Multivariate models assignment

Emily Parsons

February 4, 2020

1. Conduct an indirect ordination on the dune plant community. Specifically, visually examine a NMDS plot using the bray-curtis distance metric. Below is some code to help you develop a potential plot that emphasizes the role of the environmental variable “Moisture”. Describe how you interpret the graphic. What is the goal of creating such a plot? Does this analysis suggest any interesting findings with respect to the dune vegetation?

```
library(vegan)
```

```
## Warning: package 'vegan' was built under R version 3.5.3
```

```
## Loading required package: permute
```

```
## Warning: package 'permute' was built under R version 3.5.3
```

```
## Loading required package: lattice
```

```
## Warning: package 'lattice' was built under R version 3.5.3
```

```
## This is vegan 2.5-6
```

```
data(dune)
data(dune.env)
?dune
```

```
## starting httpd help server ...
```

```
##  done
```

```
head(dune)
```

```
##   Achimill Agrostol Airaprae Alopgeni Anthodor Bellpere Bromhord Chenalbu
## 1        1        0        0        0        0        0        0        0
## 2        3        0        0        2        0        3        4        0
## 3        0        4        0        7        0        2        0        0
## 4        0        8        0        2        0        2        3        0
## 5        2        0        0        0        4        2        2        0
## 6        2        0        0        0        3        0        0        0
##   Cirsarve Comapalu Eleopalu Elymrepe Empenigr Hyporadi Juncarti Juncbufo
## 1        0        0        0        4        0        0        0        0
## 2        0        0        0        4        0        0        0        0
## 3        0        0        0        4        0        0        0        0
## 4        2        0        0        4        0        0        0        0
## 5        0        0        0        4        0        0        0        0
## 6        0        0        0        0        0        0        0        0
##   Lolipere Planlanc Poaprat Poatriv Ranuflam Rumeacet Sagiproc Salirepe
## 1        7        0       4       2        0        0        0        0
## 2        5        0       4       7        0        0        0        0
## 3        6        0       5       6        0        0        0        0
## 4        5        0       4       5        0        0        5        0
## 5        2        5       2       6        0        5        0        0
## 6        6        5       3       4        0        6        0        0
##   Scorautu Trifprat Trifrepe Vicilath Bracruta Callcusp
## 1        0        0        0        0        0        0
## 2        5        0        5        0        0        0
## 3        2        0        2        0        2        0
## 4        2        0        1        0        2        0
## 5        3        2        2        0        2        0
## 6        3        5        5        0        6        0
```

```
head(dune.env)
```

```
##    A1 Moisture Management      Use Manure
## 1 2.8        1         SF Haypastu      4
## 2 3.5        1         BF Haypastu      2
## 3 4.3        2         SF Haypastu      4
## 4 4.2        2         SF Haypastu      4
## 5 6.3        1         HF Hayfield      2
## 6 4.3        1         HF Haypastu      2
```

```
dune
```

```
##    Achimill Agrostol Airaprae Alopgeni Anthodor Bellpere Bromhord Chenalbu
## 1         1        0        0        0        0        0        0        0
## 2         3        0        0        2        0        3        4        0
## 3         0        4        0        7        0        2        0        0
## 4         0        8        0        2        0        2        3        0
## 5         2        0        0        0        4        2        2        0
## 6         2        0        0        0        3        0        0        0
## 7         2        0        0        0        2        0        2        0
## 8         0        4        0        5        0        0        0        0
## 9         0        3        0        3        0        0        0        0
## 10        4        0        0        0        4        2        4        0
## 11        0        0        0        0        0        0        0        0
## 12        0        4        0        8        0        0        0        0
## 13        0        5        0        5        0        0        0        1
## 14        0        4        0        0        0        0        0        0
## 15        0        4        0        0        0        0        0        0
## 16        0        7        0        4        0        0        0        0
## 17        2        0        2        0        4        0        0        0
## 18        0        0        0        0        0        2        0        0
## 19        0        0        3        0        4        0        0        0
## 20        0        5        0        0        0        0        0        0
##    Cirsarve Comapalu Eleopalu Elymrepe Empenigr Hyporadi Juncarti Juncbufo
## 1         0        0        0        4        0        0        0        0
## 2         0        0        0        4        0        0        0        0
## 3         0        0        0        4        0        0        0        0
## 4         2        0        0        4        0        0        0        0
## 5         0        0        0        4        0        0        0        0
## 6         0        0        0        0        0        0        0        0
## 7         0        0        0        0        0        0        0        2
## 8         0        0        4        0        0        0        4        0
## 9         0        0        0        6        0        0        4        4
## 10        0        0        0        0        0        0        0        0
## 11        0        0        0        0        0        2        0        0
## 12        0        0        0        0        0        0        0        4
## 13        0        0        0        0        0        0        0        3
## 14        0        2        4        0        0        0        0        0
## 15        0        2        5        0        0        0        3        0
## 16        0        0        8        0        0        0        3        0
## 17        0        0        0        0        0        2        0        0
## 18        0        0        0        0        0        0        0        0
## 19        0        0        0        0        2        5        0        0
## 20        0        0        4        0        0        0        4        0
##    Lolipere Planlanc Poaprat Poatriv Ranuflam Rumeacet Sagiproc Salirepe
## 1         7        0       4       2        0        0        0        0
## 2         5        0       4       7        0        0        0        0
## 3         6        0       5       6        0        0        0        0
## 4         5        0       4       5        0        0        5        0
## 5         2        5       2       6        0        5        0        0
## 6         6        5       3       4        0        6        0        0
## 7         6        5       4       5        0        3        0        0
## 8         4        0       4       4        2        0        2        0
## 9         2        0       4       5        0        2        2        0
## 10        6        3       4       4        0        0        0        0
```

```
## 11        7        3        4        0        0        0        2        0
## 12        0        0        0        4        0        2        4        0
## 13        0        0        2        9        2        0        2        0
## 14        0        0        0        0        2        0        0        0
## 15        0        0        0        0        2        0        0        0
## 16        0        0        0        2        2        0        0        0
## 17        0        2        1        0        0        0        0        0
## 18        2        3        3        0        0        0        0        3
## 19        0        0        0        0        0        0        3        3
## 20        0        0        0        0        4        0        0        5
##    Scorautu Trifprat Trifrepe Vicilath Bracruta Callcusp
## 1         0        0        0        0        0        0
## 2         5        0        5        0        0        0
## 3         2        0        2        0        2        0
## 4         2        0        1        0        2        0
## 5         3        2        2        0        2        0
## 6         3        5        5        0        6        0
## 7         3        2        2        0        2        0
## 8         3        0        2        0        2        0
## 9         2        0        3        0        2        0
## 10        3        0        6        1        2        0
## 11        5        0        3        2        4        0
## 12        2        0        3        0        4        0
## 13        2        0        2        0        0        0
## 14        2        0        6        0        0        4
## 15        2        0        1        0        4        0
## 16        0        0        0        0        4        3
## 17        2        0        0        0        0        0
## 18        5        0        2        1        6        0
## 19        6        0        2        0        3        0
## 20        2        0        0        0        4        3
```

```
dune_pca <- rda(dune, scale=TRUE)
dune_pca
```

```
## Call: rda(X = dune, scale = TRUE)
##
##               Inertia Rank
## Total              30
## Unconstrained      30   19
## Inertia is correlations
##
## Eigenvalues for unconstrained axes:
##   PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8
## 7.032 4.997 3.555 2.644 2.139 1.758 1.478 1.316
## (Showing 8 of 19 unconstrained eigenvalues)
```

