ENTMLGY 67072 Entomological Techniques and Data Analysis

R Activity 10: Multivariate community analyses

1 Characterizing insect communities with multivariate methods

Assessing patterns of species diversity is a common approach used in ecology to understand communities. However, the use of diversity indices has its limitations.

Although these indices provide information about the number of species at different sites, this measure doesn't provide any indication as to whether those species are the same or not. For example, two insect communities may each have 10 species. The calculation for species richness indicates that these two communities are similar in terms of their diversity. What we don't know is whether the same 10 species are shared between these two communities. How similar is the overlap in species composition between these two sites? Species diversity indices measured at the site-level (i.e., alpha-diversity) are often paired with ordination methods that can assess measures of beta-diversity, or the change in species composition among sites.

Diversity metrics take data from multiple species collected at a site and distill this information into a single value. The calculated diversity metric can then be used as a response variable in a univariate model (i.e., GLM or GLMM). Multivariate models allow for the analysis of multiple environmental factors on many species simultaneously by representing relationships among variables in a low-dimensional space. These approaches are also capable of handling noisy and redundant data, as well as sparse data (i.e., many zeros from species are occur at low frequencies). This is convenient because community data tend to contain more than two variables, and it is a challenge to understand the complexity of species responses using univariate approaches. Multivariate techniques are able to represent data along a reduced number of orthogonal axes (i.e., linearly independent, uncorrelated) such that they represent (in decreasing order) the main trends of the data.

Multivariate analyses aid in addressing some of these limitations, and therefore, can provide a more holistic understanding of community responses. These methods can reveal patterns and relationships in data sets by reducing the dimensions to a more manageable form. There are two main classes of multivariate community analyses: classification and ordination. Classification centers on the placement of species and/or samples into groups, while ordination focuses on the arrangement or order of species and/or samples along gradients. To some degree, these approaches can be viewed as complementary. Both approaches require a species by sample matrix. We are going to focus on ordination approaches.

We will use two open source data sets to demonstrate how to run and interpret several common multivariate analyses. These data are ant and understory plant species collected in the Hemlock Removal Experiment at the NSF Harvard Forest Long-term Ecological Research (LTER) site (See more here: https://harvardforest1.fas.harvard.edu/exist/apps/datasets/showData.html?id=HF118). This experiment includes four treatments (Hemlock girdled, Hemlock logged, Hemlock control, and Hardwood control) each replicated across two (n = 2) 90 x 90 m plots. The goal of this experiment was to investigate the effects of hemlock mortality from the invasive hemlock woolly adelgid and preemptive hemlock removal via logging. The specific study that we are using focuses on ant community responses to these environmental changes, but there have been many studies conducted within this experiment.

Because the data are openly available on the NSF LTER website, we can directly load them into R.

```
ants <- read.csv(file="https://pasta.lternet.edu/package/data/eml/knb-lter-hfr/118/35/90ca76917fe458ee78
           header=T, na.strings=c("",".","NA"))
```

Notice that the ant data are in `long format`. Each row contains the count of one species at a specific site and date. Ant species are combined into one column. To run the multivariate analyses, we want the data represented in `wide format`. In this case, each species and their associated counts will be represented in a separate column. Each row will represent the counts of all ant species at a specific site and date. We can reshape the data set using the `reshape2` package.

```
ant.matrix <- dcast(ants, year + block + plot + treatment + trap.type ~ code,
                    sum, value.var = "abundance", na.rm = TRUE)

# change variables to factors
ant.matrix$year <- as.factor(ant.matrix$year)
ant.matrix$block <- as.factor(ant.matrix$block)
ant.matrix$plot <- as.factor(ant.matrix$plot)
ant.matrix$treatment <- as.factor(ant.matrix$treatment)
ant.matrix$trap.type <- as.factor(ant.matrix$trap.type)

levels(ant.matrix$year)
```

```
##  [1] "2003" "2004" "2005" "2006" "2007" "2008" "2009" "2010" "2011" "2012"
## [11] "2013" "2014" "2015" "2018"
```

The data set includes ant species collected over multiple years. Let's subset the data to focus on ants collected via pitfall traps in 2015 and 2018. Since it is likely that some of the ant species represented in the full data set (2003-2018) were not collected in these years (especially rarer species), let's also remove the columns of species that were not collected.

```
yr1 <- ant.matrix %>% filter(year == "2015") %>% droplevels()
yr2 <- ant.matrix %>% filter(year == "2018") %>% droplevels()

ants2 <- rbind(yr1, yr2)

ants3 <- ants2[, colSums(ants2 !=0) > 0]
```

Now let's load the understory plant data. This data set is also openly available on the NSF LTER website.

```
herb <- read.csv(file="https://pasta.lternet.edu/package/data/eml/knb-lter-hfr/106/33/417bcd2b9b90c6346
            header=T, na.strings=c("",".","NA"))
```

We need to reshape the data into `wide format`. Let's subset the shrub and herbaceous data so that we are working with the same years - 2015 and 2018. We will also remove plant species that are not very common in the data set.

```
herb$species <- as.factor(herb$species)
# species codes are provided in the csv file hf106-01-species-codes

herb.matrix <- dcast(herb, year + block + trt + plot ~ species,
                     mean, value.var = "cover", fill = 0)

yr1.herb <- herb.matrix %>% filter(year == "2015") %>% droplevels()
yr2.herb <- herb.matrix %>% filter(year == "2018") %>% droplevels()

herb2 <- rbind(yr1.herb, yr2.herb)

# remove columns of species that were not observed or observed at low frequencies
herb3 <- herb2[, colSums(herb2 !=0) > 2]
```

The ant and plant data sets are ready to go. We will use several multivariate methods to assess ant community response to the hemlock treatments. Percentage cover of understory plant species will be used as additional predictor variables to explain ant communities in these sites.

2 Ordination methods

There are two types of ordination techniques that differ based on the incorporation of environmental data: indirect and direct gradient analysis. We are able to run both types of these ordination analyses using the R package `vegan`.

2.1 Indirect gradient analysis

Indirect gradient analysis (aka unconstrained ordination) utilizes only the species by sample matrix. Any environmental data are used after the analysis to aid with the interpretation. When using an indirect gradient analysis, the approach identifies gradients solely based on the species data. Therefore, knowledge of the species and system are important for interpreting the results. Two examples of indirect gradient analysis are Nonmetric Multidimensional Scaling (NMDS) and Principal Component Analysis (PCA).

2.1.1 Nonmetric multidimensional scaling (NMDS)

Nonmetric multidimensional scaling is a commonly used technique to assess differences in species composition among samples, sites, treatments, etc. It is an iterative method that maximizes the rank order correlations between the distances in the dissimilarity matrix and the distances in low-dimensional space such that the output represents (as well as possible) the ordering among sites in species space.

NMDS is a distance-based technique that relies on a dissimilarity matrix created from incidence-based (i.e., presence/absence) or abundance-based species data. Dissimilarity matrices are essentially a measure of beta-diversity, which is the diversity `between` samples, sites, treatments, etc. The species diversity indices that we have already discussed are all measures of alpha-diversity, which is the diversity `within` a particular sample, site, treatment, etc. Gamma-diversity represents the total species diversity across all samples, sites, treatments, etc. Alpha-diversity and gamma-diversity are partitions of diversity at different scales, and beta-diversity describes the link or change between these two scales. With measures of beta-diversity, we can ask questions about how diversity changes among sites and evaluate whether sites are similar (or not) in terms of species diversity.

There are many different beta-diversity metrics, all of which have slightly different equations. The appropriate beta-diversity metric depends on the study question, data availability, and the ecology of the species. Because species richness tends to increase with sample size, beta-diversity is calculated between pairs of samples. A larger beta-diversity value means higher dissimilarity (i.e., greater difference) among those two communities in terms of their species composition. A lower beta-diversity value means lower dissimilarity among communities (i.e., the communities are more similar in terms of their species composition). A dissimilarity matrix can be calculated using the function vegdist(). The dissimilarity metric to be calculated is indicated by the `method` argument.

```r
# change ant data to presence/absence
ants4 <- ants3[,6:38] # save as a new object
ants4[ants4 > 0] <- 1 # if a value is greater than zero, replace it with 1
ants4$treatment <- ants3$treatment # add the treatments back into the data set

dis.matrix.pa <- vegdist(ants4[,1:33], method = "jaccard")
```

You can run a NMDS model in R using the metaMDS() function in the R package `vegan`. The algorithm is iterative. It starts from an initial distribution of samples in the ordination space (often this initial distribution is random), and then each iteration reshuffles the samples in search of the optimal positions of `n` samples in `k` dimensional space. Each iteration results in a slightly different solution, but the aim is to identify the best solution that minimizes the stress of the configuration. `Stress` is a measure of the mismatch between the rank order of the dissimilarities and the low dimensional solution. The default is k = 2 (i.e., 2 dimensional). Lower stress is better and ideally the stress should be below 0.2. Essentially, the stress is a measure of the fit of the model and should be reported in the results.

