Multivariate Models

Megan Sporre

February 6, 2018

```r
library(lattice)
```

```
## Warning: package 'lattice' was built under R version 3.3.3
```

```r
library(permute)
library(vegan)
```

```
## Warning: package 'vegan' was built under R version 3.3.3
```

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

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

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

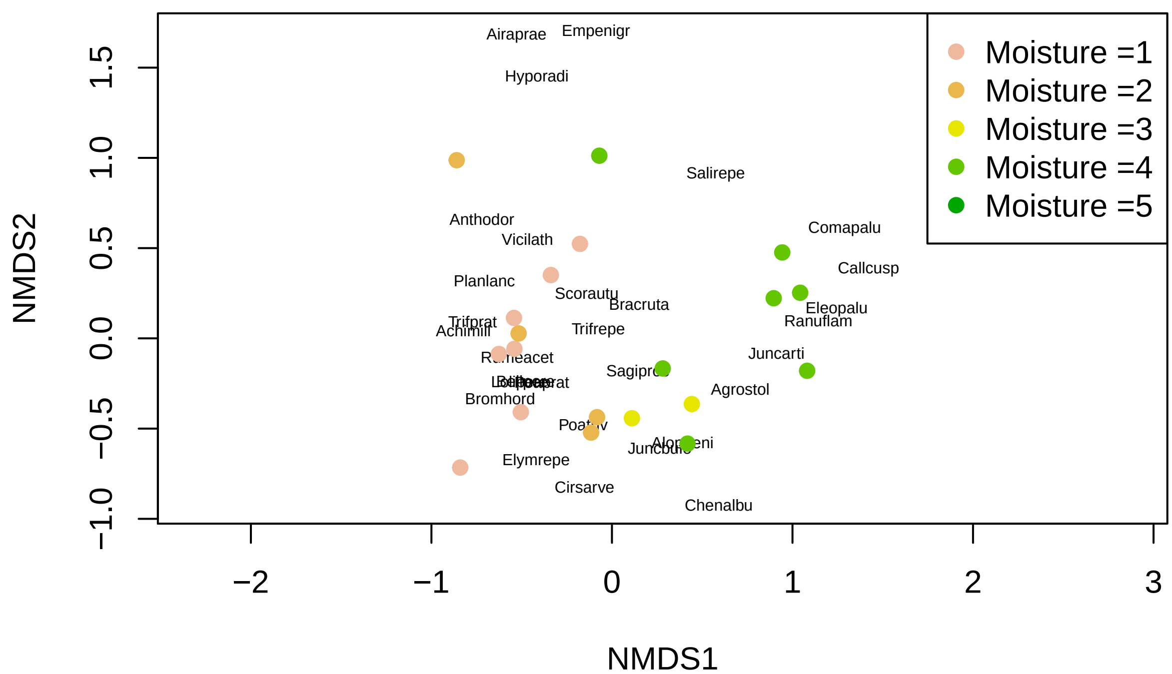
```
##  done
```

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

```
## Run 0 stress 0.1192678
## Run 1 stress 0.1192678
## ... New best solution
## ... Procrustes: rmse 5.3898e-05  max resid 0.0001614899
## ... Similar to previous best
## Run 2 stress 0.1192679
## ... Procrustes: rmse 0.0001182904  max resid 0.0003547997
## ... Similar to previous best
## Run 3 stress 0.1192678
## ... New best solution
## ... Procrustes: rmse 3.31611e-05  max resid 9.894104e-05
## ... Similar to previous best
## Run 4 stress 0.119268
## ... Procrustes: rmse 0.0001849055  max resid 0.0005765779
## ... Similar to previous best
## Run 5 stress 0.1183186
## ... New best solution
## ... Procrustes: rmse 0.02028291  max resid 0.06505355
## Run 6 stress 0.1192679
## Run 7 stress 0.1183186
## ... New best solution
## ... Procrustes: rmse 3.172206e-05  max resid 0.0001032555
## ... Similar to previous best
## Run 8 stress 0.1809578
## Run 9 stress 0.1192685
## Run 10 stress 0.1183186
## ... Procrustes: rmse 5.680085e-05  max resid 0.0001853883
## ... Similar to previous best
## Run 11 stress 0.1183187
## ... Procrustes: rmse 0.0001555455  max resid 0.0005040955
```

```
## ... Similar to previous best
## Run 12 stress 0.1192678
## Run 13 stress 0.1192685
## Run 14 stress 0.1192679
## Run 15 stress 0.1192689
## Run 16 stress 0.1809577
## Run 17 stress 0.1192678
## Run 18 stress 0.1183186
## ... Procrustes: rmse 5.003682e-05  max resid 0.0001495081
## ... Similar to previous best
## Run 19 stress 0.1183186
## ... Procrustes: rmse 6.396394e-06  max resid 2.041067e-05
## ... Similar to previous best
## Run 20 stress 0.119268
## *** Solution reached
```

```r
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)
```




```r
 #This graphic shows that moisture is important in describing dune vegetation. The goal of creating thi
```

```r
cca_tree = cca(dune~ ., data=dune.env)
RsquareAdj(cca_tree, 10)
```

```
## $r.squared
## [1] 0.7106267
##
## $adj.r.squared
## [1] 0.2643141
```

