NES8010 Quantitative Ecological Research Methods

Introduction to multivariate analysis of ecological data

## Aims

The main aims of this practical are to introduce you to doing multivariate statistical analyses using the add-on vegan analysis package to R. The package was originally designed for vegetation analysis, but its methods are suitable for any group of organisms, from genomes to vertebrates. You can use them to analyse any set of attributes x samples data (e.g. animal behaviours etc.) In particular, you will learn:

* Installation and loading of add-on packages, and have a look at the online tutorial
* Undertake unconstrained ordination using PCA, CA, DCA and NMDS
* Learn how to present ordination graphs, and their biological interpretation.

## Introduction

Multivariate techniques (MVT) are of most value where you have a large number of attributes associated with each sample in your dataset. These datasets are common in Environmental Sciences, and are often in the form of tables, for example:

* plant species x vegetation quadrats
* invertebrate species x pitfall traps
* soil chemistry x soil cores
* marine samples x benthic species

Whilst you could analyse each species or each soil chemical variable separately, you run the risk of multiple testing (give false significant P-values). One way around this is to analyse all your data simultaneously to provide a descriptive analysis of the similarity between samples (unconstrained ordination). In the next practical we will look at more complex methods that test explanatory variables in simultaneously against multiple response variables.

## Before you begin

As usual, start RStudio, then create a new R Project in an empty folder. Projects have the advantage of keeping all your data files, R scripts, output graphs etc. in one place, making it easier to reproduce your research. After creating a new R Project, please create a new R Script (click on the main menu, File -> New File… -> R Script). You are recommended to keep a record of the R commands in your R script. Remember to break up your script using blank lines, and comments # so that you can easily understand it. Putting three hash symbols ### at the end of a line creates a table of contents in RStudio making it very simple to navigate your code.

## The vegan library

To install the add-on vegan library, issue the command:

install.packages("vegan")

It is likely to install some additional packages in the background to supplement the vegan library; do not worry about this and they should install without problems on your drive.

There is an excellent 40-page tutorial to the vegan package written by Jari Oksanen on the Blackboard website; plus a short tutorial working through the basics of ordination. The vegan package also works well with the labdsv package. There is again an excellent tutorial on labdsv, at <http://ecology.msu.montana.edu/labdsv/R/> which is primarily aimed at community ecology, but also covers summarising data, GLMs etc.

Before going any further please download the file nes8010.R from Canvas, and move it from your Downloads folder into the R Project folder you have just created. Next, run the following three lines:

library(vegan)  
library(tidyverse)  
source("nes8010.R")

If you receive an error saying that nes8010.R cannot be found, check your R Project folder contains the file using File Explorer (Windows) or using Finder (Mac). You may want to check what RStudio thinks it is using as its folder by issuing getwd() in the Console window. **Note**: you will need to issue these three lines to activate the vegan, ggplot2 and nes8010 commands. The nes8010.R file contains various functions to make it easier to use R for multi-species analysis, and produce ggplot2-compatible graphs.

## Datasets

We will look at some community ecology data, of vegetation cover. The example we will work with has been widely described in a large number of publications over the last 20 years, and is from a sand dune vegetation survey. Potential explanatory data includes basic soil and management conditions. We will also use some data collected along the causeway at St Mary’s Lighthouse, but using a multivariate rather than univariate approach. Today’s dataset comprises a table of sites (samples or quadrats) by species. You may see these described as “samples” by “attributes”, because multivariate techniques are not restricted only to species data. Attributes might be biochemical composition of medicinal plants, pollution data from an environmental survey etc. The techniques available will work with any complex “messy” multi-attribute data.

The sand dune data are example datasets from the vegan package: load these with the commands

data(dune)  
data(dune.env)

for the species and environmental data. If you have problems loading them directly off the library, there are examples on the Canvas website that you can read into your workspace using the usual commands. Summarise the data using the summary command, and view the datasets on screen in the RStudio viewer. Each row represents a different site (sample quadrat for these data) and each column a different attribute (species). If you have multivariate data, R will expect you to provide the data in the form of rows=samples, columns=attributes. However, it is easy to correct if you get the table the wrong way round using the t (transpose) command to swap the rows and columns.

## IMPORTANT: format of the data

Our ecological multivariate data analyses in R uses the vegan package which assumes you have a data.frame as the input type, i.e. columns are species, and rows are sites, quadrats or similar. Thus you have a data.frame with column names and rownames. Unfortunately vegan is not (yet) 100% compatible with the tidyverse approach. The vegan package expects “wide” data, so you might have to use the pivot\_wider function from the tidyrpackage to convert your data to “wide” format, and use the rownames function to add rownames. However, typically you will initially enter a big table of species x sites in Excel for ease of data entry, and if you use read.csv (rather than read\_csv) it will automatically be setup correctly for use in vegan.