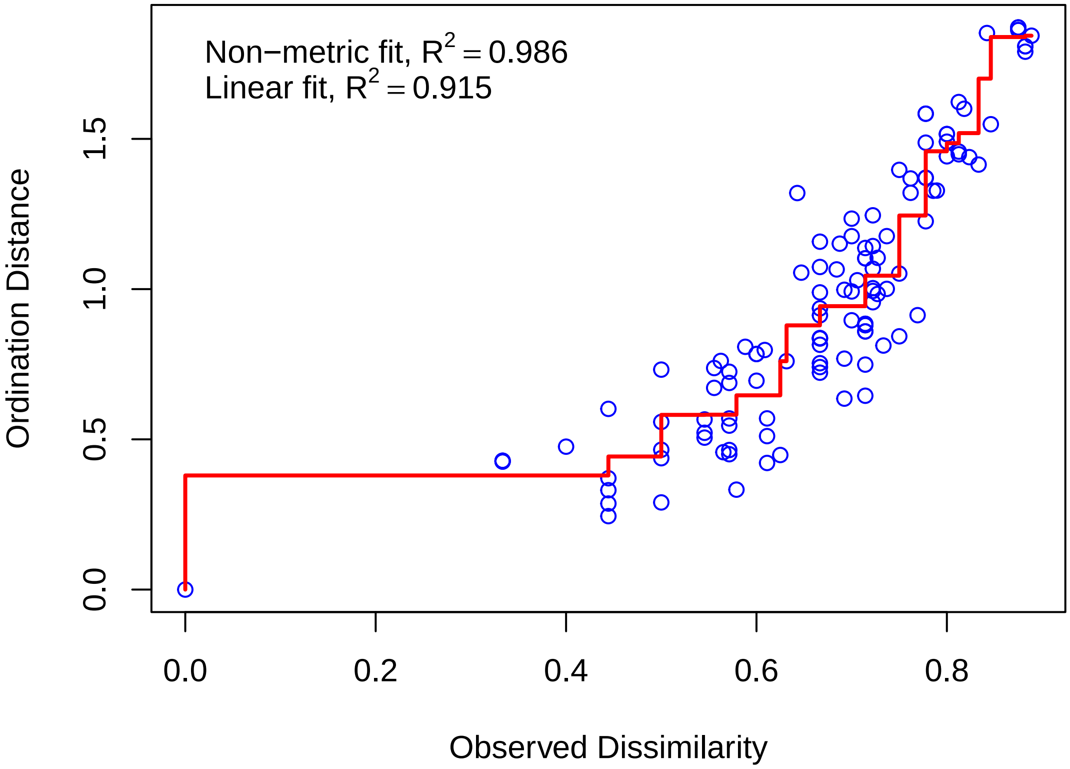
```r
nmds.ants.pa <- metaMDS(dis.matrix.pa, trymax = 500, autotransform = TRUE, k = 2)
```

```
## Run 0 stress 0.1183155
## Run 1 stress 0.1183155
## ... Procrustes: rmse 1.426668e-05  max resid 2.790632e-05
## ... Similar to previous best
## Run 2 stress 0.1305361
## Run 3 stress 0.1305361
## Run 4 stress 0.1183155
## ... Procrustes: rmse 0.0001061917  max resid 0.0001887542
## ... Similar to previous best
## Run 5 stress 0.1882071
## Run 6 stress 0.1183155
## ... New best solution
## ... Procrustes: rmse 9.820465e-06  max resid 1.71894e-05
## ... Similar to previous best
## Run 7 stress 0.1937155
## Run 8 stress 0.1183155
## ... Procrustes: rmse 2.290962e-05  max resid 4.31054e-05
## ... Similar to previous best
## Run 9 stress 0.1637823
## Run 10 stress 0.1651466
## Run 11 stress 0.1651466
## Run 12 stress 0.1580037
## Run 13 stress 0.1305361
## Run 14 stress 0.1493727
## Run 15 stress 0.1545166
## Run 16 stress 0.1703661
## Run 17 stress 0.1183155
## ... Procrustes: rmse 1.849576e-05  max resid 3.391041e-05
## ... Similar to previous best
## Run 18 stress 0.1840186
## Run 19 stress 0.1637823
## Run 20 stress 0.1637823
## *** Best solution repeated 3 times
```

```r
nmds.ants.pa
```

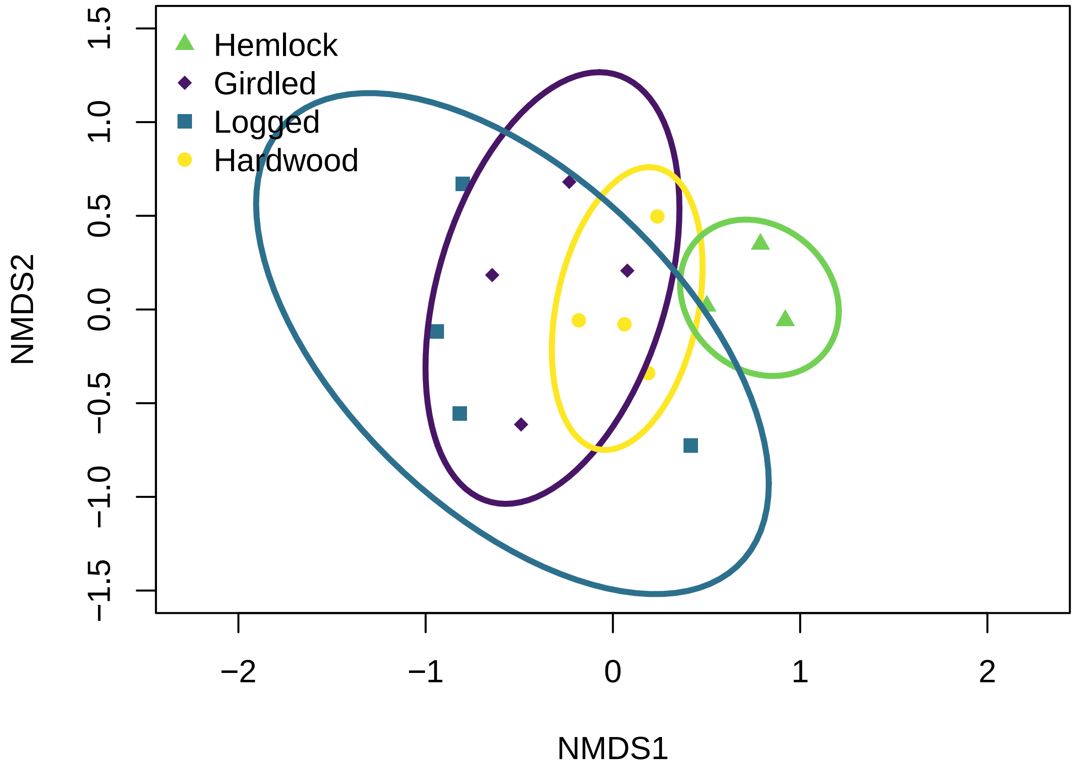
```
##
## Call:
## metaMDS(comm = dis.matrix.pa, k = 2, trymax = 500, autotransform = TRUE)
##
## global Multidimensional Scaling using monoMDS
##
## Data:     dis.matrix.pa
## Distance: jaccard
##
## Dimensions: 2
## Stress:     0.1183155
## Stress type 1, weak ties
## Best solution was repeated 3 times in 20 tries
## The best solution was from try 6 (random start)
## Scaling: centring, PC rotation, halfchange scaling
## Species: scores missing
```

```
stressplot(nmds.ants.pa)
```



```
ordiplot(nmds.ants.pa, disp = "sites", type = "n", xlim = c(-1.5, 1.5), ylim = c(-1.5, 1.5))
points(nmds.ants.pa, dis = "sites", select = which(ants3$treatment=="HemlockControl"), pch = 17, cex = 
points(nmds.ants.pa, dis = "sites", select = which(ants3$treatment=="Girdled"), pch = 18, cex = 1, col =
points(nmds.ants.pa, dis = "sites", select = which(ants3$treatment=="Logged"), pch = 15, cex = 1, col =
points(nmds.ants.pa, dis = "sites", select = which(ants3$treatment=="HardwoodControl"), pch = 16, cex =
ordiellipse(nmds.ants.pa, ants3$treatment, draw = "lines", col = c("#481567FF", "#FDE725FF", "#73D055FF
            lwd = 3, kind = "sd", conf = 0.90, label = FALSE)

legend("topleft", legend = c("Hemlock", "Girdled", "Logged", "Hardwood"),
       pch = c(17, 18, 15, 16), cex = 1, bty = "n", col = c("#73D055FF", "#481567FF", "#2D708EFF", "#FDI
```



The NMDS plot shows the distribution of sites in low-dimensional space, with the sites coded based on hemlock treatment. This plot is a visual representation of the NMDS model. The stress was 0.11, which, along with a high R2 value from the Shepard plot, indicate that the model is a good fit for the data. The R2 in the Shepard plot is determined by comparing the relationship between the final distances of the objects in the ordination and the original distances of the community data. Other than these measures of fit, there are no statistical results.

NMDS is a distance-based technique (as opposed to an eigenvector-based technique which is discussed below), so it does not maximize the variability associated with the individual ordination axes. Therefore, there is no measure of the variability explained by the axes. In fact, the direction and numeric scale of the axes are not important for the interpretation.

Because the NMDS model is a visual representation of the differences in species composition among sites, this method is often paired with permutational multivariate analysis of variance (PERMANOVA) and homogeneity of multivariate group dispersion (BETADISPER).

2.1.2 Permutational multivariate analysis of variance (PERMANOVA)

PERMANOVA tests whether the group centroid of the communities differs among a categorical grouping factor (e.g., treatments) in multivariate space. The centroids are calculated for each group and the sums of squares of deviations are determined for these points. Significance tests are conducted using permutations of the community data and compared to the sums of squares that were calculated for the observed data.

```
adonis2(dis.matrix.pa ~ ants4$treatment, permutations = 999)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = dis.matrix.pa ~ ants4$treatment, permutations = 999)
##                 Df SumOfSqs      R2      F Pr(>F)
## ants4$treatment  3   1.4433 0.41295 2.8137  0.001 ***
## Residual        12   2.0518 0.58705
## Total           15   3.4950 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
pairwise.adonis(dis.matrix.pa, ants4$treatment)
```

```
##                               pairs Df SumsOfSqs   F.Model        R2 p.value
## 1                 Logged vs Girdled  1 0.2406823 0.9936814 0.1420827   0.466
## 2          Logged vs HemlockControl  1 0.8663977 5.4361524 0.4753480   0.036
## 3         Logged vs HardwoodControl  1 0.3708324 1.8426037 0.2349480   0.048
## 4         Girdled vs HemlockControl  1 0.6411288 4.5565102 0.4316304   0.032
## 5        Girdled vs HardwoodControl  1 0.2146477 1.1756126 0.1638345   0.404
## 6 HemlockControl vs HardwoodControl  1 0.5528551 5.5425213 0.4801829   0.023
##   p.adjusted sig
## 1      1.000
## 2      0.216
## 3      0.288
## 4      0.192
## 5      1.000
## 6      0.138
```

2.1.3 Homogeneity of multivariate group dispersion (BETADISPER)

BETADISPER tests whether the dispersion of a categorical grouping factor from its spatial median is different between groups. This test is essentially a multivariate analogue of Levene's test for homogeneity of variances. Significance is determined using a permutation procedure.

