Weber_MultivariateModels

Nick Weber

2/6/2018

```
library(vegan)
```

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

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

```
## This is vegan 2.4-6
```

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

#1. Conduct an indirect ordination...

dune_mds = metaMDS(dune, trymax = 5)
```

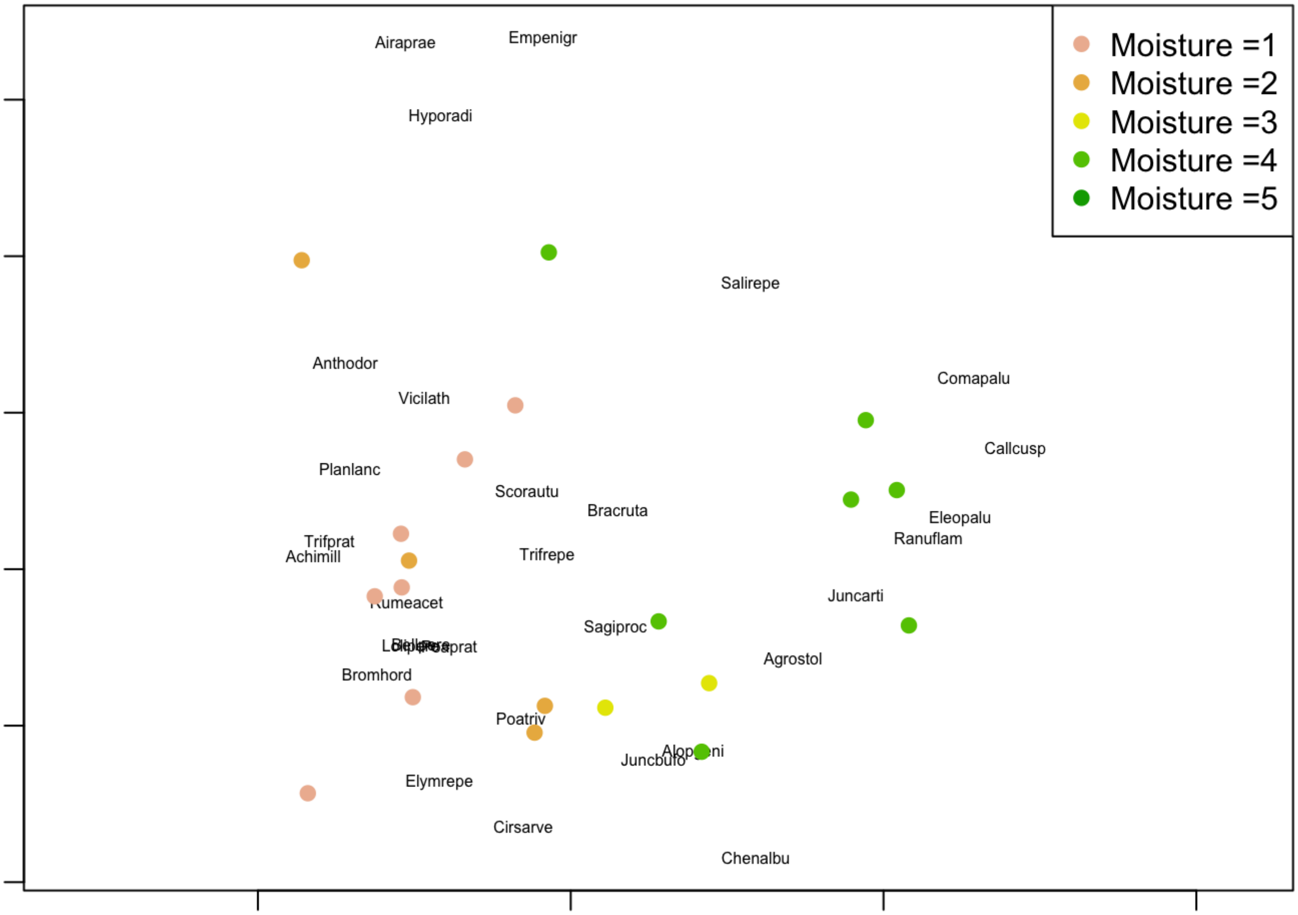
```
## Run 0 stress 0.1192678
## Run 1 stress 0.1183186
## ... New best solution
## ... Procrustes: rmse 0.02026909  max resid 0.06495097
## Run 2 stress 0.119268
## Run 3 stress 0.1183186
## ... Procrustes: rmse 2.592785e-05  max resid 8.3735e-05
## ... Similar to previous best
## Run 4 stress 0.1812932
## Run 5 stress 0.1183186
## ... Procrustes: rmse 1.069954e-05  max resid 3.400639e-05
## ... Similar to previous best
## Run 6 stress 0.2075713
## Run 7 stress 0.2035424
## Run 8 stress 0.1886532
## Run 9 stress 0.1183186
## ... Procrustes: rmse 1.859237e-05  max resid 6.599025e-05
## ... Similar to previous best
## Run 10 stress 0.1192679
## Run 11 stress 0.1183186
## ... Procrustes: rmse 7.438786e-05  max resid 0.0002065028
## ... Similar to previous best
## Run 12 stress 0.1183186
## ... Procrustes: rmse 7.361487e-06  max resid 2.350519e-05
## ... Similar to previous best
## Run 13 stress 0.1192679
## Run 14 stress 0.1809579
## Run 15 stress 0.1192684
## Run 16 stress 0.1192683
## Run 17 stress 0.1192678
## Run 18 stress 0.1809579
## Run 19 stress 0.1808921
## Run 20 stress 0.1922241
## *** Solution reached
```