Out of 30 7% is explained by PC1 etc

```
str(dune_pca)
```

```
## List of 10
##  $ colsum        : Named num [1:30] 1 1 1 1 1 1 1 1 1 1 ...
##   ..- attr(*, "names")= chr [1:30] "Achimill" "Agrostol" "Airaprae" "Alopgeni" ...
##  $ tot.chi       : num 30
##  $ Ybar          : num [1:20, 1:30] 0.037 0.407 -0.148 -0.148 0.222 ...
##   ..- attr(*, "scaled:center")= Named num [1:30] 0.8 2.4 0.25 1.8 1.05 0.65 0.75 0.05 0.1 0.2
...
##   .. ..- attr(*, "names")= chr [1:30] "Achimill" "Agrostol" "Airaprae" "Alopgeni" ...
##   ..- attr(*, "scaled:scale")= Named num [1:30] 1.24 2.683 0.786 2.628 1.701 ...
##   .. ..- attr(*, "names")= chr [1:30] "Achimill" "Agrostol" "Airaprae" "Alopgeni" ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:20] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:30] "Achimill" "Agrostol" "Airaprae" "Alopgeni" ...
##   ..- attr(*, "METHOD")= chr "PCA"
##  $ method        : chr "rda"
##  $ call          : language rda(X = dune, scale = TRUE)
##  $ pCCA          : NULL
##  $ CCA           : NULL
##  $ CA            :List of 7
##   ..$ eig    : Named num [1:19] 7.03 5 3.55 2.64 2.14 ...
##   .. ..- attr(*, "names")= chr [1:19] "PC1" "PC2" "PC3" "PC4" ...
##   ..$ poseig : NULL
##   ..$ u      : num [1:20, 1:19] 0.0456 0.2757 0.0557 0.043 0.2893 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:20] "1" "2" "3" "4" ...
##   .. .. ..$ : chr [1:19] "PC1" "PC2" "PC3" "PC4" ...
##   ..$ v      : num [1:30, 1:19] 0.27736 -0.27045 0.00162 -0.09937 0.20229 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:30] "Achimill" "Agrostol" "Airaprae" "Alopgeni" ...
##   .. .. ..$ : chr [1:19] "PC1" "PC2" "PC3" "PC4" ...
##   ..$ rank   : int 19
##   ..$ tot.chi: num 30
##   ..$ Xbar   : num [1:20, 1:30] 0.037 0.407 -0.148 -0.148 0.222 ...
##   .. ..- attr(*, "scaled:center")= Named num [1:30] 0.8 2.4 0.25 1.8 1.05 0.65 0.75 0.05 0.1
0.2 ...
##   .. .. ..- attr(*, "names")= chr [1:30] "Achimill" "Agrostol" "Airaprae" "Alopgeni" ...
##   .. ..- attr(*, "scaled:scale")= Named num [1:30] 1.24 2.683 0.786 2.628 1.701 ...
##   .. .. ..- attr(*, "names")= chr [1:30] "Achimill" "Agrostol" "Airaprae" "Alopgeni" ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:20] "1" "2" "3" "4" ...
##   .. .. ..$ : chr [1:30] "Achimill" "Agrostol" "Airaprae" "Alopgeni" ...
##   .. ..- attr(*, "METHOD")= chr "PCA"
##  $ inertia       : chr "correlations"
##  $ regularization: chr "this is a vegan::rda result object"
##  - attr(*, "class")= chr [1:2] "rda" "cca"
```

```
dune_pca$CA$eig
```

```
##        PC1        PC2        PC3        PC4        PC5        PC6 
## 7.03244773 4.99731801 3.55476518 2.64404798 2.13891282 1.75781309 
##        PC7        PC8        PC9       PC10       PC11       PC12 
## 1.47833687 1.31640259 1.10787495 0.80897860 0.74526734 0.69659526 
##       PC13       PC14       PC15       PC16       PC17       PC18 
## 0.57488468 0.35795895 0.22253446 0.21974437 0.15071227 0.13190726 
##       PC19 
## 0.06349759
```

```
sum(dune_pca$CA$eig)
```