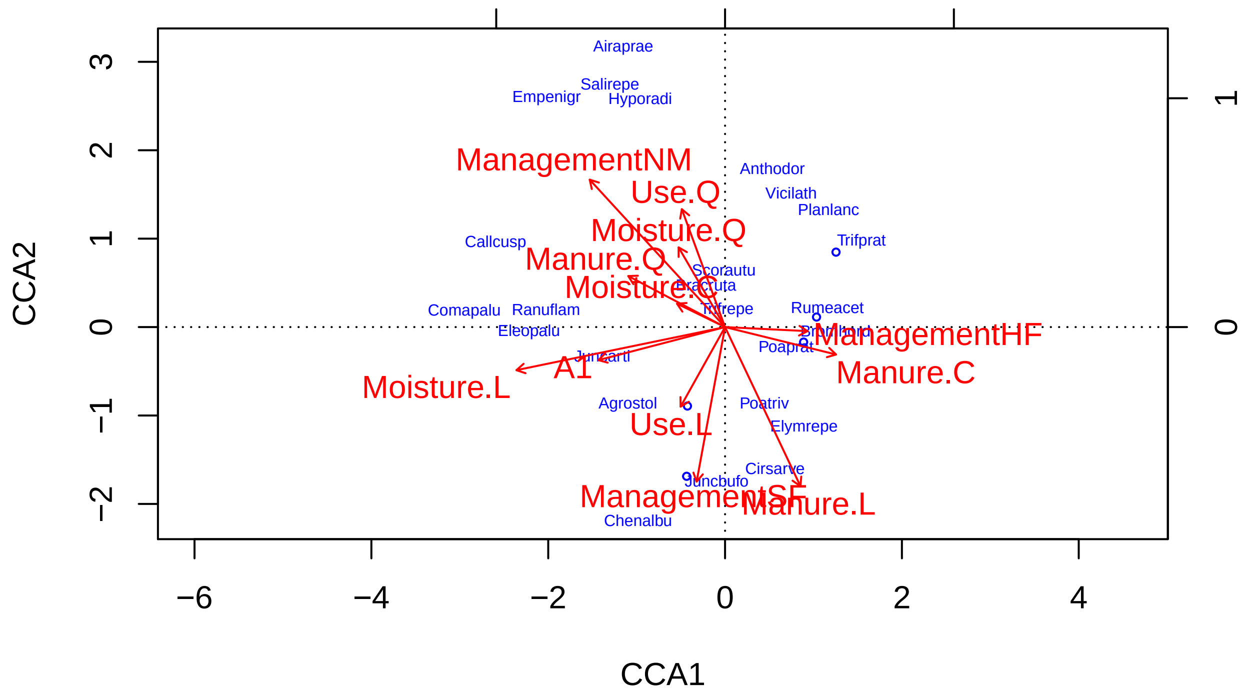
```r
anova(cca_tree, permutations = 10)
```

```
## Permutation test for cca under reduced model
## Permutation: free
## Number of permutations: 10
##
## Model: cca(formula = dune ~ A1 + Moisture + Management + Use + Manure, data = dune.env)
##          Df ChiSquare      F  Pr(>F)
## Model    12    1.5032 1.4325 0.09091 .
## Residual  7    0.6121
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(cca_tree, by='margin', permutations = 10)
```

```
## Permutation test for cca under reduced model
## Marginal effects of terms
## Permutation: free
## Number of permutations: 10
##
## Model: cca(formula = dune ~ A1 + Moisture + Management + Use + Manure, data = dune.env)
##            Df ChiSquare      F Pr(>F)
## A1          1   0.11070 1.2660 0.1818
## Moisture    3   0.31587 1.2041 0.3636
## Management  2   0.15882 0.9081 0.4545
## Use         2   0.13010 0.7439 0.7273
## Manure      3   0.25490 0.9717 0.6364
## Residual    7   0.61210
```

