Lab 3 Matrix Algebra and Ordination Part I

Lara Katz

2024-02-02

# Set up R session

## Download packages

Install and load packages

library(MVA)  
library(psych)  
library(Hmisc)  
library(vegan)  
library(StatMatch)  
library(MASS)  
library(tidyverse)  
library(MVN) # Mardia's test

# A primer of matrix algebra

Let’s start by making our own matrix:

newMatrix <- matrix(c(1,4,5,4,5,6,9,1,9),nrow=3, ncol=3)  
newMatrix

## [,1] [,2] [,3]  
## [1,] 1 4 9  
## [2,] 4 5 1  
## [3,] 5 6 9

The command c concatenates a list of numbers.

Now let’s check the dimensions of newMatrix:

dim(newMatrix)

## [1] 3 3

## Matrix addition and subtraction

## Question 1: Create a new matrix to either add to or subtract from “newMatrix.”

This new matrix should be a 3 x 3 matrix containing all ones and call it oneMatrix. (15 pts)

oneMatrix <- matrix(rep(1, times = 9), nrow = 3, ncol = 3)  
oneMatrix

## [,1] [,2] [,3]  
## [1,] 1 1 1  
## [2,] 1 1 1  
## [3,] 1 1 1

Now add oneMatrix to newMatrix:

newMatrix + oneMatrix

## [,1] [,2] [,3]  
## [1,] 2 5 10  
## [2,] 5 6 2  
## [3,] 6 7 10

Then subtract oneMatrix from newMatrix:

newMatrix - oneMatrix

## [,1] [,2] [,3]  
## [1,] 0 3 8  
## [2,] 3 4 0  
## [3,] 4 5 8

**Remember, because matrix addition and subtraction is performed on an element by element basis, matrices must have the same dimensions.**

## Scalar Multiplication

A **Scalar** is a single number. Scalar multiplication multiplies a scalar times a matrix:

3\*newMatrix

An *eigenvalue* is a scalar that is an essential component of multivariate analysis. We will explore this in a little bit.

## Matrix Multiplication

You use % to signify that are using a matrix operation. Otherwise, R will attempt the operation element by element.

oneMatrix%\*%newMatrix

order matters:

newMatrix%\*%oneMatrix

**The number of columns in the first matrix must equal the number of rows in the second matrix.**

## Matrix transposition

**Transposing** a matrix involves interchanging its rows and columns:

transMatrix<-t(newMatrix)  
transMatrix

## [,1] [,2] [,3]  
## [1,] 1 4 5  
## [2,] 4 5 6  
## [3,] 9 1 9

## Identity Matrices

An **identity matrix** is a matrix where all the diagonal terms equal one and the remaining elements equal 0:

Identity<-diag(3)  
Identity

## [,1] [,2] [,3]  
## [1,] 1 0 0  
## [2,] 0 1 0  
## [3,] 0 0 1

## Matrix Inversion

The inverse of matrix A is A-1.

invMatrix<-solve(newMatrix)  
invMatrix

## [,1] [,2] [,3]  
## [1,] -0.4148936 -0.1914894 0.4361702  
## [2,] 0.3297872 0.3829787 -0.3723404  
## [3,] 0.0106383 -0.1489362 0.1170213

Multiplying a matrix by its inverse yields an identity matrix (A x A-1 = I):

invMatrix%\*%newMatrix

Let’s round it:

round(invMatrix%\*%newMatrix,10)

## Eigenvalues and eigenvectors

Remember that an eigenvalue is a special scalar and the associated eigenvector is a vector that are key components of PCA.

eig <- eigen(newMatrix)  
eig

## eigen() decomposition  
## $values  
## [1] 15.117887 -2.553192 2.435305  
##   
## $vectors  
## [,1] [,2] [,3]  
## [1,] 0.5718107 0.8818544 0.4112631  
## [2,] 0.3014690 -0.4472324 -0.8068102  
## [3,] 0.7629869 -0.1493853 0.4241698

# Principal Component Analysis (PCA)

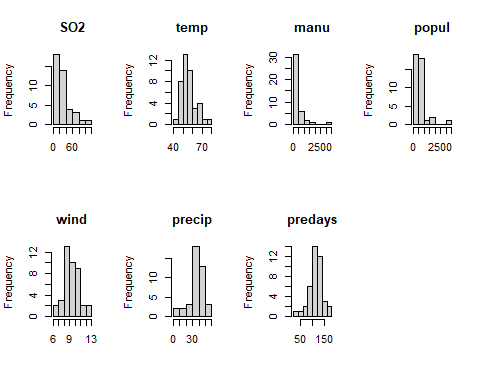
You are going to conduct a PCA on the USairpollution data that we used in Lab 2.

First let’s call in the data:

usAir <- USairpollution

Now, look at the distributions (i.e., histograms) of the variables to determine if they need to be transformed. You should be able to make the histograms and transform them based on what you learned during Lab 2.

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



Mardia’s test:

mvn(usAir, mvnTest = "mardia")

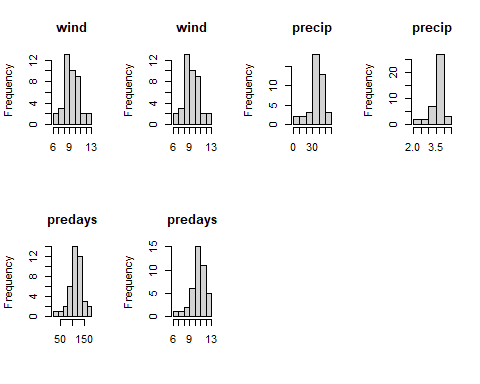
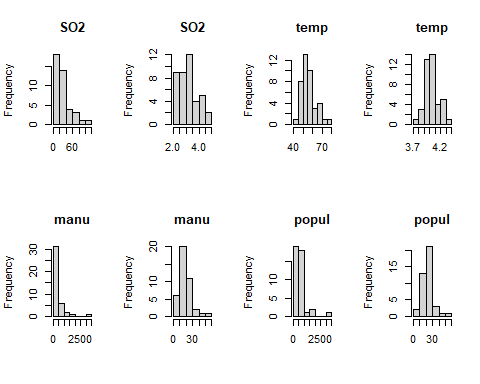
## $multivariateNormality  
## Test Statistic p value Result  
## 1 Mardia Skewness 226.612731693166 4.82491336101954e-15 NO  
## 2 Mardia Kurtosis 3.97754689564216 6.96298933924311e-05 NO  
## 3 MVN <NA> <NA> NO  
##   
## $univariateNormality  
## Test Variable Statistic p value Normality  
## 1 Anderson-Darling 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

usAir.t <- transform(usAir,  
 SO2 = log1p(SO2),   
 temp = log1p(temp),  
 manu = sqrt(manu), # square root for count data  
 popul = sqrt(popul), # count data  
 wind = wind, # no transformation, already normal  
 precip = log1p(precip),  
 predays = sqrt(predays) # count data  
 )  
head(usAir.t)

