# Assignment 4

Katie Beidler 2/6/2015

1. Carry out a PCA on the Tallgrass prairie soil dataset.

```
library(vegan)
```

```
## Loading required package: permute
## Loading required package: lattice
## This is vegan 2.2-1
```

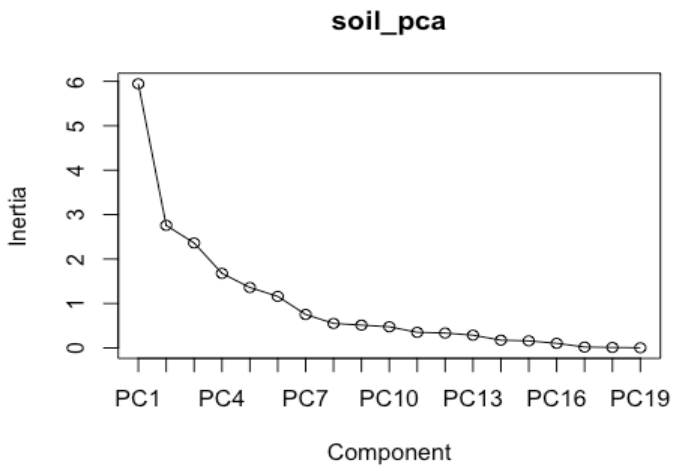
```
prairie_soil = read.csv('./TGPP_env.csv')
head(prairie_soil)
```

Which variables are most strongly loading on axis 1? Axis 2? How many axes are required to get the bulk of the variance (> 50%) of the soil variation?

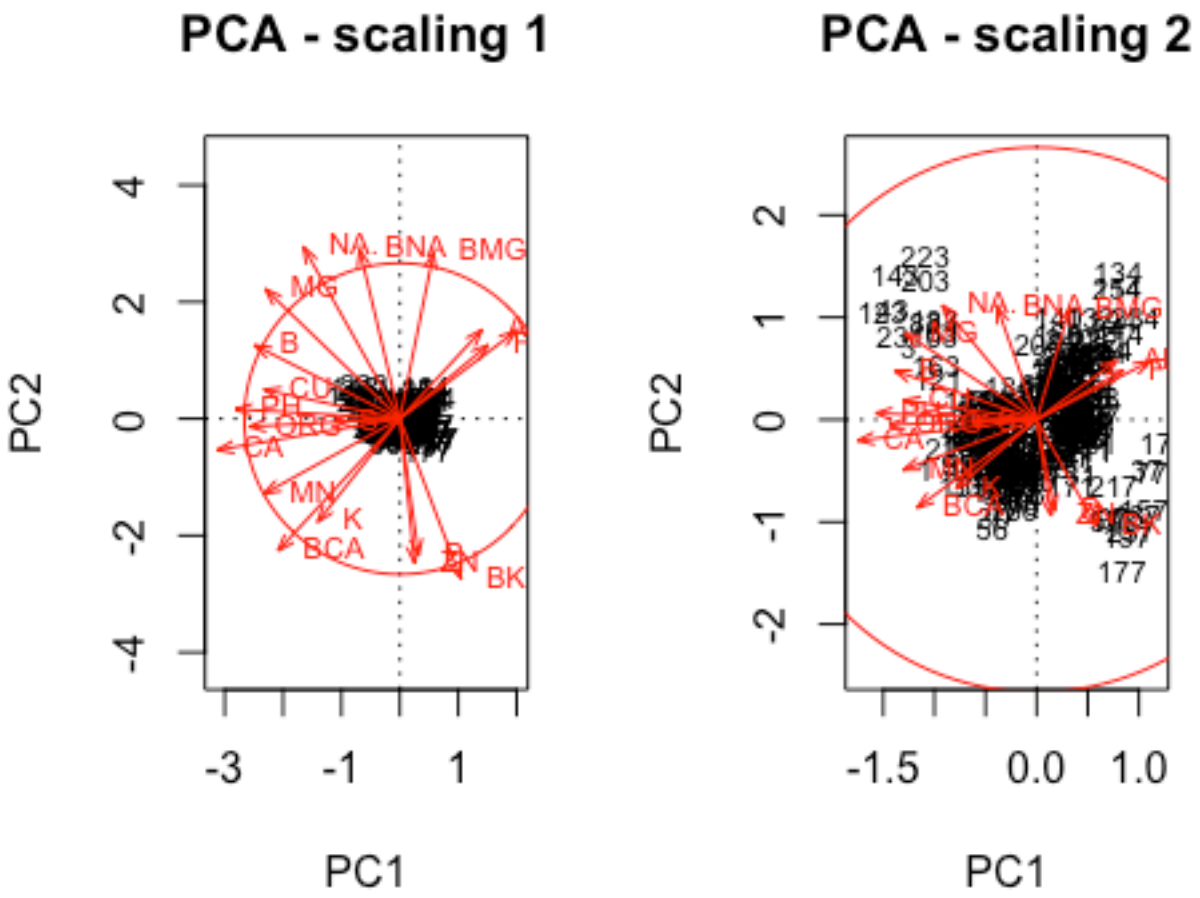
```
# Reducing the data down to only soil variables
TGPP_soil = rbind (prairie_soil[ , 17:35])
TGPP_soil_scaled = scale(TGPP_soil)
soil_pca = rda(TGPP_soil_scaled, scale=TRUE)

# PCA scores for different soil variables
soil_scores = summary(soil_pca)

# Eigen values for different axes
soil_pca
```

```
## Call: rda(X = TGPP_soil_scaled, scale = TRUE)
##
##               Inertia Rank
## Total              19
## Unconstrained      19   19
## Inertia is correlations
##
## Eigenvalues for unconstrained axes:
##   PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8
## 5.944 2.758 2.362 1.678 1.360 1.160 0.753 0.551
## (Showed only 8 of all 19 unconstrained eigenvalues)
```

```
screeplot(soil_pca, npcs = 19, type = "lines")
```



```
source("cleanplot.pca.R")
cleanplot.pca(soil_pca)
pcacircle(soil_pca)
```