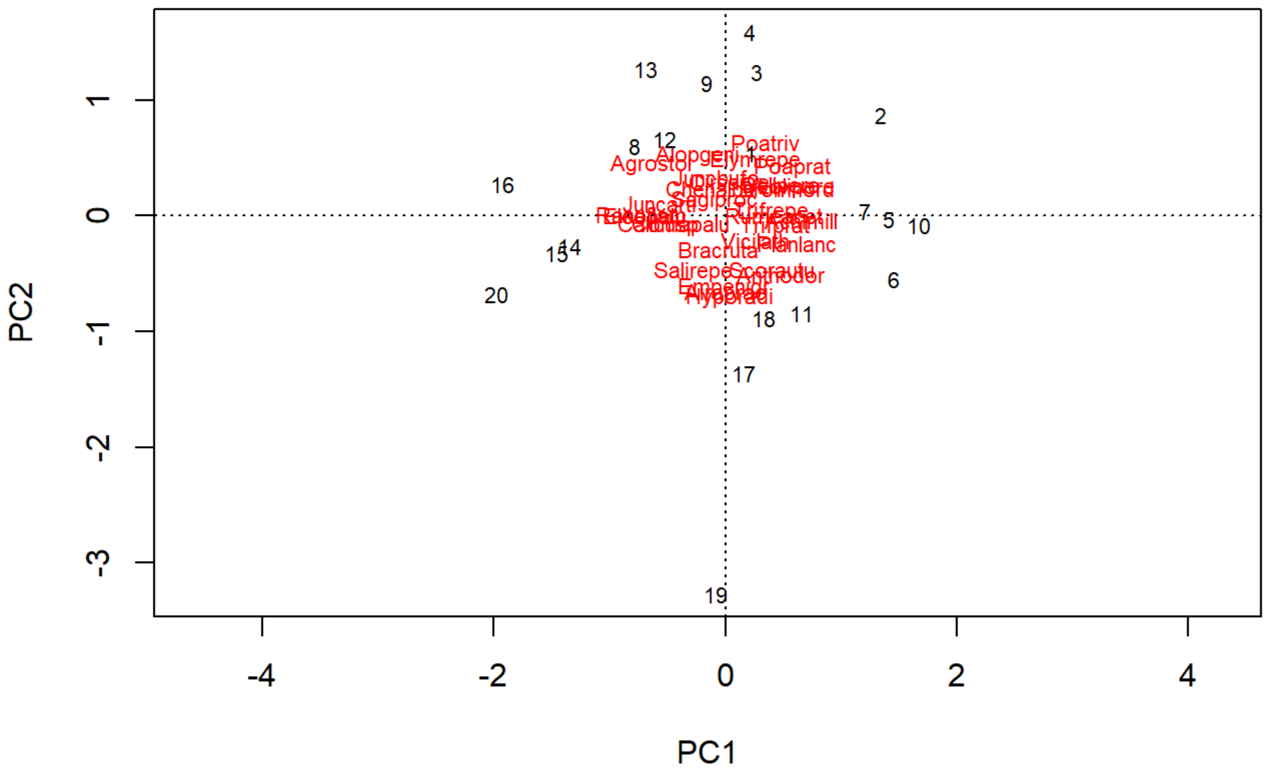
```
## [1] 30
```

```
round(dune_pca$CA$eig/dune_pca$tot.chi, 2)
```

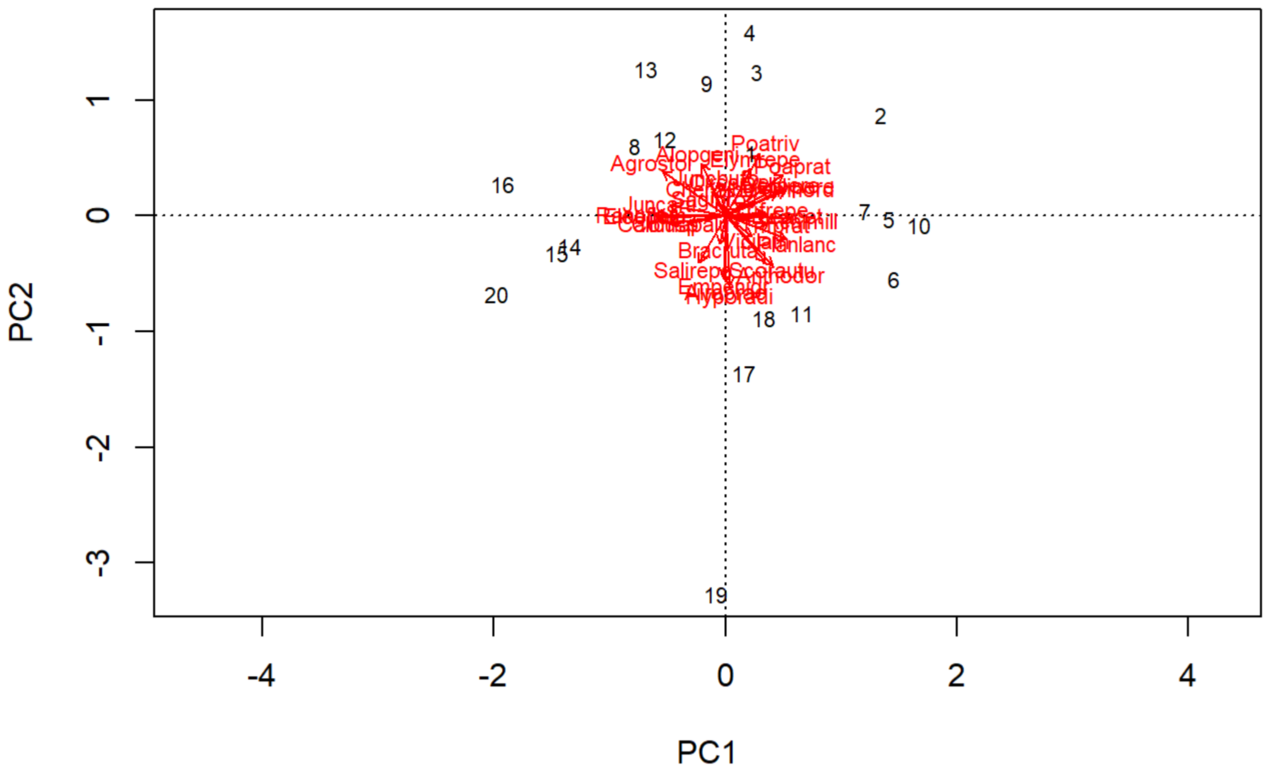
```
##  PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9 PC10 PC11 PC12 PC13 PC14 PC15 
## 0.23 0.17 0.12 0.09 0.07 0.06 0.05 0.04 0.04 0.03 0.02 0.02 0.02 0.01 0.01 
## PC16 PC17 PC18 PC19 
## 0.01 0.01 0.00 0.00
```

