

# HW4\_Margaret\_Walker

*Margaret Walker*

*February 13, 2016*

For this assignment will be analyzing data on the Vegetation and Environment in Dutch Dune Meadows.

To import the data and read the metadata run the following:

```
library(vegan)
```

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

```
## Warning: package 'permute' was built under R version 3.1.3
```

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

```
## This is vegan 2.2-1
```

```
data(dune)
```

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

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?

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

```
## Run 0 stress 0.1192678
```

```
## Run 1 stress 0.2035424
```

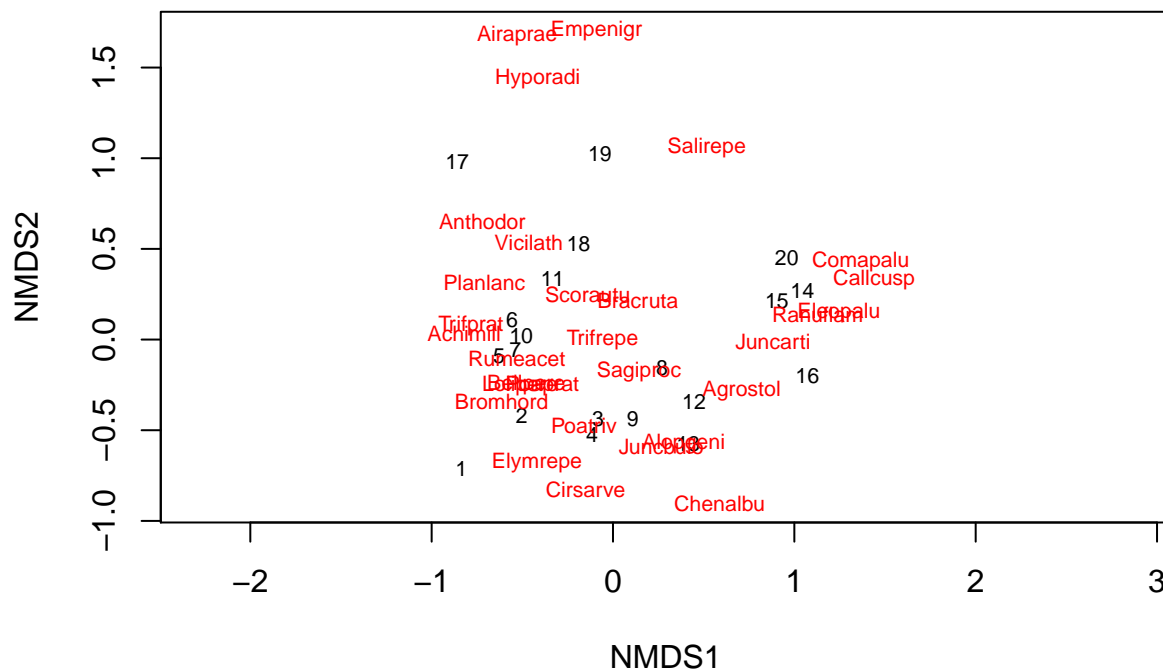
```
## Run 2 stress 0.1980523
```

```
## Run 3 stress 0.1192679
```

```
## ... procrustes: rmse 0.000160612 max resid 0.0004940658
```

```
## *** Solution reached
```

