
## 5844 5905 6026 6033 6095 6121 6137 6139   
## 4.708174 3.916004 3.552649 7.486824 3.916004 11.141153 10.301528 4.708174   
## 6185 6297 6327 6354 6487 6537 6621 6629   
## 5.226254 5.226254 11.141153 3.916004 3.916004 10.301528 10.301528 4.976982   
## 6647 6689 6695 6840 6904 6945 7003 7053   
## 10.301528 3.552649 4.708174 10.301528 11.141153 4.976982 5.226254 3.916004   
## 7061 7149 7215 7256 7290 7303 7428 7429   
## 7.486824 4.708174 4.976982 3.916004 10.301528 7.486824 4.976982 10.301528   
## 7431 7436 7440 7441 7464 7475 7487 7550   
## 3.916004 4.976982 4.708174 3.916004 4.708174 4.708174 4.976982 10.301528   
## 7563 7583 7641 7712 7714 7810 7834 7912   
## 4.817485 3.916004 4.976982 5.226254 4.817485 4.708174 10.301528 4.708174   
## 7966 8134 8230 8257 8271 8285 8303 8311   
## 10.301528 4.976982 4.976982 3.916004 11.141153 4.708174 4.708174 4.976982   
## 8345 8437 8462 8504 8555 8557 8640 8644   
## 5.226254 4.708174 3.552649 4.708174 11.141153 5.226254 3.916004 4.708174   
## 8680 8793 8892 8942 9008 9093 9166 9232   
## 7.486824 3.916004 4.817485 4.708174 4.708174 3.552649 3.916004 3.780008   
## 9404 9406 9464 9472 9547 9548 9553 9554   
## 3.916004 4.976982 4.976982 11.141153 4.708174 5.226254 3.916004 4.976982   
## 9642 9728 9762 9771 9795 9825 9886   
## 11.141153 11.141153 4.708174 5.226254 5.986573 3.552649 4.708174   
##   
## $X25\_LAc\_SwampSys  
## 157 196 199 218 283 286 289 307   
## 3.293068 3.690372 3.038252 3.286016 3.385846 3.407748 3.059652 3.404894   
## 321 355 370 396 432 469 537 564   
## 3.286016 3.286016 3.293068 3.038252 3.407748 3.690372 3.404894 3.038252   
## 618 657 666 680 727 771 779 851   
## 3.404894 3.407748 3.293068 3.690372 3.286016 3.059652 3.690372 3.059652   
## 852 874 894 900 905 995 1065 1215   
## 3.286016 3.385846 3.404894 3.286016 3.038252 3.690372 3.407748 3.690372   
## 1268 1277 1280 1301 1338 1359 1436 1458   
## 3.286016 3.286016 3.038252 3.059652 3.286016 3.038252 3.038252 3.059652   
## 1544 1560 1577 1603 1621 1633 1753 1861   
## 3.385846 3.404894 3.038252 3.385846 3.286016 3.385846 3.293068 3.385846   
## 1886 1903 1921 1971 2013 2037 2116 2212   
## 3.690372 3.059652 3.385846 3.286016 3.407748 3.407748 3.293068 3.404894   
## 2232 2266 2296 2330 2331 2357 2365 2393   
## 3.286016 3.385846 3.407748 3.293068 3.690372 3.407748 3.385846 3.385846   
## 2455 2457 2564 2639 2698 2775 2781 2792   
## 3.690372 3.385846 3.286016 3.038252 3.038252 3.690372 3.059652 3.038252   
## 2843 2846 2908 2921 2968 2985 3005 3093   
## 3.038252 3.038252 3.690372 3.690372 3.286016 3.059652 3.286016 3.286016   
## 3110 3131 3170 3255 3304 3320 3328 3383   
## 3.286016 3.293068 3.407748 3.407748 3.690372 3.059652 3.038252 3.407748   
## 3394 3464 3526 3550 3625 3705 3708 3766   
## 3.404894 3.407748 3.385846 3.059652 3.385846 3.038252 3.286016 3.059652   
## 3881 3888 3918 3946 3952 3979 4066 4098   
## 3.059652 3.690372 3.286016 3.407748 3.690372 3.286016 3.059652 3.059652   
## 4154 4233 4289 4319 4413 4441 4469 4519   
## 3.404894 3.293068 3.059652 3.690372 3.404894 3.690372 3.286016 3.293068   
## 4545 4547 4571 4577 4583 4613 4621 4637   
## 3.215951 3.286016 3.286016 3.059652 3.038252 3.690372 3.385846 3.690372   
## 4650 4672 4785 4827 4852 4885 4889 4894   
## 3.286016 3.690372 3.385846 3.404894 3.404894 3.038252 3.059652 3.385846   
## 4938 5056 5070 5094 5101 5108 5207 5252   
## 3.059652 3.690372 3.385846 3.293068 3.286016 3.385846 3.038252 3.038252   
## 5258 5259 5276 5288 5320 5323 5368 5420   
## 3.690372 3.690372 3.059652 3.690372 3.404894 3.407748 3.690372 3.385846   
## 5466 5486 5573 5591 5622 5629 5698 5789   
## 3.286016 3.286016 3.385846 3.215951 3.690372 3.385846 3.059652 3.385846   
## 5818 5927 5957 5990 6003 6025 6084 6130   
## 3.407748 3.059652 3.059652 3.286016 3.286016 3.690372 3.038252 3.286016   