```
par("mar")
```

```
## [1] 5.1 4.1 4.1 2.1
```

```
par(mar=c(1,1,1,1))

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])
legend('topright', paste("Moisture =", 1:5, sep=''),
        col=color_vect, pch=19)
```



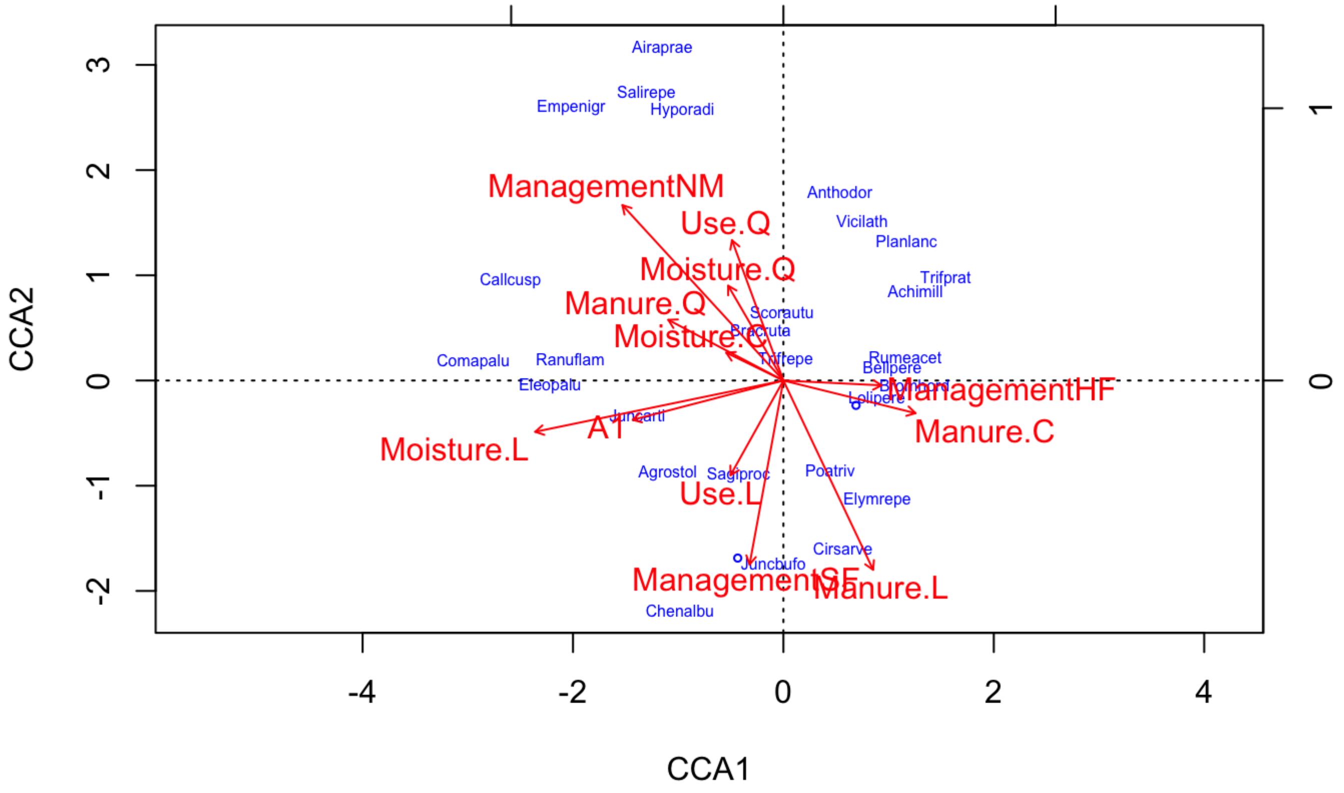
```
#On this plot, each point corresponds to 1 of the 20 sites, and the black words represent each of the 30 species. This plot is emphasizing the role of 'moisture', by color-coding each of the 20 sites according to the moisture level present at that site (aka we are post-hoc overlaying environmental information using 'moisture'). The axes are essentially hypothetical gradients. As you move to the right across the x-axis, it is clear that you are getting higher moisture levels. From this graph, we can start to infer that the species names present on the right side of the graph may be more likely found at higher moisture sites. The goal of this plot is to start to examine the correlations present in the explanatory environmental matrix.
```