## Unconstrained ordination - PCA

PCA analysis is useful for linear unconstrained ordination; we will investigate the latter here, both are available in R. Use the rda command to undertake either PCA or a constrained version, redundancy analysis (RDA) when you wish to include explanatory variables directly in the analysis. R will *automatically undertake a PCA if you only provide one set of data* without any explanatory variables to the rda command. To undertake a PCA:

dune.pca <- ordi\_pca(dune)  
summary(dune.pca)

##   
## Call:  
## rda(X = spp\_data)   
##   
## Partitioning of variance:  
## Inertia Proportion  
## Total 84.12 1  
## Unconstrained 84.12 1  
##   
## Eigenvalues, and their contribution to the variance   
##   
## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6 PC7  
## Eigenvalue 24.7953 18.1466 7.62913 7.15277 5.6950 4.33331 3.19936  
## Proportion Explained 0.2947 0.2157 0.09069 0.08503 0.0677 0.05151 0.03803  
## Cumulative Proportion 0.2947 0.5105 0.60115 0.68618 0.7539 0.80539 0.84342  
## PC8 PC9 PC10 PC11 PC12 PC13 PC14  
## Eigenvalue 2.78186 2.4820 1.85377 1.74712 1.31358 0.99051 0.637794  
## Proportion Explained 0.03307 0.0295 0.02204 0.02077 0.01561 0.01177 0.007582  
## Cumulative Proportion 0.87649 0.9060 0.92803 0.94880 0.96441 0.97619 0.983768  
## PC15 PC16 PC17 PC18 PC19  
## Eigenvalue 0.550827 0.350584 0.199556 0.148798 0.115753  
## Proportion Explained 0.006548 0.004167 0.002372 0.001769 0.001376  
## Cumulative Proportion 0.990316 0.994483 0.996855 0.998624 1.000000  
##   
## Scaling 2 for species and site scores  
## \* Species are scaled proportional to eigenvalues  
## \* Sites are unscaled: weighted dispersion equal on all dimensions  
## \* General scaling constant of scores: 6.322924   
##   
##   
## Species scores  
##   
## PC1 PC2 PC3 PC4 PC5 PC6  
## Achimill -0.6038 0.1239 0.008464 0.1596 0.4087 0.1279  
## ....   
##   
##   
## Site scores (weighted sums of species scores)  
##   
## PC1 PC2 PC3 PC4 PC5 PC6  
## 1 -0.8568 -0.1724 2.608 -1.13 0.4507 -2.491  
## ....

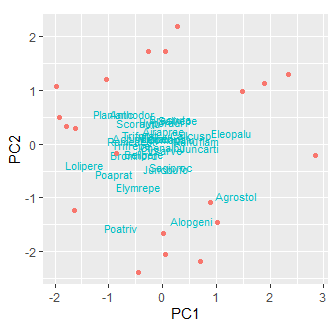
This will show a lot of information on screen, but it is the section headed Eigenvalues, and their contribution to the variance, Importance of components to which you should pay attention:

The summary will show you that 19 principal components have been extracted, which in total extract 100% of the variation - you will rarely look at more than the first 2 axes, PC1 & PC2. The summary indicates that your ordination has “captured” over 50% of the variation of the data on the first two axes - approximately 30% on PC1 and 20% on PC2. Even PC3 only accounts for 10%.

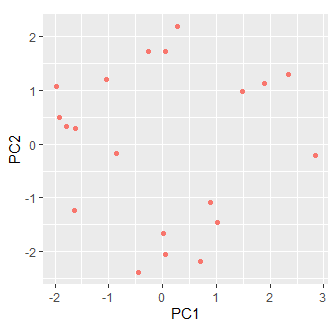
Plotting of ordination graphics is difficult, as there can sometimes be too much information to display, and there is not as yet a simple method for ggplot. At the end of this practical we will look at various plotting methods, including via ggrepel and Microsoft Excel. We will mainly focus on using the extra plotting functions in nes8010.R as these are compatible with ggplot.

You can plot the samples and attributes on the same graph, or separately:

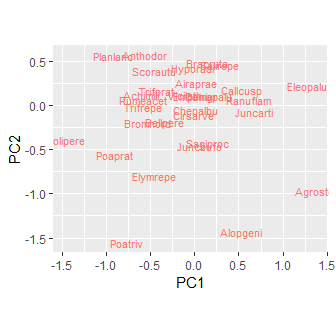
ordi\_plot(dune.pca) # Probably too cluttered to see clearly



ordi\_plot(dune.pca, display="sites")



ordi\_plot(dune.pca, display="species")



Remember: in your data what we are referring to as “sites” may be different samples, whilst the “species” may be different animal behaviour patterns, or sets of chemical contaminants. However, use the terms “sites” and “species” to obtain the plots you want. The most important species (i.e. *attributes*) in terms of differentiating between your sites (*samples*) will tend to be those with relatively high or low scores. Species in the centre of the PCA plot are often more ubiquitous, being found in most samples, and may discriminate poorly between sites. Sites far apart will have few species in common, those close together will be similar. There is a trend in these data along PC1 from sites dominated by species such as *Lolium perenne* (Lolper) and dune species such as *Elymus repens* (Elyrep) (e.g. sites 5, 7, 10) with higher scores on PC1 for on sites with species such as *Agrostis stolonifera* (Agrosto) e.g. sites 12,8,13.