## 6151 6177 6188 6253 6415 6438 6468 6509   
## 3.407748 3.286016 3.690372 3.059652 3.407748 3.215951 3.385846 3.385846   
## 6515 6538 6547 6667 6871 6894 6897 6902   
## 3.038252 3.404894 3.690372 3.059652 3.038252 3.059652 3.690372 3.404894   
## 6943 6984 7026 7097 7104 7111 7188 7227   
## 3.286016 3.385846 3.038252 3.286016 3.293068 3.407748 3.385846 3.407748   
## 7253 7368 7396 7416 7427 7433 7457 7493   
## 3.690372 3.385846 3.690372 3.038252 3.286016 3.059652 3.286016 3.286016   
## 7520 7566 7619 7699 7737 7791 7839 7864   
## 3.385846 3.038252 3.286016 3.038252 3.038252 3.286016 3.293068 3.038252   
## 7867 7956 7984 8024 8051 8054 8075 8111   
## 3.385846 3.293068 3.385846 3.038252 3.690372 3.038252 3.690372 3.059652   
## 8128 8160 8205 8212 8223 8294 8314 8316   
## 3.059652 3.385846 3.059652 3.059652 3.407748 3.038252 3.286016 3.059652   
## 8329 8361 8371 8380 8385 8390 8407 8420   
## 3.293068 3.286016 3.385846 3.385846 3.690372 3.690372 3.038252 3.293068   
## 8424 8477 8494 8518 8540 8541 8550 8551   
## 3.690372 3.385846 3.286016 3.407748 3.038252 3.059652 3.385846 3.286016   
## 8571 8582 8726 8745 8767 8771 8815 8884   
## 3.690372 3.385846 3.407748 3.286016 3.059652 3.385846 3.293068 3.385846   
## 8891 8905 8913 8944 8950 8969 8991 9083   
## 3.385846 3.385846 3.059652 3.286016 3.385846 3.059652 3.385846 3.059652   
## 9103 9149 9160 9324 9357 9366 9376 9378   
## 3.038252 3.286016 3.059652 3.690372 3.059652 3.407748 3.293068 3.286016   
## 9449 9507 9538 9541 9578 9580 9585 9597   
## 3.404894 3.690372 3.059652 3.038252 3.059652 3.404894 3.293068 3.385846   
## 9603 9615 9625 9637 9667 9680 9696 9715   
## 3.038252 3.038252 3.385846 3.385846 3.385846 3.404894 3.038252 3.690372   
## 9721 9723 9732 9734 9739 9759 9772 9777   
## 3.407748 3.407748 3.293068 3.059652 3.286016 3.286016 3.404894 3.407748   
## 9820 9871 9901 9971   
## 3.385846 3.293068 3.286016 3.690372   
##   
## $X26\_NeInt\_PineBarrens  
## 97 112 113 123 170 202 234 241   
## 7.361108 7.361108 7.361108 5.308991 3.018715 7.361108 7.361108 3.229211   
## 282 314 323 376 391 466 516 576   
## 4.394081 3.229211 3.229211 5.308991 4.901020 3.018715 5.174947 5.339849   
## 625 653 703 804 807 821 826 835   
## 3.513609 3.018715 9.647644 5.308991 5.308991 3.513609 3.495404 3.513609   
## 848 888 957 974 996 1023 1037 1177   
## 3.229211 3.229211 3.513609 3.229211 9.647644 5.308991 3.018715 5.174947   
## 1210 1219 1221 1224 1235 1264 1356 1378   
## 3.018715 5.339849 3.430985 3.430985 3.018715 3.430985 3.495404 4.901020   
## 1398 1435 1491 1513 1543 1546 1575 1579   
## 7.361108 3.513609 3.430985 7.714158 3.018715 3.229211 4.394081 3.018715   
## 1632 1655 1694 1806 1816 1835 1869 1881   
## 5.308991 7.361108 3.641153 3.229211 7.714158 4.901020 3.430985 7.714158   
## 1900 1930 1943 1944 2147 2173 2241 2272   
## 3.229211 3.229211 5.308991 3.430985 3.641153 3.495404 9.647644 3.430985   
## 2283 2340 2471 2483 2497 2592 2634 2650   
## 4.901020 3.018715 4.394081 5.174947 4.394081 3.430985 3.229211 5.339849   
## 2652 2662 2672 2766 2833 2900 2963 3047   
## 3.229211 3.018715 3.018715 3.430985 5.174947 3.229211 7.714158 3.495404   
## 3053 3065 3101 3135 3182 3190 3205 3290   
## 7.714158 3.018715 7.361108 3.229211 4.901020 7.361108 5.339849 5.174947   
## 3297 3311 3340 3347 3487 3546 3601 3639   
## 4.394081 5.174947 4.394081 3.430985 3.641153 3.641153 3.513609 3.430985   
## 3695 3786 3806 3928 3929 4061 4075 4091   
## 5.174947 5.308991 5.308991 3.018715 4.901020 4.394081 3.495404 7.714158   
## 4125 4215 4225 4244 4257 4285 4293 4296   
## 5.308991 3.018715 3.430985 3.513609 5.339849 3.641153 3.430985 7.361108   
## 4313 4338 4355 4359 4373 4411 4416 4423   
## 3.430985 3.229211 7.361108 5.174947 3.018715 3.513609 7.361108 4.394081   
## 4425 4496 4560 4573 4575 4588 4625 4678   
