Multivariate statistics
Solutions to Exercises

April 27, 2018

Contents

1 Multivariate statistics **2**

1.5 Exercises 2

1.5.1 Ordination 2

1.5.2 Analysis of Structure 1: two-table analysis 13

1.5.3 Analysis of Structure 2: variation partitioning 18

1.5.4 Analysis of Structure 3: 'experimental' systems 36

Chapter 1

Multivariate statistics

1.5 Exercises

In these exercises, we use the following colour codes:

- ■ **Easy**: make sure you complete some of these before moving on. These exercises will follow examples in the text very closely.
- ♦ **Intermediate**: a bit harder. You will often have to combine functions to solve the exercise in two steps.
- ▲ **Hard**: difficult exercises! These exercises will require multiple steps, and significant departure from examples in the text.

We suggest you complete these exercises in an **R** markdown file. This will allow you to combine code chunks, graphical output, and written answers in a single, easy-to-read file.

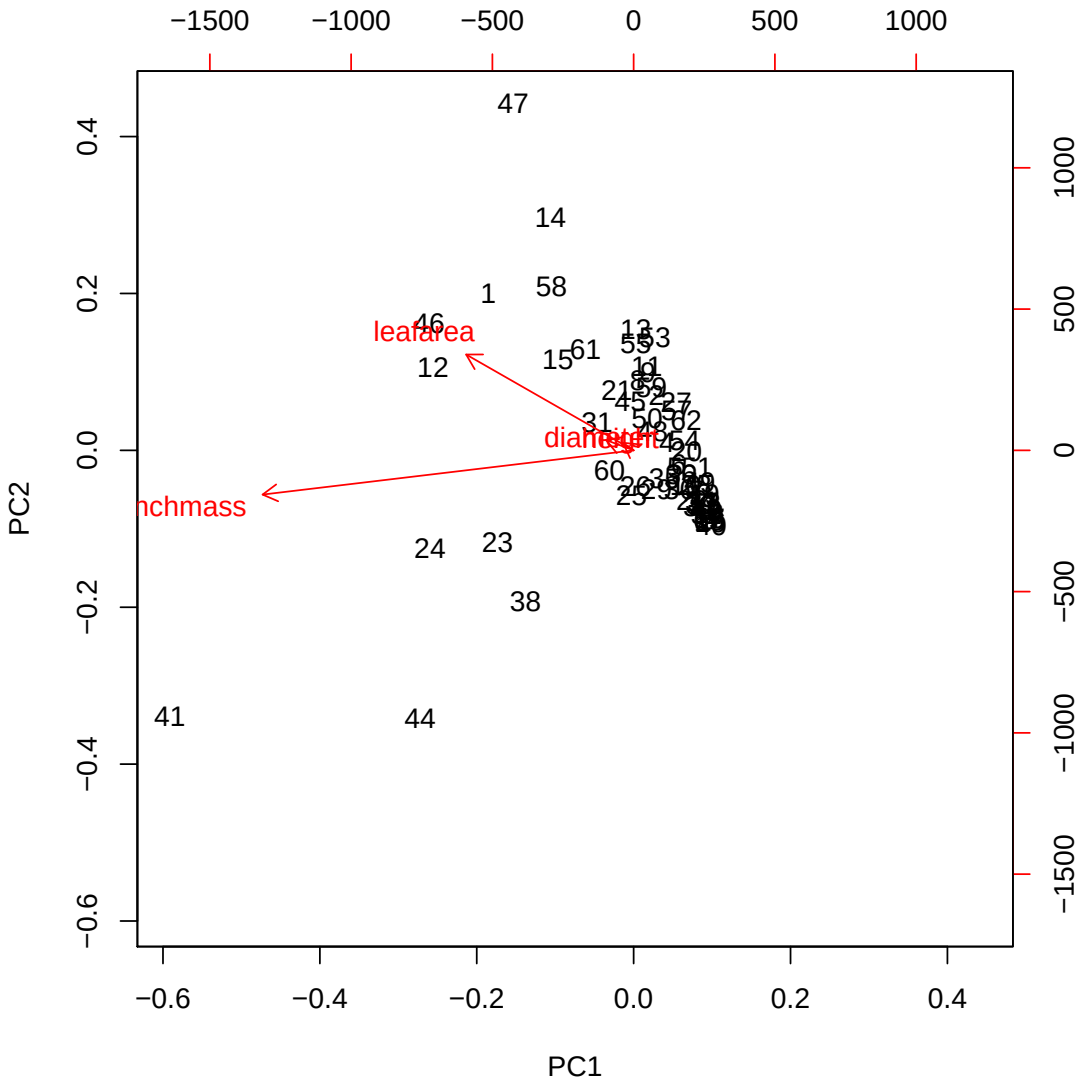
1.5.1 Ordination

1.5.1.1 Allometry data

1. ■ Perform a PCA of the data in the allometry dataset ('Allometry.csv') without changing the `scale` argument and plot the ordination result. Note that the first column contains species identities so you need to exclude this column from the analysis. See Section **??** for help, if necessary.

```
library(vegan)

## Loading required package:  permute

## Loading required package:  lattice

## This is vegan 2.4-4

# read the allometry data
allom<-read.csv('Allometry.csv')

# perform PCA on the last four columns (growth variables) of the allometry data
allom.pca<-prcomp(allom[,2:5])

# produce an ordination plot of the results
```

```
biplot(allom.pca)
```