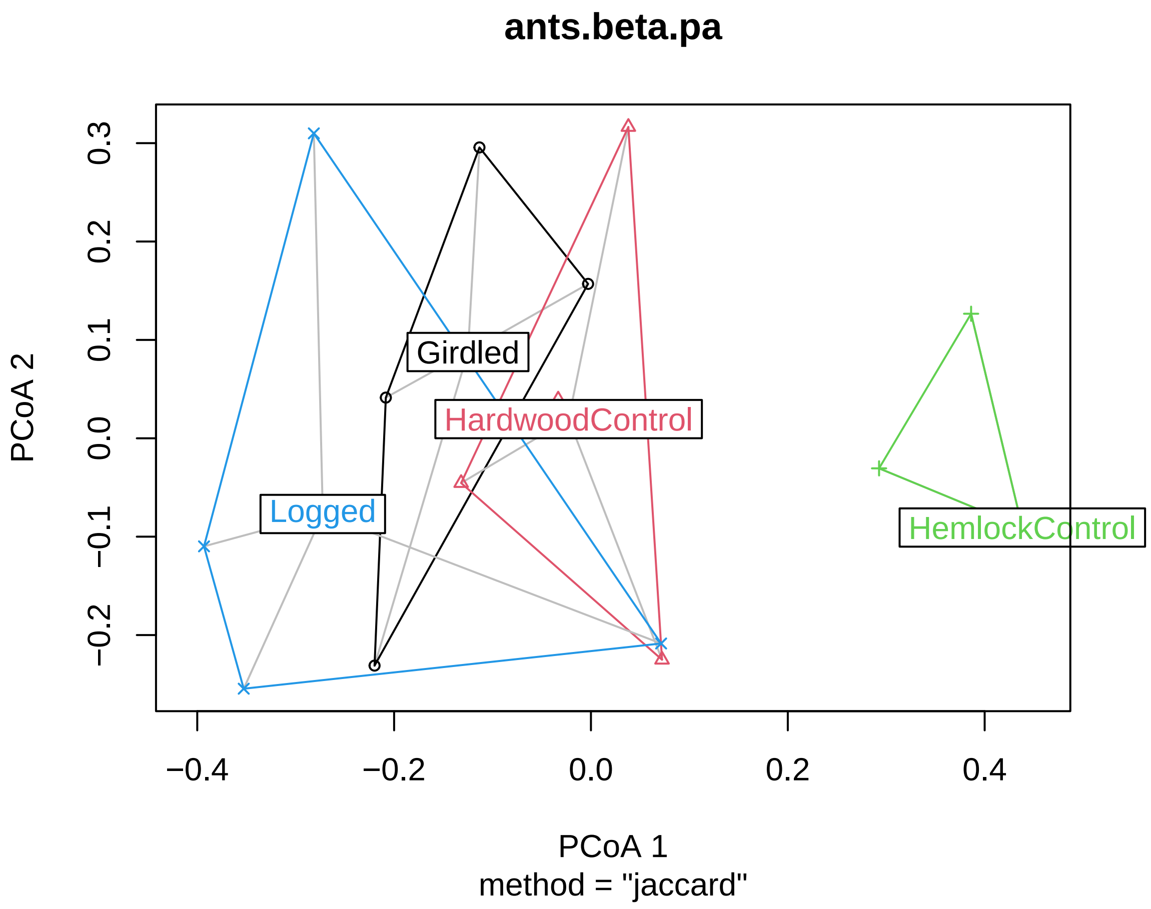
```
ants.beta.pa <- betadisper(dis.matrix.pa, ants4$treatment, type = c("median"))
```

```
## Warning in betadisper(dis.matrix.pa, ants4$treatment, type = c("median")): some
## squared distances are negative and changed to zero
```

```
anova(ants.beta.pa)
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df  Sum Sq  Mean Sq F value  Pr(>F)
## Groups     3 0.17539 0.058465  3.7923 0.04009 *
## Residuals 12 0.18500 0.015417
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
plot(ants.beta.pa)
```



2.1.4 Principal component analysis (PCA)

Principal component analysis is an eigenvalue-based ordination method that uses Euclidean distances among samples to maximize the linear correlation between the distances in the dissimilarity matrix and the distances in low dimensional space. This method is not a `statistical test`, but rather describes the correlation structure among different variables.

You can run a PCA using the function rda() in the package `vegan`. Each ordination axis is an eigenvector with an associated eigenvalue, which represents the variance explained in the model. Axes are ranked by their eigenvalues such that the order of axes is important. Therefore, the first axis has the highest eigenvalue and explains the most variation in the data. Based on the output of the model, the eigenvalues are examined for their importance (i.e., variance explained), and then the patterns of data represented by the axes are described. Report the percentage variance explained by the first two axes.

The linear nature of PCA often results in the arch effect (aka the horseshoe effect), which occurs when the data are distorted in the ordination plot producing a semicircular pattern. This makes visual interpretation of the patterns difficult. Species data are especially susceptible to the arch effect. Because of this, PCA is mostly used to characterize environmental variables among sites. For example, sample or site scores on PCA axes can be used as predictor variables in other analyses (e.g., GLMs), or environmental variables highly correlated to PCA axes can be selected for further analyses.

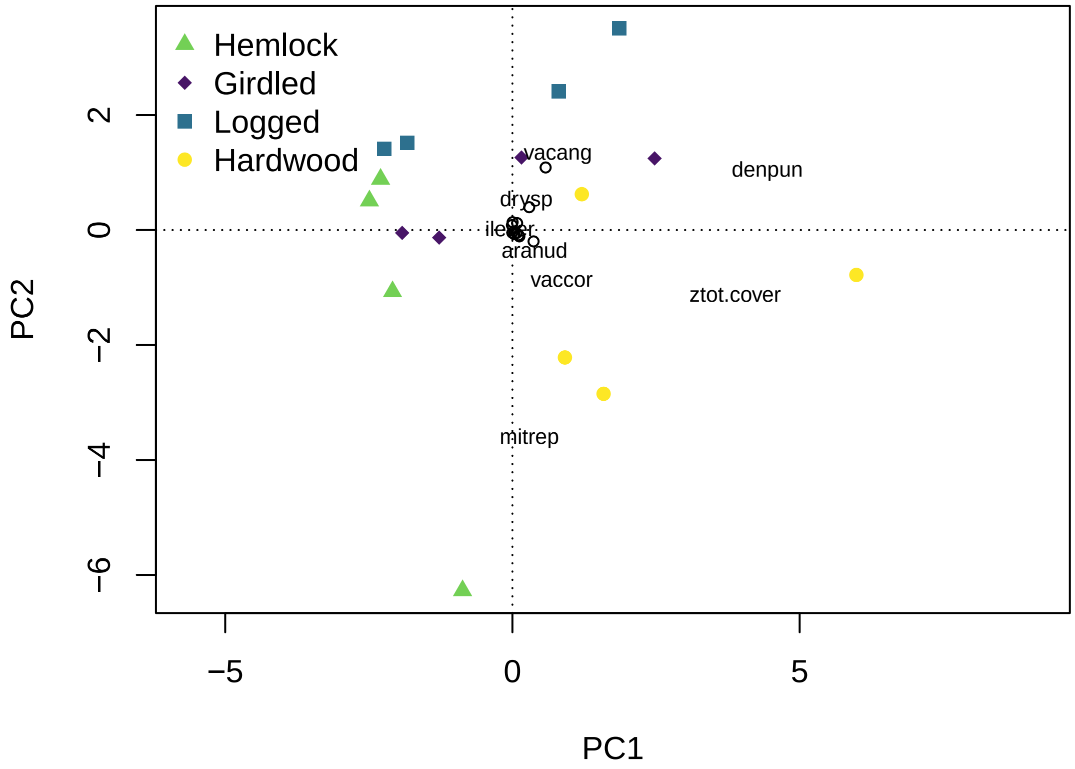
We will run a PCA using the herbaceous understory plant data. If the data set contains environmental variables that have different units, these should be standardized to have a mean of 0 and unit standard deviation, Otherwise the variable with the higher absolute value or variance will be more important in the analysis. Since all the plant data are percentage cover, we do not need to standardize (i.e., scale = FALSE).

```
herb.pca <- rda(herb3[,5:23], scale = FALSE)
# if RDA function is used without explanatory variables, then it calculates a PCA
summary(herb.pca)
```

```
##
## Call:
## rda(X = herb3[, 5:23], scale = FALSE)
##
## Partitioning of variance:
##                Inertia Proportion
## Total              439          1
## Unconstrained      439          1
##
## Eigenvalues, and their contribution to the variance
##
## Importance of components:
##                            PC1      PC2     PC3      PC4      PC5      PC6
## Eigenvalue            200.1335 106.6731 47.1189 40.23343 37.52839 3.287174
## Proportion Explained    0.4559   0.2430  0.1073  0.09164  0.08548 0.007487
## Cumulative Proportion   0.4559   0.6988  0.8062  0.89779  0.98327 0.990759
##                           PC7      PC8      PC9      PC10      PC11      PC12
## Eigenvalue            1.57620 0.875530 0.599003 0.4158937 0.3433982 0.1054584
## Proportion Explained  0.00359 0.001994 0.001364 0.0009473 0.0007822 0.0002402
## Cumulative Proportion 0.99435 0.996343 0.997708 0.9986551 0.9994373 0.9996775
##                            PC13      PC14      PC15
## Eigenvalue            0.0853128 0.0551803 1.093e-03
## Proportion Explained  0.0001943 0.0001257 2.490e-06
## Cumulative Proportion 0.9998718 0.9999975 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:  9.008381
##
##
## Species scores
##
##                   PC1      PC2        PC3       PC4       PC5       PC6
## aranud      0.385891 -0.35048 -3.912e-01  0.425973 -0.229857  0.061236
## carpen      0.081918  0.12017 -3.224e-03  0.068444  0.033449  0.035326
## denobs      0.120566 -0.11030 -2.325e-02  0.055011 -0.083041 -0.038215
## denpun      4.437194  1.02280  8.393e-01 -0.905048 -0.857601  0.254068
## drysp       0.242579  0.50762 -1.051e+00 -1.554751  1.171519  0.036443
## hupluc     -0.002268 -0.04691  6.937e-05  0.007906 -0.002459  0.028693
## ilever     -0.041904  0.02078 -3.777e-02 -0.049513 -0.001359 -0.064839
## lysbor      0.030863 -0.07130 -1.495e-01 -0.122120  0.029951  0.088368
## lysqua     -0.008423  0.08546  3.715e-02  0.046400  0.056738  0.001758
## maican      0.367786 -0.19762 -2.201e-01  0.166357 -0.204208 -0.078599
## medvir      0.107156 -0.09029 -1.845e-01  0.074573 -0.017451 -0.111079
## mitrep      0.295461 -3.62513  1.187e+00 -0.546034  0.775407  0.141089
## ruball      0.292084  0.39693 -6.156e-01 -0.655354  0.746259 -0.048040
## rubhis      0.578005  1.08815  8.758e-02  0.168342  1.335217  0.147782
## rubida      0.003059  0.13624 -2.190e-01 -0.228720  0.179590 -0.077066
## uvuses      0.071196 -0.05402 -9.302e-02  0.167878  0.005862  0.002524
## vacang      0.792238  1.28380  1.279e+00  1.290031  1.268744  0.042059
## vaccor      0.858964 -0.88542 -1.588e+00  1.022042  0.033407  0.508678
## ztot.cover  3.879222 -1.12488 -7.926e-01  0.642815  0.371196 -0.439762
##
##
## Site scores (weighted sums of species scores)
##
##           PC1      PC2     PC3       PC4     PC5     PC6
## sit1   2.4751  1.24614 -4.2569 -4.509491  4.3655 -0.1168
## sit2   0.9141 -2.21739 -2.4164  2.775156 -0.4720 -3.5330
## sit3  -2.2954  0.88540 -0.4336 -0.748498 -0.8061  0.6717
## sit4  -1.8336  1.51397 -1.1016 -0.629729  0.5132 -1.7435
## sit5  -1.2755 -0.13171 -0.7635  0.816624 -0.4643 -5.2137
## sit6   1.2081  0.62333  1.0888 -0.286262 -2.8293 -0.8295
## sit7  -2.4897  0.50892  0.6366  0.005362 -1.4813  1.6175
## sit8   1.8613  3.50753  3.5659  4.439392  4.8250 -0.4752
## sit9   0.1575  1.26161 -0.5960 -2.513436 -0.5754  1.0706
## sit10  1.5872 -2.84940 -4.2559  4.347351 -0.2239  4.2243
## sit11 -0.8680 -6.27253  3.6119 -2.029502  3.4602  0.5903
## sit12 -2.2286  1.41310 -0.3247  0.186097 -0.8728  1.6113
## sit13 -1.9194 -0.04983  0.1162  0.068827 -0.6847 -1.1818
## sit14  5.9878 -0.78152  2.2853 -1.491791 -3.6083 -0.7443
## sit15 -2.0870 -1.07111  1.4189 -0.155539 -0.5508  0.9406
## sit16  0.8062  2.41347  1.4250 -0.274559 -0.5949  3.1114
```