```
#2. Carry out a direct ordination using CCA...

cca_dune <- cca(dune ~ ., data=dune.env)
RsquareAdj(cca_dune, 100) #0.26
```

```
## $r.squared
## [1] 0.7106267
##
## $adj.r.squared
## [1] 0.2381094
```

```
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')
```

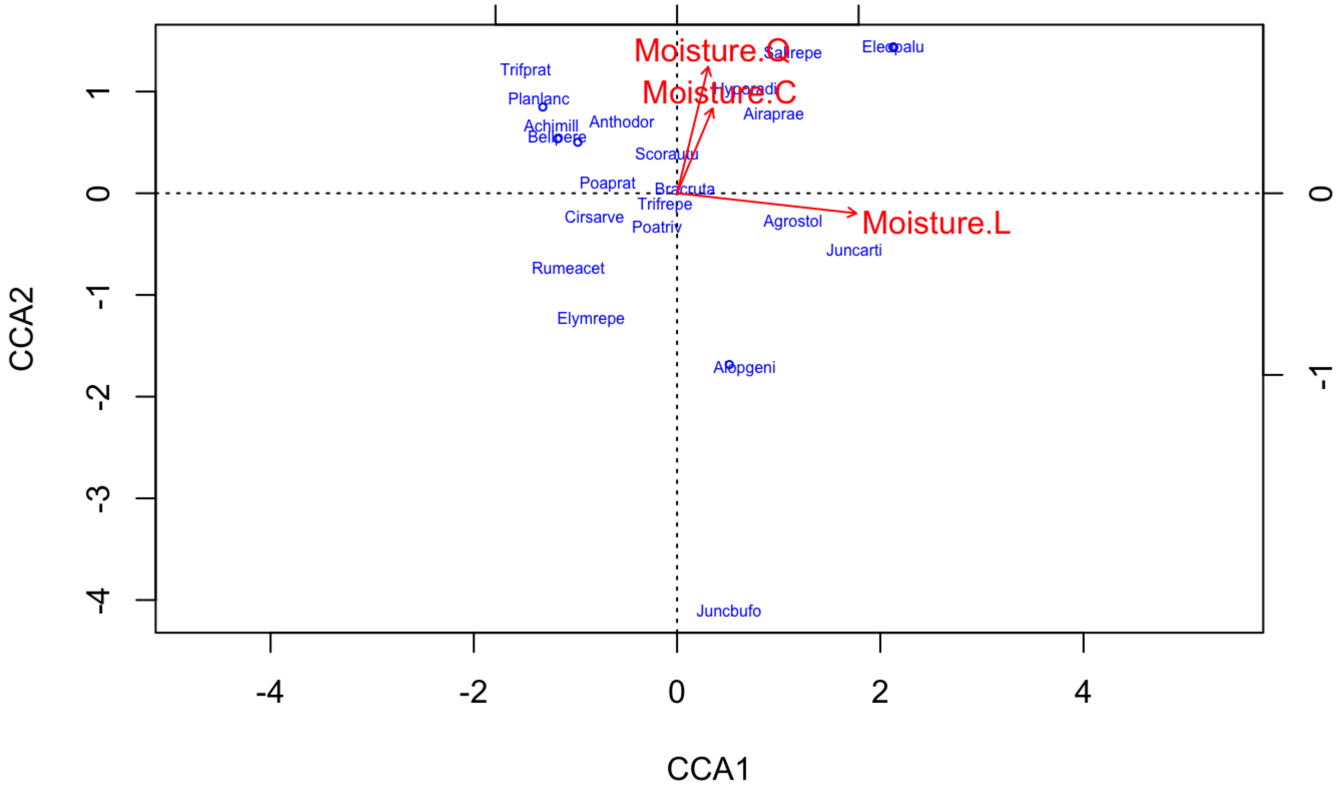


```
#In this plot, 'Moisture.L' is weighing heavily on CCA1, indicating that this variable explains a larger proption of the variance associated with the CCA1 axis. The location of the species relative to each environmental variable indicates how strongly a species is associated with a given variable. The majority of the variation associated with axis 2 is explained by the variables 'Management' and 'Manure'.

cca_dune_1 <- cca(dune ~ Moisture, data=dune.env)
RsquareAdj(cca_dune_1, 100) #0.17
```