2. ■ In the previous ordination, the length of each vector varies extensively. This is because the variance in 'branchmass' and 'leafarea' is much larger than that of 'diameter' and 'height'. Use the `var` function to calculate the variance for each variable in `allom` to confirm this. Then use the `scale` argument to repeat the analysis after standardising the variables. (See Section **??** for help, if necessary.) What effect did this have on the vector lengths?

```
library(vegan)

# read the allometry data
allom<-read.csv('Allometry.csv')

# estimate variance for each of the four growth parameters
var(allom$diameter)
```

```
## [1] 373.2473
```

```
var(allom$height)
```

```
## [1] 130.3961
```

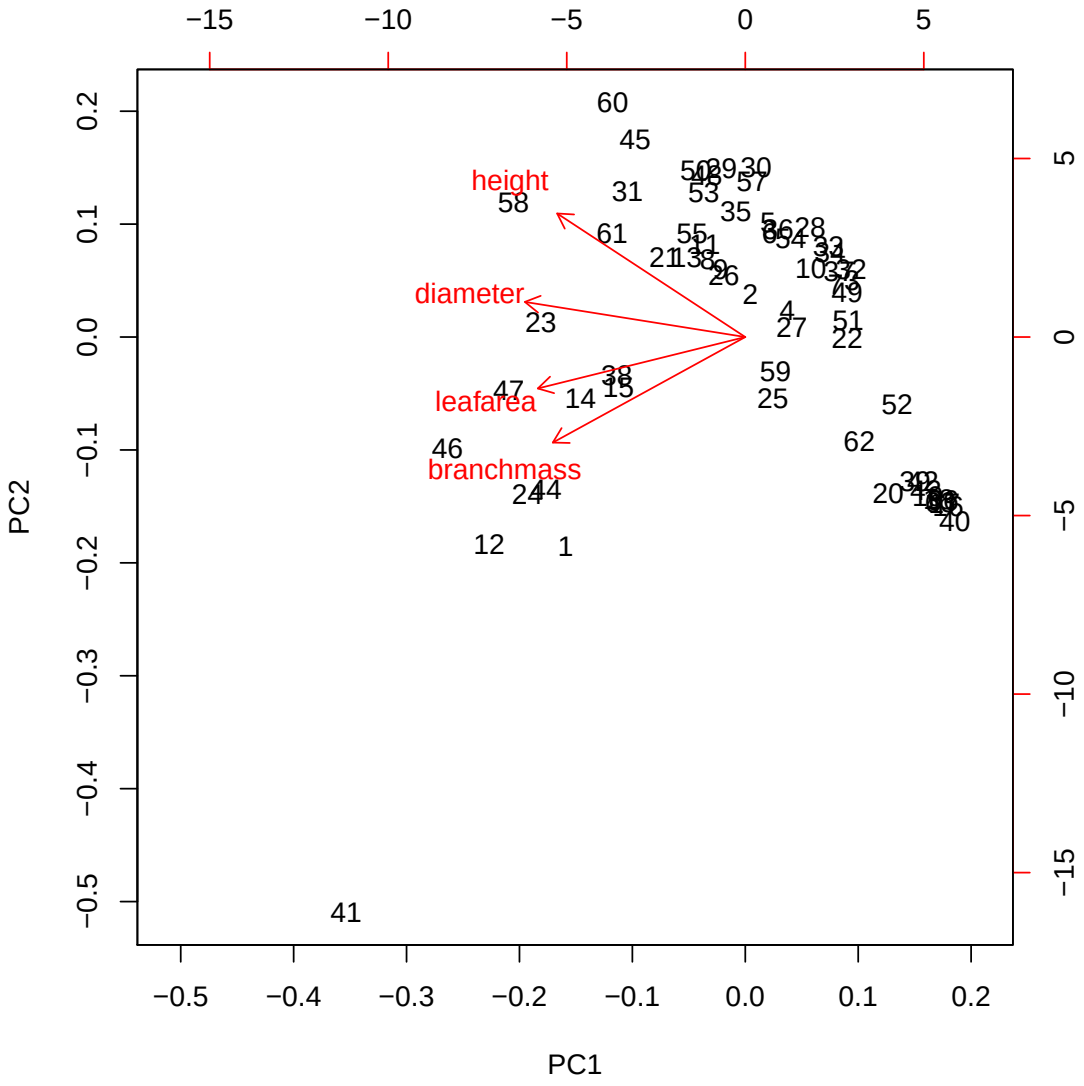
```
var(allom$leafarea)
```

```
## [1] 11573.64
```

```
var(allom$branchmass)
```

```
## [1] 43385.5
```