PC1 and PC2 explain 23% and 17% of the vaiance

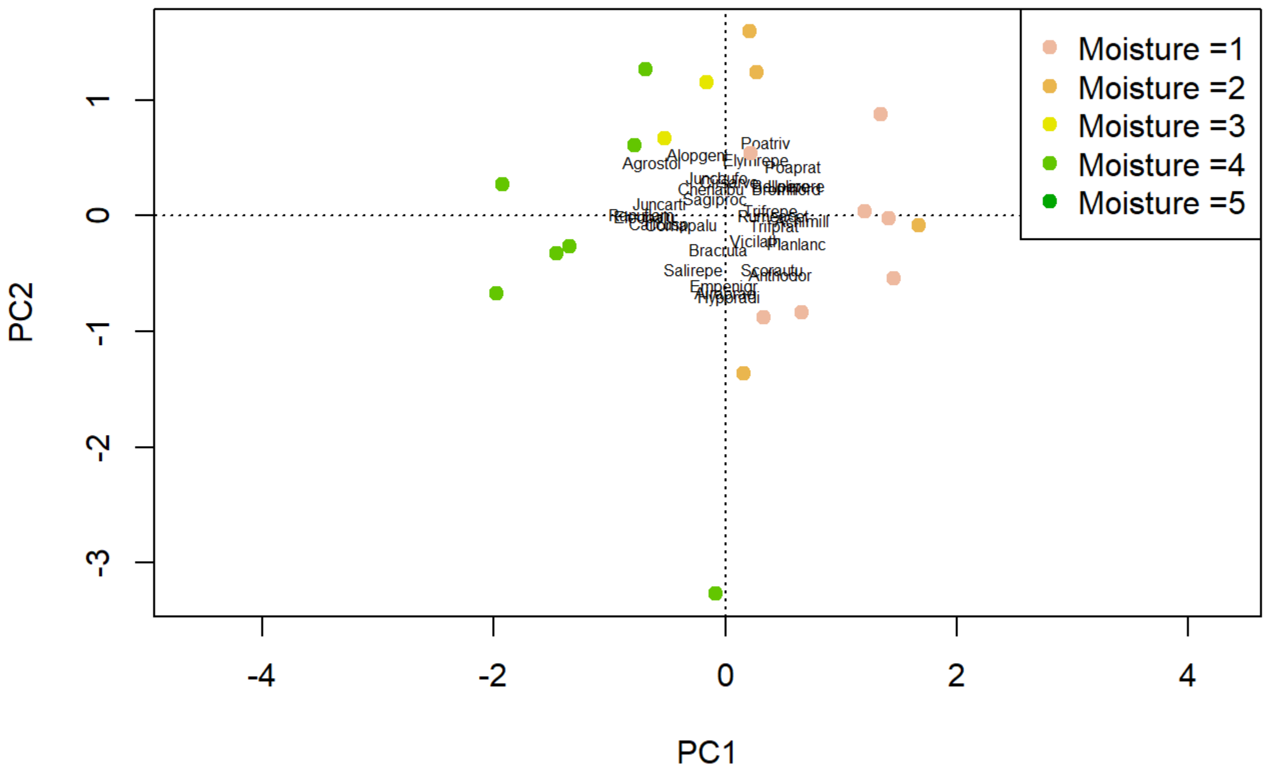
```
plot(dune_pca)
```




```
biplot(dune_pca)
```




```
plot(dune_pca, type='n')
 text(dune_pca, 'sp', cex=.5)
 # generate vector of colors
 color_vect = rev(terrain.colors(6))[-1]
 points(dune_pca, 'sites', pch=19,
        col=color_vect[dune.env$Moisture])
 legend('topright', paste("Moisture =", 1:5, sep=''),
        col=color_vect, pch=19)
```



Putting our user-defined distance matrix. . . putting it in ordination space…. so the computer can find the best arragement between the rank orders of the two distances.

points close have similar composition, and plots further apart are different.

if a variable (species) plots close to a sample, that indicates that the sample has high values of that variable. so this is telling us which species is more common towards higher or lower moisture

It appears that most of the variation is explained by PC2? The moisture increases as it goes along the x-axis.