## 5.339849 4.901020 7.714158 3.513609 5.308991 3.229211 4.901020 5.174947   
## 4797 4815 4821 4829 4874 4881 4911 4932   
## 3.430985 5.339849 3.495404 4.901020 5.339849 3.229211 3.641153 3.018715   
## 4946 5000 5083 5197 5203 5225 5285 5338   
## 3.430985 3.229211 4.901020 3.229211 3.430985 3.229211 5.308991 4.394081   
## 5378 5381 5398 5452 5465 5512 5517 5547   
## 4.394081 3.495404 7.714158 5.339849 3.018715 7.361108 5.174947 9.647644   
## 5567 5578 5598 5656 5716 5724 5763 5796   
## 5.339849 5.174947 3.229211 3.513609 7.714158 5.174947 3.430985 7.714158   
## 5850 5906 5987 6041 6209 6227 6264 6322   
## 3.018715 7.361108 3.430985 5.339849 7.714158 3.229211 3.018715 3.430985   
## 6337 6361 6414 6425 6443 6491 6517 6532   
## 3.495404 5.339849 4.394081 3.229211 3.229211 9.647644 5.308991 4.394081   
## 6553 6578 6580 6599 6617 6650 6719 6735   
## 4.394081 3.513609 5.308991 4.901020 3.430985 5.174947 3.229211 3.430985   
## 6753 6764 6782 6784 6795 6817 6826 6848   
## 3.641153 7.714158 3.018715 3.495404 5.308991 7.361108 3.229211 3.641153   
## 6855 6868 6912 6985 7051 7059 7182 7251   
## 5.174947 3.513609 4.901020 4.394081 3.229211 3.229211 5.174947 9.647644   
## 7252 7254 7279 7430 7474 7532 7543 7594   
## 3.513609 3.229211 5.308991 3.495404 7.714158 3.495404 3.495404 5.308991   
## 7633 7697 7727 7739 7740 7744 7756 7785   
## 3.018715 3.430985 3.430985 9.647644 7.714158 3.641153 4.394081 4.901020   
## 7807 7830 7859 7860 7882 7899 7904 7910   
## 5.174947 3.495404 5.174947 5.174947 3.018715 4.901020 3.229211 7.714158   
## 7926 7993 8043 8102 8129 8139 8146 8159   
## 7.361108 7.361108 4.901020 3.495404 3.495404 3.495404 3.229211 5.174947   
## 8165 8181 8201 8306 8333 8336 8392 8423   
## 4.394081 4.901020 7.361108 5.308991 3.495404 5.308991 9.647644 3.229211   
## 8459 8483 8492 8536 8610 8665 8679 8746   
## 4.394081 7.714158 4.901020 3.641153 7.361108 4.394081 3.018715 5.339849   
## 8765 8874 8883 8973 8990 8996 9023 9045   
## 3.430985 3.018715 3.513609 7.714158 7.361108 3.229211 5.339849 3.018715   
## 9053 9073 9213 9214 9226 9250 9260 9263   
## 5.308991 3.018715 5.339849 3.430985 4.394081 3.430985 5.308991 5.308991   
## 9283 9292 9305 9410 9443 9451 9460 9480   
## 3.229211 5.174947 3.018715 5.339849 3.513609 4.394081 5.308991 5.308991   
## 9555 9564 9590 9620 9729 9733 9751 9810   
## 3.229211 5.174947 4.394081 5.174947 5.339849 7.714158 3.018715 4.901020   
## 9817 9839 9916 9951   
## 5.339849 3.430985 3.430985 3.430985   
##   
## $X27\_AcAp\_WdHeath.Krummholz  
## 252 277 626 694 988 1014 1046 1052   
## 6.772746 11.782103 6.772746 3.235447 13.175952 11.782103 3.235447 4.841990   
## 1154 1243 1335 1411 1430 1462 1471 1507   
## 8.203458 4.841990 4.841990 4.841990 4.841990 4.841990 5.746058 4.841990   
## 1653 1689 1746 1873 1933 2010 2059 2172   
## 6.056389 11.782103 3.235447 3.619434 12.475932 6.056389 4.841990 5.849178   
## 2262 2302 2341 2374 2395 2505 2506 2532   
## 3.619434 3.037537 4.841990 3.037537 11.782103 3.405817 5.746058 4.841990   
## 2563 2569 2588 2817 2832 2834 2850 2870   
## 4.841990 13.175952 13.303359 4.841990 3.037537 4.841990 14.628746 5.849178   
## 2916 2960 2989 2991 3006 3246 3288 3330   
## 4.841990 13.303359 3.405817 4.841990 4.841990 3.037537 3.235447 4.841990   
## 3523 3671 3727 3843 3877 3951 3954 3980   
## 11.782103 3.235447 9.869475 13.175952 8.203458 6.056389 15.367197 15.367197   
## 4004 4034 4076 4121 4210 4214 4315 4388   
## 11.782103 4.841990 11.782103 5.746058 3.037537 15.367197 5.849178 6.772746   
## 4392 4399 4488 4552 4586 4663 4713 4726   
## 5.746058 3.235447 6.056389 4.841990 12.475932 13.303359 8.203458 13.303359   
## 4730 4772 4848 4851 5004 5076 5103 5115   
## 3.235447 5.849178 4.841990 6.772746 3.235447 3.037537 13.303359 15.367197   
## 5271 5275 5317 5345 5363 5453 5494 5509   