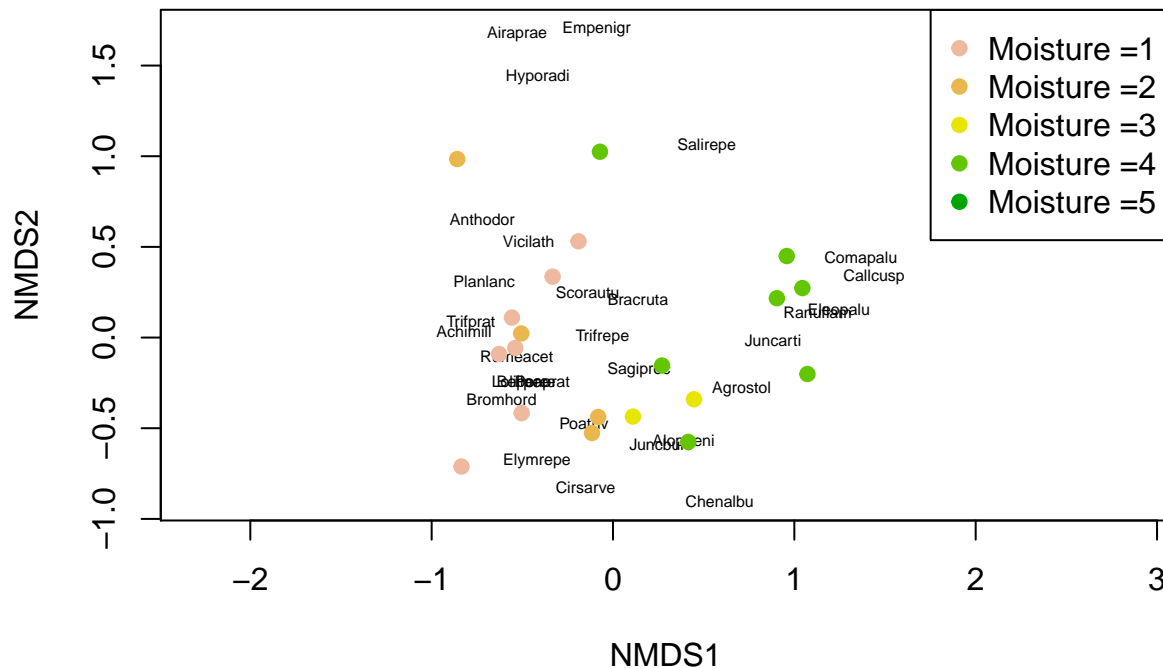
```
plot(dune_mds, type="t")
```



```
dune_mds
```

```
##
## Call:
## metaMDS(comm = dune)
##
## global Multidimensional Scaling using monoMDS
##
## Data:      dune
## Distance: bray
##
## Dimensions: 2
## Stress:    0.1192678
## Stress type 1, weak ties
## Two convergent solutions found after 3 tries
## Scaling: centring, PC rotation, halfchange scaling
## Species: expanded scores based on 'dune'
```

```
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)
```



I started by created NMDS object for the dune data set. I then plotted this object, and it gave me a plot of all of the species for NMDS1 vs. NMDS2. This plot is hard to decipher other than seeing which species are grouped together. This original plot gives us an idea of how different species are related according to the NMDS. That is, we can assume that species close together are more related while those farther away are less related. The next plot separated the species based on their moisture values. This graph shows us that the species grouped together on the NMDS plot have similar moisture values. For example, the species that correspond to high moisture areas (4 and 5) are lumped together on the right side of the graph. This leads me to believe that moisture may be an environmental factor that is important in describing the dune vegetation community. Furthermore, we may want to evaluate all of the environmental factors and see what environmental conditions are driving the community.

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.