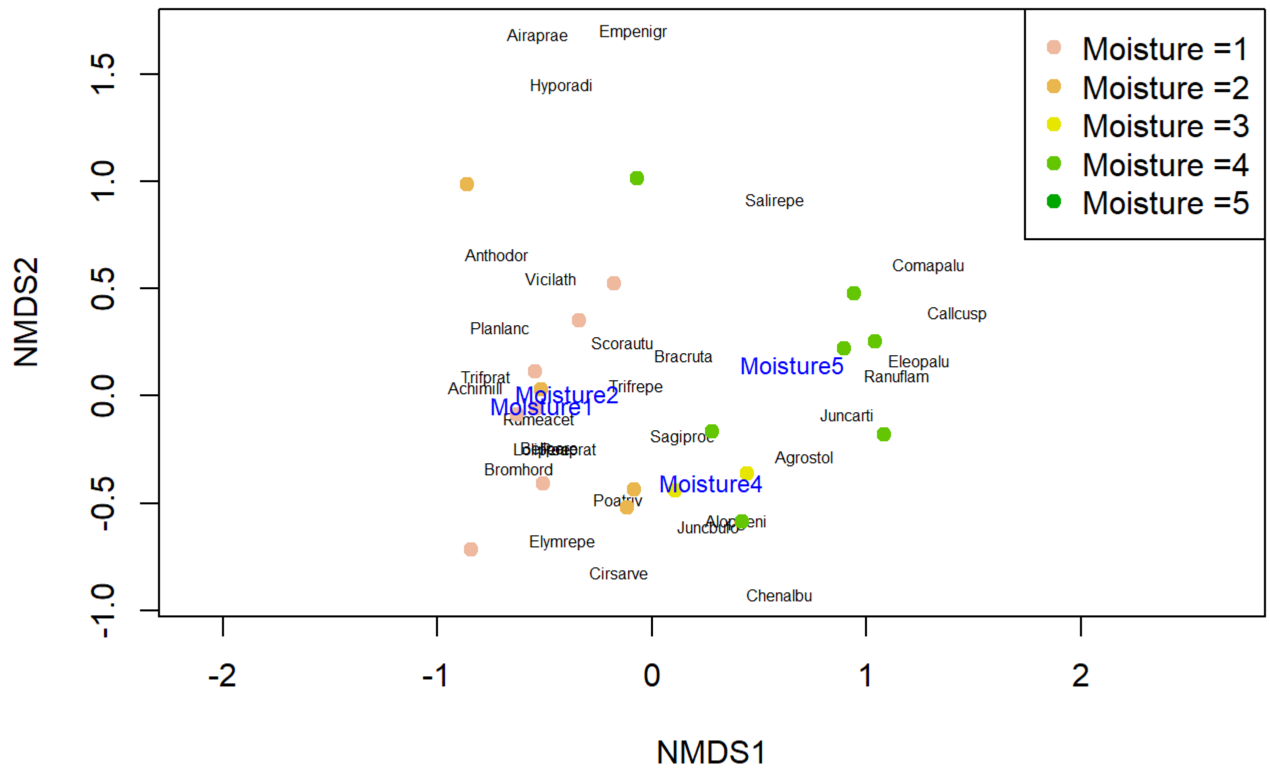
```
dune_mds <- metaMDS(dune)
```

```
## Run 0 stress 0.1192678
## Run 1 stress 0.1183186
## ... New best solution
## ... Procrustes: rmse 0.02027027  max resid 0.06495624
## Run 2 stress 0.119268
## Run 3 stress 0.1183186
## ... New best solution
## ... Procrustes: rmse 5.23479e-06  max resid 1.204835e-05
## ... Similar to previous best
## Run 4 stress 0.1192678
## Run 5 stress 0.1183186
## ... Procrustes: rmse 2.037384e-05  max resid 6.375787e-05
## ... Similar to previous best
## Run 6 stress 0.1192679
## Run 7 stress 0.1183187
## ... Procrustes: rmse 5.588802e-05  max resid 0.0001807808
## ... Similar to previous best
## Run 8 stress 0.2045511
## Run 9 stress 0.1192679
## Run 10 stress 0.1183186
## ... Procrustes: rmse 4.889369e-06  max resid 1.246414e-05
## ... Similar to previous best
## Run 11 stress 0.1183186
## ... Procrustes: rmse 7.344584e-05  max resid 0.0001576426
## ... Similar to previous best
## Run 12 stress 0.119268
## Run 13 stress 0.1922252
## Run 14 stress 0.1183186
## ... Procrustes: rmse 3.815798e-05  max resid 0.0001163425
## ... Similar to previous best
## Run 15 stress 0.1812981
## Run 16 stress 0.1183186
## ... Procrustes: rmse 3.998817e-05  max resid 0.0001131281
## ... Similar to previous best
## Run 17 stress 0.1192679
## Run 18 stress 0.1183186
## ... Procrustes: rmse 3.794412e-05  max resid 0.0001202983
## ... Similar to previous best
## Run 19 stress 0.1886532
## Run 20 stress 0.1192678
## *** Solution reached
```

```
dune_fit <- envfit(dune_mds ~ A1 + Moisture, data=dune.env, perm=999)
dune_fit
```

```
##
## ***VECTORS
##
##      NMDS1   NMDS2     r2 Pr(>r)
## A1 0.96474 0.26320 0.3649  0.017 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
##
## ***FACTORS:
##
## Centroids:
##             NMDS1   NMDS2
## Moisture1 -0.5101 -0.0403
## Moisture2 -0.3938  0.0139
## Moisture4  0.2765 -0.4033
## Moisture5  0.6561  0.1476
##
## Goodness of fit:
##              r2 Pr(>r)
## Moisture 0.5014  0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
```

```
plot(dune_mds, type='n')
 text(dune_mds, 'sp', cex=.5)
 # generate vector of colors
 color_vect <- rev(terrain.colors(6))[-1]
 points(dune_mds, 'sites', pch=19,
        col=color_vect[dune.env$Moisture])
 plot(dune_fit, p.max = 0.01, cex=.75)
 legend('topright', paste("Moisture =", 1:5, sep=''),
        col=color_vect, pch=19)
```



2. Carry out a direct ordination using CCA in order to test any potential hypotheses that you developed after examining the MDS plot. Specifically, carry out a test of the entire model (i.e., including all constrained axes) and also carry out tests at the scale of individual explanatory variables you included in your model if you included more than one variable. Plot your results.