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




```r
#moisture explains the majority of variation aling the x-axis and managment and moisture explain most o
```

```r
cca_moist = cca(dune~ Moisture, data=dune.env)
RsquareAdj(cca_moist, 10)
```

```
## $r.squared
## [1] 0.2970359
##
## $adj.r.squared
## [1] 0.1451072
```

```r
cca_manure = cca(dune~ Manure, data=dune.env)
RsquareAdj(cca_manure, 100)
```

```
## $r.squared
## [1] 0.2891171
##
## $adj.r.squared
## [1] 0.1020495
```

```r
cca_management = cca(dune~ Management, data=dune.env)
RsquareAdj(cca_manure, 100)
```

```
## $r.squared
## [1] 0.2891171
##
## $adj.r.squared
## [1] 0.1147119
```

```r
cca_moistman = cca(dune~ Moisture+Management, data=dune.env)
RsquareAdj(cca_moistman, 100)
```

```
## $r.squared
## [1] 0.4738772
##
## $adj.r.squared
## [1] 0.240114
```

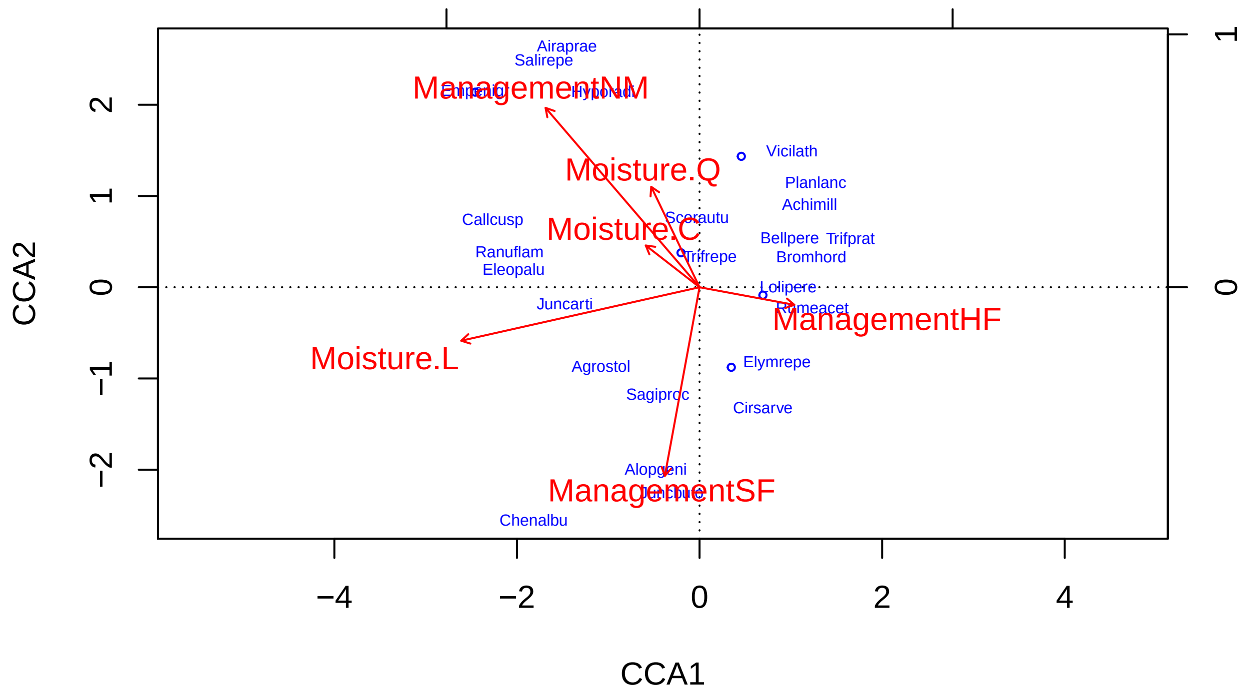
```r
anova(cca_moistman, permutations = 100)
```

```
## 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
```

```r
anova(cca_moistman, 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 0.04950 *
## Management  3   0.37407 1.4565 0.07921 .
## Residual   13   1.11289
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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