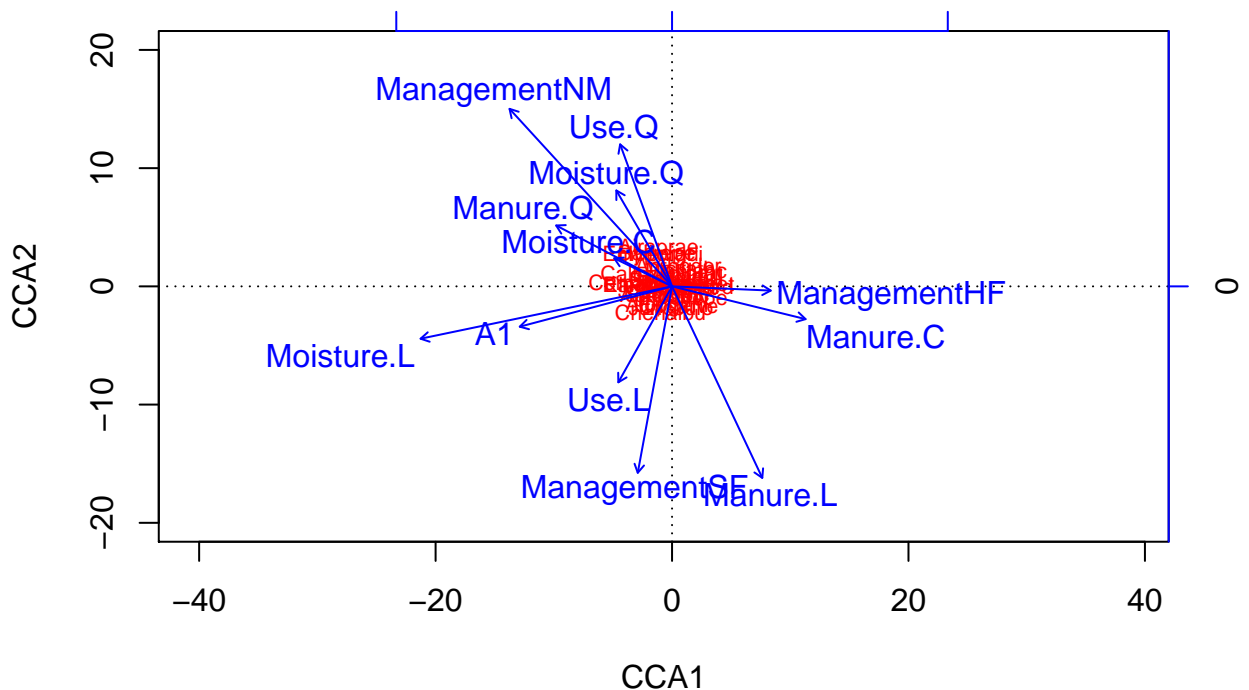
I will start out by running a CCA with all of the variables since based on the NMDS I believe that the environment is driving the dune vegetation community. See code below:

```
dune_cca = cca(dune ~ A1 + Moisture + Management + Use + Manure, data = dune.env)
dune_cca
```

```
## 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 mean squared contingency coefficient
## 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
```

```
plot(dune_cca, ylim = c(-20, 20), display = c("sp", "bp"), scaling = 1)
```



```
anova(dune_cca)
```

```
## 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 0.027 *
## Residual 7      0.6121
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(dune_cca, by = "margin")
```

```
## 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 0.233
## Moisture     3   0.31587 1.2041 0.230
## Management   2   0.15882 0.9081 0.581
## Use          2   0.13010 0.7439 0.788
## Manure       3   0.25490 0.9717 0.510
## Residual     7   0.61210
```

```
library(dummies)
```

```
## dummies-1.5.6 provided by Decision Patterns
```

```
moist <- dummy(dune.env$Moisture)
uses <- dummy(dune.env$Use)
manage <- dummy(dune.env$Management)
man <- dummy(dune.env$Manure)
varpart(dune, dune.env$A1, moist, dune.env[, c(uses, manage, man)])
```

```
## Warning: collinearity detected in X2: mm = 4, m = 3
```

```
## Warning: collinearity detected in X3: mm =60, m =1
```

```
## Warning: collinearity detected in cbind(X1,X2): mm = 5, m = 4
```

```
## Warning: collinearity detected in cbind(X1,X3): mm = 61, m = 1
```

```
## Warning: collinearity detected in cbind(X2,X3): mm = 64, m = 4
```

```
## Warning: collinearity detected in cbind(X1,X2,X3): mm = 65, m = 4
```

```
##
## Partition of variation in RDA
##
## Call: varpart(Y = dune, X = dune.env$A1, moist, dune.env[, c(uses,
## manage, man)])
##
## Explanatory tables:
## X1: dune.env$A1
## X2: moist
## X3: dune.env[, c(uses, manage, man)]
##
## No. of explanatory tables: 3
## Total variation (SS): 1598.4
##           Variance: 84.124
## No. of observations: 20
##
## Partition table:
##           Df R.square Adj.R.square Testable
```

```

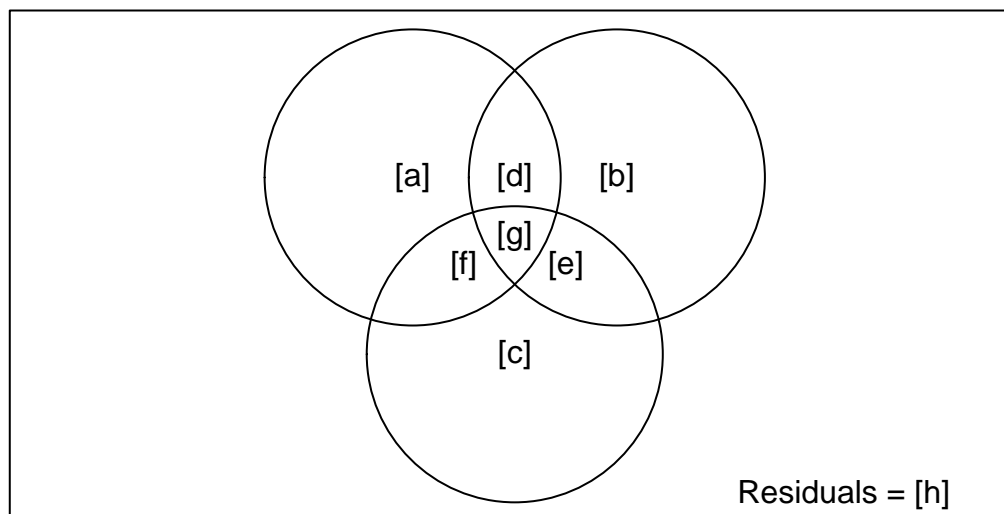
## [a+d+f+g] = X1      1  0.09646      0.04627      TRUE
## [b+d+e+g] = X2      3  0.32674      0.20050      TRUE
## [c+e+f+g] = X3      1  0.09646      0.04627      TRUE
## [a+b+d+e+f+g] = X1+X2 4  0.35382      0.18150      TRUE
## [a+c+d+e+f+g] = X1+X3 1  0.09646      0.04627      TRUE
## [b+c+d+e+f+g] = X2+X3 4  0.35382      0.18150      TRUE
## [a+b+c+d+e+f+g] = All 4  0.35382      0.18150      TRUE
## Individual fractions
## [a] = X1 | X2+X3      0              0.00000      FALSE
## [b] = X2 | X1+X3      3              0.13524      TRUE
## [c] = X3 | X1+X2      0              0.00000      FALSE
## [d]                   0              0.00000      FALSE
## [e]                   0              0.00000      FALSE
## [f]                   0             -0.01900      FALSE
## [g]                   0              0.06526      FALSE
## [h] = Residuals              0.81850      FALSE
## Controlling 1 table X
## [a+d] = X1 | X3      0              0.00000      FALSE
## [a+f] = X1 | X2      1             -0.01900      TRUE
## [b+d] = X2 | X3      3              0.13524      TRUE
## [b+e] = X2 | X1      3              0.13524      TRUE
## [c+e] = X3 | X1      0              0.00000      FALSE
## [c+f] = X3 | X2      1             -0.01900      TRUE
## ---
## Use function 'rda' to test significance of fractions of interest

## Warning: collinearity detected: redundant variable(s) between tables X1, X3
## results are probably incorrect: remove redundant variable(s) and repeat the analysis

## Warning: collinearity detected: redundant variable(s) between tables X1, X2, X3
## results are probably incorrect: remove redundant variable(s) and repeat the analysis