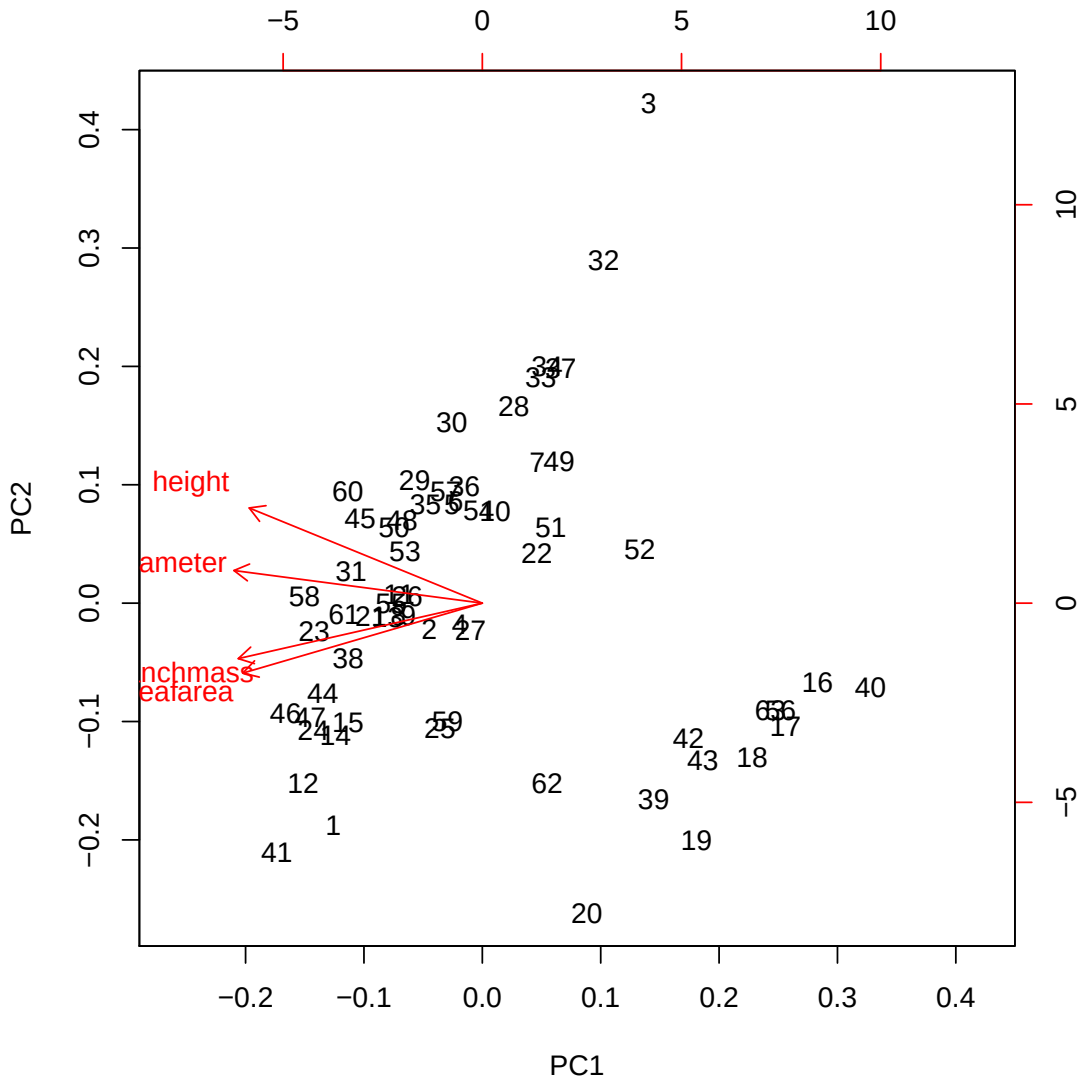
```
# repeat PCA on these variables but scale each variable relative to
# its standard deviation (scale=T)
allom.pca<-prcomp(allom[,2:5],scale=T)

# produce an ordination plot of the results
biplot(allom.pca)
```