The summary output of the PCA is lengthy. It provides eigenvalues and the proportion of variance explained for each axis as well as species and site scores for the first six PCA axes. Although many PCA axes are generated, the focus is usually on the first two or three axes that explain the most variation in the data. In this case, axis 1 and 2 explain 69% of the variation in the understory plant data, with axis 1 explaining the majority (45%).

```r
ordiplot (herb.pca, display = 'sites', type = 'n')
points(herb.pca, dis = "sites", select = which(herb3$trt=="hemlock"), pch = 17, cex = 1, col = "#73D055
points(herb.pca, dis = "sites", select = which(herb3$trt=="girdled"), pch = 18, cex = 1, col = "#481567
points(herb.pca, dis = "sites", select = which(herb3$trt=="logged"), pch = 15, cex = 1, col = "#2D708EF
points(herb.pca, dis = "sites", select = which(herb3$trt=="hardwood"), pch = 16, cex = 1, col = "#FDE72
orditorp(herb.pca, display = 'sp')
legend("topleft", legend = c("Hemlock", "Girdled", "Logged", "Hardwood"),
       pch = c(17, 18, 15, 16), cex = 1, bty = "n", col = c("#73D055FF", "#481567FF", "#2D708EFF", "#FD
```



The majority of the variance is explained by PCA axis 1. Based on how the sites fall along axis 1, we can see that it ranges from hemlock dominated forests to hardwood dominated forests. This suggests that there is a change in the understory plant community along this forest gradient.

2.2 Direct gradient analysis

Direct gradient analysis (aka constrained ordination) utilizes environmental data in addition to a species by site matrix. When using a direct gradient analysis, the approach assesses whether species composition is related to the measured environmental factors. Therefore, these methods can be used to test hypotheses, which are carried out using permutation procedures. Two commonly used constrained ordination methods are Redundancy Analysis (RDA) and Canonical Correspondence Analysis (CCA).

2.2.1 Redundancy analysis (RDA)

Redundancy analysis combines multiple regression and principal component analysis (PCA) to model multivariate response variables as a function of a set of environmental predictors. RDA is most useful when environmental gradients are short because it assumes linear relationships among species and measured environmental gradients.

This method applies multiple regression of species abundance data as a function of the environmental variables to produce two new matrices: one with predicted values and one with residuals. Constrained and unconstrained ordination axes are extracted from these new matrices using PCA. These axes are linear and orthogonal combinations of the environmental predictor variables that best explain the variation in the species matrix.

We can run an RDA using the rda() function in the package `vegan`. To differentiate this analysis from PCA, we will specify our environmental data set of understory plant species.

```r
ants.rda <- rda(ants3[,6:38] ~ ., herb3[,5:23])
summary(ants.rda)
```

```
##
## Call:
## rda(formula = ants3[, 6:38] ~ aranud + carpen + denobs + denpun +      drysp + hupluc + ilever + lys
##
## Partitioning of variance:
##               Inertia Proportion
## Total            1862          1
## Constrained      1862          1
## Unconstrained       0          0
##
## Eigenvalues, and their contribution to the variance
##
## Importance of components:
##                           RDA1     RDA2     RDA3     RDA4     RDA5     RDA6
## Eigenvalue            861.2686 635.5632 211.3264 64.71583 53.89369 19.03651
## Proportion Explained    0.4625   0.3413   0.1135  0.03475  0.02894  0.01022
## Cumulative Proportion   0.4625   0.8037   0.9172  0.95194  0.98088  0.99110
##                           RDA7    RDA8     RDA9     RDA10     RDA11     RDA12
## Eigenvalue            8.253510 3.09128 1.931927 1.3615812 0.8908122 0.5335600
## Proportion Explained  0.004432 0.00166 0.001037 0.0007311 0.0004783 0.0002865
## Cumulative Proportion 0.995536 0.99720 0.998234 0.9989647 0.9994430 0.9997295
##                           RDA13     RDA14     RDA15
## Eigenvalue            0.3455534 1.163e-01 4.201e-02
## Proportion Explained  0.0001855 6.242e-05 2.256e-05
## Cumulative Proportion 0.9999150 1.000e+00 1.000e+00
##
## Accumulated constrained eigenvalues
## Importance of components:
##                           RDA1     RDA2     RDA3     RDA4     RDA5     RDA6
## Eigenvalue            861.2686 635.5632 211.3264 64.71583 53.89369 19.03651
## Proportion Explained    0.4625   0.3413   0.1135  0.03475  0.02894  0.01022
## Cumulative Proportion   0.4625   0.8037   0.9172  0.95194  0.98088  0.99110
##                           RDA7    RDA8     RDA9     RDA10     RDA11     RDA12
## Eigenvalue            8.253510 3.09128 1.931927 1.3615812 0.8908122 0.5335600
## Proportion Explained  0.004432 0.00166 0.001037 0.0007311 0.0004783 0.0002865
## Cumulative Proportion 0.995536 0.99720 0.998234 0.9989647 0.9994430 0.9997295
##                           RDA13     RDA14     RDA15
## Eigenvalue            0.3455534 1.163e-01 4.201e-02
## Proportion Explained  0.0001855 6.242e-05 2.256e-05
## Cumulative Proportion 0.9999150 1.000e+00 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:  12.92824
##
##
## Species scores
##
##              RDA1       RDA2       RDA3      RDA4      RDA5       RDA6
## aphful  -4.277915 -6.3410773  0.2980717 -0.276053 -0.386502  0.0397680
## aphpic  -6.581412  3.2459839  1.9746181  0.071757  0.341102 -0.0147992
## camchr  -0.007751 -0.0202265  0.0130177 -0.025831 -0.014105  0.0052507
## camher   0.037347  0.0003816 -0.0002881 -0.041477 -0.008984  0.0095186
## camnea  -0.041647  0.0377180 -0.0706437  0.054579 -0.056330 -0.0875132
## camnov  -0.077647 -0.1003481  0.1251478 -0.128936 -0.114634 -0.0156374
## campen  -0.645562  0.5765607 -0.2502863  1.695409 -1.169806  0.5804393
## crelin  -0.036270  0.0099423  0.0522617  0.010485 -0.005940  0.0004809
## forase  -3.820856  1.2654243 -3.7675771 -0.120951  0.155642 -0.0399649
## forinc  -0.059370 -0.2895788  0.0153416  0.212975  0.179348 -0.0017404
## forint  -0.029685 -0.1447894  0.0076708  0.106488  0.089674 -0.0008702
## forlas   0.009396 -0.0219057 -0.0048199 -0.007375 -0.004092 -0.0021310
## forneo1 -0.076573 -0.3430937 -0.0821697  0.404991 -0.088190 -0.3150412
## forpal  -0.009832 -0.0461239 -0.0135982  0.061546 -0.010584 -0.0457560
## forrub  -0.072539  0.0198847  0.1045233  0.020970 -0.011880  0.0009618
## forsub1 -0.094995  0.0395425  0.1249701  0.073041 -0.099470 -0.0935345
## forsub2 -0.028183 -0.0216831 -0.0423186  0.004472 -0.063205 -0.0317387
## forsub3 -0.452854  0.2957110  0.3607787  0.400228 -0.336598 -0.5044105
## lasame  -0.049870 -0.0928165  0.0062681  0.149332 -0.122862 -0.1807576
## lasaph   0.177501 -1.8917878 -0.1194825  1.452212  1.573835 -0.0894451
## lasbre  -0.016018  0.0456603  0.0960027 -0.031459  0.059698 -0.0081467
## lassub1 -0.398965  0.1093658  0.5748782  0.115335 -0.065340  0.0052901
## lasumb  -0.021216  0.3495674 -0.0694802  0.416545 -0.425917 -0.8356368
## myrame1 -0.095688 -0.0251078  0.0295028  0.022629  0.034467  0.0088602
## myrdet  -0.114123  0.0597420  0.0177245  0.101599 -0.108685 -0.1320173
## myrpun  -0.052549 -0.3436170 -0.2479617  0.218855 -0.467538 -0.4650847
## myrscu  -0.009705 -0.0418454 -0.0459084  0.113646 -0.091534 -0.1366878
## ponpen  -0.066387 -0.0959078 -0.0497700 -0.007661 -0.015568  0.0271643
## steimp  -0.030460 -0.1866179 -0.0151900  0.046557  0.065792  0.0157670
## stesch   0.023560 -0.0017897 -0.0110223  0.017094  0.028624 -0.0029802
## tapses  -0.049475 -0.2413156  0.0127847  0.177479  0.149457 -0.0014503
## temamb  -0.036270  0.0099423  0.0522617  0.010485 -0.005940  0.0004809
## temlon  -0.440476  0.2870758 -0.3189026  0.011216  0.192912  0.0449758
```

```
##
##
## Site scores (weighted sums of species scores)
##
##           RDA1     RDA2     RDA3    RDA4     RDA5     RDA6
## row1  -6.06207  1.66176  8.73498  1.7525 -0.99281  0.08038
## row2  -6.98197  5.58906 -6.40702  0.4143  4.11500  0.57139
## row3   1.14928  2.42045  1.79484 -2.1827  2.44029 -0.05157
## row4   1.25876  3.20412  1.70522  1.4728 -0.30847 -4.40772
## row5   2.08074  0.02126 -0.01605 -2.3108 -0.50052  0.53031
## row6   0.01056  0.35755 -2.70016  4.3540 -6.76496 -7.59914
## row7   2.64374  2.72911 -0.24894  5.8938 -4.50849  9.06534
## row8   2.28771  0.90873  0.08281 -1.7574 -0.11702 -0.67615
## row9   0.84620  1.49247  1.82771 -1.7526  2.74266 -0.36050
## row10 -4.72100 -3.98165 -4.37295 -3.6066 -3.79903  2.29435
## row11  0.94028  0.63050  1.37366 -2.8667  0.99192  0.31224
## row12  3.93776 -0.29914 -1.84226  2.8570  4.78424 -0.49810
## row13 -1.29543 -3.38065  2.17576 -4.3174 -2.35746  0.87761
## row14 -1.65384 -8.06667  0.42736  5.9328  4.99603 -0.04848
## row15  3.98892  0.37440 -1.72938 -2.6503 -0.03742  0.26623
## row16  1.57037 -3.66131 -0.80560 -1.2326 -0.68397 -0.35618
##
##
## Site constraints (linear combinations of constraining variables)
##
##           RDA1     RDA2     RDA3    RDA4     RDA5     RDA6
## row1  -6.06207  1.66176  8.73498  1.7525 -0.99281  0.08038
## row2  -6.98197  5.58906 -6.40702  0.4143  4.11500  0.57139
## row3   1.14928  2.42045  1.79484 -2.1827  2.44029 -0.05157
## row4   1.25876  3.20412  1.70522  1.4728 -0.30847 -4.40772
## row5   2.08074  0.02126 -0.01605 -2.3108 -0.50052  0.53031
## row6   0.01056  0.35755 -2.70016  4.3540 -6.76496 -7.59914
## row7   2.64374  2.72911 -0.24894  5.8938 -4.50849  9.06534
## row8   2.28771  0.90873  0.08281 -1.7574 -0.11702 -0.67615
## row9   0.84620  1.49247  1.82771 -1.7526  2.74266 -0.36050
## row10 -4.72100 -3.98165 -4.37295 -3.6066 -3.79903  2.29435
## row11  0.94028  0.63050  1.37366 -2.8667  0.99192  0.31224
## row12  3.93776 -0.29914 -1.84226  2.8570  4.78424 -0.49810
## row13 -1.29543 -3.38065  2.17576 -4.3174 -2.35746  0.87761
## row14 -1.65384 -8.06667  0.42736  5.9328  4.99603 -0.04848
## row15  3.98892  0.37440 -1.72938 -2.6503 -0.03742  0.26623
## row16  1.57037 -3.66131 -0.80560 -1.2326 -0.68397 -0.35618
##
##
## Biplot scores for constraining variables
##
##            RDA1     RDA2     RDA3      RDA4      RDA5      RDA6
## aranud -0.69596 -0.12164 -0.63864  0.011463 -0.005952  0.013981
## carpen  0.04321 -0.12508 -0.09656  0.267349  0.225209 -0.384110
## denobs -0.60513 -0.10496 -0.53605  0.250127  0.189311 -0.158777
## denpun -0.28803 -0.55994  0.12827  0.462170  0.153219 -0.252232
## drysp  -0.32372  0.25539  0.81733 -0.007973  0.089089 -0.076471
## hupluc -0.11704 -0.02560 -0.17437 -0.192120 -0.318491 -0.225261
## ilever -0.11491 -0.15805  0.37652 -0.401883 -0.101650  0.065501
## lysbor -0.34228  0.02477  0.30344 -0.178282  0.215113 -0.135542
## lysqua  0.35012  0.11851  0.02579  0.029316  0.090282 -0.306247
## maican -0.75041 -0.25654 -0.47016  0.225813 -0.014849 -0.124597
## medvir -0.79859  0.32872 -0.37144  0.036880  0.235812  0.008485
## mitrep -0.13683 -0.10192 -0.06815 -0.209189  0.096473  0.122194
## ruball -0.37197  0.22758  0.73570  0.153739 -0.021679 -0.125894
## rubhis -0.03468  0.06267  0.41112 -0.079030 -0.033016 -0.118513
## rubida -0.20292  0.31818  0.57477  0.154947 -0.024236 -0.292721
```