```r
#The moisture and managment model has the best r-squared.
```

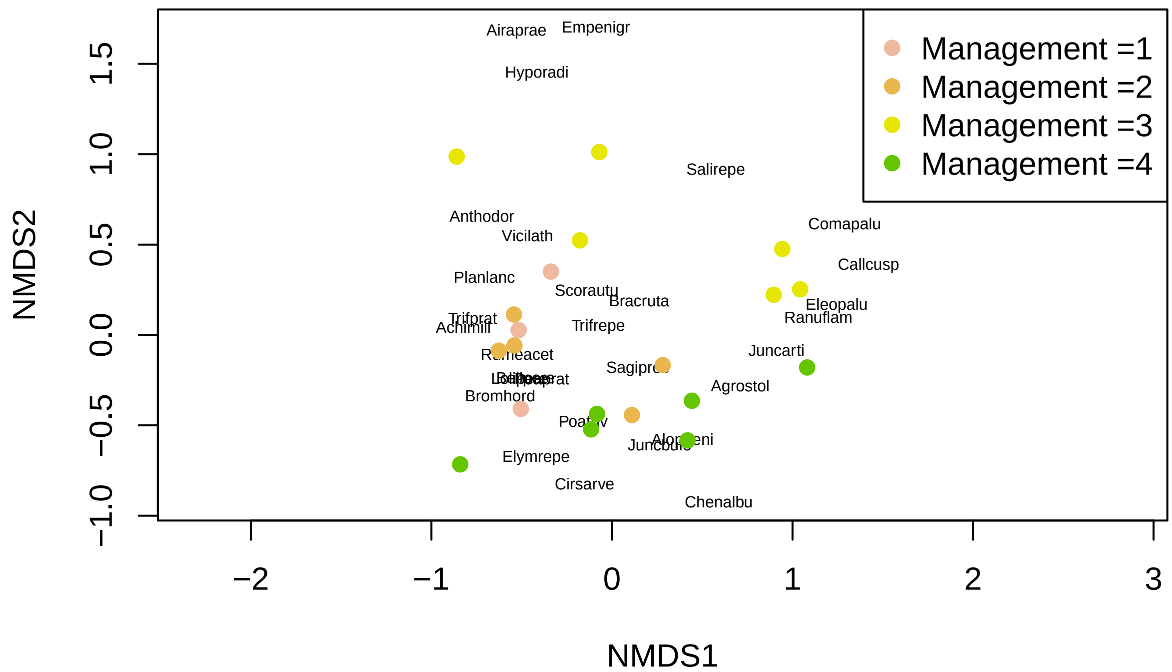
The two analyses agree with one another, moisture is a significant variable in explaining dunes. In the CCA, management also had a slight effect on dunes. We did not test the effect of managment in the NMDS of dune vegetation. Upon running NMDS for Management we see that it explains variation in the y-axis although there is not as clear of a delineation.

```r
dune_mdsman=metaMDS(dune, trymax=5)
```

```
## Run 0 stress 0.1192678
## Run 1 stress 0.1192682
## ... Procrustes: rmse 0.0003495902  max resid 0.001076421
## ... Similar to previous best
## Run 2 stress 0.1812934
## Run 3 stress 0.1183186
## ... New best solution
## ... Procrustes: rmse 0.02027368  max resid 0.06497106
## Run 4 stress 0.1812966
## Run 5 stress 0.1183186
## ... Procrustes: rmse 5.558027e-05  max resid 0.0001322035
## ... Similar to previous best
## Run 6 stress 0.1192678
## Run 7 stress 0.1183186
## ... New best solution
## ... Procrustes: rmse 2.641697e-05  max resid 6.714371e-05
## ... Similar to previous best
## Run 8 stress 0.1808912
```

```
## Run 9 stress 0.1192678
## Run 10 stress 0.1808911
## Run 11 stress 0.1183186
## ... Procrustes: rmse 1.785163e-05  max resid 4.242629e-05
## ... Similar to previous best
## Run 12 stress 0.1808919
## Run 13 stress 0.1192682
## Run 14 stress 0.1192679
## Run 15 stress 0.1183186
## ... Procrustes: rmse 0.0001065622  max resid 0.0003342585
## ... Similar to previous best
## Run 16 stress 0.1192692
## Run 17 stress 0.1192679
## Run 18 stress 0.1192681
## Run 19 stress 0.1183186
## ... New best solution
## ... Procrustes: rmse 8.289489e-06  max resid 2.009208e-05
## ... Similar to previous best
## Run 20 stress 0.1192679
## *** Solution reached
```

```r
plot(dune_mdsman, type='n')
 text(dune_mdsman, 'sp', cex=.5)
 # generate vector of colors
 color_vect = rev(terrain.colors(6))[-1]
 points(dune_mdsman, 'sites', pch=19,
        col=color_vect[dune.env$Management])
 legend('topright', paste("Management =", 1:4, sep=''),
        col=color_vect, pch=19)
```



Management =1
Management =2
Management =3
Management =4
Airaprae
Empenigr
Hyporadi
Salirepe
Anthodor
Vicilath
Comapalu
Planlanc
Callcusp
Scorautu
Bracruta
Eleopalu
Ranuflam
Trifprat
Achimill
Trifrepe
Juncarti
Sagipro
Agrostol
Bromhord
Elymrepe
Cirsarve
Chenalbu
NMDS1
NMDS2