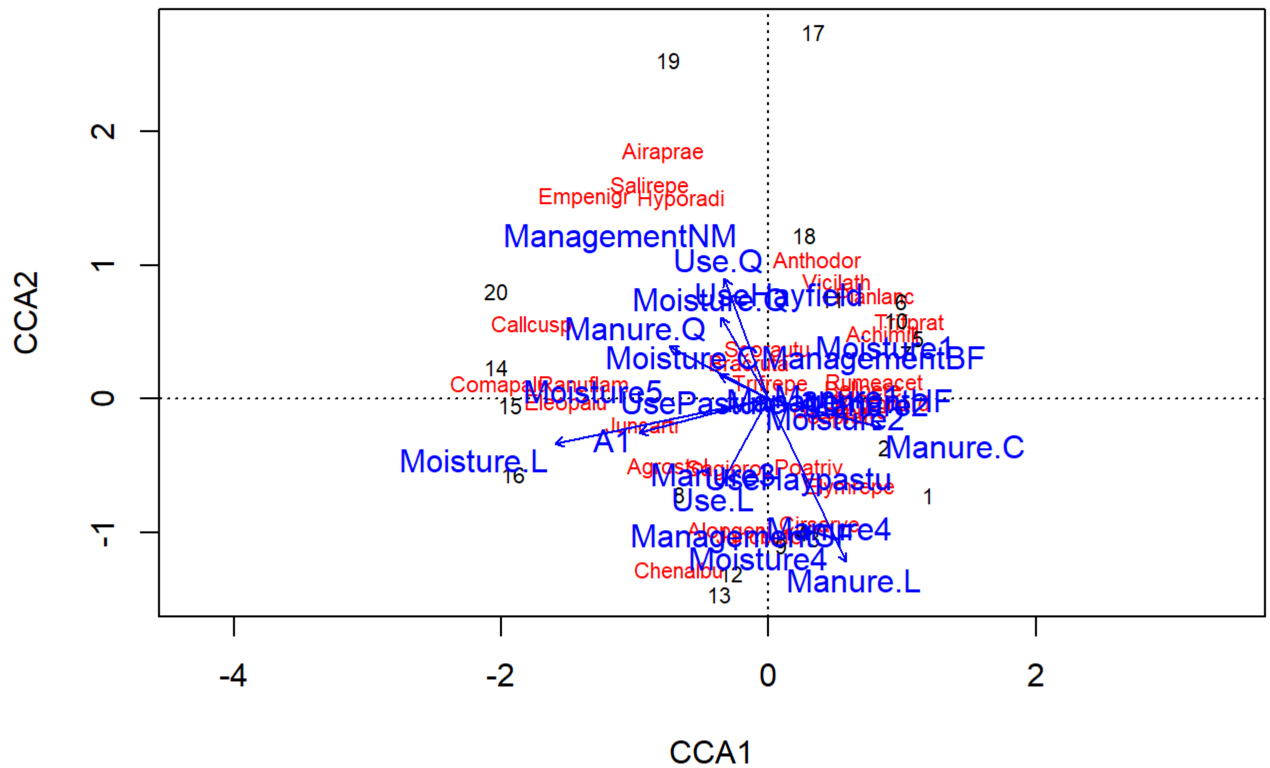
```
cca_dune <- cca(dune~., data=dune.env)
cca_dune
```

```
## Call: cca(formula = dune ~ A1 + Moisture + Management + Use +
## Manure, data = dune.env)
##
##               Inertia Proportion Rank
## Total          2.1153     1.0000
## Constrained    1.5032     0.7106   12
## Unconstrained  0.6121     0.2894    7
## Inertia is scaled Chi-square
## Some constraints were aliased because they were collinear (redundant)
##
## Eigenvalues for constrained axes:
##   CCA1   CCA2   CCA3   CCA4   CCA5   CCA6   CCA7   CCA8   CCA9  CCA10
## 0.4671 0.3410 0.1761 0.1532 0.0953 0.0703 0.0589 0.0499 0.0318 0.0260
##  CCA11  CCA12
## 0.0228 0.0108
##
## Eigenvalues for unconstrained axes:
##     CA1     CA2     CA3     CA4     CA5     CA6     CA7
## 0.27237 0.10876 0.08975 0.06305 0.03489 0.02529 0.01798
```

```
RsquareAdj(cca_dune,30)
```

```
## $r.squared
## [1] 0.7106267
##
## $adj.r.squared
## [1] 0.1980259
```

```
plot(cca_dune)
```



The adjusted R-squared is a modified version of R-squared that has been adjusted for the number of predictors in the model.

The adjusted R-squared increases only if the new term improves the model more than would be expected by chance.

It decreases when a predictor improves the model by less than expected by chance.

Not doing so great with random values?

```
anova(cca_dune, permutations=999)
```

```
## Permutation test for cca under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: cca(formula = dune ~ A1 + Moisture + Management + Use + Manure, data = dune.env)
##          Df ChiSquare      F Pr(>F)
## Model    12    1.5032 1.4325  0.025 *
## Residual  7    0.6121
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(cca_dune, by = 'margin', permutatinos = 999)
```