Again, the output is very lengthy. It contains information about the partitioned variance among the constrained and unconstrained axes, as well as the total inertia (i.e., total variance explained by the model, both constrained and unconstrained axes combined). Similar to PCA, it shows the eigenvalues and proportion of variance explained by each RDA axis. Again, although there are many RDA axes, we are only concerned with the first two or three axes that explain the most variation in the ant species data. We can see that the first two axes explain 80% of the variance (46.2% for axis 1 and 34.1% for axis 2).

The importance of the environmental variables can also be assessed. The ordistep() function can be used to build constrained ordination models from the RDA using a stepwise model selection procedure. Significance is determined using permutation tests. It identifies the best or minimum adequate model. This method is one way to test hypotheses with RDA.

First, a null model needs to be created without any environmental predictor variables (mod0). Then, we specify that the scope should be the RDA model with all the environmental predictors. The permutation procedure creates a reduced model with only those environmental variables determined to be important for the community matrix.

```
mod0.rda <- rda(ants3[,6:38] ~ 1, herb3[,5:23])
ants.rda.red <- ordistep(mod0.rda, scope = formula(ants.rda), direction = "both",
                         permutations = how(nperm = 199))
```

```
##
## Start: ants3[, 6:38] ~ 1
##
##              Df    AIC      F Pr(>F)
## + medvir      1 116.57 7.5117  0.005 **
## + maican      1 117.50 6.2888  0.005 **
## + ztot.cover  1 117.82 5.8941  0.005 **
## + aranud      1 118.29 5.3211  0.010 **
## + vaccor      1 118.24 5.3759  0.015 *
## + denobs      1 119.68 3.7087  0.025 *
## + uvuses      1 119.66 3.7284  0.030 *
## + drysp       1 120.89 2.4146  0.065 .
## + denpun      1 120.72 2.5932  0.095 .
## + ruball      1 120.94 2.3650  0.125
## + rubida      1 121.86 1.4495  0.250
## + lysqua      1 122.39 0.9475  0.455
## + lysbor      1 122.32 1.0180  0.475
## + ilever      1 122.83 0.5433  0.605
## + vacang      1 123.06 0.3347  0.675
## + rubhis      1 123.08 0.3190  0.875
## + mitrep      1 123.20 0.2124  0.890
## + carpen      1 123.22 0.1900  0.960
## + hupluc      1 123.20 0.2144  0.985
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Step: ants3[, 6:38] ~ medvir
##
##          Df    AIC      F Pr(>F)
## - medvir  1 121.44 7.5117  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##              Df    AIC      F Pr(>F)
## + ztot.cover  1 112.23 6.3213  0.005 **
## + maican      1 111.90 6.7256  0.010 **
## + denpun      1 114.27 4.0035  0.020 *
## + aranud      1 114.44 3.8228  0.035 *
## + ruball      1 115.14 3.1087  0.075 .
## + denobs      1 115.85 2.4096  0.080 .
## + drysp       1 114.91 3.3410  0.090 .
## + rubida      1 115.98 2.2790  0.105
## + vaccor      1 115.25 2.9915  0.150
## + uvuses      1 117.30 1.0708  0.385
## + ilever      1 117.49 0.9076  0.410
## + lysbor      1 117.64 0.7782  0.475
## + rubhis      1 117.81 0.6297  0.625
## + lysqua      1 117.87 0.5777  0.695
## + mitrep      1 118.29 0.2312  0.870
## + carpen      1 118.27 0.2453  0.900
## + hupluc      1 118.24 0.2668  0.905
## + vacang      1 118.49 0.0641  0.985
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: ants3[, 6:38] ~ medvir + ztot.cover
##
##              Df    AIC      F Pr(>F)
## - ztot.cover  1 116.57 6.3213  0.005 **
## - medvir      1 117.82 7.8924  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##          Df    AIC      F Pr(>F)
## + drysp   1 108.42 5.2502  0.015 *
## + ruball  1 108.57 5.0895  0.020 *
## + rubida  1 110.22 3.4158  0.030 *
## + maican  1 110.90 2.7712  0.060 .
## + aranud  1 111.04 2.6482  0.070 .
## + ilever  1 111.79 1.9721  0.115
## + rubhis  1 112.08 1.7261  0.115
## + denobs  1 112.21 1.6098  0.160
## + uvuses  1 112.38 1.4677  0.175
## + vaccor  1 112.67 1.2288  0.255
## + vacang  1 112.57 1.3069  0.275
## + lysbor  1 112.93 1.0151  0.400
## + lysqua  1 113.06 0.9073  0.460
## + denpun  1 113.36 0.6707  0.650
## + carpen  1 113.61 0.4728  0.795
## + hupluc  1 113.81 0.3146  0.890
## + mitrep  1 114.04 0.1391  0.985
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: ants3[, 6:38] ~ medvir + ztot.cover + drysp
##
##              Df    AIC       F Pr(>F)
## - drysp       1 112.23  5.2502  0.010 **
## - ztot.cover  1 114.91  8.3964  0.005 **
## - medvir      1 116.26 10.1938  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##          Df    AIC      F Pr(>F)
## + vacang  1 107.93 1.8539  0.105
## + ilever  1 107.47 2.2248  0.110
## + maican  1 107.69 2.0461  0.125
## + aranud  1 108.62 1.3117  0.245
## + vaccor  1 108.26 1.5918  0.250
## + lysqua  1 108.71 1.2445  0.290
## + rubhis  1 108.90 1.1001  0.315
## + denpun  1 108.89 1.1009  0.360
## + ruball  1 109.08 0.9644  0.410
## + rubida  1 109.23 0.8485  0.425
## + carpen  1 109.54 0.6213  0.620
## + uvuses  1 109.78 0.4487  0.755
## + lysbor  1 109.75 0.4689  0.805
## + hupluc  1 109.76 0.4629  0.810
## + denobs  1 109.95 0.3307  0.855
## + mitrep  1 110.02 0.2822  0.890
```

```
summary(ants.rda.red) # which environmental variables are important
```

```
##
## Call:
## rda(formula = ants3[, 6:38] ~ medvir + ztot.cover + drysp, data = herb3[,      5:23])
##
## Partitioning of variance:
##               Inertia Proportion
## Total          1862.4     1.0000
## Constrained    1295.1     0.6954
## Unconstrained   567.3     0.3046
##
## Eigenvalues, and their contribution to the variance
##
## Importance of components:
##                           RDA1     RDA2      RDA3      PC1      PC2      PC3
## Eigenvalue            728.4875 383.8433 182.73903 281.3954 141.1647 61.33528
## Proportion Explained    0.3912   0.2061   0.09812   0.1511   0.0758  0.03293
## Cumulative Proportion   0.3912   0.5973   0.69539   0.8465   0.9223  0.95522
##                            PC4      PC5       PC6      PC7      PC8       PC9
## Eigenvalue            38.44300 19.77811 16.003765 4.645384 2.037175 1.2009730
## Proportion Explained   0.02064  0.01062  0.008593 0.002494 0.001094 0.0006449
## Cumulative Proportion  0.97586  0.98648  0.995070 0.997565 0.998659 0.9993035
##                           PC10      PC11      PC12
## Eigenvalue            0.76365 0.3767743 1.567e-01
## Proportion Explained  0.00041 0.0002023 8.415e-05
## Cumulative Proportion 0.99971 0.9999159 1.000e+00
##
## Accumulated constrained eigenvalues
## Importance of components:
##                           RDA1     RDA2     RDA3
## Eigenvalue            728.4875 383.8433 182.7390
## Proportion Explained    0.5625   0.2964   0.1411
## Cumulative Proportion   0.5625   0.8589   1.0000
##
## 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:  12.92824
##
##
## Species scores
##
##               RDA1      RDA2      RDA3        PC1       PC2       PC3
## aphful  -2.3081752 -5.365229  0.408122 -4.9439034 -0.267530  0.010784
## aphpic  -6.5973239  1.636688  1.876170 -0.0539063 -2.606926  0.114538
## camchr   0.0149467  0.005839 -0.001540 -0.0486617 -0.025800 -0.011451
## camher   0.0053595  0.005388 -0.018304  0.0212351  0.055811 -0.043935
## camnea  -0.0475211  0.007303 -0.064492  0.0296170 -0.037427  0.051182
## camnov   0.0495257  0.043442  0.054709 -0.2779247 -0.160402 -0.044852
## campen  -0.5022330  0.136300 -0.273167  0.3371128 -1.058098  1.804701
## crelin  -0.0364549  0.003273  0.056658 -0.0025660 -0.009365  0.011170
## forase  -3.9401029  0.322435 -3.436498 -0.1902339 -1.033393 -0.252019
## forinc  -0.0473010 -0.302301 -0.028288 -0.0627297  0.164855  0.176878
## forint  -0.0236505 -0.151151 -0.014144 -0.0313648  0.082428  0.088439
## forlas   0.0111154 -0.004397  0.003731 -0.0207371  0.016502 -0.004384
## forneo1 -0.0237551 -0.399133 -0.116874 -0.0440263  0.032619  0.374764
## forpal  -0.0055288 -0.057853 -0.016476  0.0006622  0.008345  0.055174
## forrub  -0.0729097  0.006546  0.113316 -0.0051319 -0.018729  0.022339
## forsub1 -0.0954477 -0.029161  0.146226  0.0546968 -0.074812  0.063143
## forsub2 -0.0116225 -0.032276 -0.025901 -0.0117362 -0.041243  0.005850
## forsub3 -0.5013344  0.048019  0.470004  0.2172579 -0.251545  0.367550
## lasame  -0.0186489 -0.145575 -0.001583  0.0124915 -0.065192  0.139699
## lasaph   0.1021082 -1.494017 -0.096945 -0.6219014  1.779233  1.251257
## lasbre  -0.0606727  0.053703  0.132878  0.0194112  0.082201 -0.040869
## lassub1 -0.4010035  0.036004  0.623238 -0.0282257 -0.103011  0.122867
## lasumb   0.0293857  0.097279 -0.020457  0.3384111 -0.488902  0.392499
## myrame1 -0.0816221 -0.019029  0.009434 -0.0542999 -0.006855  0.028993
## myrdet  -0.1180762  0.006380  0.037062  0.0356022 -0.075286  0.099216
## myrpun   0.0373890 -0.405073 -0.130609 -0.0533482 -0.126024  0.201756
## myrscu  -0.0008195 -0.072792 -0.039999  0.0228965 -0.029915  0.106561
## ponpen  -0.0358378 -0.099996 -0.032994 -0.0561617 -0.016749 -0.010208
## steimp  -0.0159976 -0.189078 -0.023523 -0.0515458  0.082170  0.027431
## stesch   0.0164531  0.015064 -0.001260 -0.0047682  0.033355  0.016553
## tapses  -0.0394175 -0.251918 -0.023573 -0.0522747  0.137379  0.147398
## temamb  -0.0364549  0.003273  0.056658 -0.0025660 -0.009365  0.011170
## temlon  -0.5205062  0.178650 -0.326203  0.0410251 -0.023972 -0.010523
##
```

```
##
## Site scores (weighted sums of species scores)
##
##           RDA1     RDA2     RDA3     PC1     PC2      PC3
## row1  -6.62275   0.3506  9.77783 -0.4289 -1.5652  1.86690
## row2  -8.64928   4.7774 -6.63021  1.2339  0.1394 -0.03434
## row3   0.75002   3.4885  1.73114 -1.1186  1.3556 -1.49394
## row4   0.75469   4.3859  1.62896  2.6218 -3.4813  1.27421
## row5   2.22119   0.7093 -0.16125  1.1831  3.1094 -2.44776
## row6  -0.01672   0.2020 -2.79620  1.8581 -3.1974  4.29446
## row7   2.33869   4.1286 -0.63368  1.3758 -3.5804  7.02128
## row8   2.25823   1.8724 -0.08031  6.7124 -1.4136 -3.63598
## row9   0.59786   2.2295  1.83761  0.9183  3.8261 -2.17443
## row10 -4.21662  -6.4802 -4.20344 -3.8197 -3.6959 -3.31665
## row11  0.89627   1.1732  1.36681  2.7296 -2.2147 -3.69963
## row12  4.10090   0.7843 -2.21581 -0.7970  5.5749  2.76669
## row13 -0.61581  -4.5054  2.50646 -8.1333 -4.3121 -1.91391
## row14 -0.31448 -10.7228  0.80689 -1.7474  4.5923  4.92721
## row15  4.14565   1.7173 -2.13359  0.8778  2.1048 -2.70132
## row16  2.37217  -4.1106 -0.80122 -3.4660  2.7581 -0.73279
##
##
## Site constraints (linear combinations of constraining variables)
##
##          RDA1    RDA2     RDA3     PC1     PC2      PC3
## row1  -6.0930  0.5471  9.46979 -0.4289 -1.5652  1.86690
## row2  -8.7298  3.7176 -6.84754  1.2339  0.1394 -0.03434
## row3   0.5378  4.4662  0.24671 -1.1186  1.3556 -1.49394
## row4   1.8565  1.6931  1.72540  2.6218 -3.4813  1.27421
## row5   0.2986  0.3002 -1.01977  1.1831  3.1094 -2.44776
## row6   0.3936 -1.2485 -1.96582  1.8581 -3.1974  4.29446
## row7   3.2541  2.6844 -1.43827  1.3758 -3.5804  7.02128
## row8   1.5389 -4.0182 -0.03755  6.7124 -1.4136 -3.63598
## row9  -1.0119  2.1072  3.18483  0.9183  3.8261 -2.17443
## row10 -2.3361 -4.1461 -2.36328 -3.8197 -3.6959 -3.31665
## row11  1.4588 -1.7585  1.19184  2.7296 -2.2147 -3.69963
## row12  2.7500  2.5178 -0.21057 -0.7970  5.5749  2.76669
## row13  2.4982  0.9759 -0.25745 -8.1333 -4.3121 -1.91391
## row14 -1.3176 -8.4211 -0.78801 -1.7474  4.5923  4.92721
## row15  3.0442  1.3178 -1.51396  0.8778  2.1048 -2.70132
## row16  1.8578 -0.7349  0.62365 -3.4660  2.7581 -0.73279
##
##
## Biplot scores for constraining variables
##
##               RDA1    RDA2     RDA3 PC1 PC2 PC3
## medvir     -0.9204  0.1611 -0.35613   0   0   0
## ztot.cover -0.6986 -0.7146 -0.03642   0   0   0
## drysp      -0.3825  0.2379  0.89282   0   0   0
```