```

```
showvarparts(3)
```



```

r2_adj_cca = function(cca_obj, nperm, analytical = FALSE) {
  ## From Eq. 4 and 5 of Peres-Neto et al. 2006 - Ecology Returns a vector of
  ## R2, R2adj Arguments cca_obj: the output of rda or cca that was specified
  ## using a model formula nperm: the number of permutations to perform, if
  ## nperm not specified the analytical r2 and/or r2adj is returned Note: for
  ## CCA only the permutation based r2 adj is unbiased explanatory matrix
  r2 = cca_obj$CCA$tot.chi/cca_obj$tot.chi
  n = nrow(cca_obj$CCA$Xbar)
  if (missing(nperm)) {
    # eq 4 Peres-Neto
    p = cca_obj$CCA$rank
    out = c(r2, 1 - (((n - 1)/(n - p - 1)) * (1 - r2)))
  } else {
    if (nperm <= 0)
      stop("nperm argument must either be a positive integer or not specified")
    if (!any(grepl("~", cca_obj$call)))
      stop("The model object must be specified using a model formula rather than providing individ")
    rand.r2 = rep(NA, nperm)
    Y_string = as.character(cca_obj$terms[[2]])
    Y = eval(parse(text = Y_string))
    for (i in 1:nperm) {
      Yrand = Y[sample(n), ]
      # assign(Y_string, Yrand) cca.rand = eval(cca_obj$call)
      cca_obj$call[2] = sub("comm", "Yrand", cca_obj$call[2])
      cca.rand = eval(parse(text = paste(cca_obj$call[1], "(", cca_obj$call[2],
        ", data=", cca_obj$call[3], ")", sep = "")))
      # cca.rand = update(cca_obj, Yrand ~ .)
      rand.r2[i] = cca.rand$CCA$tot.chi/cca_obj$tot.chi
      if (i%100 == 0)
        print(i)
    }
    # Eq 5 Peres-Neto
    out = c(r2, 1 - ((1 - r2)/(1 - mean(rand.r2))))
  }
  names(out) = c("r2", "r2adj")
  return(out)
}
r2_adj_cca(dune_cca, 100)