## SO2 temp manu popul wind precip predays  
## Albany 3.850148 3.883624 6.63325 10.77033 8.8 3.536893 11.618950  
## Albuquerque 2.484907 4.056989 6.78233 15.62050 8.9 2.171337 7.615773  
## Atlanta 3.218876 4.135167 19.18333 22.29350 9.1 3.898735 10.723805  
## Baltimore 3.871201 4.025352 25.00000 30.08322 9.6 3.745023 10.535654  
## Buffalo 2.484907 3.873282 19.77372 21.51743 12.4 3.613886 12.884099  
## Charleston 3.465736 4.028917 5.91608 8.42615 6.5 3.731699 12.165525

Compare histograms:

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



## Question 2: What variable did you transform, and what transformation did you use? (15 pts)

*I transformed SO2, temp, and precip using a log transformation because they are continuous variables. I transformed manu, popul, and predays using a square root transformation because these columns were made up of count data.*

If you do transform any variables, use the transformed data matrix going forward.

Next, apply a z-score standardization:

ZusAir <- scale(usAir.t)

# Running the PCA:

You are going to use the package princomp function in the stats package. Take some time to read about this function:

?princomp

Run the princomp function:

usAir\_pca <- princomp(ZusAir, cor = F)

cor = F, because you are using the covariance matrix instead of the correlation matrix.

**It should be noted that the covariance matrix of a z-standardized data matrix is equivalent to the correlation matrix of the unscaled data.**

Let’s look at a summary of our PCA:

summary(usAir\_pca)

## Importance of components:  
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5  
## Standard deviation 1.6078438 1.2694590 1.1059738 0.9925472 0.52876233  
## Proportion of Variance 0.3785415 0.2359735 0.1791082 0.1442541 0.04093991  
## Cumulative Proportion 0.3785415 0.6145150 0.7936232 0.9378773 0.97881721  
## Comp.6 Comp.7  
## Standard deviation 0.30669542 0.224946333  
## Proportion of Variance 0.01377338 0.007409411  
## Cumulative Proportion 0.99259059 1.000000000

Notice that the summary gives the standard deviation instead of the eigenvalue (variance). Let’s calculate the eigenvalues using what we know about the relationship between standard deviation and variance (var = sd^2):

eigenVal<- (usAir\_pca$sdev\*sqrt(41/40))^2  
eigenVal

## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7   
## 2.64979076 1.65181428 1.25375748 1.00977862 0.28657934 0.09641363 0.05186587

The *sqrt(41/40)* is to correct for the fact that princomp calculates variances with the divisor N instead of N-1 as is customary. This adjustment will allow direct comparison with “hand” calculated eigenvalues using the function eigen below. Note that these hand calculations are just to help you understand what is going on ‘under the hood’ in the model.

Let’s make the PCA table with the eigenvalues instead of the standard deviations:

propVar<-eigenVal/sum(eigenVal)  
cumVar<-cumsum(propVar)  
pca\_Table<-t(rbind(eigenVal,propVar,cumVar))  
pca\_Table

## eigenVal propVar cumVar  
## Comp.1 2.64979076 0.378541538 0.3785415  
## Comp.2 1.65181428 0.235973469 0.6145150  
## Comp.3 1.25375748 0.179108212 0.7936232  
## Comp.4 1.00977862 0.144254089 0.9378773  
## Comp.5 0.28657934 0.040939906 0.9788172  
## Comp.6 0.09641363 0.013773376 0.9925906  
## Comp.7 0.05186587 0.007409411 1.0000000

**This calculation and table are just to show you that the eigenvalues and the output from princomp, the standard deviations, are the same thing. YOU DO NOT NEED TO DO THIS WHEN YOU RUN A PCA - it is just to help you gain a deeper understanding of the model.**

the factor loadings from princomp:

loadings(usAir\_pca)

##   
## Loadings:  
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7  
## SO2 0.444 0.142 0.214 0.508 0.650 0.195 0.133  
## temp -0.354 -0.181 -0.664 0.303 0.537 -0.123  
## manu 0.484 -0.434 -0.150 -0.153 -0.720  
## popul 0.389 -0.537 -0.252 -0.247 0.657  
## wind 0.283 0.112 -0.846 0.374 0.222   
## precip 0.222 0.440 -0.632 -0.101 0.184 -0.551 0.102  
## predays 0.402 0.523 -0.128 -0.496 0.543   
##   
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7  
## SS loadings 1.000 1.000 1.000 1.000 1.000 1.000 1.000  
## Proportion Var 0.143 0.143 0.143 0.143 0.143 0.143 0.143  
## Cumulative Var 0.143 0.286 0.429 0.571 0.714 0.857 1.000

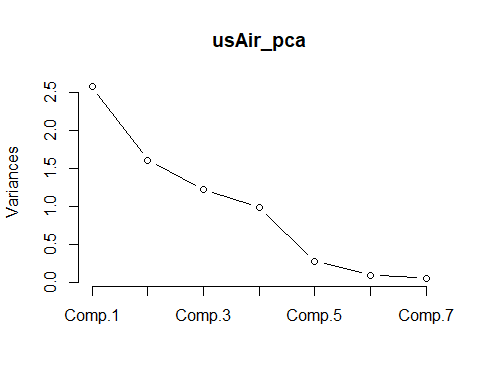
and the factor scores:

scores(usAir\_pca)