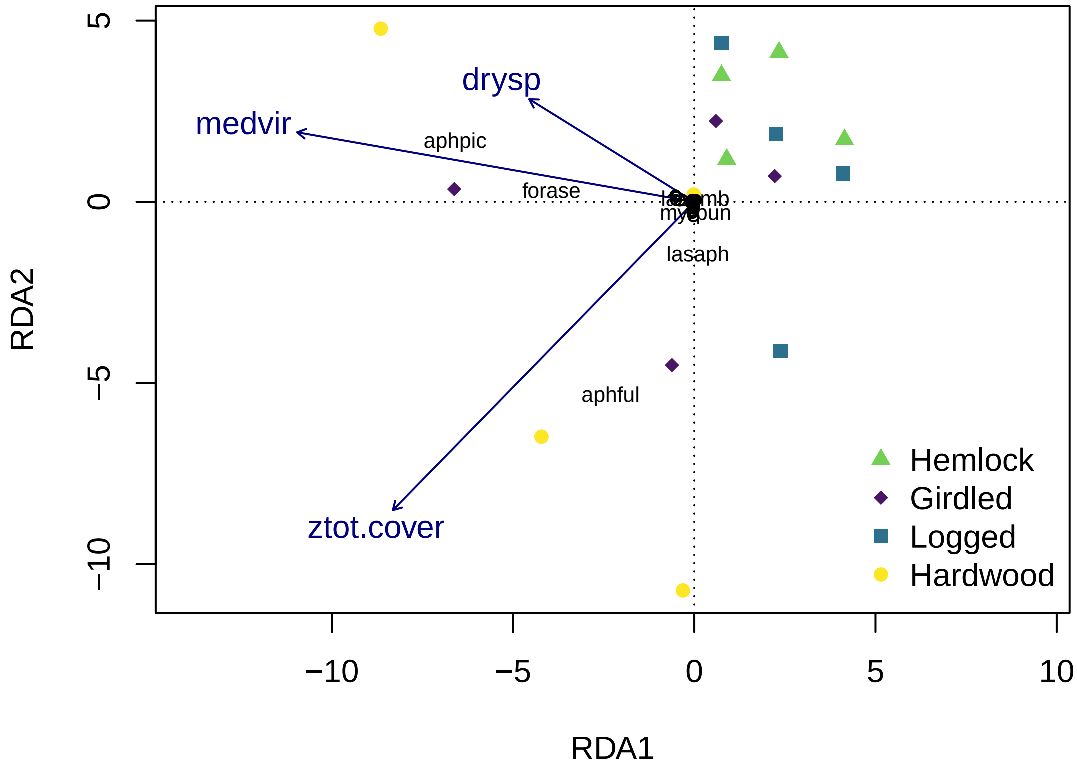
Permutation tests also can be used to test the overall fit of the reduced model. The anova() function can assess the importance of the RDA axes for explaining the community data.

```
anova(ants.rda.red, by = 'axis') # which axes are important
```

```
## Permutation test for rda under reduced model
## Forward tests for axes
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = ants3[, 6:38] ~ medvir + ztot.cover + drysp, data = herb3[, 5:23])
##          Df Variance       F Pr(>F)
## RDA1      1   728.49 15.4095  0.001 ***
## RDA2      1   383.84  8.1194  0.003 **
## RDA3      1   182.74  3.8654  0.005 **
## Residual 12   567.30
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

RDA axis 1, axis 2, and axis 3 of the reduced model were found to be important. The reduced model explains less variation in ant species compared to the full RDA model. Axis 1 explains 40% of the variance and axis 2 explains 22%. The reduced model identified three environmental variables to be important for explaining ant communities - the total cover of understory vegetation (ztot.cover), as well as the percentage cover of medvir and drysp. Medvir is the species code for *Medeola virginiana*, a plant species in the lily family known commonly as Indian cucumber. Drysp is the species code for *Dryopteris carthusiana*, a native fern species commonly known as spinulose woodfern. We can plot the reduced model with sites categorized by the hemlock treatment to describe the overall patterns in ant species and understory plant data.

