Multivariate statistics

Aims of this practical

1. Understand when ordinations are appropriate

2. Carry out ordination procedures

- Principal component analysis (PCA)
- Non-metric multidimensional scaling (NMDS)

3. Relate environmental variables to species community data

4. Report your results

Coding you will learn:

Procedures conducted

Name	**Data type**
PCA	Species data or environmental data
NMDS	Species data with fitted environmental data

Other useful code…
protest
envfit
Diversity metric calculation: *specnumber* and *diversity*
simper
anosim
adonis

Principal component analysis (PCA) – species data

Set up your script like previous weeks with the code you can copy from weeks 1-5 (see below). The first example we will look at is the built in dataset *varespec*. You will need to install and load the "**vegan**" package in R-Studio first (use *.libPaths()* if on Uni PC).

Once vegan is loaded you can load the dataset *varespec*. If you want some information about this dataset type **?varespec**. You'll see that the data contains estimates of lichen vegetation cover for 44 lichen species from 24 sites (rows)

If we want to explore the species composition per site we could compare each separate pair of species individually, but this would soon end up with a huge number of graphs, correlations and models. Therefore we will look at all species simultaneously using the ordination technique PCA.

```
# DD/MM/YY  Your Name
# Data: various
# Analysis: community ordination
.libPaths(“C:/Rpackages”) # need to create Rpackages folder if using Uni PC
library(vegan)

# clears r console
rm(list=ls())
# set working directory
setwd("/path/to/my/data/My_R_Space")

data("varespec") # load built in data
```

Running the PCA:
Running the ordination in R is simple if you have set up your data well (this built in dataset is nicely set up, no missing, mistaken or odd values). We have several options for running PCA in R, for this course we will use the **rda** function (**prcomp** is another option).

```
# PCA analysis using rda method
lichen.pca <- rda(varespec)
lichen.pca  # gives the basic PCA result
summary(lichen.pca) #
```

The first line runs the PCA and creates an object called lichen.pca that holds the results. Calling this object in the second line of code above gives the basic results, the total variation (also known as inertia) explained by the ordination (1826) and the amount of variation explained by each ordination axis (also called an Eigenvalue). What we would look for here is the first two or three axis explaining the vast majority of the variation.

In the third line of code above we use the summary command to get the full results of the

PCA. This gives us all 23 axes and their contribution to the total variation explained. It also gives us the PCA axis scores for each species and for each site in the dataset. We can then visualise these scores using the plot function.

Plotting the PCA:

```
plot(lichen.pca,  scaling = -1)
plot(lichen.pca, display = "species", scaling = -1)
plot(lichen.pca, display = "sites", scaling = -1)
biplot(lichen.pca, scaling = -1)
```

The first plot shows both the sites and species scores for the first two ordination axes. The second plot displays just the species while the third plot is just the sites. The fourth plot is a biplot, this shows both species and sites but gives arrows indicating the direction of influence of each species on the position of sites.

Which plot you use in a report depends on your aims, if you are only interested in the site differences in composition you might opt for the plot that only displays the sites. If you want to report which sites are more associated with different species then the biplot would be more appropriate. You would also need to report the variation explained by each of the axis in your plot. In this case we could calculate this from those give in the results or simply look at the 'Proportion Explained' in the summary.

Variation explained by axis one = 982.98/1826
Variation explained by axis one = 464.30/1826

Environmental variables:
We can also explore the influence of multiple environmental variables on our species composition by using the **envfit** function in the vegan package. The *varespec* dataset also comes with an environmental dataset called *varechem*, load this now and have a quick look at the variables.

The function **envfit** will not only test the significance of each environmental variable against the community composition calculated in the PCA it will also provide a very easy method of plotting environmental vectors on to our ordination plot.

Run the **envfit** function as show below and then run the **ef** object and look at your results, how many environmental variables are having a significant influence on the species composition in the PCA?