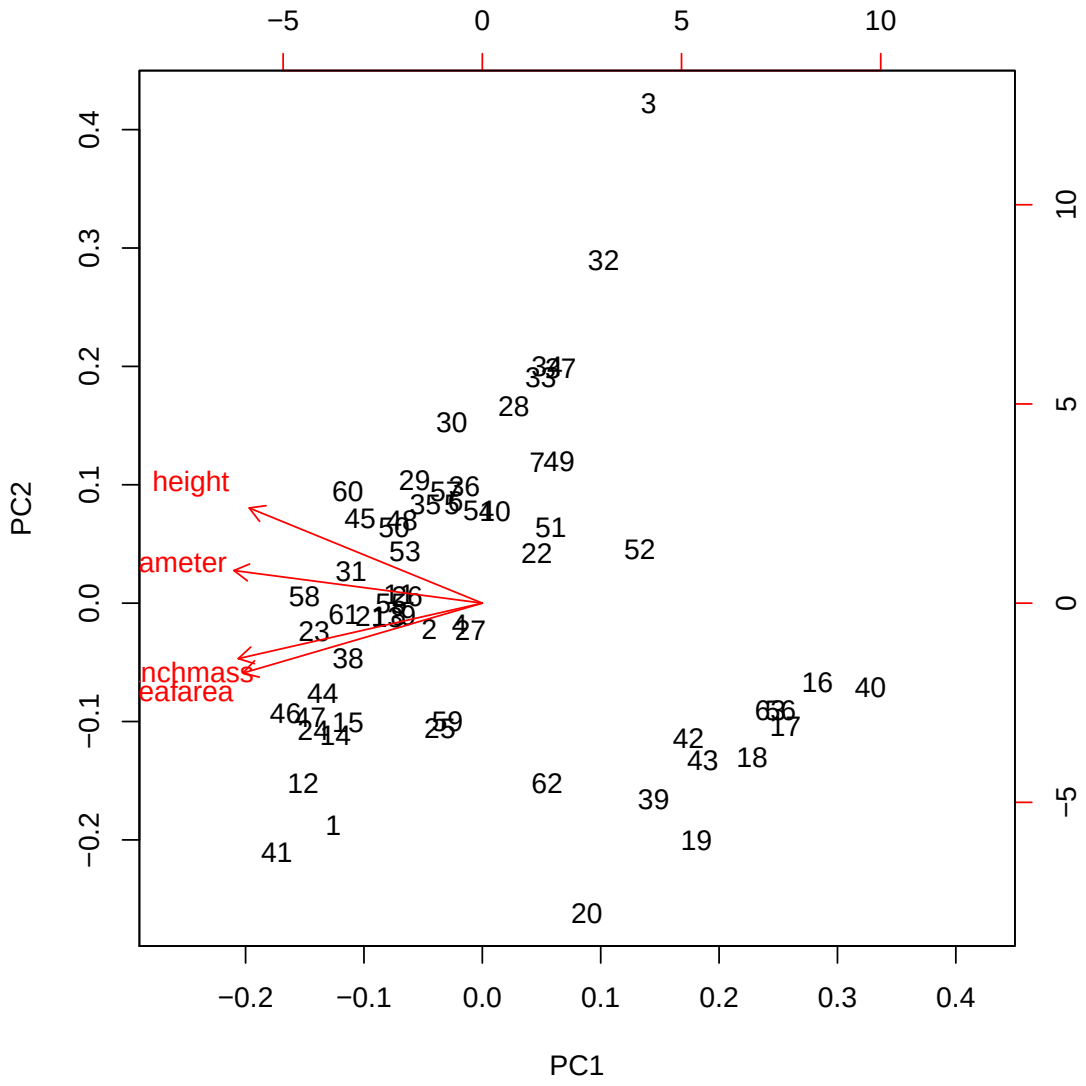
```
# vector lengths are much more similar than for the default analysis (scale = F)
```

3. ♦ The ordination plot shows very strong overlap for most of the sites and a few very divergent sites. In addition, the plot is partly off-centre. These properties indicate that there is lack of normality in the response variables. Log-transform the variables to see the effects on the ordination, trying this two ways: (a) using `log()` to generate new vectors containing transformed data and (b) using `decostand()` with an appropriate 'method' argument). See Section **??** for help, if necessary.

```
library(vegan)

# read the allometry data
allom<-read.csv('Allometry.csv')

# log-transform the data prior to PCA
# using log() on the four columns containing growth data, nested within the call to prcomp()
allom.pca.log1<-prcomp(log(allom[,2:5]),scale=T)
biplot(allom.pca.log1)
```




```
# using decostand() (not exactly 'log', see help pages;
# ignore warning message), nested in prcomp()
allom.pca.log2<-prcomp(decostand(allom[,2:5],'log'),scale=T)

## Warning:  non-integer data:  divided by smallest positive value

biplot(allom.pca.log2)
```



1.5.1.2 Endophyte data

1. ■ Read in the data from 'endophytes.csv'; see Section **??** (p. **??**) for a description of the data. Use the `decorana` function to calculate gradient lengths and determine whether PCA is appropriate for these data (see Section **??** for help, if necessary).

```
# read in endophyte community data
endo<-read.csv('endophytes.csv')

# estimate gradient length
decorana(endo)
```