## 13.303359 5.849178 11.782103 5.746058 12.475932 4.841990 3.619434 3.619434   
## 5543 5661 5822 6053 6091 6277 6290 6296   
## 6.772746 3.235447 4.841990 13.303359 5.746058 5.849178 6.772746 14.628746   
## 6326 6452 6474 6603 6734 6787 6832 7066   
## 4.841990 5.849178 5.746058 4.841990 3.405817 12.475932 3.619434 4.841990   
## 7093 7125 7198 7209 7258 7443 7471 7510   
## 5.849178 13.175952 4.841990 11.782103 6.772746 4.841990 3.405817 6.772746   
## 7512 7526 7540 7760 7932 7948 8047 8076   
## 11.782103 5.746058 4.841990 4.841990 4.841990 3.619434 9.869475 3.619434   
## 8082 8144 8170 8243 8247 8268 8482 8584   
## 4.841990 4.841990 5.746058 3.405817 5.849178 3.619434 3.405817 11.782103   
## 8609 8756 8791 8803 8810 8866 8937 9094   
## 4.841990 9.869475 3.235447 3.619434 9.869475 9.869475 4.841990 4.841990   
## 9110 9114 9168 9177 9549 9631 9672 9726   
## 11.782103 4.841990 6.056389 6.772746 4.841990 5.746058 3.405817 4.841990   
## 9803 9805 9864 9929   
## 15.367197 6.772746 12.475932 3.235447   
##   
## $X28\_Bor\_JackPine.BlackSpruce  
## 144 215 225 402 433 464 682 683   
## 8.517336 8.517336 7.642294 8.517336 8.517336 8.517336 8.517336 8.517336   
## 761 787 881 989 1181 1414 1467 1539   
## 7.642294 8.517336 8.517336 8.517336 7.642294 8.517336 7.642294 8.517336   
## 1661 1736 1878 2014 2098 2120 2123 2161   
## 8.517336 7.642294 7.642294 7.642294 8.711692 8.711692 8.517336 8.517336   
## 2177 2256 2309 2342 2565 2607 2619 2714   
## 8.517336 7.642294 7.642294 7.642294 8.517336 7.642294 8.517336 8.517336   
## 2720 2729 2760 3045 3572 3575 3619 3679   
## 8.711692 7.642294 8.517336 8.517336 8.517336 8.517336 7.642294 7.642294   
## 3716 3746 3773 4041 4152 4156 4252 4391   
## 8.517336 8.517336 8.711692 8.517336 8.517336 8.517336 8.517336 8.517336   
## 4456 4710 4862 4866 4879 4913 5009 5097   
## 7.642294 7.642294 8.711692 7.642294 7.642294 7.642294 8.517336 8.517336   
## 5281 5347 5471 5581 5699 5764 5811 5878   
## 8.711692 7.642294 8.517336 8.517336 7.642294 7.642294 7.642294 8.711692   
## 6014 6076 6218 6260 6494 6573 6602 6630   
## 8.711692 8.711692 8.711692 8.517336 8.517336 7.642294 8.711692 8.517336   
## 6818 6837 6839 7012 7366 7413 7515 7824   
## 8.517336 8.517336 8.517336 8.711692 8.517336 8.517336 8.711692 8.517336   
## 7873 7896 7920 8173 8213 8628 8901 9002   
## 8.517336 8.517336 8.517336 8.517336 8.711692 8.517336 8.517336 8.517336   
## 9024 9104 9139 9219 9242 9298 9382 9481   
## 8.517336 8.711692 8.517336 8.517336 8.517336 8.517336 8.517336 7.642294   
## 9513 9693 9944   
## 8.517336 8.517336 8.517336   
##   
## $X29\_AcAp\_AlpineTundra  
## 173 180 209 227 236 269 331 352   
## 5.739082 6.233434 7.246635 4.450847 5.739082 4.450847 7.246635 7.246635   
## 441 447 455 468 475 513 589 643   
## 6.120851 4.678907 7.429613 3.454060 3.376535 7.300658 4.450847 7.246635   
## 776 802 843 938 948 1060 1151 1253   
## 7.429613 3.376535 3.376535 5.739082 7.246635 4.678907 5.340989 7.246635   
## 1276 1308 1510 1520 1568 1628 1629 1631   
## 7.429613 4.450847 7.308909 6.120851 6.120851 7.246635 4.450847 6.120851   
## 1659 1697 1782 1843 1851 1935 2027 2039   
## 5.739082 3.376535 6.233434 7.246635 7.429613 7.246635 4.678907 3.376535   
## 2089 2138 2156 2218 2251 2255 2258 2351   
## 5.639469 7.246635 3.376535 7.246635 7.300658 3.376535 7.246635 7.300658   
## 2367 2493 2522 2615 2647 2731 2754 2788   
## 5.340989 7.246635 3.376535 7.246635 7.246635 7.300658 6.120851 7.246635   
## 2827 2931 2946 2958 3007 3086 3133 3176   
## 3.376535 3.376535 7.246635 4.678907 7.429613 4.450847 5.639469 4.450847   
## 3276 3380 3417 3474 3506 3562 3612 3636   
## 3.376535 5.739082 7.429613 5.739082 6.233434 5.739082 6.120851 5.639469   
## 3662 3731 3789 3797 3822 3828 3898 3925   
## 3.376535 7.246635 7.429613 5.340989 7.429613 6.233434 6.120851 