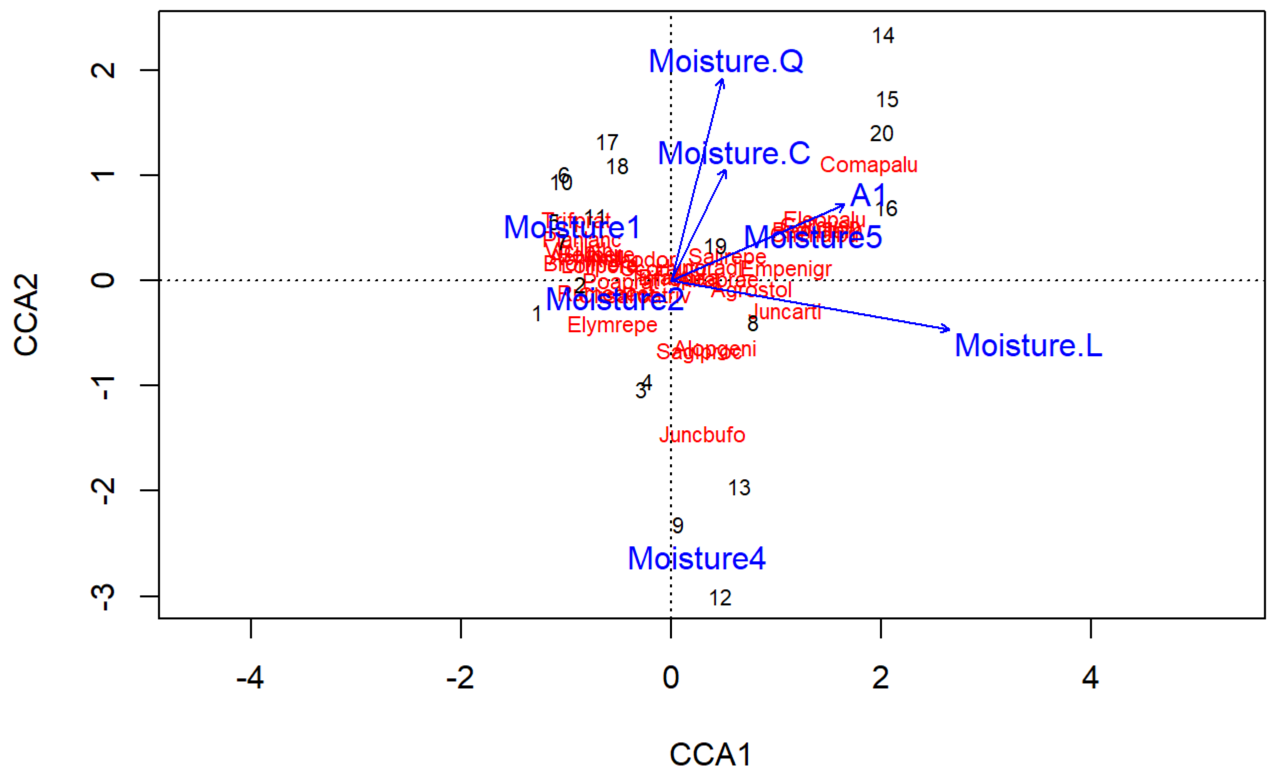
```
## Permutation test for cca under reduced model
## Marginal effects of terms
## Permutation: free
## Number of permutations: 999
##
## Model: cca(formula = dune ~ A1 + Moisture + Management + Use + Manure, data = dune.env)
##            Df ChiSquare      F Pr(>F)
## A1          1   0.11070 1.2660  0.220
## Moisture    3   0.31587 1.2041  0.211
## Management  2   0.15882 0.9081  0.568
## Use         2   0.13010 0.7439  0.764
## Manure      3   0.25490 0.9717  0.503
## Residual    7   0.61210
```

tests each variable individually. fits everythig first. fits the resids. and then fits in everything else. so while nothing is great it looks like A1(numeric vector of thickness of soil A1 horizon?) and moisture are the strongest predictors of our response variable out of all listed

```
cca_dune2 <- cca(dune~Moisture + A1, data = dune.env)
cca_dune2
```

```
## Call: cca(formula = dune ~ Moisture + A1, data = dune.env)
##
##               Inertia Proportion Rank
## Total          2.1153     1.0000
## Constrained    0.7437     0.3516    4
## Unconstrained  1.3715     0.6484   15
## Inertia is scaled Chi-square
##
## Eigenvalues for constrained axes:
##   CCA1   CCA2   CCA3   CCA4
## 0.4314 0.1350 0.1066 0.0706
##
## Eigenvalues for unconstrained axes:
##    CA1    CA2    CA3    CA4    CA5    CA6    CA7    CA8    CA9   CA10
## 0.3843 0.2140 0.1601 0.1226 0.0989 0.0902 0.0751 0.0605 0.0539 0.0456
##   CA11   CA12   CA13   CA14   CA15
## 0.0209 0.0154 0.0125 0.0096 0.0081
```

```
plot(cca_dune2)
```