```
## $r.squared
## [1] 0.2970359
##
## $adj.r.squared
## [1] 0.172184
```

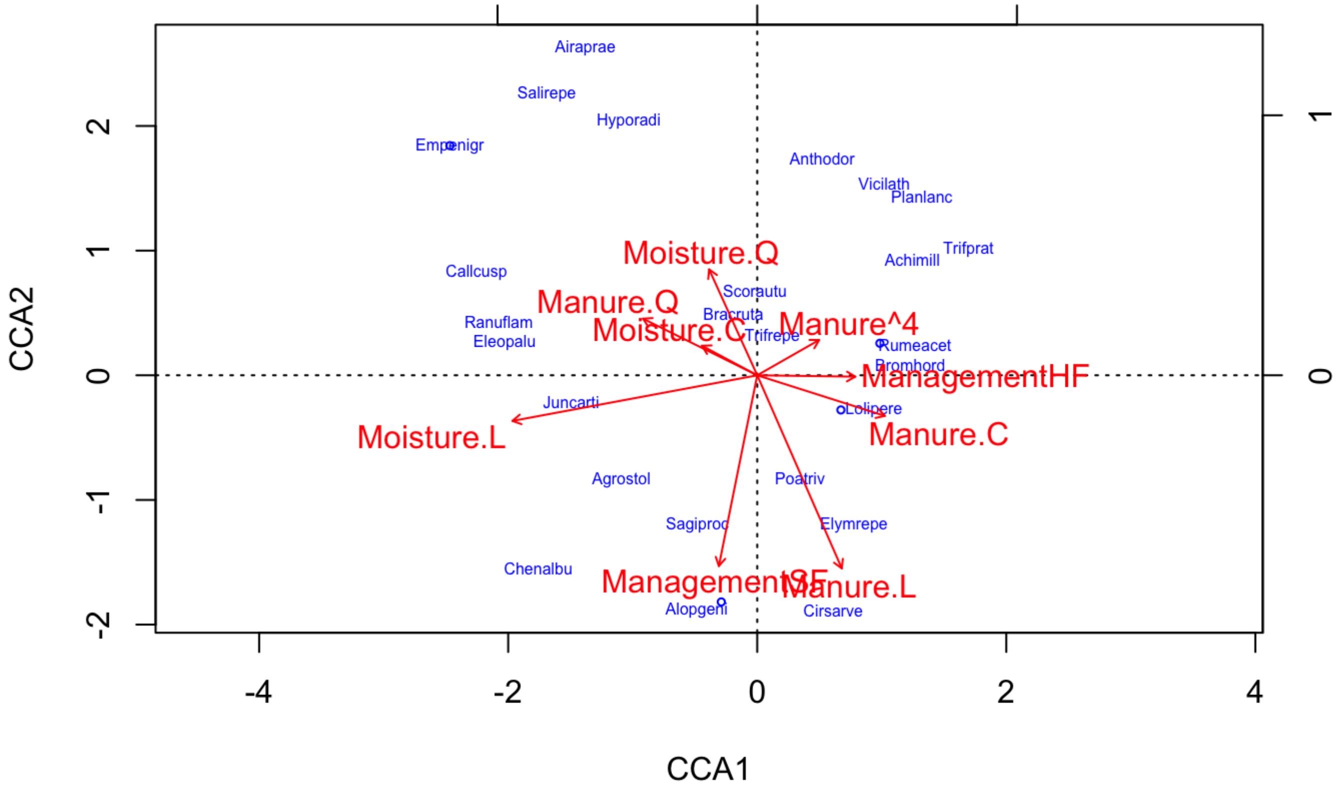
```
plot(cca_dune_1, type='n', scaling=1)
orditorp(cca_dune_1, display='sp', cex=0.5, scaling=1, col='blue')
text(cca_dune_1, display='bp', col='red')
```



```
cca_dune_2 <- cca(dune ~ Moisture + Manure + Management, data=dune.env)
RsquareAdj(cca_dune_2, 100) #0.21
```