3.454060   
## 3988 4015 4019 4029 4030 4083 4139 4153   
## 4.450847 7.429613 3.454060 6.233434 6.233434 7.246635 3.454060 7.246635   
## 4218 4232 4239 4270 4301 4306 4378 4395   
## 7.246635 6.120851 7.246635 5.739082 3.376535 5.739082 5.739082 5.739082   
## 4467 4474 4505 4572 4662 4696 4776 4780   
## 7.246635 6.233434 7.246635 7.246635 7.429613 7.429613 3.454060 7.246635   
## 4823 4858 4887 4922 4953 4957 4996 4998   
## 7.308909 5.739082 7.246635 7.246635 6.120851 5.739082 3.376535 7.246635   
## 5002 5149 5152 5212 5260 5279 5283 5489   
## 7.246635 7.300658 7.429613 5.739082 7.246635 7.246635 5.739082 5.739082   
## 5508 5642 5654 5667 5729 5730 5815 5939   
## 6.120851 3.376535 4.450847 7.246635 7.300658 5.639469 3.376535 7.246635   
## 5975 6012 6072 6111 6113 6176 6215 6216   
## 7.429613 6.233434 6.120851 7.308909 7.246635 7.300658 5.739082 3.376535   
## 6355 6362 6430 6439 6465 6478 6505 6522   
## 7.429613 7.308909 7.308909 7.246635 6.120851 4.450847 7.246635 5.739082   
## 6552 6596 6805 6827 6880 6899 6923 6993   
## 7.429613 3.376535 5.739082 7.246635 7.429613 7.429613 5.739082 7.246635   
## 7043 7048 7108 7109 7244 7283 7334 7370   
## 7.308909 7.246635 4.450847 7.429613 3.376535 7.429613 7.246635 3.454060   
## 7446 7542 7544 7569 7612 7631 7640 7661   
## 3.454060 7.429613 7.246635 7.246635 7.308909 7.300658 7.246635 5.739082   
## 7674 7755 7763 7801 7866 7923 7961 8007   
## 7.246635 7.246635 3.376535 4.450847 7.308909 3.376535 4.450847 7.246635   
## 8029 8061 8081 8250 8297 8330 8339 8353   
## 7.429613 7.429613 6.120851 3.376535 6.120851 7.246635 7.246635 7.246635   
## 8363 8393 8445 8448 8520 8618 8708 8737   
## 7.246635 7.246635 6.120851 6.120851 7.246635 7.246635 5.739082 7.246635   
## 8747 8770 8800 8825 8843 8872 8875 8920   
## 7.246635 7.246635 7.429613 7.308909 5.739082 3.376535 5.739082 5.739082   
## 8952 8960 9048 9183 9313 9337 9408 9426   
## 7.246635 6.120851 6.120851 7.429613 7.246635 7.246635 7.246635 7.300658   
## 9432 9433 9487 9540 9640 9678 9699 9724   
## 7.246635 5.739082 4.450847 7.246635 7.246635 3.376535 3.376535 7.429613   
## 9806 9845 9908 9922 9949   
## 7.429613 4.678907 7.246635 3.376535 3.376535   
##   
## $sand  
## 110 120 126 148 206 244 285 319   
## -3.708526 -3.042808 -3.708526 -3.042808 -3.042808 -3.617410 -3.708526 -4.937971   
## 349 406 525 602 608 627 639 710   
## -4.937971 -3.484674 -3.227575 -3.484674 -4.066695 -3.416462 -3.351058 -3.042808   
## 720 978 1123 1146 1261 1318 1371 1375   
## -3.617410 -3.042808 -3.351058 -3.042808 -3.227575 -3.416462 -3.042808 -3.416462   
## 1386 1454 1466 1487 1493 1532 1595 1599   
## -3.665454 -3.446835 -3.708526 -3.042808 -3.728579 -3.227575 -3.351058 -3.524382   
## 1635 1766 1780 1833 1836 1837 1862 1866   
## -3.963176 -3.416462 -3.617410 -3.446835 -3.227575 -3.227575 -3.227575 -3.484674   
## 1924 1979 2093 2223 2307 2311 2321 2354   
## -3.446835 -3.351058 -3.148170 -4.066695 -4.066695 -3.446835 -3.416462 -3.524382   
## 2416 2428 2476 2515 2586 2657 2734 2853   
## -3.708526 -3.446835 -3.708526 -4.066695 -3.351058 -3.708526 -3.227575 -3.963176   
## 2861 3032 3099 3118 3186 3196 3357 3374   
## -3.351058 -3.708526 -3.227575 -3.351058 -4.066695 -3.617410 -3.963176 -3.963176   
## 3433 3629 3653 3680 3698 3782 3942 4138   
## -3.665454 -3.708526 -3.617410 -3.524382 -3.446835 -3.351058 -3.617410 -3.416462   
## 4164 4223 4226 4263 4267 4330 4397 4404   
## -3.708526 -3.728579 -3.042808 -3.042808 -3.416462 -3.963176 -3.484674 -3.227575   
## 4464 4504 4667 4822 4844 4855 4895 5172   
## -3.524382 -3.484674 -3.042808 -3.963176 -3.416462 -3.446835 -3.708526 -3.617410   
## 5289 5434 5513 5556 5655 5787 5885 5916   
## -3.524382 -4.066695 -3.351058 -3.446835 -3.227575 -3.617410 -3.963176 -3.963176   
## 5934 6007 6099 6167 6313 6317 6432 6577   
## -3.227575 -3.416462 -3.708526 -3.963176 -3.351058 -3.708526 -3.351058 -3.708526   
## 6697 6728 6767 6859 6861 6866 6954 6962   
## -3.708526 -3.617410 -3.227575 -3.227575 -3.617410 -3.484674 -3.708526 -4.066695   
## 7015 7019 7028 7049 7297 7352 7407 7499   
## -3.227575 -3.708526 -3.617410 -3.728579 -3.728579 -3.708526 -3.484674 -3.042808   
## 7558 7664 7685 7782 7853 7906 7938 7954   
## -3.227575 -3.484674 -3.484674 -3.042808 -3.227575 -3.524382 -3.963176 -3.617410   
## 7974 8078 8093 8207 8323 8388 8583 8643   
## -3.446835 -3.708526 -3.617410 -3.617410 -3.042808 -3.708526 -3.042808 -3.708526   
## 8664 8681 8697 8757 8766 8851 8887 8964   
## -3.617410 -3.484674 -3.484674 -4.937971 -3.416462 -3.665454 -4.937971 -3.351058   
## 8989 9025 9035 9174 9330 9370 9466 9482   
## -3.708526 -4.066695 -3.963176 -3.351058 -3.148170 -3.148170 -4.066695 -3.524382   
## 9606 9633 9808 9868 9894 9956   
## -4.937971 -3.446835 -3.351058 -3.351058 -3.524382 -3.665454   
##   
## $silt  
## 319 349 1623 4591 5215 5294 8757 8887   
## 3.179968 3.179968 3.130955 3.130955 3.130955 3.130955 3.179968 3.179968   
## 9606   
## 3.179968   
##   
## $slope  
## 277 513 694 907 988 1014 1046 1048   
## 4.349354 4.483151 3.150882 3.125921 5.340810 4.349354 3.150882 3.392857   
## 1052 1151 1243 1323 1335 1385 1411 1430   
## 3.389620 4.144403 3.389620 3.000587 3.389620 3.147574 3.389620 3.389620   
## 1471 1507 1612 1653 1675 1689 1746 1933   
## 3.298240 3.389620 3.467685 3.371309 3.000587 4.349354 3.150882 3.598580   
## 2010 2012 2059 2089 2142 2222 2251 2254   
## 3.371309 3.000587 3.389620 3.434802 3.147574 3.467685 4.483151 3.000587   
## 2341 2351 2367 2384 2395 2506 2532 2563   
## 3.389620 4.483151 4.144403 3.467685 4.349354 3.298240 3.389620 3.389620   
## 2569 2588 2621 2731 2817 2834 2850 2916   
## 5.340810 3.930422 3.000587 4.483151 3.389620 3.389620 4.118048 3.389620   
## 2960 2991 3006 3028 3133 3288 3330 3331   
## 3.930422 3.389620 3.389620 3.000587 3.434802 3.150882 3.389620 3.000587   
## 3523 3561 3636 3648 3671 3727 3797 3843   
## 4.349354 3.467685 3.434802 3.000587 3.150882 4.997419 4.144403 5.340810   
## 3951 3954 3980 4004 4034 4076 4121 4147   
## 3.371309 3.703886 3.703886 4.349354 3.389620 4.349354 3.298240 3.147574   
## 4187 4214 4392 4399 4487 4488 4552 4586   
## 3.125921 3.703886 3.298240 3.150882 3.000587 3.371309 3.389620 3.598580   
## 4661 4663 4726 4730 4843 4848 4901 5004   
## 3.000587 3.930422 3.930422 3.150882 3.000587 3.389620 3.467685 3.150882   
## 5034 5103 5115 5149 5191 5271 5317 5345   
## 3.467685 3.930422 3.703886 4.483151 3.000587 3.930422 4.349354 3.298240   
## 5363 5453 5608 5661 5729 5730 5786 5821   
## 3.598580 3.389620 3.000587 3.150882 4.483151 3.434802 3.000587 3.000587   
## 5822 5861 6053 6091 6134 6176 6296 6326   
## 3.389620 3.125921 3.930422 3.298240 3.000587 4.483151 4.118048 3.389620   
## 6356 6474 6603 6641 6787 6844 7066 7125   
## 3.000587 3.298240 3.389620 3.000587 3.598580 3.147574 3.389620 5.340810   
## 7198 7209 7213 7243 7296 7443 7454 7512   
## 3.389620 4.349354 3.147574 3.467685 3.467685 3.389620 3.000587 4.349354   
## 7526 7540 7631 7760 7789 7794 7848 7932   
## 3.298240 3.389620 4.483151 3.389620 3.147574 3.467685 3.000587 3.389620   
## 7999 8047 8082 8144 8170 8321 8584 8609   
## 3.125921 4.997419 3.389620 3.389620 3.298240 3.467685 4.349354 3.389620   
## 8707 8749 8756 8791 8810 8850 8866 8894   
## 3.000587 3.125921 4.997419 3.150882 4.997419 3.467685 4.997419 3.000587   
## 8921 8937 8958 9013 9094 9110 9114 9168   
## 3.000587 3.389620 3.147574 3.467685 3.389620 4.349354 3.389620 3.371309   
## 9256 9426 9470 9549 9586 9631 9726 9746   
## 3.467685 4.483151 3.000587 3.389620 3.125921 3.298240 3.389620 3.467685   
## 9803 9864 9927 9929 9937 9980   
## 3.703886 3.598580 3.125921 3.150882 3.000587 3.000587