## Determining number of axes to keep

You now have 7 PC axes. Which ones give us vital information and which ones can we toss? One method for selecting the number of Axes is a **Scree plot**:

plot(usAir\_pca, type="lines")



How about the **latent root criterion (i.e., keep components with eigenvalues > 1)** and the **relative percent variance criteria**. Check the summary of the PCA explore this:

eigenVal

## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7   
## 2.64979076 1.65181428 1.25375748 1.00977862 0.28657934 0.09641363 0.05186587

pca\_Table

## eigenVal propVar cumVar  
## Comp.1 2.64979076 0.378541538 0.3785415  
## Comp.2 1.65181428 0.235973469 0.6145150  
## Comp.3 1.25375748 0.179108212 0.7936232  
## Comp.4 1.00977862 0.144254089 0.9378773  
## Comp.5 0.28657934 0.040939906 0.9788172  
## Comp.6 0.09641363 0.013773376 0.9925906  
## Comp.7 0.05186587 0.007409411 1.0000000

summary(usAir\_pca)

## Importance of components:  
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5  
## Standard deviation 1.6078438 1.2694590 1.1059738 0.9925472 0.52876233  
## Proportion of Variance 0.3785415 0.2359735 0.1791082 0.1442541 0.04093991  
## Cumulative Proportion 0.3785415 0.6145150 0.7936232 0.9378773 0.97881721  
## Comp.6 Comp.7  
## Standard deviation 0.30669542 0.224946333  
## Proportion of Variance 0.01377338 0.007409411  
## Cumulative Proportion 0.99259059 1.000000000

## Question 3: How many axes you should keep and why? (15 points)

* *Latent root criterion: This stopping rule drops eigenvalues less than one. Components 5, 6, and 7 would be dropped here, but since there are < 20 variables this method will keep a conservative number of components.*
* *Scree plot criterion: The scree plot criterion indicates the maximum number of components to extract by plotting the eigenvalues against the component number. The scree plot begins to level off after PCA4, so components 1 through 4 should be retained.*
* *Relative percent variance criterion: According to this criterion I would keep the first four PCA axes because, cumulatively, they explain 93.8% of the variation in the data. I could also just keep the first three PCA axes if I needed to simplify the results for a secondary model because they explain > 70% of the variation in the data (79.4%).*

## Significance of factor loadings.

While many use the “rule of thumb” where a loading > 0.30 dictates an “important” variable. Another method for determining significance of factor loadings is bootstrapping. Details and comparisons of the many way to assess significance of factor loadings are presented in Peres-Neto et al. (2003), which is supplemental reading for this week. Here we will run the method that they found to have the lowest type I error rates, Bootstrapped eigenvector. For reference, this is the method 6 in Peres-Neto et al. (2003).

sigpca2<-function (x, permutations=1000, ...)  
{  
 pcnull <- princomp(x, ...)  
 res <- pcnull$loadings  
 out <- matrix(0, nrow=nrow(res), ncol=ncol(res))  
 N <- nrow(x)  
 for (i in 1:permutations) {  
 pc <- princomp(x[sample(N, replace=TRUE), ], ...)  
 pred <- predict(pc, newdata = x)  
 r <- cor(pcnull$scores, pred)  
 k <- apply(abs(r), 2, which.max)  
 reve <- sign(diag(r[k,]))  
 sol <- pc$loadings[ ,k]  
 sol <- sweep(sol, 2, reve, "\*")  
 out <- out + ifelse(res > 0, sol <= 0, sol >= 0)  
 }  
 out/permutations  
}  
  
set.seed(4)  
   
sigpca2(ZusAir, permutations=1000)

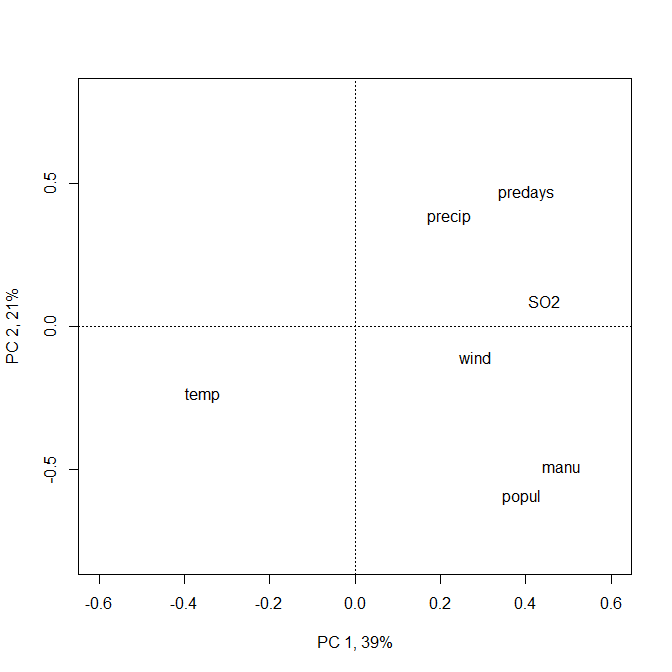
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7  
## SO2 0.002 0.287 0.165 0.133 0.014 0.058 0.091  
## temp 0.023 0.371 0.159 0.346 0.027 0.005 0.259  
## manu 0.023 0.171 0.346 0.268 0.269 0.240 0.031  
## popul 0.074 0.147 0.280 0.388 0.026 0.396 0.037  
## wind 0.045 0.390 0.479 0.166 0.023 0.011 0.492  
## precip 0.214 0.194 0.158 0.468 0.074 0.009 0.261  
## predays 0.017 0.124 0.379 0.450 0.021 0.014 0.287