```
## $r.squared
## [1] 0.5774129
##
## $adj.r.squared
## [1] 0.2159814
```

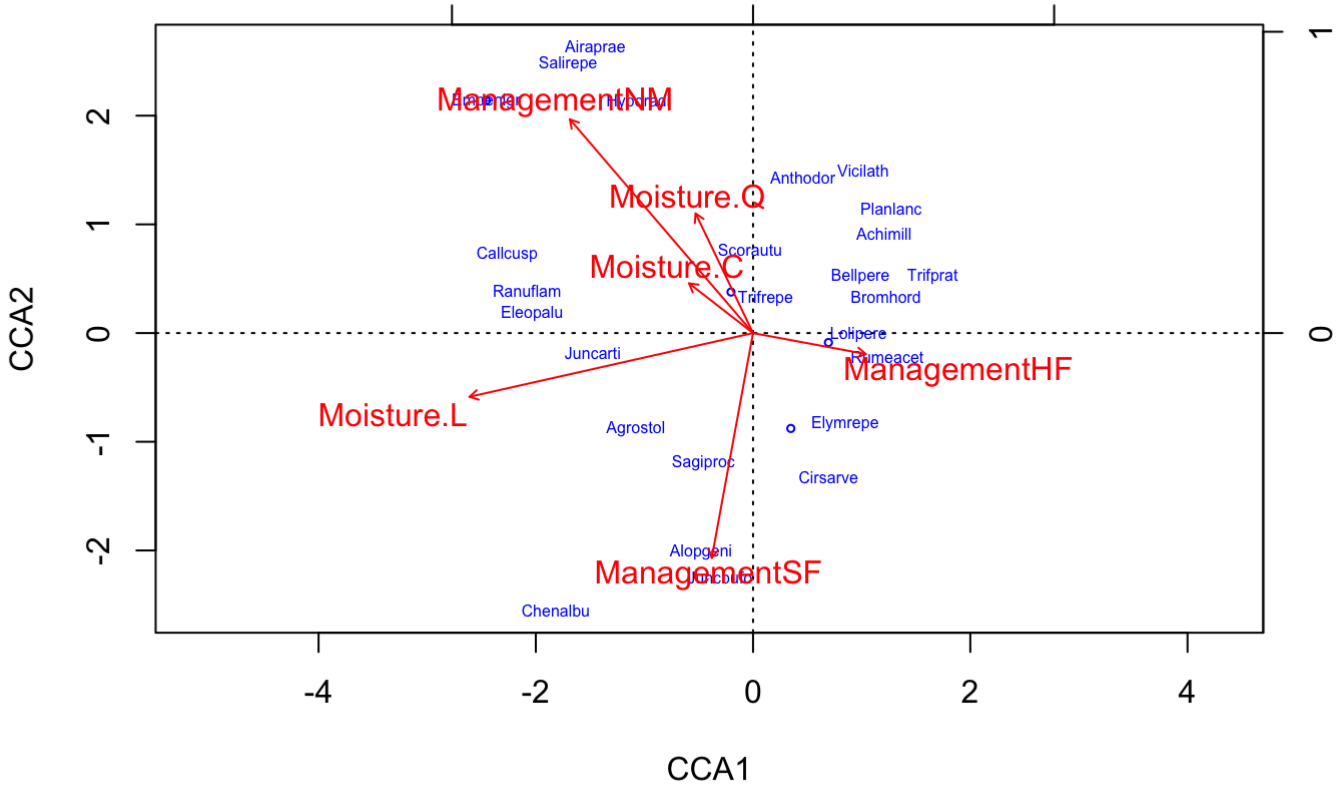
```
plot(cca_dune_2, type='n', scaling=1)
orditorp(cca_dune_2, display='sp', cex=0.5, scaling=1, col='blue')
text(cca_dune_2, display='bp', col='red')
```



```
cca_dune_3 <- cca(dune ~ Moisture + Management, data=dune.env)
RsquareAdj(cca_dune_3, 100) #0.24
```

```
## $r.squared
## [1] 0.4738772
##
## $adj.r.squared
## [1] 0.2451795
```

```
plot(cca_dune_3, type='n', scaling=1)
orditorp(cca_dune_3, display='sp', cex=0.5, scaling=1, col='blue')
text(cca_dune_3, display='bp', col='red')
```



```
#2. cont'd...

#Running permutation test on entire model (cca_dune).
anova(cca_dune, permutations = 100)
```

```
## Permutation test for cca under reduced model
## Permutation: free
## Number of permutations: 100
##
## Model: cca(formula = dune ~ A1 + Moisture + Management + Use + Manure, data = dune.en
v)
##          Df ChiSquare      F Pr(>F)
## Model    12    1.5032 1.4325 0.0297 *
## Residual  7    0.6121
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(cca_dune, by='margin', permutations = 100)
```

```
## Permutation test for cca under reduced model
## Marginal effects of terms
## Permutation: free
## Number of permutations: 100
## 
## Model: cca(formula = dune ~ A1 + Moisture + Management + Use + Manure, data = dune.en
v)
##            Df ChiSquare      F Pr(>F)
## A1          1   0.11070 1.2660 0.2574
## Moisture    3   0.31587 1.2041 0.1980
## Management  2   0.15882 0.9081 0.6040
## Use         2   0.13010 0.7439 0.7426
## Manure      3   0.25490 0.9717 0.4554
## Residual    7   0.61210
```

```
#Running permutation test on best-fitting model (cca_dune_3).