### Interactive labelling of plots

You will often find that the plots are difficult to read or the species labels overlap. You can improve your plots in two ways:

1. If you use the ordi\_plot() function (from nes8010.R) rather than the default plot() function, you will create a ggplot2-compatible plot. This means that you can easily change its theme, axis labels etc in the usual way.
2. You can use the ordi\_identify() function (from nes8010.R) to interactively label only selected points. This is easiest to do if you store your plot in an R object, using the geom="point" option for points, and then display it. The ordi\_identify() function can then be used with the mouse to click on the points you wish to label. It isn’t perfect (I’ve written it) but you might find it useful.

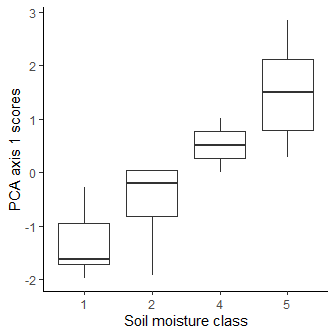
dune.pca <- ordi\_pca(dune)  
dune\_pca\_plt <- ordi\_plot(dune.pca, display="species", geom="point")  
dune\_pca\_plt # Displays your plot  
ordi\_identify(dune\_pca\_plt) # Hit Esc key to exit

You can extract individual scores separately; e.g. to extract PC1 site scores you can use the scores() function, with choices=1 for just the first axis

dune\_pc1 <- scores(dune.pca, display="sites", choices=1)

This can be compared with some of your explanatory data, for example:

ggplot(dune.env, aes(x=Moisture, y=dune\_pc1)) +  
 geom\_boxplot() +  
 xlab("Soil moisture class") +  
 ylab("PCA axis 1 scores") +  
 theme\_classic() # Compare with theme\_bw()



Moisture class 5 (wetter) has higher PC1 scores than moisture class 1. **Investigate the trends in Management, Manure, depth of soil horizon (A1) and land use (Use) by plotting in similar ways**. **Note:** some of the variables in dune.env are categorical, such as soil moisture class, whereas others are on a continuous scale. Use the summary command and ?dune.env to learn more about the data. If it is on a continuous scale geom\_boxplot is **not** correct. Which ggplot geom should you use?

Remember that the impact of the explanatory variables on individual species can be evaluated by comparison with the ‘species’ plot of PC1 v PC2, and the positions of species at high or low extreme ends of the axis. **Which species are associated with deeper soils (A1) or moister conditons?**

Whilst this method of doing an unconstrained analysis then comparing the scores with environmental factors is useful, you will find that including the environment in the ordination itself, via a constrained analysis, is much more powerful. **Question:** why might incorporating explanatory data after doing an unconstrained ordination be weaker than a constrained ordination?? (*Hint*: in regression what are the assumptions about independence between data points?) We will cover constrained ordination in the next practical.

## Arch effect in PCA

PCA is a linear ordination method - if there is a large amount of species turnover (i.e. many sites with few species in common) one result is the “horseshoe” or “arch” effect in the plots.

This can sometimes be corrected by rescaling or standardising the data before analysis, or by using an alternative analytical method. Common ones include unimodal techniques such as correspondence analysis (CA) and detrended correspondence analysis (DECORANA or DCA), or non-metric methods such as non-metric multidimensional scaling (NMDS). There has been criticism that DCA is sometimes too severe in its attempt to remove the arch effect, so we will focus on CA and NMDS. If you wish to use DCA, it is available in vegan as under the decorana command, but today we will focus on CA and NMDS.

We will use another built-in dataset available from the vegan package: varespec. This describes different species of lichens in reindeer-grazed pastures in Scandinavia (use ?varespec for details about the data). Begin by doing a PCA and note the “arch” effect:

data(varespec)

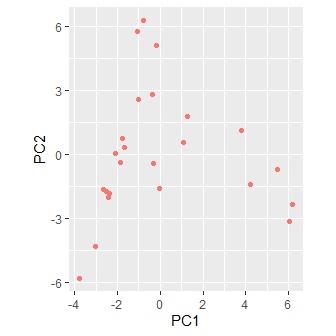
(As usual you can inspect the data in the RStudio viewer to see the dataframe).