#Piece-by-piece (read along step-by-step with pg. 2350 section 6) Bootstrapped eigenvector (V vectors)” of Peres-Neto et al. 2003.  
pcnull<-princomp(ZusAir)  
res <- pcnull$loadings  
out <- matrix(0, nrow=nrow(res), ncol=ncol(res))  
N <- nrow(ZusAir)  
pc<-princomp(ZusAir[sample(N, replace=TRUE), ])  
pred <- predict(pc, newdata = ZusAir)   
r <- cor(pcnull$scores, pred)  
k <- apply(abs(r), 2, which.max)  
reve <- sign(diag(r[k,]))  
sol <- pc$loadings[ ,k]  
sol <- sweep(sol, 2, reve, "\*")  
out <- out + ifelse(res > 0, sol <= 0, sol >= 0)

## PCA plots

Plot out the factor loadings for the first 2 PC axes:

plot(usAir\_pca$loadings,type="n",xlab="PC 1, 39%", ylab="PC 2, 21%",ylim=c(-.8,.8), xlim=c(-.6,.6))  
  
text(usAir\_pca$loadings, labels=as.character(colnames(ZusAir)), pos=1, cex=1)  
abline(h = 0, lty = 3)  
abline(v = 0, lty = 3)



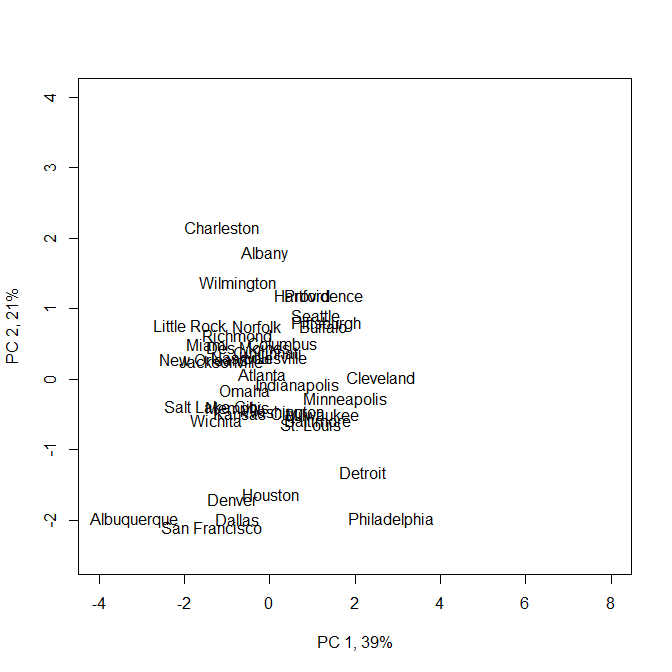
## Question 4: How do you interpret these axes? Come up with a name for each. (15 pts)

*PCA1 is being driven primarily by population parameters (manu and popul) and temperature (temp). PCA2 is primarily being driven by precipitation parameters (predays and precip) and population parameters (popul and manu). I would name PCA1 “Manufacturing-Temperature” and PCA2 “PrecipitationDays-Population.”*

**Close the plot window after viewing**

Let’s now plot the PC score for each sample (city):

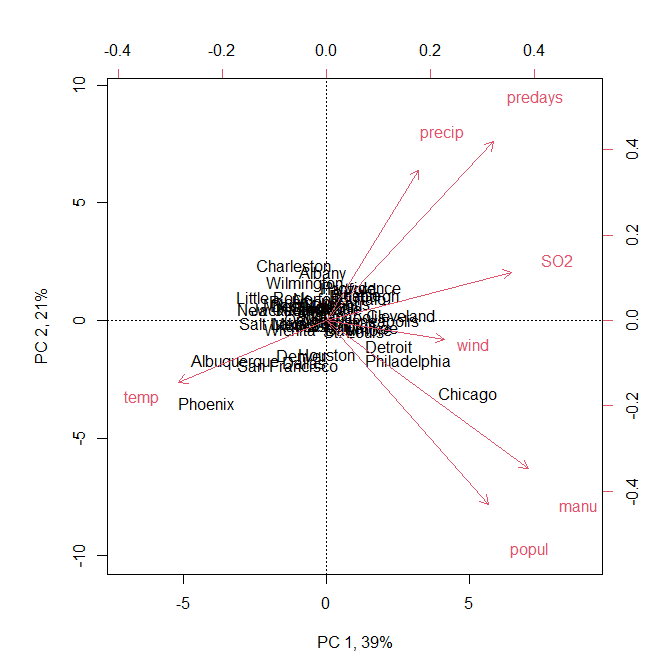
plot(usAir\_pca$scores,type="n",xlab="PC 1, 39%", ylab="PC 2, 21%",ylim=c(-2.5,4), xlim=c(-4,8))  
text(usAir\_pca$scores, labels=as.character(rownames(ZusAir)), pos=1, cex=1)



And now all together in a biplot:

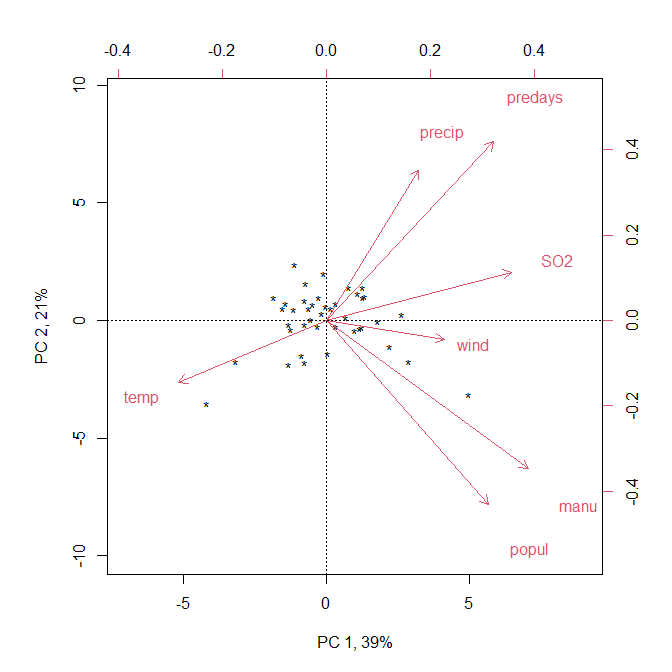
?biplot

biplot(usAir\_pca$scores,usAir\_pca$loading,xlab="PC 1, 39%", ylab="PC 2, 21%",expand= 4, ylim=c(-10,9.5), xlim=c(-7,9))  
abline(h = 0, lty = 3)  
abline(v = 0, lty = 3)



to replace city names with a symbol:

biplot(usAir\_pca$scores,usAir\_pca$loading, expand= 4, xlabs= rep("\*",41),xlab="PC 1, 39%", ylab="PC 2, 21%",ylim=c(-10,9.5), xlim=c(-7,9))  
abline(h = 0, lty = 3)  
abline(v = 0, lty = 3)



# Eigen Analysis

You can also just simply use the Eigen analysis function, eigen and calculate your own scores by hand. Note that I am showing this for illustration for those of you who want to have a deeper understanding of the method.