anova(cca_dune_3, permutations = 100)
```

```
## Permutation test for cca under reduced model
## Permutation: free
## Number of permutations: 100
## 
## Model: cca(formula = dune ~ Moisture + Management, data = dune.env)
##          Df ChiSquare      F   Pr(>F)
## Model     6    1.0024 1.9515 0.009901 **
## Residual 13    1.1129
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(cca_dune_3, by='margin', permutations = 100)
```

```
## Permutation test for cca under reduced model
## Marginal effects of terms
## Permutation: free
## Number of permutations: 100
## 
## Model: cca(formula = dune ~ Moisture + Management, data = dune.env)
##            Df ChiSquare      F  Pr(>F)
## Moisture    3   0.39854 1.5518 0.01980 *
## Management  3   0.37407 1.4565 0.05941 .
## Residual   13   1.11289
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#3. The two analyses do agree with eachother. The first analysis in Question 1 suggested that 'moisture' was explaining a good portion of the variation in the first axis. The second analysis in Question 2 also suggested that 'moisture' was explaining the majority of the variation in axis 1, and also suggested that 'Management' explained the marjority of the variation in axis 2. Finally, the anova() in analysis 2 suggested that 'Moisture' was having a significant effect on dune species distribution.

#I found the second analysis to be more useful. I was able to arrive at the variables 'Moisture' and 'Management' explaining the majority of the variation, and could then confirm that result by plotting those variables on an NMDS plot.

#The indirect ordination below shows that 'Management' is explaining a good portion of the variation in the second axis.