Now run the plotting functions shown below. Try changing the p.max values and re-run the plotting code to see what this does to your plotted environmental vectors.

```
data(varechem) # load built in dataset
ef <- envfit(lichen.pca, varechem, permu = 999)
ef

plot(lichen.pca, display = "sites")
plot(ef, p.max = 0.05)
```

Non-metric multidimensional scaling (NMDS)

The next dataset we will use for this practical is call ‘*marsh.csv*’ and can be downloaded from Moodle. Load this data in R and explore the variable names and attributes. This dataset contains species and environmental data from three saltmarsh sites in Ireland. The first column gives the individual quadrat names; the next 32 columns contain the species data and the remaining 12 columns are environmental data and site factors (marsh name and type).

The ‘type’ column has two categories, natural and managed. Try plotting two or three of the environmental variables against site ‘type’. Do you notice any obvious differences between natural and managed marsh?

Next pick one of the species columns and try plotting this against two or three of the environmental variables. Do you find any strong patterns?

If we carried on plotting like this to explore the data we would end up with a large number of plots, and if you followed each plot with a statistical test (e.g. t-tests and/or regressions) we would have a massive amount of confusing interpretation to carry out!

The vegan package allows us to calculate some basic diversity metrics using the **specnumber** and **diversity** functions. These are really useful ways to calculate simple metrics (see code below).

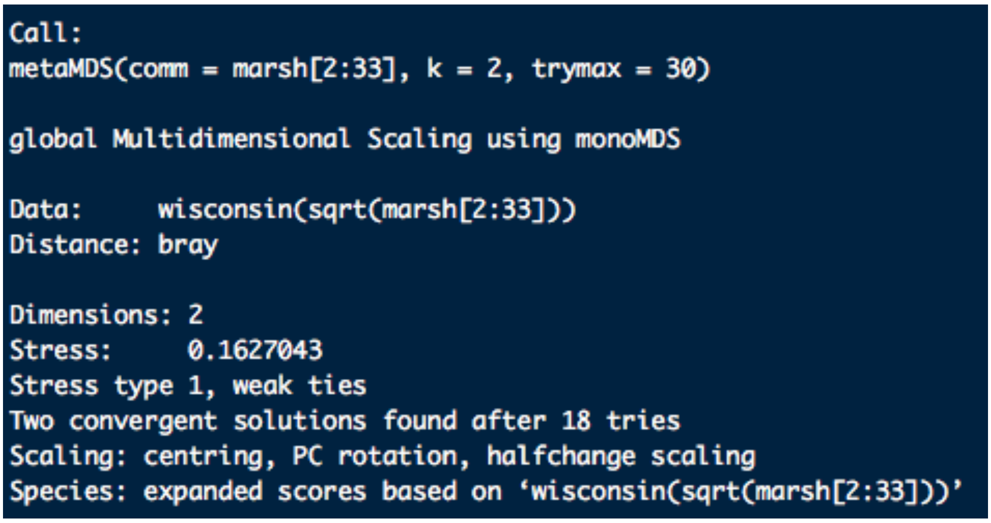
```
# diversity metrics: species richness, Shannon diversity
spp.rich <- specnumber(marsh[2:33], MARGIN=1) # 2:33 selects just the species data
shannon <- diversity(marsh[2:33])
```

These objects can now easily be added to our “marsh” dataset using the **cbind** function. Once you have added these objects to the marsh dataset create two boxplots to view species richness and Shannon diversity for marsh type.

Now we will carry out a NMDS ordination on the species data. The metaMDS code below runs the ordination, k=2 restricts the analysis to two axes and trymax=30 limits the number of runs/iterations to 30 (for very large datasets you can reduce this to speed up the computation time).

```
## NMDS ordination
marsh.nmds <- metaMDS(marsh[2:33], k=2, trymax = 30)
marsh.nmds  # look at the results output
scores(marsh.nmds)
```

When you run “marsh.nmds” you can view the results output.



```
Call:
metaMDS(comm = marsh[2:33], k = 2, trymax = 30)

global Multidimensional Scaling using monoMDS

Data:     wisconsin(sqrt(marsh[2:33]))
Distance: bray

Dimensions: 2
Stress:     0.1627043
Stress type 1, weak ties
Two convergent solutions found after 18 tries
Scaling: centring, PC rotation, halfchange scaling
Species: expanded scores based on ‘wisconsin(sqrt(marsh[2:33]))’
```

The important things to note here are the Stress value and the Distance used. Stress is a way of assessing the fit of the ordination, the lower the stress the better. A general rule of thumb for NMDS is that stress values above 0.2 suggest the ordination should be treated with caution and those above 0.3 suggest the placement of points on the plot is very unlikely to be representing the true dissimilarity.