```
ordiplot(ants.rda.red, display = c('si', 'cn'), type = 'n')
points(ants.rda.red, dis = "sites", select = which(herb3$trt=="hemlock"), pch = 17, cex = 1, col = "#73
points(ants.rda.red, dis = "sites", select = which(herb3$trt=="girdled"), pch = 18, cex = 1, col = "#48
points(ants.rda.red, dis = "sites", select = which(herb3$trt=="logged"), pch = 15, cex = 1, col = "#2D7
points(ants.rda.red, dis = "sites", select = which(herb3$trt=="hardwood"), pch = 16, cex = 1, col = "#F
text(ants.rda.red, display = 'cn', col = 'navy', cex = 1)
orditorp(ants.rda.red, display = 'sp')
legend("bottomright", legend = c("Hemlock", "Girdled", "Logged", "Hardwood"),
       pch = c(17, 18, 15, 16), cex = 1, bty = "n", col = c("#73D055FF", "#481567FF", "#2D708EFF", "#FD
```



2.2.2 Canonical correspondence analysis (CCA)

Canonical correspondence analysis is similar to RDA, but there are a couple differences. CCA combines multiple regression and correspondence analysis (CA) to model multivariate response variables as a function of a set of environmental predictors. CCA is most useful when environmental gradients are long because it assumes unimodal relationships among species and measured environmental gradients.

We can run a CCA using the cca() function in the package `vegan`.

As you can see, the output looks very similar to RDA, and you can interpret it the same way. We can see that the first two axes explain 42% of the variance (27.5% for axis 1 and 15.2% for axis 2). We can use the same permutation procedures to assess the importance of the environmental variables via a reduced model as well as the importance of the axes.

```
mod0.cca <- cca(ants3[,6:38] ~ 1, herb3[,5:23])
ants.cca.red <- ordistep(mod0.cca, scope = formula(ants.cca), direction = "both",
                         permutations = how(nperm = 199))
```

```
##
## Start: ants3[, 6:38] ~ 1
##
##               Df    AIC      F Pr(>F)
## + medvir       1 78.899 2.6219  0.015 *
## + uvuses       1 78.960 2.5580  0.030 *
## + aranud       1 79.260 2.2504  0.030 *
## + denobs      1 79.510 1.9991  0.045 *
## + denpun      1 79.516 1.9933  0.045 *
## + maican      1 79.623 1.8864  0.050 *
## + rubida      1 79.799 1.7128  0.090 .
## + drysp       1 79.757 1.7535  0.110
## + ruball      1 79.975 1.5405  0.195
## + vaccor      1 80.019 1.4983  0.210
## + ztot.cover  1 80.381 1.1509  0.235
## + rubhis      1 80.423 1.1117  0.265
## + ilever      1 80.352 1.1791  0.335
## + carpen      1 80.431 1.1044  0.340
## + lysqua      1 80.850 0.7139  0.730
## + mitrep      1 81.056 0.5251  0.795
## + lysbor      1 80.931 0.6397  0.815
## + hupluc      1 80.915 0.6536  0.825
## + vacang      1 81.249 0.3513  0.825
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: ants3[, 6:38] ~ medvir
##
##          Df    AIC      F Pr(>F)
## - medvir  1 79.645 2.6219   0.01 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##              Df    AIC      F Pr(>F)
## + denobs      1 78.410 2.1876  0.045 *
## + maican      1 78.693 1.9213  0.050 *
## + denpun      1 78.613 1.9968  0.080 .
## + rubida      1 78.680 1.9340  0.085 .
## + drysp       1 78.643 1.9679  0.120
## + ruball      1 78.891 1.7382  0.165
## + ztot.cover  1 79.090 1.5559  0.195
## + uvuses      1 79.447 1.2346  0.220
## + aranud      1 79.233 1.4261  0.245
## + ilever      1 79.494 1.1925  0.315
## + rubhis      1 79.526 1.1646  0.320
## + carpen      1 79.545 1.1475  0.320
## + hupluc      1 79.835 0.8933  0.575
## + vaccor      1 80.141 0.6309  0.630
## + lysbor      1 80.070 0.6912  0.725
## + mitrep      1 80.275 0.5169  0.770
## + lysqua      1 80.094 0.6709  0.785
## + vacang      1 80.633 0.2175  0.910
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: ants3[, 6:38] ~ medvir + denobs
##
##          Df    AIC      F Pr(>F)
## - denobs  1 78.899 2.1876  0.025 *
## - medvir  1 79.510 2.7788  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##              Df    AIC      F Pr(>F)
## + denpun      1 78.543 1.4856  0.125
## + hupluc      1 78.662 1.3854  0.185
## + rubida      1 78.654 1.3923  0.205
## + uvuses      1 78.823 1.2518  0.275
## + drysp       1 78.836 1.2408  0.285
## + ruball      1 78.832 1.2443  0.290
## + ztot.cover  1 78.935 1.1592  0.310
## + ilever      1 79.072 1.0470  0.370
## + carpen      1 79.188 0.9529  0.500
## + aranud      1 79.357 0.8169  0.595
## + lysqua      1 79.390 0.7904  0.610
## + rubhis      1 79.385 0.7940  0.625
## + mitrep      1 79.548 0.6643  0.675
## + vaccor      1 79.551 0.6619  0.705
## + maican      1 79.659 0.5769  0.750
## + lysbor      1 79.565 0.6512  0.825
## + vacang      1 80.125 0.2155  0.905
```

```
summary(ants.cca.red) # which environmental variables are important
```

```
##
## Call:
## cca(formula = ants3[, 6:38] ~ medvir + denobs, data = herb3[,      5:23])
##
## Partitioning of scaled Chi-square:
##               Inertia Proportion
## Total          1.4126     1.0000
## Constrained    0.3942     0.2791
## Unconstrained  1.0184     0.7209
##
## Eigenvalues, and their contribution to the scaled Chi-square
##
## Importance of components:
##                         CCA1   CCA2    CA1    CA2     CA3     CA4     CA5
## Eigenvalue            0.2290 0.1652 0.2876 0.1815 0.11295 0.11101 0.09125
## Proportion Explained  0.1621 0.1169 0.2036 0.1285 0.07996 0.07858 0.06460
## Cumulative Proportion 0.1621 0.2791 0.4827 0.6112 0.69113 0.76972 0.83432
##                           CA6     CA7     CA8     CA9    CA10     CA11     CA12
## Eigenvalue            0.07008 0.05133 0.04359 0.03418 0.02224 0.007354 0.003594
## Proportion Explained  0.04961 0.03634 0.03086 0.02420 0.01574 0.005206 0.002544
## Cumulative Proportion 0.88393 0.92027 0.95113 0.97533 0.99107 0.996275 0.998819
##                           CA13
## Eigenvalue            0.001668
## Proportion Explained  0.001181
## Cumulative Proportion 1.000000
##
## Accumulated constrained eigenvalues
## Importance of components:
##                        CCA1   CCA2
## Eigenvalue            0.229 0.1652
## Proportion Explained  0.581 0.4190
## Cumulative Proportion 0.581 1.0000
```

```
##
## Scaling 2 for species and site scores
## * Species are scaled proportional to eigenvalues
## * Sites are unscaled: weighted dispersion equal on all dimensions
##
##
## Species scores
##
##              CCA1      CCA2      CA1      CA2       CA3      CA4
## aphful   0.288100  0.002256 -0.34599 -0.29796 -0.137370 -0.01844
## aphpic  -0.197362  0.262456  0.22671 -0.09610  0.321857  0.10973
## camchr   0.662790  0.768691 -0.34024 -1.48251 -0.428561  0.06640
## camher   0.003710  1.074740 -0.49145 -0.90147 -0.881914  0.55925
## camnea  -0.287695 -0.878371  1.08258  0.32920 -0.635724 -0.99226
## camnov   0.442725  0.682341 -0.10525 -0.94138 -0.115848 -0.20211
## campen   0.110134  0.120716  0.71212  0.78909 -0.333076  0.69066
## crelin  -0.655370  1.380789 -0.10845  0.36469  0.742183 -0.86620
## forase  -1.094279 -0.704610  0.21308 -0.15057 -0.483181  0.05873
## forinc   0.886434 -1.619376 -1.05811  0.39674  0.995526  0.12421
## forint   0.886434 -1.619376 -1.05811  0.39674  0.995526  0.12421
## forlas   0.662790  0.768691 -1.27111 -0.20600 -1.754858 -0.31183
## forneo1  0.765598 -1.136878  0.23390  0.28938  0.017665 -0.69090
## forpal   0.772941 -1.272990  0.27491  0.41594  0.049539 -0.74499
## forrub  -0.655370  1.380789 -0.10845  0.36469  0.742183 -0.86620
## forsub1  0.002596  0.509642  0.54022  0.25654  0.030547 -1.03781
## forsub2  0.296280 -0.670946  0.78604 -0.24598 -0.963864 -0.56723
## forsub3 -0.116113  0.355131  0.39534  0.49337 -0.002973 -0.82814
## lasame   0.564541 -0.447444  0.46863  0.11012 -0.170165 -0.82311
## lasaph   0.752795 -0.503066 -1.54457  1.20689  0.065233 -0.07625
## lasbre  -0.128106  1.135950 -0.34592 -0.22879  1.494830  0.03714
## lassub1 -0.655370  1.380789 -0.10845  0.36469  0.742183 -0.86620
## lasumb   0.401884 -0.014173  1.61685  0.63881  0.185641 -0.85407
## myrame1 -0.322031 -0.062950 -0.36873 -0.15094  0.298718 -0.10599
## myrdet  -0.252396 -0.133356  0.77313  0.35869 -0.219876 -1.05389
## myrpun   0.525313 -0.187122  0.13026  0.04808 -0.772622 -0.61873
## myrscu   0.716195 -1.099797  0.94142  0.42554 -0.423455 -1.17960
## ponpen   0.250887 -0.816651 -0.37661 -0.48583 -0.355677  0.36123
## steimp   0.610685 -0.534135 -0.66379 -0.30893  0.074451  0.27286
## stesch   0.662790  0.768691 -2.81705  3.75265 -1.103249 -0.40478
## tapses   0.886434 -1.619376 -1.05811  0.39674  0.995526  0.12421
## temamb  -0.655370  1.380789 -0.10845  0.36469  0.742183 -0.86620
## temlon  -1.850916 -0.551968  0.01296  0.04486 -0.119780  0.17075
##
##
## Site scores (weighted averages of species scores)
##
##            CCA1    CCA2      CA1     CA2     CA3      CA4
## row1  -0.349159  1.6675 -0.10845  0.3647  0.7422 -0.86620
## row2  -2.174179 -0.8632  0.03188  0.1173 -0.1143  0.25163
## row3  -0.428208  1.2667  0.51608 -0.8447  2.0055  0.84219
## row4  -0.005495  1.0521  2.00095  0.9125  1.1046 -0.38007
## row5   0.402082  0.9883 -0.49145 -0.9015 -0.8819  0.55925
## row6   0.166371 -0.7484  1.60794  0.4351 -0.8964 -1.61420
## row7  -0.062362  1.0734  1.80497  2.1454 -0.9071  4.00817
## row8   0.278190  0.7690  0.35081 -0.4251 -0.2504 -0.39986
## row9  -0.002082  1.3371 -0.40529 -0.3772  1.6830  0.26298
## row10 -0.612175 -0.6882 -0.03586 -0.9271 -1.0313  0.47974
## row11  0.102902  0.8297  0.32627 -1.3181  1.2026  0.81022
## row12  2.101706 -1.1640 -2.81705  3.7526 -1.1032 -0.40478
## row13  0.678220  0.6777 -0.34024 -1.4825 -0.4286  0.06640
## row14  1.690620 -1.6977 -1.05811  0.3967  0.9955  0.12421
## row15  1.065324  0.1569 -0.91284 -1.5860 -1.2569  0.03888
## row16  1.444643 -0.2981 -1.27111 -0.2060 -1.7549 -0.31183
##
##
## Site constraints (linear combinations of constraining variables)
##
##            CCA1      CCA2      CA1     CA2     CA3      CA4
## row1  -0.65537  1.380789 -0.10845  0.3647  0.7422 -0.86620
## row2  -2.18198 -0.781905  0.03188  0.1173 -0.1143  0.25163
## row3   0.35680 -0.002907  0.51608 -0.8447  2.0055  0.84219
## row4   0.66279  0.768691  2.00095  0.9125  1.1046 -0.38007
## row5   0.00371  1.074740 -0.49145 -0.9015 -0.8819  0.55925
## row6   0.65945 -0.926603  1.60794  0.4351 -0.8964 -1.61420
## row7   0.66279  0.768691  1.80497  2.1454 -0.9071  4.00817
## row8   0.66279  0.768691  0.35081 -0.4251 -0.2504 -0.39986
## row9   0.00371  1.074740 -0.40529 -0.3772  1.6830  0.26298
## row10 -0.06689 -0.415289 -0.03586 -0.9271 -1.0313  0.47974
## row11  1.01588 -0.308955  0.32627 -1.3181  1.2026  0.81022
## row12  0.66279  0.768691 -2.81705  3.7526 -1.1032 -0.40478
## row13  0.66279  0.768691 -0.34024 -1.4825 -0.4286  0.06640
## row14  0.88643 -1.619376 -1.05811  0.3967  0.9955  0.12421
## row15  0.66279  0.768691 -0.91284 -1.5860 -1.2569  0.03888
## row16  0.66279  0.768691 -1.27111 -0.2060 -1.7549 -0.31183
##
##
## Biplot scores for constraining variables
##
##            CCA1    CCA2 CA1 CA2 CA3 CA4
## medvir -0.9503 -0.3114   0   0   0   0
## denobs -0.4212 -0.9070   0   0   0   0
```