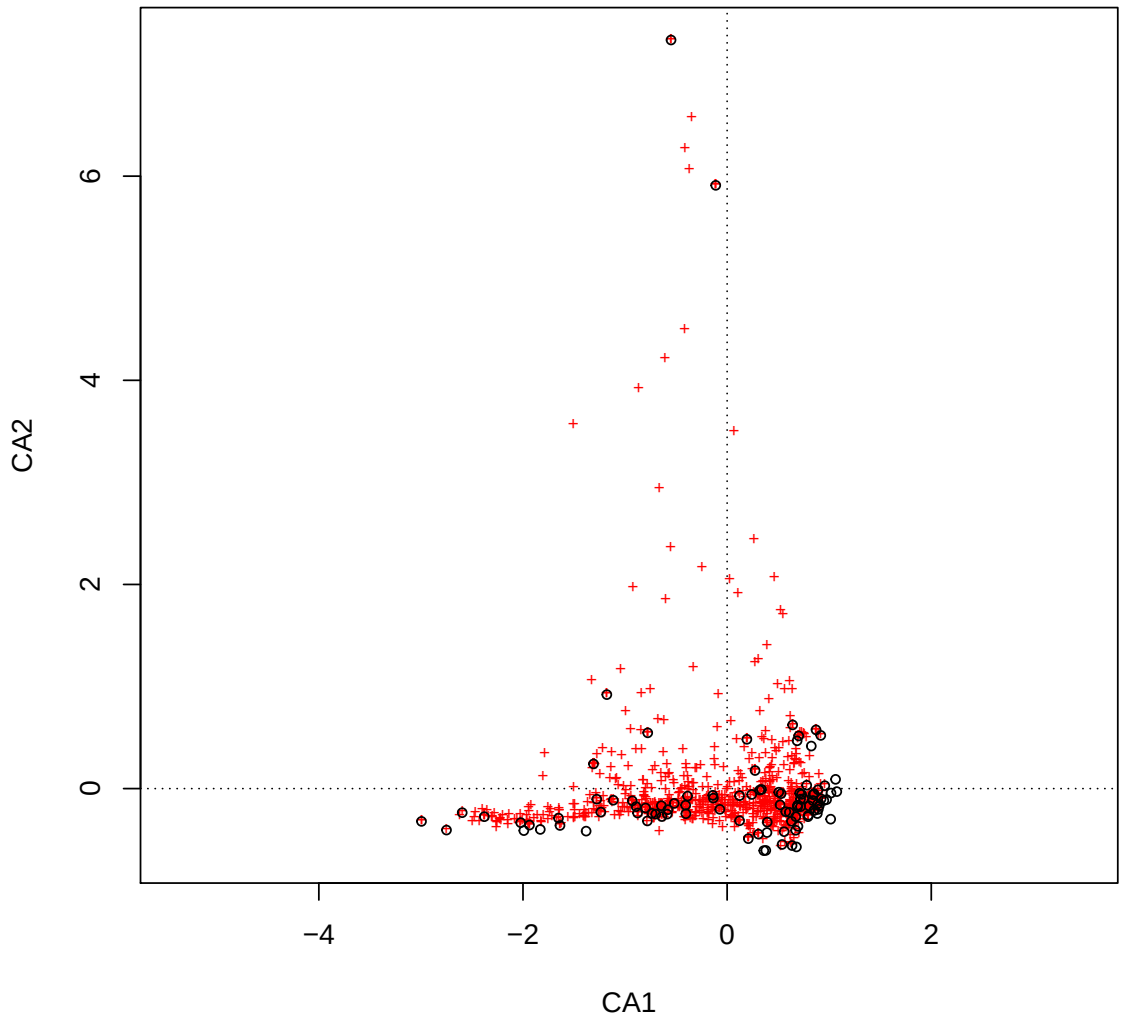
```
##
## Call:
## decorana(veg = endo)
##
```

```
## Detrended correspondence analysis with 26 segments.
## Rescaling of axes with 4 iterations.
##
##                   DCA1   DCA2   DCA3   DCA4
## Eigenvalues     0.5160 0.3277 0.2663 0.2971
## Decorana values 0.5406 0.4279 0.2738 0.2552
## Axis lengths    3.6757 4.0614 2.7659 2.9553
```

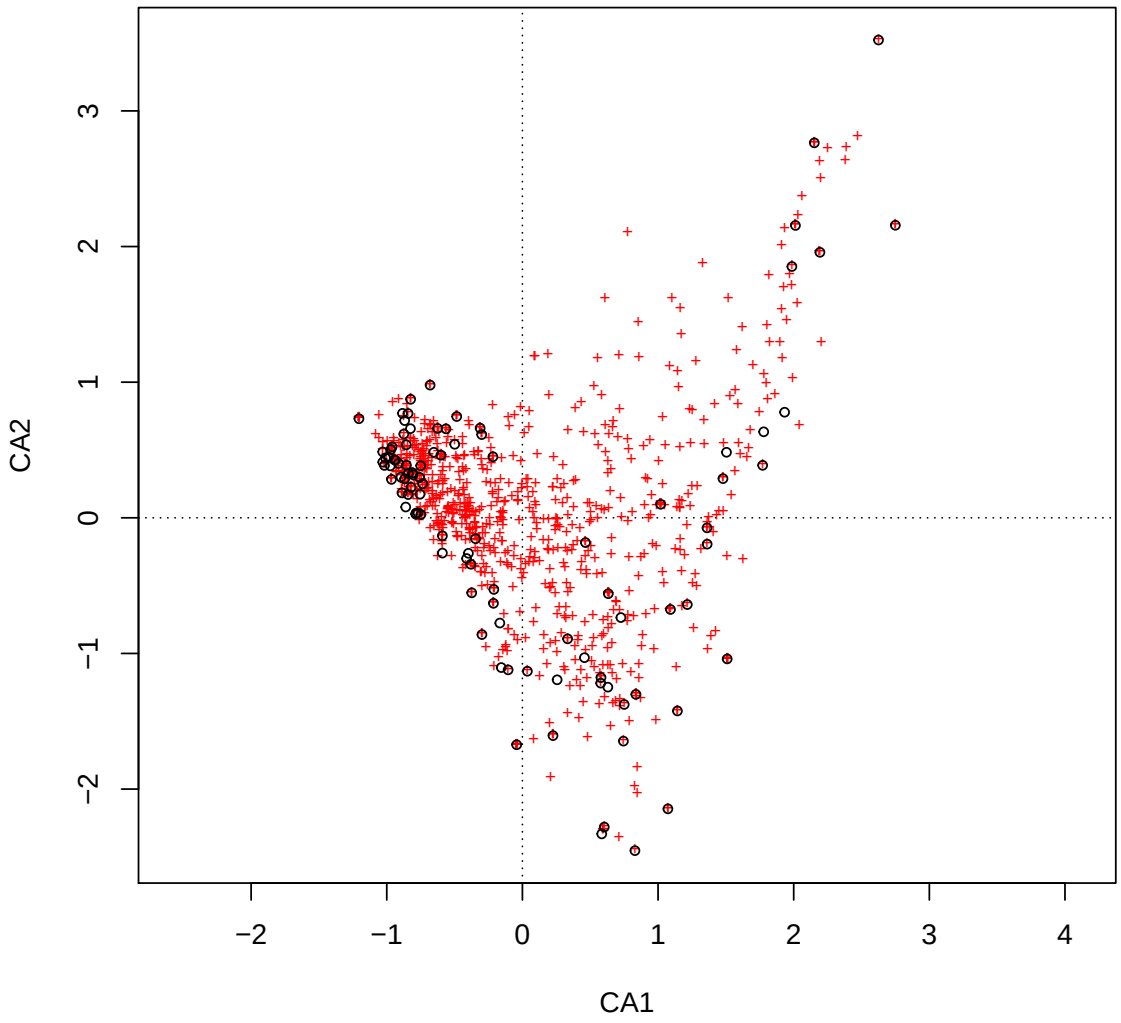
```
# the gradient (axis) lengths are around or greater than 3,
# suggesting that PCA is not appropriate
```