```

```
## [1] 100
```

```
##          r2          r2adj
## 0.7106267 0.0000000
```

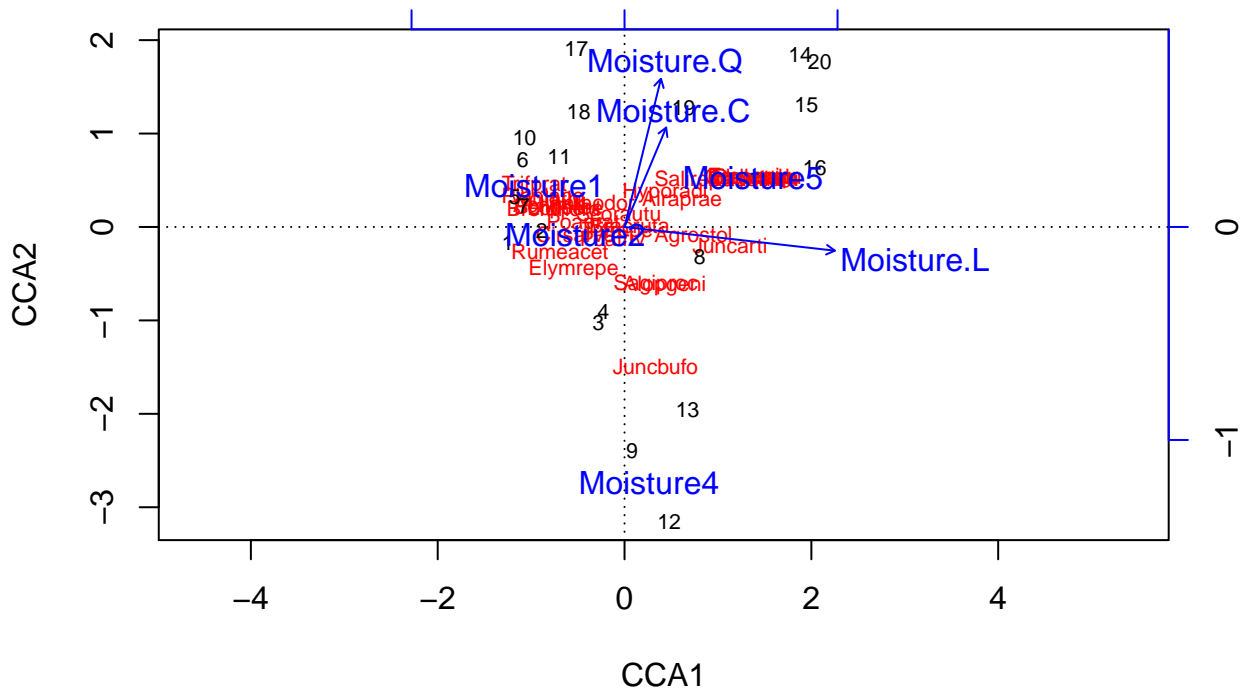
The plot is very hard to interpret due to the variables with different factors. Using the `anova()` function we can get an idea of the fit of the overall model compared to a random matrix. Based on the p-value (0.029) and the residual variance being smaller than the model variance we can conclude that the model does a pretty good job of explaining the dune vegetation. Furthermore, the R2 is about 71% which is pretty good. That is, it appears that A1, moisture, management, use and manure play some role in the dune vegetation community. Next, I ran the `anova()` function again using the `by="margin"` argument to get an idea of the individual importance of the variables. None of the p-values seem significant, but the chisquare value for moisture is the largest which would lead me to believe that moisture describes the most variance in the dune vegetation followed next by manure. The `varpart()` function also allowed me to compare the partition of

the different variables. I separated the variables into A1, moisture and landuse (management, manure, uses). Once again we still see that moisture is most important with an adjusted R2 of 0.20. Furthermore, when running a cca of must moisture and manure we get smaller p-values and larger chisquare values for those individual variables than when we ran the full model. The R2 for the individual models is 29.7% and 28.9% respectively. This leads me to believe that we may have included too many variables originally and we may get more information from a simpler model. See code below for the two additional models:

```
cca_moisture <- cca(dune~Moisture, data=dune.env)
anova(cca_moisture)
```

```
## Permutation test for cca under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: cca(formula = dune ~ Moisture, data = dune.env)
##           Df ChiSquare      F Pr(>F)
## Model      3   0.62831 2.2536 0.002 **
## Residual  16   1.48695
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(cca_moisture)
```



```
r2_adj_cca(cca_moisture, 100)
```

```
## [1] 100
```