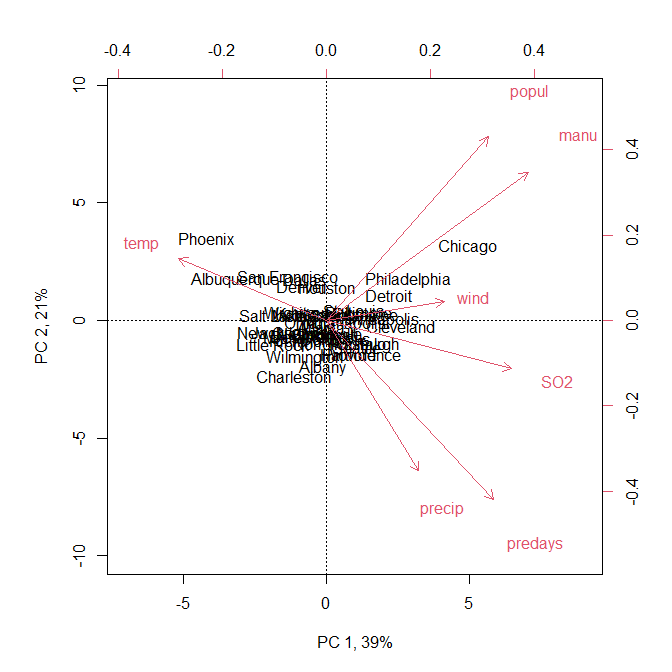
eig<-eigen(cov(ZusAir))

Extract the first two eigenvectors (because that is what we are interested in plotting):

eigVec<-as.matrix(eig$vectors[,1:2])  
rownames(eigVec) <- rownames(cov(ZusAir))

Then simply multiply each eigenvector times the matrix of standardized observation values (ZusAir) and plot!

scores<-t(rbind(eigVec[,1]%\*%t(ZusAir),eigVec[,2]%\*%t(ZusAir)))#hand calculated scores  
  
# and plot  
  
biplot(scores,eigVec,xlab="PC 1, 39%", ylab="PC 2, 21%",expand= 4, ylim=c(-10,9.5), xlim=c(-7,9))  
abline(h = 0, lty = 3)  
abline(v = 0, lty = 3)



## Question 5: Does your individual dataset (the one used in Lab 2) meet the assumptions of PCA? Is PCA an analysis you could use on your data? (40 pts)

*I cannot use PCA on my data because, even with variable transformations (below), they do not meet the assumption of multivariate normality. The majority of the data points are independent pseudo-absence points generated by terra, but the Bridle Shiner survey locations were sampled in person and most were not random (i.e., spatial autocorrelation). Taking a stratified random sample of the in-person survey locations would allow the data to meet the independent sample assumption. The data do meet some of the data requirements for PCA: There are more samples (n = 10,000) than there are columns (n = 38), which meets the rule of thumb that n 3x38. The majority of the variables are continuous, three are categorical, and the response variable is binary (I removed the binary response variable when I standardized the data). Every sample entity is measured for the same set of variables and there are no NA values.*

Read in dataset and define data types:

bds <- read.csv("./Data/Katz\_BDS\_data.csv", header = TRUE, row.names = 1)  
bds <- bds[,-2]  
bds$catchment <- as.factor(bds$catchment) # categorical variables  
bds$lith <- as.factor(bds$lith)  
bds$marine <- as.factor(bds$marine)  
bds$ph <- bds$ph/10 # convert integer raster values back to pH scale

Convert proportional data to percentages between 0 and 1:

for (i in 9:35){ # proportion of HUC12 covered by LANDFIRE cover class  
 bds[,i] <- bds[,i]/100 # convert to percentages between 0 and 1  
}  
summary(bds[,9:35])

Transform data based on data type (log-transformation for continuous, arcsine square root for percentages):