**Question 5: Do you detect any outliers? For which variables? (5 pts)**

*I detected many outliers across all of the variables except for coast, X01\_water, X03\_LAc\_NHardwd, and X10\_Ap\_Hemlock.N\_Hardwood.*

### Multivariate outliers

**we will come back to this…**

## Distance and Dissimilarity

As we know from lecture, multivariate data with *p* variables are visually represented by a collection of points forming a data cloud in *p*-dimensional space. The shape, clumping, and dispersion of the data cloud contains information we seek to describe. Several distance and dissimilarity measures are used to calculate the distance between data points.

### Euclidean Distance:

**Euclidean** distance is one of the most commonly used distance measures. It is normally preceded by column standardization (e.g. z standardization). Let’s calculate Euclidean distance for the US air pollution data set. You will use the function *vegdist* from the *vegan* (vegetation analysis) package. Look up *vegdist* to see the different indices available in this package.

?vegdist

First, z standardization:

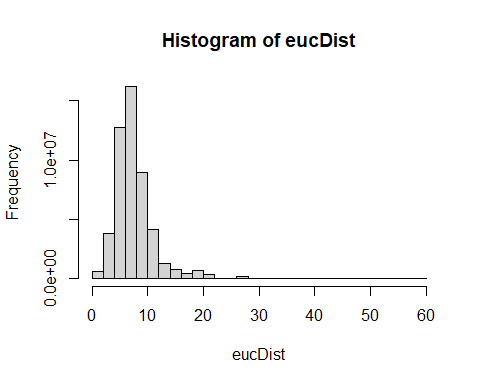
scaledData <- scale(bds.num)

Then calculate distance:

eucDist <- vegdist(scaledData,"euclidean")

Let’s look at a histogram of distances:

hist(eucDist)



mean(eucDist)

## [1] 7.471683

max(eucDist)

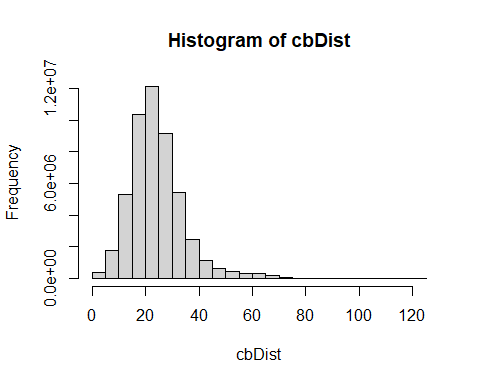
## [1] 58.42512

**Question 6: What does this frequency distribution tell you about pollution conditions across these 41 cities? (5 pts)**

*The histogram tells us that most sites/points are under 20 units away from one another in the data cloud. The majority of observations are clustered between approximately 5 and 10 on the histogram, with a mean of 7.47. The distribution of Euclidean distances is left-skewed, and the maximum Euclidean distance is 58.43.*

### City-block (Manhattan) distance

cbDist <- vegdist(scaledData,"manhattan")  
  
#Let’s look at a histogram of distances:  
  
hist(cbDist)



mean(cbDist)

## [1] 24.20289

max(cbDist)

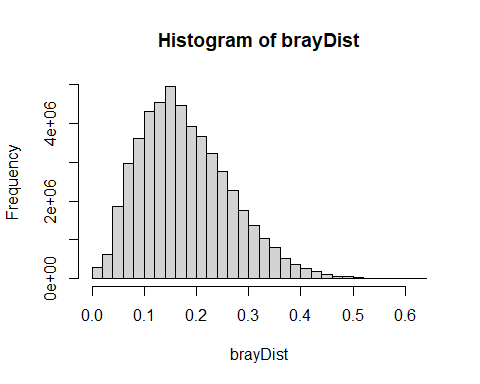
## [1] 122.0872

**Question 7: How does this distribution compare to Euclidean distance? (5 pts)**

*This histogram is also left-skewed. The mean (24.20) is higher than that of the Euclidean distance distribution, and the maximum distance is much higher (122.09).*

### Bray-Curtis dissimilarity

brayDist <- vegdist(bds.num,"bray")  
  
#Histogram:  
  
hist(brayDist)



**Back to multivariate outliers!**

Your goal here is to examine deviations of the sample average distances to other samples. We will use **Bray-Curtis** distance:

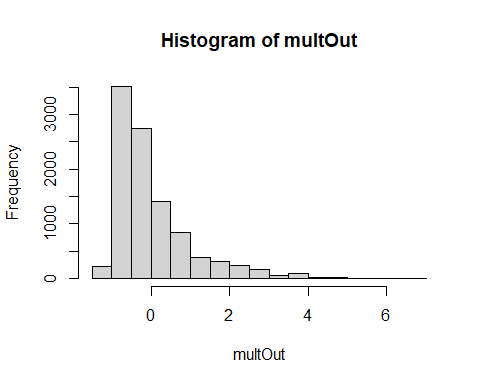
brayDist <- vegdist(bds.num,"bray")

Next, calculate column means. These column means represent the average dissimilarity of each city to all other cities. You want to know if any cities are on average more than 3 standard deviation units (z scores). To achieve this, z-transform the averages:

multOut <- scale(colMeans(as.matrix(brayDist)))

Plot a histogram and look for observations >3 sd units:

hist(multOut)



You can find the points that are outliers with:

multOut [multOut >3,]