dune_mds = metaMDS(dune, trymax = 5)
```

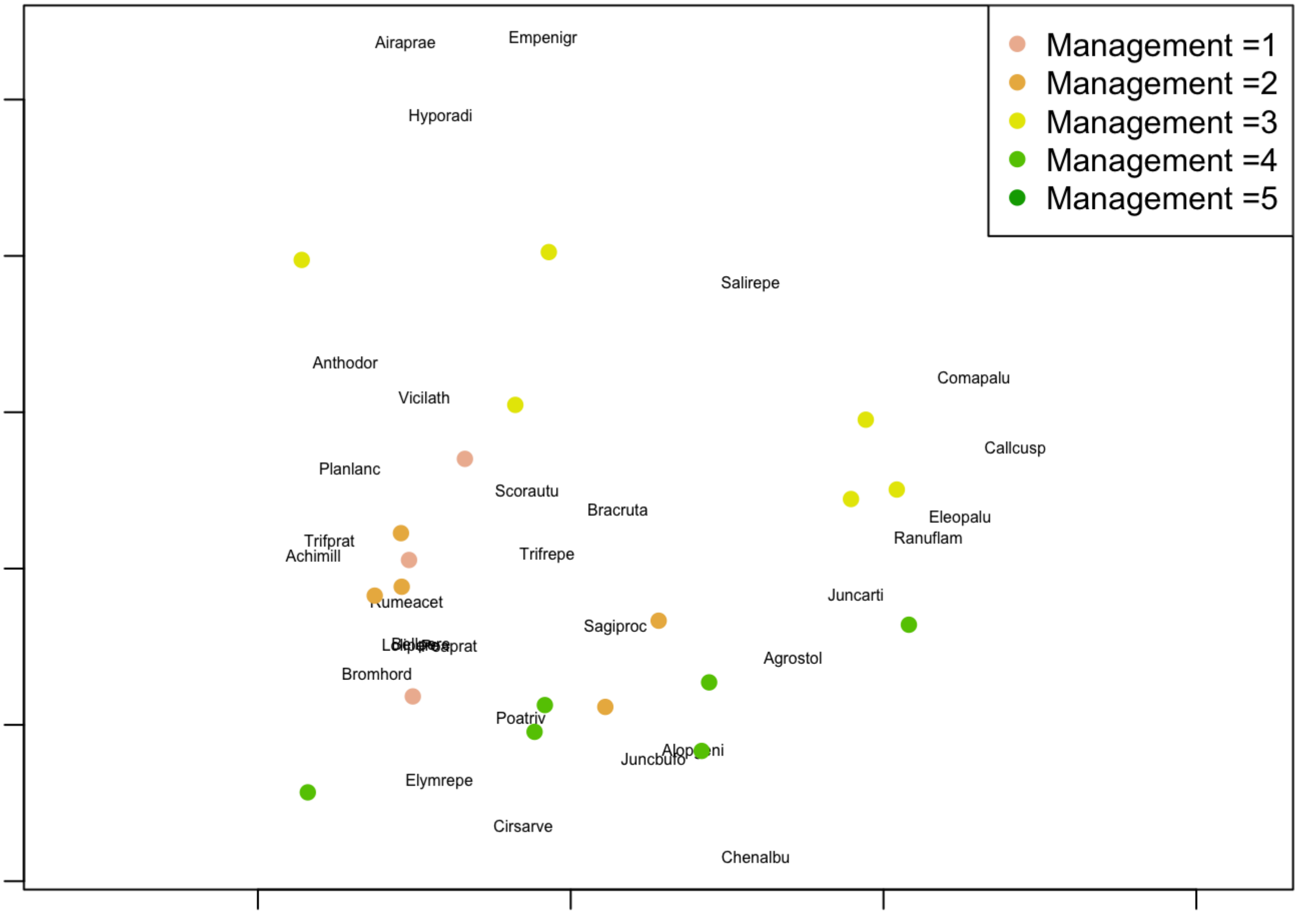
```
## Run 0 stress 0.1192678
## Run 1 stress 0.1192679
## ... Procrustes: rmse 0.0001277808  max resid 0.0003934782
## ... Similar to previous best
## Run 2 stress 0.1183186
## ... New best solution
## ... Procrustes: rmse 0.02027381  max resid 0.0649742
## Run 3 stress 0.119268
## Run 4 stress 0.1183186
## ... New best solution
## ... Procrustes: rmse 4.856957e-05  max resid 0.0001642136
## ... Similar to previous best
## Run 5 stress 0.1183186
## ... Procrustes: rmse 3.480091e-05  max resid 0.0001029482
## ... Similar to previous best
## Run 6 stress 0.1192679
## Run 7 stress 0.1192679
## Run 8 stress 0.1812937
## Run 9 stress 0.1192678
## Run 10 stress 0.1183186
## ... Procrustes: rmse 2.185123e-05  max resid 7.289272e-05
## ... Similar to previous best
## Run 11 stress 0.1192685
## Run 12 stress 0.1192679
## Run 13 stress 0.1183186
## ... Procrustes: rmse 1.727919e-05  max resid 6.025904e-05
## ... Similar to previous best
## Run 14 stress 0.2045511