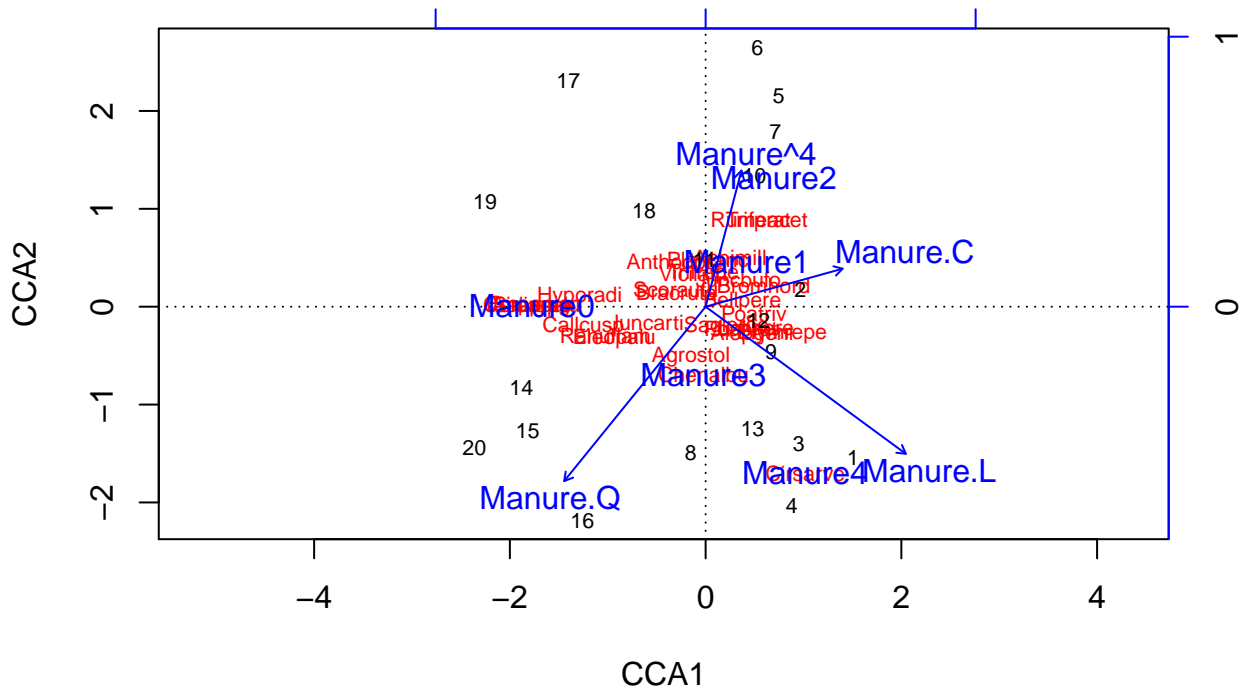
```
##           r2      r2adj
## 0.2970359 0.0000000
```



```
cca_manure <- cca(dune~Manure, data=dune.env)
anova(cca_manure)
```

```
## Permutation test for cca under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: cca(formula = dune ~ Manure, data = dune.env)
##           Df ChiSquare      F Pr(>F)
## Model      4   0.61156 1.5251 0.023 *
## Residual 15   1.50370
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(cca_manure)
```



```
r2_adj_cca(cca_manure, 100)
```

```
## [1] 100
```

```
##           r2      r2adj
## 0.2891171 0.0000000
```

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?

I believe the two analyses complement one another pretty well. The first, was an exploratory analysis of the data. This original analysis led us to look further into the environmental variables that were lumping the different dune species together, especially moisture. When we used the cca analysis we saw that environmental

factors were important in describing the dune community. Originally, it appears that all of the different environmental factors may be important. However, when looking further, using the `varparts()` and `anova` by margin it appears that moisture is the most important variable for the dune vegetation. This complements what we found earlier using the NMDS. However, we need to be careful about overfitting our model as we may have done originally. Based on the NMDS I think we should have ran a CCA with just moisture along as we did later on. I believe the CCA is more informing than the NMDS. You can also run more tests such as `anova()` and `varparts()` on the CCA to get a better idea of the important variables.