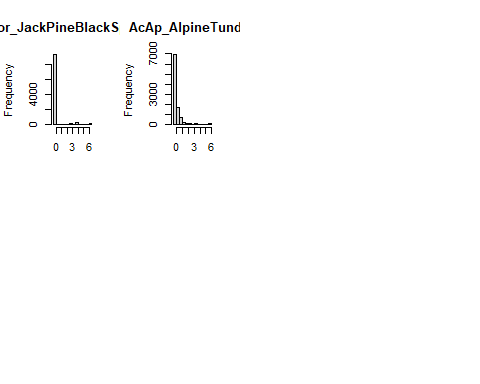
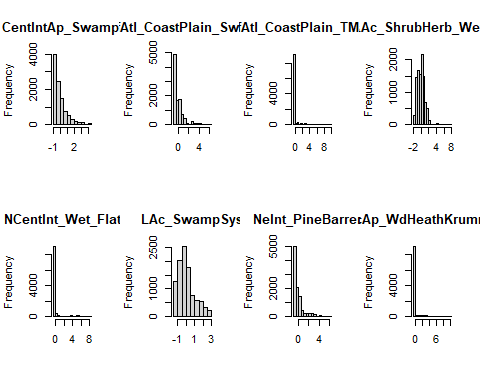
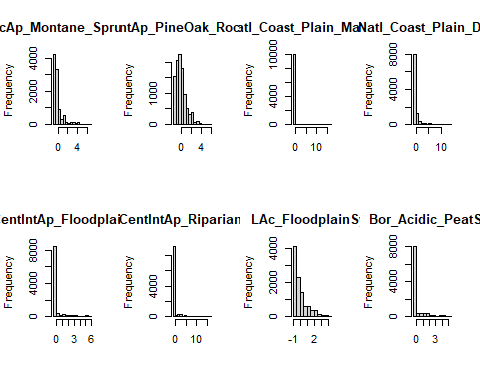
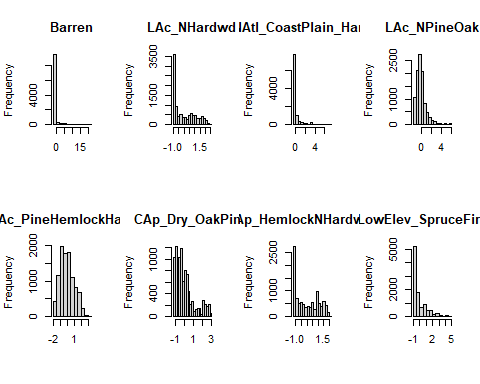
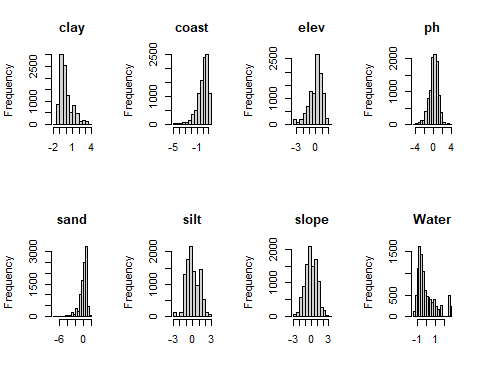
bds.t <- transform(bds,  
 pab = pab, # binary response, no transformation  
 catchment = catchment, # categorical  
 clay = log1p(clay), # log transformation  
 coast = log1p(coast), # log transformation  
 elev = log1p(elev), # log transformation  
 lith = lith,  
 marine = marine,  
 ph = log1p(ph), # log transformation  
 Water = asin(sqrt(X01\_water)), # arcsine square root for percentages  
 Barren = asin(sqrt(X02\_Barren)),  
 LAc\_NHardwd = asin(sqrt(X03\_LAc\_NHardwd)),  
 NAtl\_CoastPlain\_Hardwd = asin(sqrt(X05\_NAtl\_CoastPlain\_Hardwd)),  
 LAc\_NPineOak = asin(sqrt(X07\_LAc\_NPine.Oak)),  
 LAc\_PineHemlockHardwd = asin(sqrt(X08\_LAc\_Pine.Hemlock.Hardwd)),  
 CAp\_Dry\_OakPine = asin(sqrt(X09\_CAp\_Dry\_Oak.Pine)),  
 Ap\_HemlockNHardwood = asin(sqrt(X10\_Ap\_Hemlock.N\_Hardwood)),  
 Ac\_LowElev\_SpruceFirHardwd = asin(sqrt(X11\_Ac\_Low.Elev\_Spruce.Fir.Hardwood)),  
 AcAp\_Montane\_SpruceFir = asin(sqrt(X12\_AcAp\_Montane\_Spruce.Fir)),  
 CentAp\_PineOak\_Rocky\_Wd = asin(sqrt(X13\_CentAp\_Pine.Oak\_Rocky\_Wd)),  
 Natl\_Coast\_Plain\_Maritime = asin(sqrt(X14\_NAtl\_Coast\_Plain\_Maritime)),  
 Natl\_Coast\_Plain\_Dune = asin(sqrt(X15\_NAtl\_Coast\_Plain\_Dune)),  
 CentIntAp\_FloodplainSys = asin(sqrt(X16\_CentIntAp\_FloodplainSys)),  
 CentIntAp\_RiparianSys = asin(sqrt(X17\_CentIntAp\_RiparianSys)),  
 LAc\_FloodplainSys = asin(sqrt(X18\_LAc\_FloodplainSys)),  
 Bor\_Acidic\_PeatSys = asin(sqrt(X19\_Bor\_Acidic\_PeatSys)),  
 CentIntAp\_SwampSys = asin(sqrt(X20\_CentIntAp\_SwampSys)),  
 GulfAtl\_CoastPlain\_SwampSys = asin(sqrt(X21\_GulfAtl\_CoastPlain\_SwampSys)),  
 GulfAtl\_CoastPlain\_TMarshSys = asin(sqrt(X22\_GulfAtl\_CoastPlain\_TMarshSys)),  
 LAc\_ShrubHerb\_WetlSys = asin(sqrt(X23\_LAc\_Shrub.Herb\_WetlSys)),  
 NCentInt\_Wet\_Flatwd = asin(sqrt(X24\_NCentInt\_Wet\_Flatwd)),  
 LAc\_SwampSys = asin(sqrt(X25\_LAc\_SwampSys)),  
 NeInt\_PineBarrens = asin(sqrt(X26\_NeInt\_PineBarrens)),  
 AcAp\_WdHeathKrummholz = asin(sqrt(X27\_AcAp\_WdHeath.Krummholz)),  
 Bor\_JackPineBlackSpruce = asin(sqrt(X28\_Bor\_JackPine.BlackSpruce)),  
 AcAp\_AlpineTundra = asin(sqrt(X29\_AcAp\_AlpineTundra)),  
 sand = log1p(sand), # log transformation  
 silt = log1p(silt), # log transformation  
 slope = log1p(slope) # log transformation  
 )  
bds.t <- (bds.t[,-c(9:35)]) # remove original columns

Scale data:

require(tidyverse)  
bds.num <- bds.t %>% select(-pab, -catchment, -lith, -marine)  
scaledData <- scale(bds.num)

Look at histograms:

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



Mardia’s test for multivariate normality (scaled data):

mvn(scaledData, mvnTest = "mardia")