## Run 15 stress 0.1808911
## Run 16 stress 0.1183186
## ... Procrustes: rmse 2.197358e-05  max resid 7.196308e-05
## ... Similar to previous best
## Run 17 stress 0.1886533
## Run 18 stress 0.1183186
## ... Procrustes: rmse 3.067781e-05  max resid 0.000102643
## ... Similar to previous best
## Run 19 stress 0.1183186
## ... Procrustes: rmse 3.473404e-06  max resid 1.15365e-05
## ... Similar to previous best
## Run 20 stress 0.1192679
## *** Solution reached
```

```
par("mar")
```

```
## [1] 5.1 4.1 4.1 2.1
```

```
par(mar=c(1,1,1,1))

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$Management])
legend('topright', paste("Management =", 1:5, sep=''),
        col=color_vect, pch=19)
```


● Management =1
● Management =2
● Management =3
● Management =4
● Management =5
Airaprae
Empenigr
Hyporadi
Salirepe
Anthodor
Comapalu
Vicilath
Callcusp
Planlanc
Scorautu
Bracruta
Eleopalu
Ranuflam
Trifprat
Achimill
Trifrepe
Juncarti
Rumeacet
Sagiproc
Lolipere
Bellpere
Poaprat
Agrostol
Bromhord
Poatriv
Juncbufo
Alopgeni
Elymrepe
Cirsarve
Chenalbu