##   
## Call:  
## rda(X = spp\_data)   
##   
## Partitioning of variance:  
## Inertia Proportion  
## Total 1826 1  
## Unconstrained 1826 1  
##   
## Eigenvalues, and their contribution to the variance   
##   
## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6  
## Eigenvalue 982.9788 464.3040 132.25052 73.9337 48.41829 37.00937  
## Proportion Explained 0.5384 0.2543 0.07244 0.0405 0.02652 0.02027  
## Cumulative Proportion 0.5384 0.7927 0.86519 0.9057 0.93220 0.95247  
## PC7 PC8 PC9 PC10 PC11 PC12  
## Eigenvalue 25.72624 19.70557 12.274191 10.435361 9.350783 2.798400  
## Proportion Explained 0.01409 0.01079 0.006723 0.005716 0.005122 0.001533  
## Cumulative Proportion 0.96657 0.97736 0.984083 0.989799 0.994921 0.996454  
## PC13 PC14 PC15 PC16 PC17  
## Eigenvalue 2.327555 1.3917180 1.2057303 0.8147513 0.3312842  
## Proportion Explained 0.001275 0.0007623 0.0006604 0.0004463 0.0001815  
## Cumulative Proportion 0.997729 0.9984910 0.9991515 0.9995977 0.9997792  
## PC18 PC19 PC20 PC21 PC22  
## Eigenvalue 0.1866564 1.065e-01 6.362e-02 2.521e-02 1.652e-02  
## Proportion Explained 0.0001022 5.835e-05 3.485e-05 1.381e-05 9.048e-06  
## Cumulative Proportion 0.9998814 9.999e-01 1.000e+00 1.000e+00 1.000e+00  
## PC23  
## Eigenvalue 4.590e-03  
## Proportion Explained 2.514e-06  
## Cumulative Proportion 1.000e+00  
##   
## Scaling 2 for species and site scores  
## \* Species are scaled proportional to eigenvalues  
## \* Sites are unscaled: weighted dispersion equal on all dimensions  
## \* General scaling constant of scores: 14.31485   
##   
##   
## Species scores  
##   
## PC1 PC2 PC3 PC4 PC5 PC6  
## Callvulg -0.147 0.3483 -0.2506 0.603 0.3961 -0.4608  
## ....   
##   
##   
## Site scores (weighted sums of species scores)  
##   
## PC1 PC2 PC3 PC4 PC5 PC6  
## 18 -1.027 2.592 -1.539 -3.231 -1.105 -2.203  
## ....

varespec.pca <- ordi\_pca(varespec)  
summary(varespec.pca)

As usual, the summary command produces a lot of output: focus on the percentage of variation explained by PC1 and PC2. This suggests almost 80% of the variation (0.7927) in the dataset has been explained by the PC1 plus PC2; always be suspicious when the explained variation is so high in a PCA. Another problem is immediately apparent when you plot the site scores, which are in the shape of a distinct “arch”:

ordi\_plot(varespec.pca, display="sites")



## Correspondence Analysis (CA) & non-metric multidimensional scaling (NMDS)

This arch effect can sometimes be reduced by CA, DCA or NMDS.

### Correspondence analysis

Correspondence analysis is a **unimodal** equivalent to PCA; its main difference is that it assumes the abundance of species rises and falls along each ordination axis, rather than changing in a roughly straight line. If using the basic vegan functions it is available via the cca function, but for greater consistancy there is a wrapper function in nes8010.R for it called ordi\_ca()

##   
## Call:  
## cca(X = spp\_data)   
##   
## Partitioning of scaled Chi-square:  
## Inertia Proportion  
## Total 2.083 1  
## Unconstrained 2.083 1  
##   
## Eigenvalues, and their contribution to the scaled Chi-square   
##   
## Importance of components:  
## CA1 CA2 CA3 CA4 CA5 CA6 CA7  
## Eigenvalue 0.5249 0.3568 0.2344 0.19546 0.17762 0.12156 0.11549  
## Proportion Explained 0.2520 0.1713 0.1125 0.09383 0.08526 0.05835 0.05544  
## Cumulative Proportion 0.2520 0.4233 0.5358 0.62962 0.71489 0.77324 0.82868  
## CA8 CA9 CA10 CA11 CA12 CA13 CA14  
## Eigenvalue 0.08894 0.07318 0.05752 0.04434 0.02546 0.01710 0.014896  
## Proportion Explained 0.04269 0.03513 0.02761 0.02129 0.01222 0.00821 0.007151  
## Cumulative Proportion 0.87137 0.90650 0.93411 0.95539 0.96762 0.97583 0.982978  
## CA15 CA16 CA17 CA18 CA19 CA20  
## Eigenvalue 0.010160 0.007830 0.006032 0.004008 0.002865 0.0019275  
## Proportion Explained 0.004877 0.003759 0.002896 0.001924 0.001375 0.0009253  
## Cumulative Proportion 0.987855 0.991614 0.994510 0.996434 0.997809 0.9987341  
## CA21 CA22 CA23  
## Eigenvalue 0.0018074 0.0005864 0.0002434  
## Proportion Explained 0.0008676 0.0002815 0.0001168  
## Cumulative Proportion 0.9996017 0.9998832 1.0000000  
##   
## Scaling 2 for species and site scores  
## \* Species are scaled proportional to eigenvalues  
## \* Sites are unscaled: weighted dispersion equal on all dimensions  
##   
##   
## Species scores  
##   
## CA1 CA2 CA3 CA4 CA5 CA6  
## Callvulg 0.04004 -1.739 0.1011 -2.333 0.1057 1.456  
## ....   
##   
##   
## Site scores (weighted averages of species scores)  
##   
## CA1 CA2 CA3 CA4 CA5 CA6  
## 18 -0.3926 -2.365 1.247 1.453 0.8775 -1.076  
## ....