## 110 126 244 285 313 319 349 395   
## 3.701844 3.702237 3.854322 3.716873 3.007819 6.485073 6.480941 3.000095   
## 406 525 602 608 627 639 655 707   
## 3.802926 4.107455 3.864265 4.980593 3.994138 3.584157 3.573978 3.015550   
## 720 879 1123 1217 1261 1318 1375 1386   
## 3.851563 3.219841 3.592703 3.013871 4.149358 3.985972 3.990850 4.178719   
## 1427 1454 1466 1493 1526 1532 1551 1595   
## 3.597959 3.486940 3.681551 4.566020 3.028061 4.098130 3.162208 3.577535   
## 1599 1635 1766 1780 1833 1836 1837 1862   
## 3.912190 4.628771 3.981147 3.828226 3.548032 4.121005 4.151240 4.126758   
## 1866 1924 1938 1979 2074 2093 2223 2307   
## 3.814107 3.484493 3.186648 3.581180 3.160216 3.787761 4.947259 4.976691   
## 2311 2321 2354 2416 2428 2476 2490 2515   
## 3.507094 3.979390 3.926987 3.686382 3.536844 3.762059 3.022284 4.982627   
## 2586 2657 2734 2853 2858 2861 2969 3032   
## 3.612168 3.743624 4.086675 4.687989 3.561420 3.642122 3.583597 3.743491   
## 3099 3118 3184 3186 3196 3357 3374 3433   
## 4.108441 3.592126 3.020832 4.959821 3.820732 4.641507 4.611308 4.143542   
## 3621 3629 3653 3680 3698 3782 3797 3942   
## 3.576102 3.734212 3.836509 3.895151 3.507997 3.607590 3.006880 3.858958   
## 4039 4138 4146 4164 4223 4267 4311 4330   
## 3.216026 4.028739 3.168606 3.679430 4.548354 3.965863 3.003132 4.676390   
## 4397 4404 4464 4504 4605 4732 4756 4822   
## 3.814496 4.121622 3.906196 3.816372 3.030255 3.041085 3.018885 4.690383   
## 4844 4855 4895 5048 5167 5172 5226 5227   
## 3.994327 3.546548 3.753039 3.022848 3.127100 3.846478 3.208492 3.024150   
## 5289 5434 5473 5513 5526 5556 5626 5655   
## 3.901054 4.973215 3.020741 3.621069 3.027135 3.491800 3.142801 4.127909   
## 5746 5787 5854 5885 5916 5934 6007 6089   
## 3.023068 3.868798 3.024488 4.587380 4.684235 4.101009 3.995715 3.006466   
## 6099 6138 6167 6219 6313 6317 6390 6432   
## 3.682394 3.177187 4.706273 3.006389 3.595027 3.706822 3.023446 3.627571   
## 6501 6577 6697 6728 6767 6859 6861 6866   
## 3.167846 3.726235 3.682678 3.848589 4.124210 4.156209 3.831373 3.817241   
## 6954 6962 6992 7015 7019 7028 7049 7297   
## 3.720225 4.956561 3.196274 4.132934 3.664794 3.986941 4.567341 4.550690   
## 7352 7376 7407 7549 7558 7664 7685 7718   
## 3.747386 3.026140 3.877075 3.179184 4.129068 3.858554 3.842322 3.194380   
## 7761 7853 7906 7938 7939 7954 7974 8078   
## 3.015199 4.161127 3.903405 4.611089 3.134264 3.856936 3.542322 3.752858   
## 8093 8198 8207 8374 8388 8643 8664 8681   
## 3.869279 3.015853 3.829681 3.120268 3.741951 3.699050 3.844925 3.802644   
## 8697 8757 8766 8812 8851 8887 8964 8989   
## 3.804622 6.451803 4.014260 3.142951 4.202418 6.506063 3.577307 3.720430   
## 9025 9035 9174 9261 9330 9370 9462 9466   
## 4.986657 4.641295 3.568761 3.589342 3.810937 3.789473 3.144216 4.969462   
## 9482 9494 9606 9633 9738 9808 9868 9894   
## 3.891853 3.014371 6.461128 3.508444 3.119097 3.601074 3.586233 3.889989   
## 9956   
## 4.193399

Another possibility is to determine which observation are > 3 standard deviations from the mean. Using Bray-Curtis distance again:

Calculate column means:

colBray <- colMeans(as.matrix(brayDist))

Calculate the mean of the column means:

mBray <- mean(colBray)

Calculate the standard deviation:

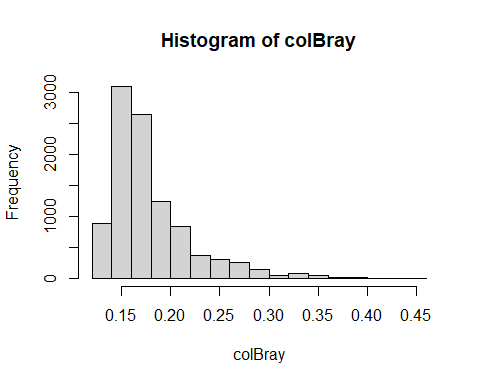
stdBray <- sd(colBray)

… 3 standard deviations

threeSD <- stdBray \* 3 + mBray

plot a histogram and look for observations >3 sd:

hist(colBray)



Find the outliers:

outliers <- colBray[colBray > threeSD]  
length(outliers)

## [1] 193

*There are 193 outlier points/sites.*