2. ♦ Perform CA these data and plot the results. Notice the strong skew in the data along both axes. Try again after standardising the community matrix using the `decostand` function (try the 'hellinger' and 'max' methods). Notice the undesireable parabolic pattern in the ordination and strong skew; this suggests that CA is not an improvement over PCA (common for data matrices that contain many zeros, collected along long environmental gradients).

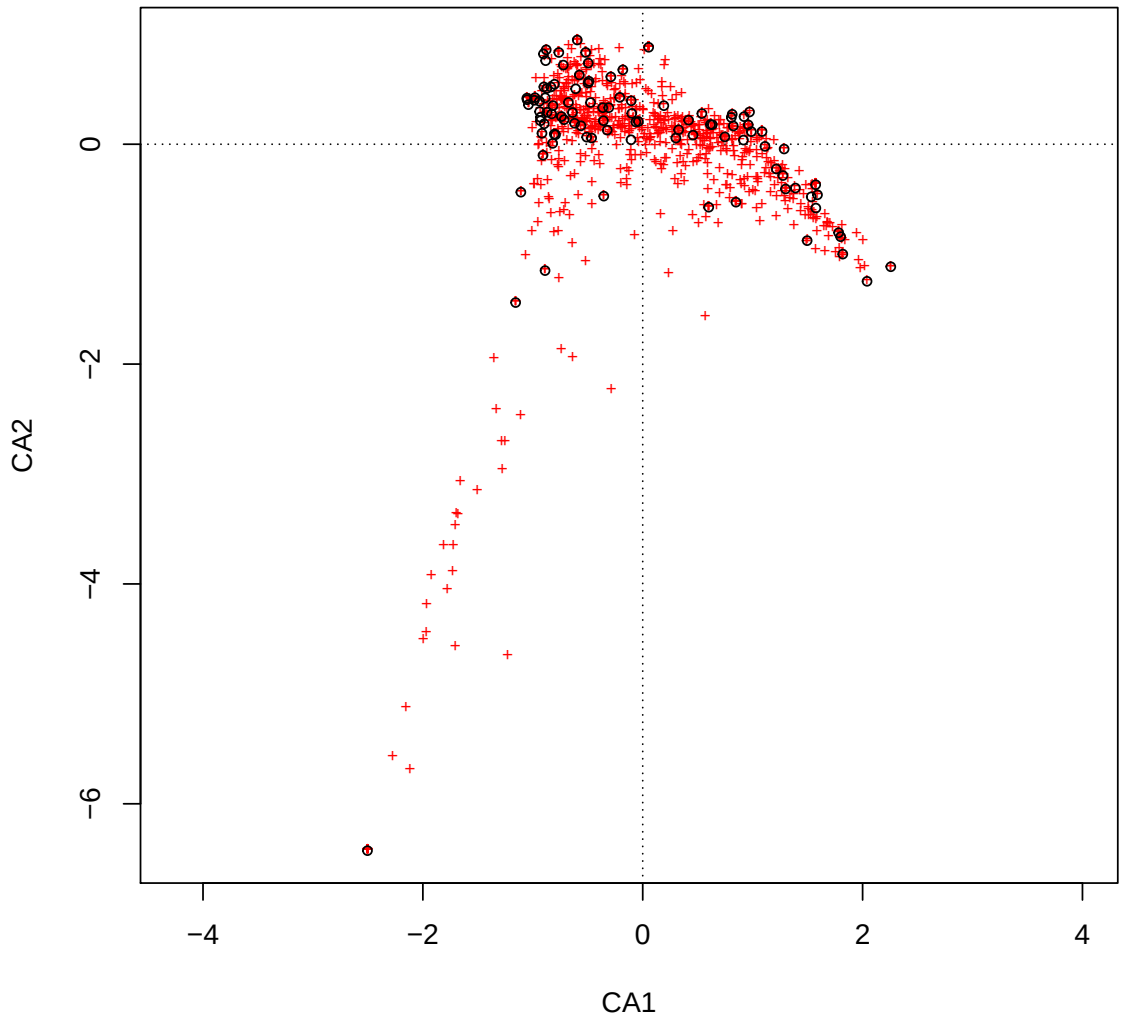
```
# read in endophyte community data
endo<-read.csv('endophytes.csv')

# plot cca results using raw data and following two different standardisation
# approaches
plot(vegan::cca(endo))
```




```
plot(vegan::cca(decostand(endo, method='hellinger')))
```




```
plot(vegan::cca(decostand(endo, method='max')))
```




```
# how many cells in the matrix are zeros?
summary(as.numeric(endo == 0))
```