varespec.ca <- ordi\_ca(varespec)  
summary(varespec.ca)

The percentage variation explained by the first two axes (42.33 %) is now more typical of an ordination of ecological data. To confirm that the arch effect has been removed plot the samples; interpretation of species is the same as for PCA:

ordi\_plot(varespec.ca, display="sites")  
ordi\_plot(varespec.ca, display="species")

Remember, if you want to interactively identify points on your graphs, you can use the geom="point" option, save the plot, and then ordi\_identify().

### Non-metric multidimensional scaling

NMDS uses a different approach to fit an ordination, and is more concerned with the relative ranking of samples along the ordination axes, rather than the actual distances between them. It uses a different analytical approach to come to a solution, which may involve rescaling the data. The vegan library has the metaMDS function which automates many of the issues. As it uses an iterative method to come to a solution, your output is likely to differ slightly from mine, but this is the output that I obtained. For consistencty in function naming, you may find it easier to call with ordi\_nmds():

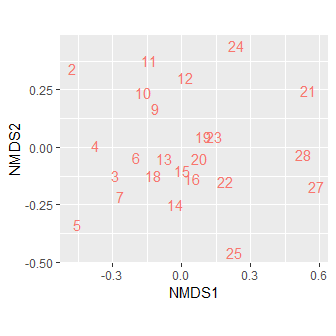
varespec.nmds <- ordi\_nmds(varespec)

## Square root transformation  
## Wisconsin double standardization  
## Run 0 stress 0.1843196   
## Run 1 stress 0.1825658   
## ... New best solution  
## ... Procrustes: rmse 0.04162539 max resid 0.1518001   
## Run 2 stress 0.1982376   
## Run 3 stress 0.2028828   
## Run 4 stress 0.18458   
## Run 5 stress 0.1967393   
## Run 6 stress 0.2028828   
## Run 7 stress 0.2279369   
## Run 8 stress 0.2127526   
## Run 9 stress 0.2048307   
## Run 10 stress 0.2093088   
## Run 11 stress 0.2109002   
## Run 12 stress 0.2154149   
## Run 13 stress 0.2187601   
## Run 14 stress 0.1975409   
## Run 15 stress 0.2085949   
## Run 16 stress 0.2092456   
## Run 17 stress 0.1825658   
## ... Procrustes: rmse 2.640659e-05 max resid 8.769071e-05   
## ... Similar to previous best  
## Run 18 stress 0.2328562   
## Run 19 stress 0.18458   
## Run 20 stress 0.2115175   
## \*\*\* Solution reached

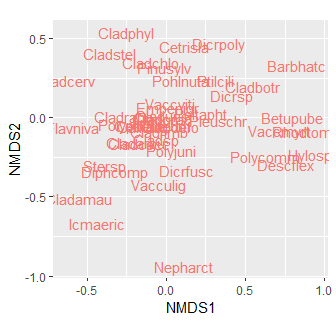
Here the command has decided that the best solution is to standardise the data before the NMDS. NMDS only returns 2 axes and does not provide a measure of how much variation is explained by these compared to the other axes (since there are not any other axes to compare with). The “stress” value from the analysis is usually reported; the lower the stress the better the NMDS. However there is not a single stress value which might be viewed as “good” or “poor”. Instead if you were to do several NMDS analyses (e.g. with different transformations of the raw data) you would select the one with the lowest stress. Fortunately R automatically selects the best transformation for your raw data.