- Soil variables most strongly loading on axis 1 include: pH, organic matter, calcium, magnisium, manganese, copper, boron and saturation of calcium.
- Soil variables most strongly loading on axis 2 include: phosphorus, sodium, saturation of magnesium, saturation of potassium, saturation of sodium, and zinc.
- To get the bulk of the variance (>50%) you would need to include the first 3 axes (58%). It is hard to tell from the Scree Plot which principle components to include, the amount of variance explained drops off quickly after PC6.

How and why might you use the soil PCA axes rather than the raw soil variables in a model?

- Variables correlated with each principle component or axis may be grouped and characterized. For instance, the variables that load with PC1 include factors that affect soil quality (organic matter and pH). PC1 may serve as a new variable related to soil quality. By creating a new variable that includes multiple raw variables, the use of PCA axes can help reduce the number of terms in a model.

2.Examine the following for loop, and then complete the exercises

```
data(iris)
head(iris)

##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
```

```
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
sp_ids = unique(iris$Species)

output = matrix(0, nrow=length(sp_ids), ncol=ncol(iris)-1)
rownames(output) = sp_ids
colnames(output) = names(iris[ , -ncol(iris)])

for(i in seq_along(sp_ids)) {
   iris_sp = subset(iris, subset=Species == sp_ids[i], select=-Species)
   for(j in 1:(ncol(iris_sp))) {
      x = 0
      y = 0
      if (nrow(iris_sp) > 0) {
         for(k in 1:nrow(iris_sp)) {
            x = x + iris_sp[k, j]
              # add together the cells in a column for each sp
            y = y + 1 # for each species subset add the number of rows in column 1
         }
         output[i, j] = x / y
      }
   }
}
output
```

```
##            Sepal.Length Sepal.Width Petal.Length Petal.Width
## setosa            5.006       3.428        1.462       0.246
## versicolor        5.936       2.770        4.260       1.326
## virginica         6.588       2.974        5.552       2.026
```

Describe the values stored in the object output. In other words what did the loops create?

- The values stored in the object output are means or Avg. sepal/ petal length and width for each species.

Describe using pseudo-code how output was calculated, for example,

- Loop through ... the unique species IDs in the iris data set and subset the data for each species-remove the now unessecary species column to make the new data frame (iris_sp)
    - Loop through ... all of the columns in the newly subsetted data frame (iris_sp) and set x & y equal to zero if the row number is greater than zero (aka exclude the header)
        - Loop through...all of the rows in iris_sp and sum the value of the cells for each species- making a new object x- that contains the sum + 0, also, make a new object y that sums the number of rows in column 1 for each species

The variables in the loop were named so as to be vague. How can the objects output, x, and y could be renamed such that it is clearer what is occurring in the loop.

- x could be renamed to: sum_sp
- y could be renamed to: total_individuals

Is it possible to accomplish the same task using fewer lines of code? Please suggest one other way to calculate output that decreases the number of loops by 1.

```
sp_ids = unique(iris$Species)
mean_table = matrix(0, nrow=length(sp_ids), ncol=ncol(iris)-1)
rownames(mean_table) = sp_ids
colnames(mean_table) = names(iris[ , -ncol(iris)])
```

```
for(i in seq_along(sp_ids)) {
  iris_sp = subset(iris, subset=Species == sp_ids[i], select=-Species)
     for(j in 1:ncol(iris_sp)) {
       output[i,j] = sum(iris_sp[,j])/length(iris_sp[,j])

     }
 }
output
```

```
##            Sepal.Length Sepal.Width Petal.Length Petal.Width
## setosa            5.006       3.428        1.462       0.246
## versicolor        5.936       2.770        4.260       1.326
## virginica         6.588       2.974        5.552       2.026
```