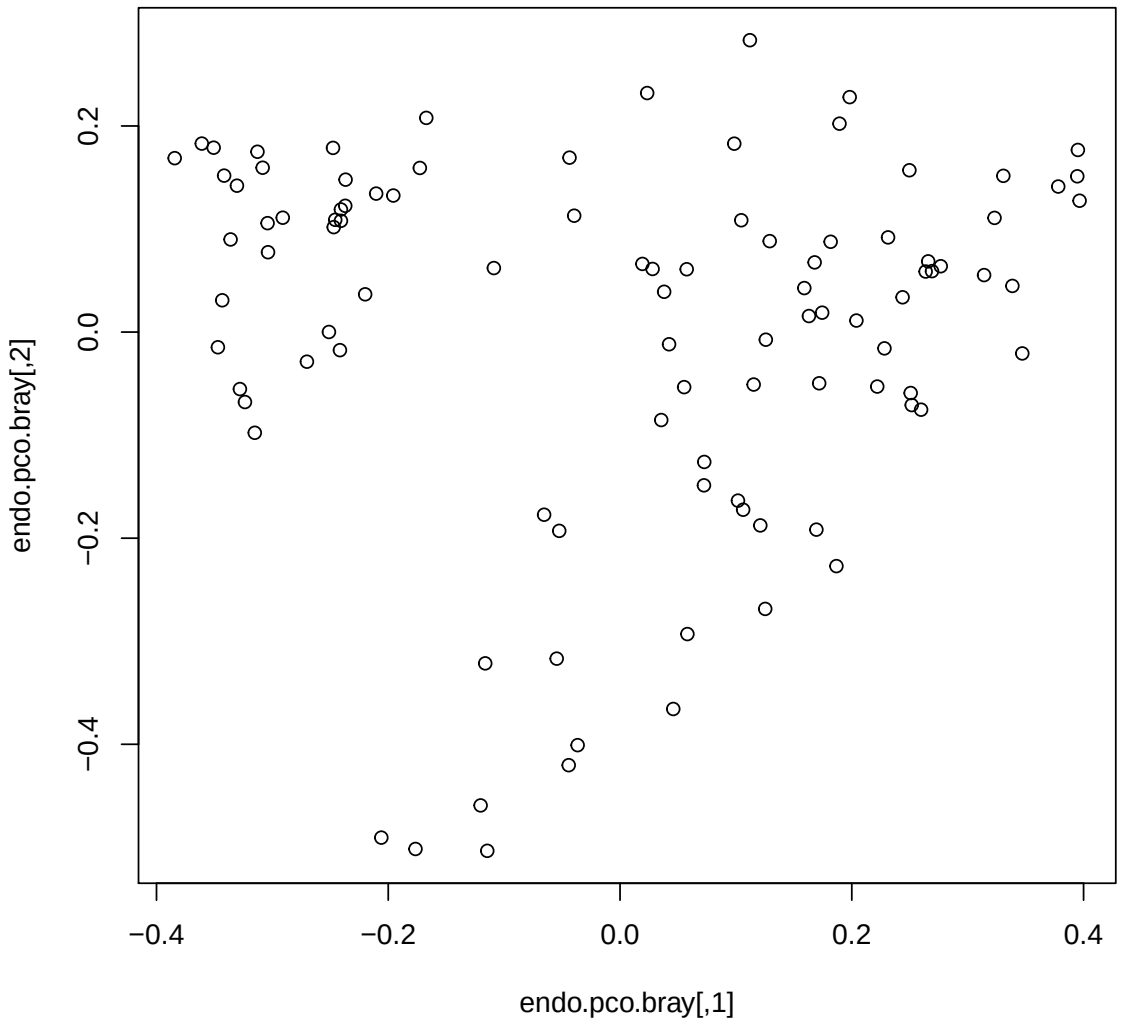
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  1.0000  1.0000  0.8688  1.0000  1.0000
```