To plot the results you also need to specify the type="text" option to display the text on the graphs:

ordi\_plot(varespec.nmds, display="sites", geom="text")

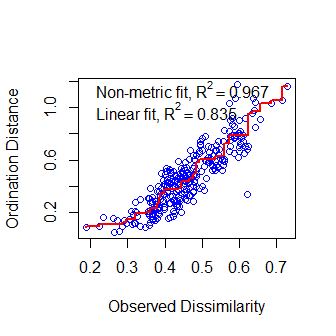


ordi\_plot(varespec.nmds, display="species", geom="text")



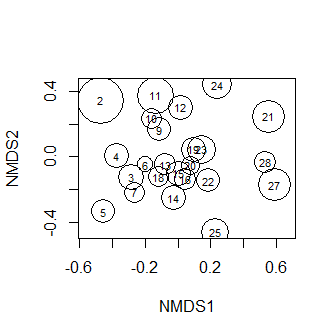
The lower the stress value, the better the NMDS ordination has been at capturing the original multivariate data into 2 dimensions. It is often useful to create a “stressplot” (also known as a “Shepard diagram”) after an NMDS to understand how well it has fitted the data; ideally the data should form an approximately straight line scatter:

stressplot(varespec.nmds)



You can check how well individual sites or species have been fitted in the ordination in NMDS (and other ordination techniques) using the goodness of fit commands. The bigger the circle the better the fit:

gof <- goodness(varespec.nmds)  
plot(varespec.nmds, display="sites", type="text")  
points(varespec.nmds, display="sites", cex=gof\*100)



### Detrended correspondence analysis

The decorana command can be used for DCA; you might want to try this in your own time after today’s practical, and compare the graphs for PCA, CA, DCA and NMDS. However, generally NMDS is recommended rather than DCA but you will see the latter in many older scientific papers.

# Ordination graphics

Good graphical display is essential to help you interpret your ordination results. Here we look at interactive labelling of points in R, the use of the ggrepel package, or exporting results back to Excel. Some of the graphs you have used are with additional functions (in nes8010.R) to make it easier to interface with ggplot2.

### Using ggrepel for your ordination graphs

The additional package ggrepel is very useful to reduce labels overlapping:

install.packages("ggrepel")  
library(ggrepel)

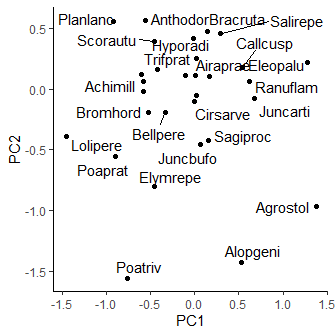
You will need to extract your ordination scores from your ordination output; here we will again look at the species scores. Unfortunately the default return from the scores command is a matrix and ggplot expects a data.frame (data.frames and tibbles are more versatile for plotting)

dunespp.sco <- data.frame(scores(dune.pca, display="species"))

We can use ggplot in the usual way, building up layer by layer, but take advantage of extra commands provided by ggrepel to reduce overlap between labels:

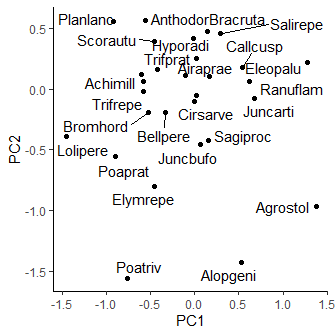
ggplot(dunespp.sco, aes(x=PC1, y=PC2, label=rownames(dunespp.sco))) +  
 geom\_point() +  
 geom\_text\_repel() +  
 theme\_classic()

## Warning: ggrepel: 6 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps



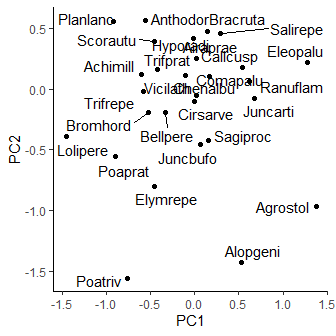
If this is still too crowded, you can drop the labels for certain species, such as *Empetrum nigrum* in the centre of the plot. You’ll find it easier if you convert your PCA scores into a tibble so that you no longer have to refer to rownames and can use the filter command to filter out any rows you don’t want. The != symbol means “not equal to”. Remember that in any ordination diagram, species near the centre of the plot tend to be relatively ubiquitous and found in most samples, whereas those nearer the edge will be of more ecological interest, as they occur in some samples, but not in others.

# Convert into a tibble, changing rownames to a new column SppID  
dunespp.sco <- as\_tibble(rownames\_to\_column(data.frame(dunespp.sco), var="SppID"))  
  
ggplot(filter(dunespp.sco, SppID != "Empenigr"), aes(x=PC1, y=PC2, label=SppID)) +  
 geom\_point() +  
 geom\_text\_repel() +  
 theme\_classic()



If you wanted to drop both *Empetrum nigrum* and *Rumex acetosa* it is usually easiest to have them in a list and precede SppID with ! (for “NOT”) and use the %in% operator:

ggplot(filter(dunespp.sco, !SppID %in% c("Empenigr", "Rumeacet")),  
 aes(x=PC1, y=PC2, label=SppID)) +  
 geom\_point() +  
 geom\_text\_repel() +  
 theme\_classic()



### Exporting ordination scores to Excel

Whilst R gives a lot of control on plotting, it may require extra coding or new commands to learn. Some students prefer to export data to Excel and use its plotting facilities. This shows you how to produce a species PCA plot, based on your original dune dataset; you can use the same method for samples plots.