The distance is the dissimilarity measure used in the ordination, in this case we went for the default Bray Curtis distance. Run “?metaMDS” to see other distance measures you could have used.

R can very easily calculate other dissimilarity measures and these can be used in the NMDS analysis. For example, you could use the following code to first calculate a dissimilarity matrix using the gower distance and then use this in the metaMDS function.
gower.dist <- vegdist(marsh[2:33], method="gower", upper=TRUE) # beta diversity matrix
gower.nmds <- metaMDS(gower.dist, k=2, trymax = 30)
It all depends on which aspects of the composition you want to emphasis in the analysis.

We should also check the NMDS scores, as we selected k=2 we only get axis 1 and 2 scores, all the possible variation in the community data is fitted on to these two axes.

Lets look at some plotting options. The vegan package gives us some really nice plotting tools such as oriellipse, ordihull and oridispider. See the code below for a couple of these and try out ordihull for yourselves.

```
## NMDS plot
plot(marsh.nmds, display = "sites")
with(marsh, ordiellipse(marsh.nmds, site, col=4, lwd=2, draw = "polygon", kind = c("sd")))
with(marsh, ordispider(marsh.nmds, site, label=TRUE))
```

Ordiellipse has drawn polygons around the site centroids. This polygon represents the standard deviation of point scores for that site. Ordispider has drawn a line from each point back to the site centroid so you can visualise the spread of each site.

Try changing the “site” part of the ordiellipse and oridspider with “type”. What can you tell from this plot?

Lets add the NMDS axis 1 and 2 scores to the main “marsh” dataset

```
## Housekeeping: add nmds scores to ‘marsh’ dataset
marsh.nmds.scores<-scores(marsh.nmds)
marsh<-cbind(marsh, marsh.nmds.scores)
names(marsh)
```

Simper is a nice easy tool to investigate which species are having the greatest effect on community composition between pairs of sites.

```
# simper - discriminating species between groups
simp<-simper(marsh[2:33], marsh$type, permutations = 999)
summary(simp)
```

After running the summary(simp) code you will get a table telling you each species contribution (contr) to the composition differences; for this data Atr.port is the species causing most of the variation between managed and natural sites. The av.a and av.b tells you the average abundances per group. The final column gives you a p-value so you can tell which species are significantly contributing to the variation between site types.

The next test we can easily undertake using our NMDS ordination is an Analysis of Similarities (anosim). This provides a way to test significantly whether two groups (site types in our case) are significant different.

```
# anosim – Analysis of similarity
sim.type<-anosim(marsh[2:33],  marsh$type, permutations = 999, distance = "bray")
summary(sim.type)
```

The key results to take away from the summary are the ANOSIM statistic and the significance (p-value). There are other ways to calculate similar statistics so if this is something you will do in the future look at the function adonis, the technique of multivariate analysis of variance (MANOVA) and especially the R-package “mvabund”.

Environmental variables:

Now we want to check out the environmental variables in the dataset, we can do this with the **envfit** function. The variables measured were all taken to learn something about the erosion/stability potential on coastal marsh and were carried out along a strong tidal gradient so you would expect a fair amount of correlation between these variables. What code could you use to look at the correlation in these variables in one single plot? Below we will again use the **envfit** function to look at the environmental data.

```
# envfit: Environmental variables fitted to an ordination
ef <- envfit(marsh.nmds, marsh[34:43], na.rm = TRUE, permu = 999)
ef
# Plot the fitted vectors
plot(marsh.nmds, display = "sites")
plot(ef, p.max = 0.05)
```

Ok, if we run the **envfit** object, **ef**, we get the results of our fitted vectors (env variables). In this nicely constructed example all the variables are highly significant (p<0.001). We can view the vectors with the above plotting code. You can see from the plot of environmental vectors that many are correlated and are mainly acting on axis 1 of the ordination. This is a great example of where we could use PCA to reduce the environmental data to a single compound variable (we will look at that soon).