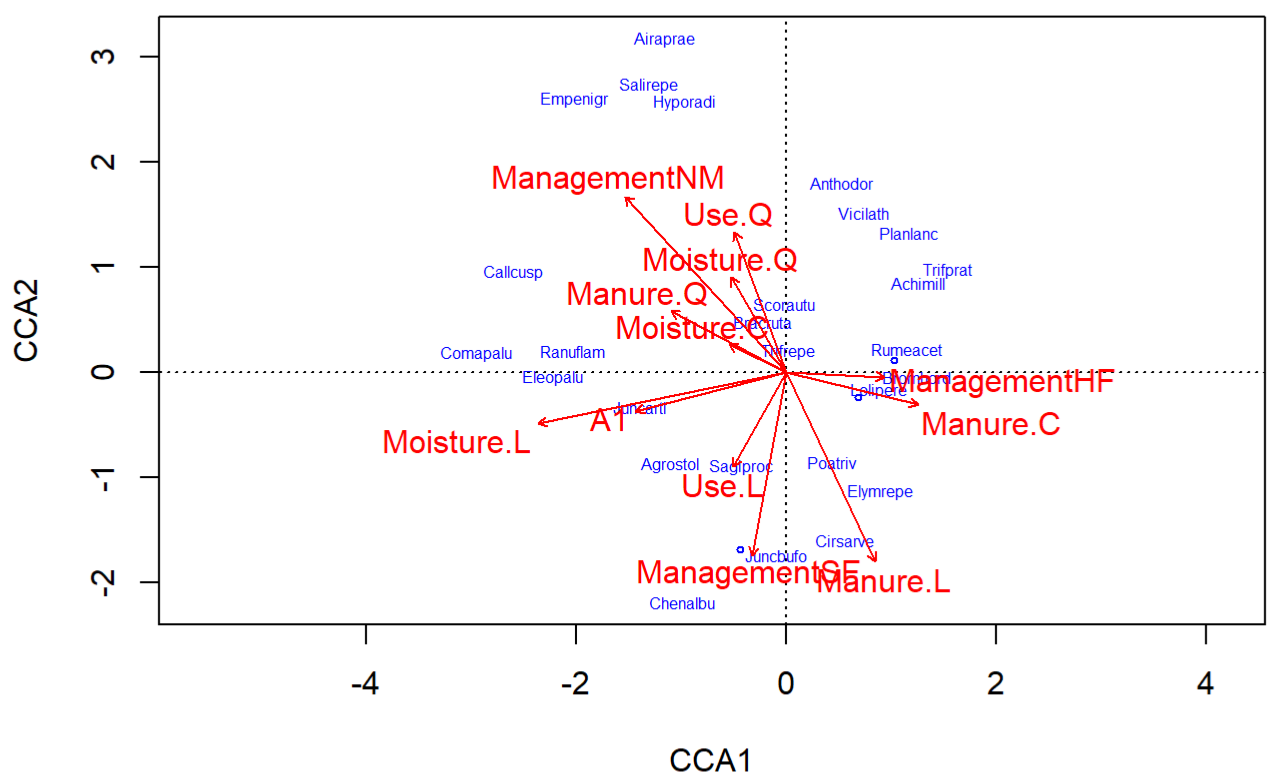

```
anova(cca_dune2, my = 'margin', permutations = 999)
```

```
## Permutation test for cca under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: cca(formula = dune ~ Moisture + A1, data = dune.env)
##          Df ChiSquare      F Pr(>F)
## Model     4   0.74374 2.0335  0.004 **
## Residual 15   1.37153
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

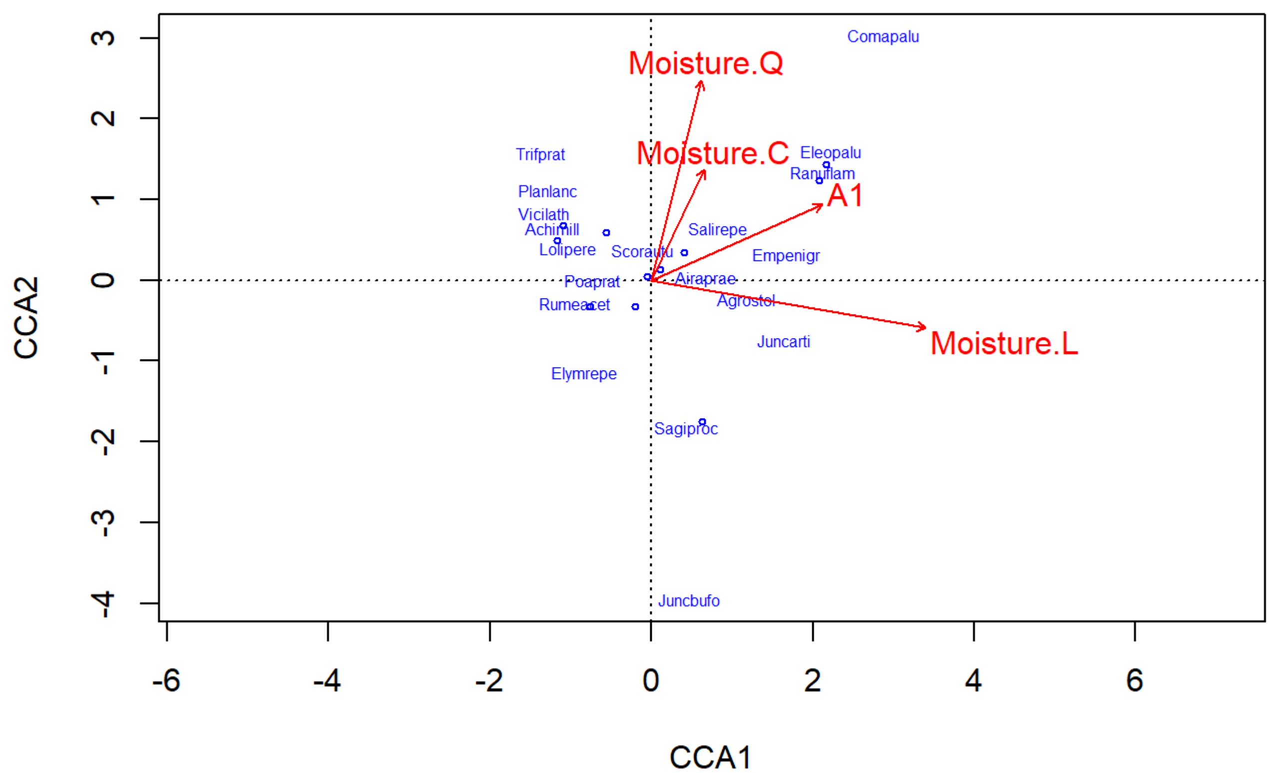
```
anova(cca_dune2, by ='axis', permutations = 999)
```

```
## Permutation test for cca under reduced model
## Forward tests for axes
## Permutation: free
## Number of permutations: 999
##
## Model: cca(formula = dune ~ Moisture + A1, data = dune.env)
##          Df ChiSquare      F Pr(>F)
## CCA1      1   0.43144 4.7186  0.001 ***
## CCA2      1   0.13503 1.4768  0.493
## CCA3      1   0.10663 1.1662  0.598
## CCA4      1   0.07063 0.7724  0.676
## Residual 15   1.37153
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(cca_dune, type='n', scaling = 1)
orditorp(cca_dune, display='sp', cex=0.5, scaling = 1, col='blue')
text(cca_dune, display='bp', col='red')
```




```
plot(cca_dune2, type='n', scaling = 1)
orditorp(cca_dune2, display='sp', cex=0.5, scaling = 1, col='blue')
text(cca_dune2, display='bp', col='red')
```



3. Do your two analyses agree with one another or complement one another or do these two analyses seem to be suggesting different take home messages? Which analysis do you find to be more useful?

The two analyes seem to complement each other in a useful way. The NMDS orientation gave a broader look at the general trends in the data, showing the best places to begin to pick apart for further constrained analysis. The CCA analysis allowed for a more detailed fit of the data to the orientations put forth in the first analysis. It further showed what was suggested by the first analysis, that Moisture and A1 are perhaps the strongest predictors of variation in the data. Very cool