## $multivariateNormality  
## Test Statistic p value Result  
## 1 Mardia Skewness 1470099.9834051 0 NO  
## 2 Mardia Kurtosis 1023.13441480936 0 NO  
## 3 MVN <NA> <NA> NO  
##   
## $univariateNormality  
## Test Variable Statistic p value Normality  
## 1 Anderson-Darling clay 394.0066 <0.001 NO   
## 2 Anderson-Darling coast 284.7068 <0.001 NO   
## 3 Anderson-Darling elev 150.2363 <0.001 NO   
## 4 Anderson-Darling ph 60.6150 <0.001 NO   
## 5 Anderson-Darling sand 414.8435 <0.001 NO   
## 6 Anderson-Darling silt 89.8063 <0.001 NO   
## 7 Anderson-Darling slope 16.6699 <0.001 NO   
## 8 Anderson-Darling Water 515.0883 <0.001 NO   
## 9 Anderson-Darling Barren 3374.6704 <0.001 NO   
## 10 Anderson-Darling LAc\_NHardwd 553.6798 <0.001 NO   
## 11 Anderson-Darling NAtl\_CoastPlain\_Hardwd 1868.1637 <0.001 NO   
## 12 Anderson-Darling LAc\_NPineOak 295.5882 <0.001 NO   
## 13 Anderson-Darling LAc\_PineHemlockHardwd 89.0778 <0.001 NO   
## 14 Anderson-Darling CAp\_Dry\_OakPine 457.9486 <0.001 NO   
## 15 Anderson-Darling Ap\_HemlockNHardwood 366.5684 <0.001 NO   
## 16 Anderson-Darling Ac\_LowElev\_SpruceFirHardwd 1056.7777 <0.001 NO   
## 17 Anderson-Darling AcAp\_Montane\_SpruceFir 1369.5958 <0.001 NO   
## 18 Anderson-Darling CentAp\_PineOak\_Rocky\_Wd 286.5584 <0.001 NO   
## 19 Anderson-Darling Natl\_Coast\_Plain\_Maritime 3822.8117 <0.001 NO   
## 20 Anderson-Darling Natl\_Coast\_Plain\_Dune 2088.0407 <0.001 NO   
## 21 Anderson-Darling CentIntAp\_FloodplainSys 2629.9858 <0.001 NO   
## 22 Anderson-Darling CentIntAp\_RiparianSys 3052.7322 <0.001 NO   
## 23 Anderson-Darling LAc\_FloodplainSys 489.6470 <0.001 NO   
## 24 Anderson-Darling Bor\_Acidic\_PeatSys 2111.3944 <0.001 NO   
## 25 Anderson-Darling CentIntAp\_SwampSys 643.0533 <0.001 NO   
## 26 Anderson-Darling GulfAtl\_CoastPlain\_SwampSys 1047.4052 <0.001 NO   
## 27 Anderson-Darling GulfAtl\_CoastPlain\_TMarshSys 3102.0165 <0.001 NO   
## 28 Anderson-Darling LAc\_ShrubHerb\_WetlSys 55.0068 <0.001 NO   
## 29 Anderson-Darling NCentInt\_Wet\_Flatwd 3168.5796 <0.001 NO   
## 30 Anderson-Darling LAc\_SwampSys 197.6993 <0.001 NO   
## 31 Anderson-Darling NeInt\_PineBarrens 1298.3802 <0.001 NO   
## 32 Anderson-Darling AcAp\_WdHeathKrummholz 3038.3646 <0.001 NO   
## 33 Anderson-Darling Bor\_JackPineBlackSpruce 3290.3170 <0.001 NO   
## 34 Anderson-Darling AcAp\_AlpineTundra 1460.1162 <0.001 NO   
##   
## $Descriptives  
## n Mean Std.Dev Median  
## clay 10000 -1.176815e-15 1 -0.33794720  
## coast 10000 4.323910e-16 1 0.24206266  
## elev 10000 1.040609e-16 1 0.24166314  
## ph 10000 2.562818e-15 1 0.07671340  
## sand 10000 8.948340e-16 1 0.25210538  
## silt 10000 -1.053198e-15 1 -0.17203301  
## slope 10000 2.276857e-17 1 -0.05928892  
## Water 10000 -1.782934e-17 1 -0.36585441  
## Barren 10000 6.533676e-18 1 -0.18834861  
## LAc\_NHardwd 10000 3.258904e-17 1 -0.35524372  
## NAtl\_CoastPlain\_Hardwd 10000 3.360763e-17 1 -0.42595422  
## LAc\_NPineOak 10000 2.976314e-17 1 -0.17544858  
## LAc\_PineHemlockHardwd 10000 -1.204661e-16 1 -0.12772850  
## CAp\_Dry\_OakPine 10000 -2.011910e-17 1 -0.31193735  
## Ap\_HemlockNHardwood 10000 3.310061e-17 1 -0.16605341  
## Ac\_LowElev\_SpruceFirHardwd 10000 5.865471e-17 1 -0.54985775  
## AcAp\_Montane\_SpruceFir 10000 4.084758e-17 1 -0.39744950  
## CentAp\_PineOak\_Rocky\_Wd 10000 -1.233656e-17 1 -0.21441239  
## Natl\_Coast\_Plain\_Maritime 10000 2.899692e-18 1 -0.08919197  
## Natl\_Coast\_Plain\_Dune 10000 -4.128675e-17 1 -0.37317879  
## CentIntAp\_FloodplainSys 10000 -2.728416e-18 1 -0.34329319  
## CentIntAp\_RiparianSys 10000 -1.821021e-17 1 -0.23104039  
## LAc\_FloodplainSys 10000 2.407803e-17 1 -0.28966935  
## Bor\_Acidic\_PeatSys 10000 -9.304753e-18 1 -0.43417854  
## CentIntAp\_SwampSys 10000 -4.014171e-17 1 -0.39087249  
## GulfAtl\_CoastPlain\_SwampSys 10000 6.622133e-18 1 -0.47788002  
## GulfAtl\_CoastPlain\_TMarshSys 10000 -1.436430e-17 1 -0.22076130  
## LAc\_ShrubHerb\_WetlSys 10000 -5.980082e-17 1 0.02213274  
## NCentInt\_Wet\_Flatwd 10000 1.548169e-17 1 -0.23032783  
## LAc\_SwampSys 10000 -8.031024e-17 1 -0.23353800  