Another way to plot environmental variables on to your ordination plots is to use surface plotting via the **ordisurf** function. Like topographical mapping lines, ordisurf will plot an environmental variable as a surface over your ordination! Awesome right! Lets have a go!

```
# Fit environmental surface to ordination
plot(marsh.nmds, display = "sites", type="n")
with(marsh, ordihull(marsh.nmds, type, col = "black",lty = 2))
with(marsh, ordispider(marsh.nmds, type, label=TRUE))
with(marsh, ordisurf(marsh.nmds, Bio.mass, add = TRUE, col = "cornflowerblue"))
```

What a great plot! What we have done is plot all the variation in saltmarsh vegetation composition for both managed and natural marsh, we have also demarcated the total compositional variation for each marsh type using ordihull and ordispider and then shown a clear gradient of biomass as a surface! Wow! This shows that sites that have greater axis 1 scores have greater biomass scores than those with lower axis 1 scores, and that natural sites mostly have greater biomass than managed sites. Again, wow! Using these techniques you are able to clearly visualise and summarise patterns using massive multivariate datasets.

There are many plotting options we can use, have a go at changing the plotting functions, colours etc. Below is another option for plotting points with different colours.

```
plot(marsh.nmds, display = "sites", type="n")
points(marsh.nmds.scores[marsh$type=="managed",],pch=16, col="cornflowerblue")
points(marsh.nmds.scores[marsh$type=="natural",],pch=16, col="green")
with(marsh, ordisurf(marsh.nmds, Bio.mass, add = TRUE, col = "red"))
```

For the next part of the practical we will try to reduce all the environmental data down to a single compound variable that indicates the erosion/stability potential of coastal marsh. To do this we need to use a PCA on only the environmental data. We can also quickly plot the PCA to investigate differences in only environmental factors [note: Environmental factors are now species in this PCA].

```
# PCA on env variable: data reduction
env.pca <- rda(marsh[34:43], scale=TRUE)
summary(env.pca)
biplot(env.pca)
```

Ok, again we can see that the majority of the environmental vectors are running along the first axis. Also, from the summary we can see that axis 1 explains the vast majority of the explained variance (eigenvalue). This is good news, we can now extract the PCA scores and use these in further analysis.

```
# Extract PCA scores
env.pca.scores<-scores(env.pca)
env.pca.scores$sites
marsh<-cbind(marsh, env.pca.scores$sites)
names(marsh)

names(marsh)[48] <- "stability.pc1"  # change PC1 column name
```

We can change the name of the PC1 variable to ‘stability.pc1’ as this compound viable is telling us something about the marsh stability. **Try plotting this new variable with the species richness variable** (spp.rich) we calculated and combined with the “marsh” dataset earlier. You could also try modeling this relationship.

Remember from the lecture that NMDS can be calculated with other dissimilarity measures. We used the default Bray Curtis dissimilarity for the NMDS above, now lets try it with the binary (presence/absence) Jaccard’s dissimilarity. This measure will convert the abundance data to 1’s and 0’s.

```
# Jaccard dissimilarity
marsh.jacc<-vegdist(marsh[2:33], "jaccard", binary=TRUE) # dissimilarity calculation for species
# data columns

# Rerun the ordination
jacc.nmds <- metaMDS(marsh.jacc, k=2, trymax = 30)
plot(jacc.nmds, display = "sites")
with(marsh, ordiellipse(jacc.nmds, site, col=4, lwd=2, draw = "polygon", kind = c("sd")))
with(marsh, ordispider(jacc.nmds, site, label=TRUE))
```

You should be able to see from this new ordination that differences between marsh types are smaller when using Jaccard's dissimilarity (i.e. there is more overlap in polygons).

We can use Procrustes rotation to evaluate the two different NMDS ordinations (Bray Curtis and Jaccards). This test will tell us how strongly correlated these two ordinations are, the higher the Procrustes correlation value the more similar the ordinations are. With the code below we get a value of 0.85 suggesting that they are highly correlated.

```
proT1<-protest(marsh.nmds, jacc.nmds, permutations = 999)
proT1
plot(proT1)
```