```
anova(ants.cca.red, by = 'axis') # which axes are important
```

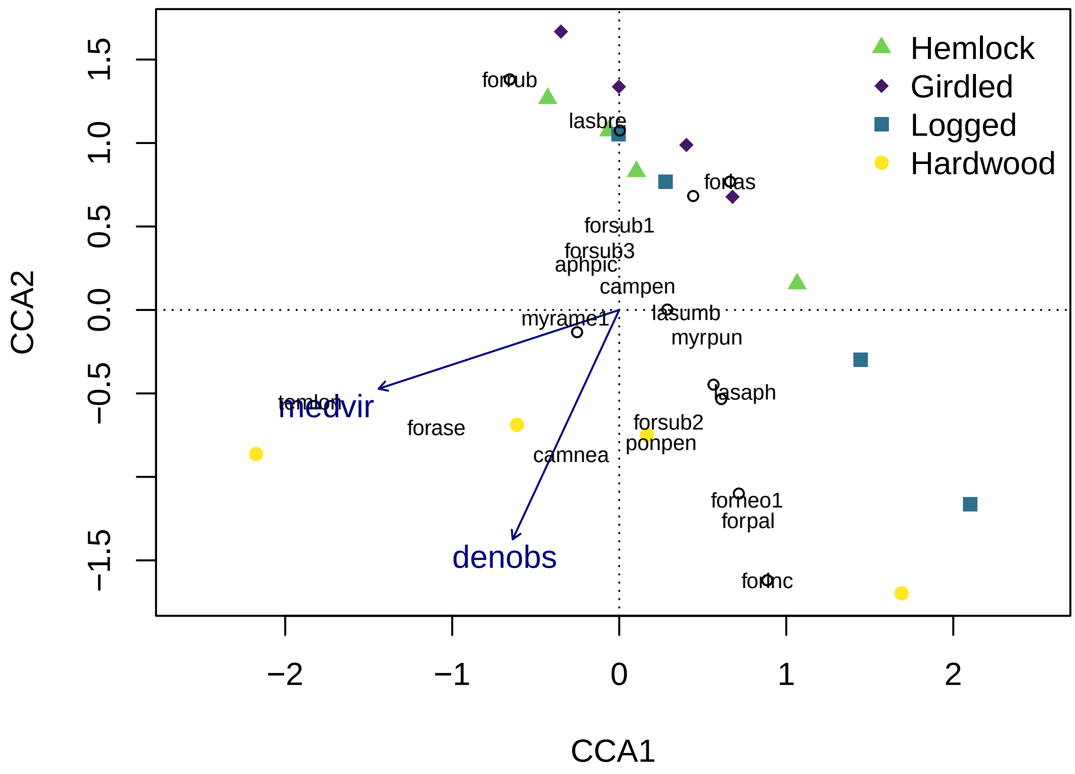
```
## Permutation test for cca under reduced model
## Forward tests for axes
## Permutation: free
## Number of permutations: 999
##
## Model: cca(formula = ants3[, 6:38] ~ medvir + denobs, data = herb3[, 5:23])
##          Df ChiSquare      F Pr(>F)
## CCA1      1   0.22901 2.9233  0.015 *
## CCA2      1   0.16518 2.1086  0.027 *
## Residual 13   1.01839
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

CCA axis 1 and axis 2 of the reduced model were found to be important. The reduced CCA model explained less variation in ant species compared to the full CCA model. Axis 1 explains 16.2% of the variance and axis

2 explains 11.6%. The reduced model identified two environmental variables to be important for explaining ant communities - percentage cover of denobs and medvir. As in the RDA, medvir is the species code for *Medeola virginiana*, a plant species in the lily family known commonly as Indian cucumber. Denobs is the species code for *Dendrolycopodium obscurum*, a plant species in the clubmoss family known as rare clubmoss or ground pine.

We can plot the reduced model with sites categorized by the hemlock treatment to describe the overall patterns in ant species and understory plant data.

```
ordiplot(ants.cca.red, display = c('si', 'cn'), type = 'n')
points(ants.cca.red, dis = "sites", select = which(herb3$trt=="hemlock"), pch = 17, cex = 1, col = "#73
points(ants.cca.red, dis = "sites", select = which(herb3$trt=="girdled"), pch = 18, cex = 1, col = "#48
points(ants.cca.red, dis = "sites", select = which(herb3$trt=="logged"), pch = 15, cex = 1, col = "#2D7
points(ants.cca.red, dis = "sites", select = which(herb3$trt=="hardwood"), pch = 16, cex = 1, col = "#F
text(ants.cca.red, display = 'cn', col = 'navy', cex = 1)
orditorp (ants.cca.red, display = 'sp')
legend("topright", legend = c("Hemlock", "Girdled", "Logged", "Hardwood"),
       pch = c(17, 18, 15, 16), cex = 1, bty = "n", col = c("#73D055FF", "#481567FF", "#2D708EFF", "#FD
```



2.2.3 How to select which method (CCA or RDA) to use

The most appropriate model to use depends on the length of the environmental gradients represented in the study. CCA assumes that the environmental variables represent a unimodal distribution and that sites sampled are representative of both ends of this gradient. RDA assumes a linear distribution such that sites sampled represent only a portion of the unimodal distribution. Generally, studies that implement experimental treatments and/or are smaller in scale will have linear environmental gradients. This is because smaller scale studies usually do not incorporate the full range of environmental variation experienced by species in a community to produce a unimodal response.

The length of the gradient can be estimated based on the data using a Detrended Correspondence Analysis (DCA). DCA is an indirect gradient method that was originally developed to remove the arch effect from Correspondence Analysis (CA). This method is no longer used for community analysis except for the purpose of identifying gradient lengths. DCA repeatedly splits the main axes into segments until the best solution is found. The length of the first axis can be used to determine whether an RDA (linear, axis length < 4) or CCA (unimodal, axis length > 4) model is best for the data set.

```
DCA <- decorana(ants3[,6:38])
DCA
```

```
##
## Call:
## decorana(veg = ants3[, 6:38])
##
## Detrended correspondence analysis with 26 segments.
## Rescaling of axes with 4 iterations.
## Total inertia (scaled Chi-square): 1.4126
##
##                        DCA1   DCA2    DCA3    DCA4
## Eigenvalues          0.3548 0.1835 0.14205 0.09275
## Additive Eigenvalues 0.3548 0.1828 0.13337 0.05265
## Decorana values      0.3884 0.1422 0.05087 0.02241
## Axis lengths         2.1980 1.2314 1.11022 1.26230
```

The length of DCA axis 1 is less than 4, so RDA is the more appropriate analysis for the ant data set.

3 R Activity

1. We will use another open source data set from the NSF Harvard Forest Long-term Ecological Research (LTER) site. These data are spiders collected in the Hemlock Removal Experiment. Remember, this experiment includes four treatments (Hemlock girdled, Hemlock logged, Hemlock control, and Hardwood control) each replicated across two ($n = 2$) 90 x 90 m plots. Load the `HarvardForest_spiders` data into R. We will characterize spider communities among these four treatments. Load the `HarvardForest_HerbLayer` data into R. We will also assess the relationship among spiders and understory plants.

2. Before running any analyses, you will have to do some data wrangling. First, create a new variable `abundance` by summing the counts of adult male and female spiders. Next, change the data set from `long` format to `wide` format using spider genus as the taxonomic resolution (i.e., each column should be a spider genus). Be sure that the new data frame includes the predictor variables `block`, `plot`, `replicate`, `treatment`, and `sampling method`. Then, remove any columns that do not have count data (some genera are indicated as immatures with imm), as well as three columns with unidentified spiders (LinytoID, LinyToID, and unk_toID). Change the variables `treatment` and `sampling method` to factors. Calculate a dissimilarity matrix for the spider data using the bray-curtis method. Provide the distance matrix output.

3. Run a nonmetric multidimensional scaling (NMDS) model using the metaMDS() function on the spider genera data. Provide a figure showing the treatment groups with 95% confidence interval ellipses. Report the model stress.

4. Run a permutational multivariate analysis of variance (PERMANOVA) using adonis2(). Model the bray-curtis distance matrix as a function of `treatment`. Provide the PERMANOVA output.

5. Interpret the output (using the figure and PERMANOVA results).

6. Now, some data wrangling for the understory herb data. Change the data set from `long` format to `wide` format using species as the taxonomic resolution (i.e., each column should be a plant species). Be sure that the new data frame includes the predictor variables `year`, `block`, `trt`, `plot`, and `subplot`. Then, subset the data by year 2008, so it aligns with the spider data. Lastly, double check that the number of samples per plot aligns between the spider and plant data sets. Each plot should have 10 samples (indicated as replicates or subplots).

7. Run a detrended correspondence analysis (DCA) to determine the most appropriate constrained model for these data. Provide the output. Use the original data set with abundances.

8. Run the appropriate model (RDA or CCA). Provide a figure.

9. Interpret the results.