**Note**: It is always much better to try and create your plots in RStudio rather than Excel. Using RStudio gives you a completely **reproducible** way of generating plots automatically. Excel relies on click-and-point mouse and menus, and so is less reliable. The following advice should merely considered as optional should you wish to use Excel.

If you have not already done so, first make sure that RStudio is looking in your BIO8010 folder by selecting:

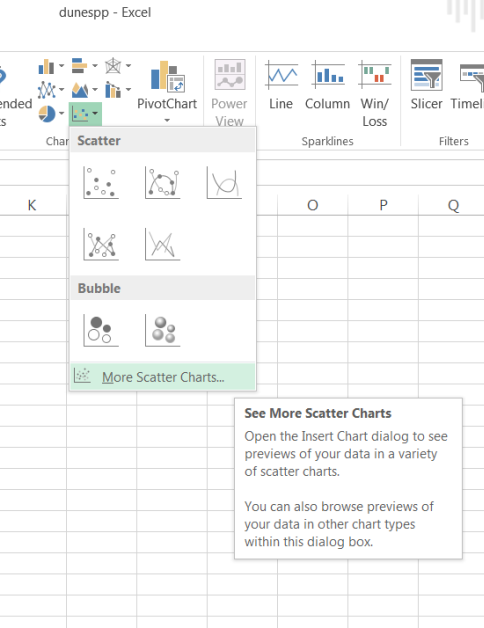
**Session -> Set Working Directory -> Choose Directory.**

and choosing the correct folder. Then extract the **species** scores for both PC1 and PC2, and then write them to a CSV file.

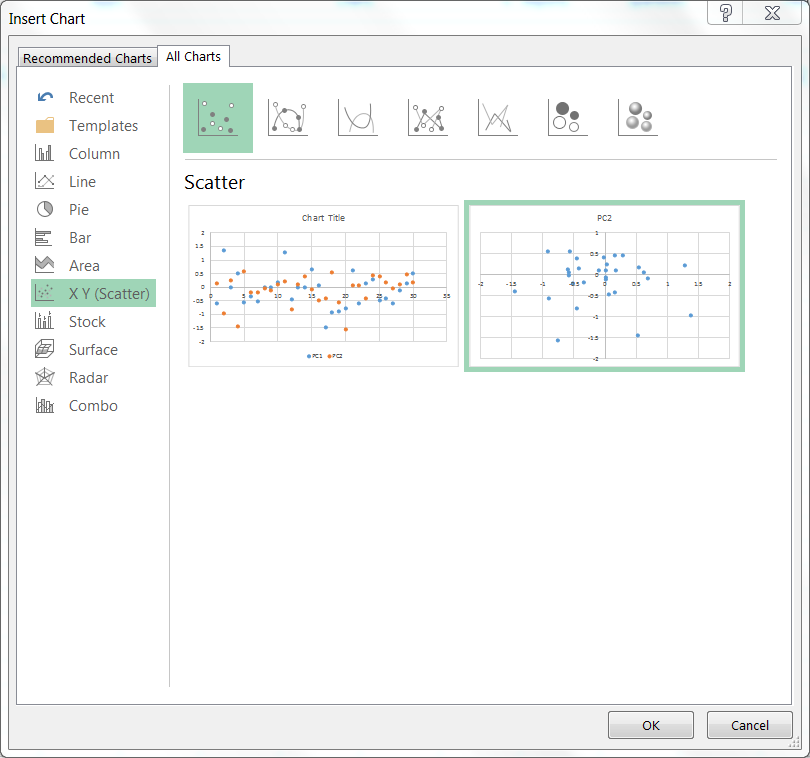
dunespp.sco <- scores(dune.pca, display="species")  
write.csv(dunespp.sco, "dunespp.csv") # Use write.csv rather than write\_csv

If you go to your NES8010 folder in File Explorer you should see the file dunespp.csv listed; depending on your Microsoft Windows configuration, it might only be listed as dunespp. Double-click on it and by default it will open in Microsoft Excel. After the data are loaded, you need to insert a chart.