## NeInt\_PineBarrens 10000 2.529204e-17 1 -0.47920801  
## AcAp\_WdHeathKrummholz 10000 -9.414992e-18 1 -0.24424578  
## Bor\_JackPineBlackSpruce 10000 2.991531e-19 1 -0.25843618  
## AcAp\_AlpineTundra 10000 1.928498e-17 1 -0.49028551  
## Min Max 25th 75th  
## clay -1.70706364 3.633503 -0.70001354 0.401148107  
## coast -4.96670424 1.300961 -0.47763207 0.751299560  
## elev -3.22457641 2.056560 -0.62897741 0.734525153  
## ph -4.47988596 3.782812 -0.55989537 0.609281328  
## sand -7.08925089 1.551365 -0.40481704 0.690161464  
## silt -2.89635519 2.854138 -0.79983925 0.800612645  
## slope -2.52251500 3.160389 -0.67939639 0.758142123  
## Water -1.34770861 2.614660 -0.70146831 0.489007052  
## Barren -0.18834861 21.920208 -0.18834861 -0.188348609  
## LAc\_NHardwd -0.95421411 2.433893 -0.95421411 0.821009205  
## NAtl\_CoastPlain\_Hardwd -0.42595422 7.026418 -0.42595422 -0.055768549  
## LAc\_NPineOak -1.29888082 5.998294 -0.62058790 0.328412261  
## LAc\_PineHemlockHardwd -1.51156319 3.025624 -0.75564862 0.736150546  
## CAp\_Dry\_OakPine -1.06010372 2.893975 -0.75474765 0.341058984  
## Ap\_HemlockNHardwood -1.16655929 2.008577 -1.03227082 0.836299034  
## Ac\_LowElev\_SpruceFirHardwd -0.67120833 4.507430 -0.67120833 0.471031300  
## AcAp\_Montane\_SpruceFir -0.55224079 6.931622 -0.55224079 -0.009712615  
## CentAp\_PineOak\_Rocky\_Wd -1.08696194 5.994131 -0.76244567 0.377990484  
## Natl\_Coast\_Plain\_Maritime -0.08919197 16.530171 -0.08919197 -0.089191967  
## Natl\_Coast\_Plain\_Dune -0.37317879 13.063061 -0.37317879 -0.129554174  
## CentIntAp\_FloodplainSys -0.34329319 5.524994 -0.34329319 -0.343293193  
## CentIntAp\_RiparianSys -0.23104039 16.316880 -0.23104039 -0.231040389  
## LAc\_FloodplainSys -0.97613966 4.075604 -0.69527578 0.367139898  
## Bor\_Acidic\_PeatSys -0.43417854 5.097003 -0.43417854 -0.387560438  
## CentIntAp\_SwampSys -0.82466503 4.242597 -0.72396936 0.494720704  
## GulfAtl\_CoastPlain\_SwampSys -0.62044628 6.425342 -0.62044628 0.288550504  
## GulfAtl\_CoastPlain\_TMarshSys -0.22076130 9.535339 -0.22076130 -0.220761302  
## LAc\_ShrubHerb\_WetlSys -1.67392607 7.617638 -0.84679500 0.609093999  
## NCentInt\_Wet\_Flatwd -0.23032783 8.155918 -0.23032783 -0.230327831  
## LAc\_SwampSys -1.45818901 2.789402 -0.67915689 0.452854145  
## NeInt\_PineBarrens -0.57498334 6.015587 -0.57498334 0.149019108  
## AcAp\_WdHeathKrummholz -0.24424578 10.004718 -0.24424578 -0.244245784  
## Bor\_JackPineBlackSpruce -0.25843618 6.423541 -0.25843618 -0.258436183  
## AcAp\_AlpineTundra -0.49028551 5.771801 -0.49028551 0.108448873  
## Skew Kurtosis  
## clay 1.21389751 0.9722056  
## coast -1.56652144 3.2351084  
## elev -0.96112292 0.9203044  
## ph -0.53656626 1.4515493  
## sand -1.96326505 5.4310026  
## silt 0.17566411 -0.3704002  
## slope 0.04515633 -0.2879537  
## Water 1.29340916 0.8700766  
## Barren 7.75215265 86.0676763  
## LAc\_NHardwd 0.65323021 -0.9563786  
## NAtl\_CoastPlain\_Hardwd 3.71949403 15.9678537  
## LAc\_NPineOak 2.04409064 7.1195940  
## LAc\_PineHemlockHardwd 0.40947346 -0.6573998  
## CAp\_Dry\_OakPine 1.26033166 0.7865864  
## Ap\_HemlockNHardwood 0.29287744 -1.3738279  
## Ac\_LowElev\_SpruceFirHardwd 1.80780358 2.9793772  
## AcAp\_Montane\_SpruceFir 2.83967678 9.0931306  
## CentAp\_PineOak\_Rocky\_Wd 1.46567143 3.1274244  
## Natl\_Coast\_Plain\_Maritime 11.50465515 133.4110670  
## Natl\_Coast\_Plain\_Dune 4.74394984 31.3193013  
## CentIntAp\_FloodplainSys 3.46031362 12.1003460  
## CentIntAp\_RiparianSys 8.18687226 102.5656187  
## LAc\_FloodplainSys 1.41498106 1.5944484  
## Bor\_Acidic\_PeatSys 2.75591008 7.6063207  
## CentIntAp\_SwampSys 1.80068796 3.3498489  
## GulfAtl\_CoastPlain\_SwampSys 2.64990830 8.3869167  
## GulfAtl\_CoastPlain\_TMarshSys 6.53019396 49.4620613  
## LAc\_ShrubHerb\_WetlSys 1.00423264 3.8365173  
## NCentInt\_Wet\_Flatwd 5.37972728 30.8236637  
## LAc\_SwampSys 0.84408263 0.1644315  
## NeInt\_PineBarrens 2.51869313 6.8191763  
## AcAp\_WdHeathKrummholz 5.66162114 37.7268774  
## Bor\_JackPineBlackSpruce 4.21947348 18.3871398  
## AcAp\_AlpineTundra 3.69901954 15.6954128