Click on the **Insert** tab, select **Scatter** and choose the **More Scatter Charts…** option as shown below:

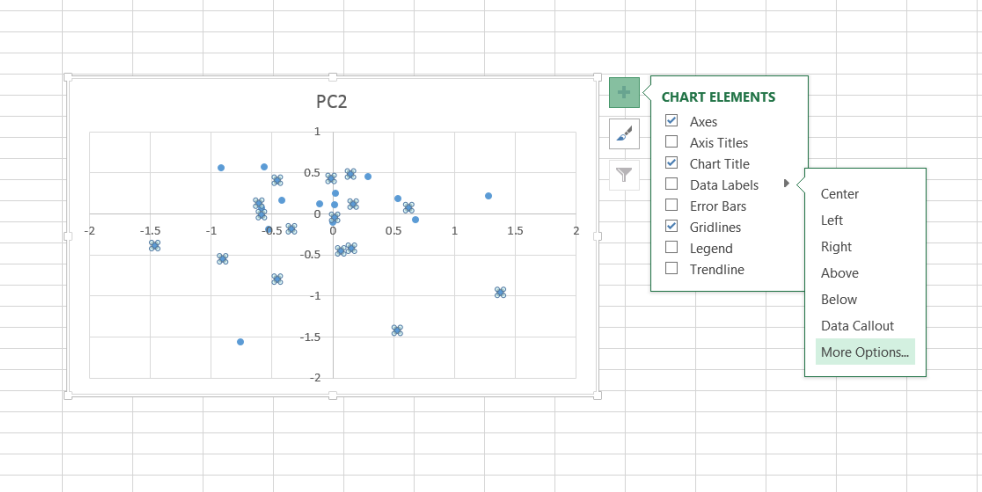


Under the **XY (Scatter)** option, ensure that you pick the option that gives you a true x and y graph, on the right:

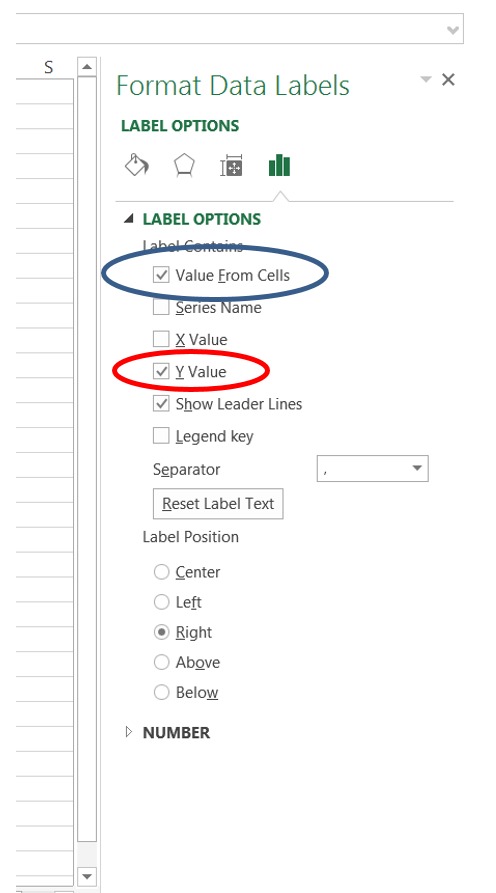


Click OK to insert the graph.

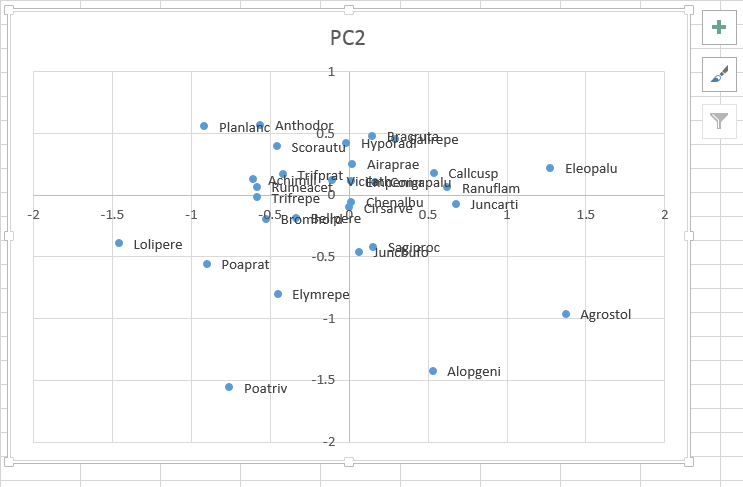
Finally to add species labels to this graph, click once on one of the points in the graph to select all the points, and click on the large + symbol to the right of the graph. This gives you access to various options, click on the arrow to the right of Data Labels as shown below, and select **More Options…**



This allows you to select column A as the range for your species labels. Click on **Values from Cells**, highlighted in blue in the diagram below:



This will produce a pop-up window called **Data Label Range**. Select the list of species names in column A (ignore the blank A1 cell) in cells A2:A31 and click OK. Unclick **Y value** (shown in red above) to stop the PC2 numbers being displayed. Your graph should now look similar to the following:



Where species names overlap it is easy in Microsoft Excel to move them with the mouse, change font size or font colour etc. Of course you will also want to change the chart title from the default of PC2 to something more sensible. The main problem with Excel is that it involves export/import and lots of mouse-clicks, and so can be harder to remember.