```
# 87% of cells in the matrix equal zero (species is absent)
```

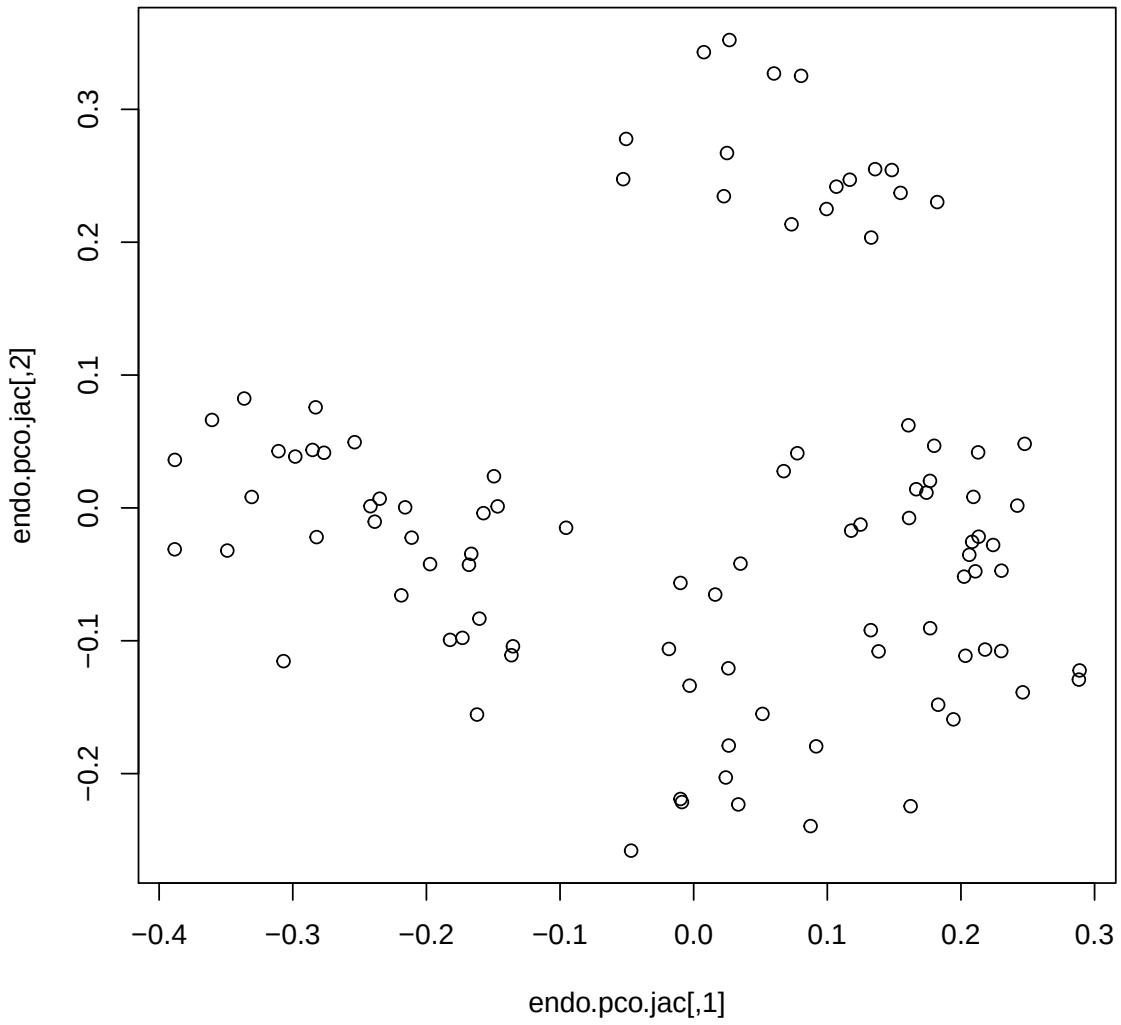
3. ♦ Perform PCoA on these data, using the 'hellinger' method for the `decostand` function and 'bray' method for the `vegdist()` function, and plot the results. See Section **??** for help, if necessary. Repeat as before but use the `binary` argument in the `vegdist` function to convert the matrix to a 'presence/absence' matrix.

```
# read in endophyte community data
endo<-read.csv('endophytes.csv')

# PCoA with Bray-Curtis dissimilarities
endo.pco.bray<-wcmdscale(vegdist(endo,method='bray'))
```

```
# plot PCoA result
plot(endo.pco.bray)
```




```
# PCoA with Jaccard index (species presence / absence)
endo.pco.jac<-wcmdscale(vegdist(endo,method='jaccard',binary=T))
# plot PCoA result
plot(endo.pco.jac)
```



1.5.2 Analysis of Structure 1: two-table analysis

1.5.2.1 Endophyte data

1. ♦ Look at the help page for the `capscale` function. Use `capscale` to perform distance-based RDA (constrained PCoA) using the continuous variables in 'endophytes_env.csv' (percentC, percentN, CNratio) as predictors, then plot the results. First use the `envfit` function to determine which variables to include in db-RDA (Section **??**).

```
# First look at the help page with: ?capscale

# read in endophyte community data
endo<-read.csv('endophytes.csv')
# read in matrix of predictor variables
endo.env<-read.csv('endophytes_env.csv')
```

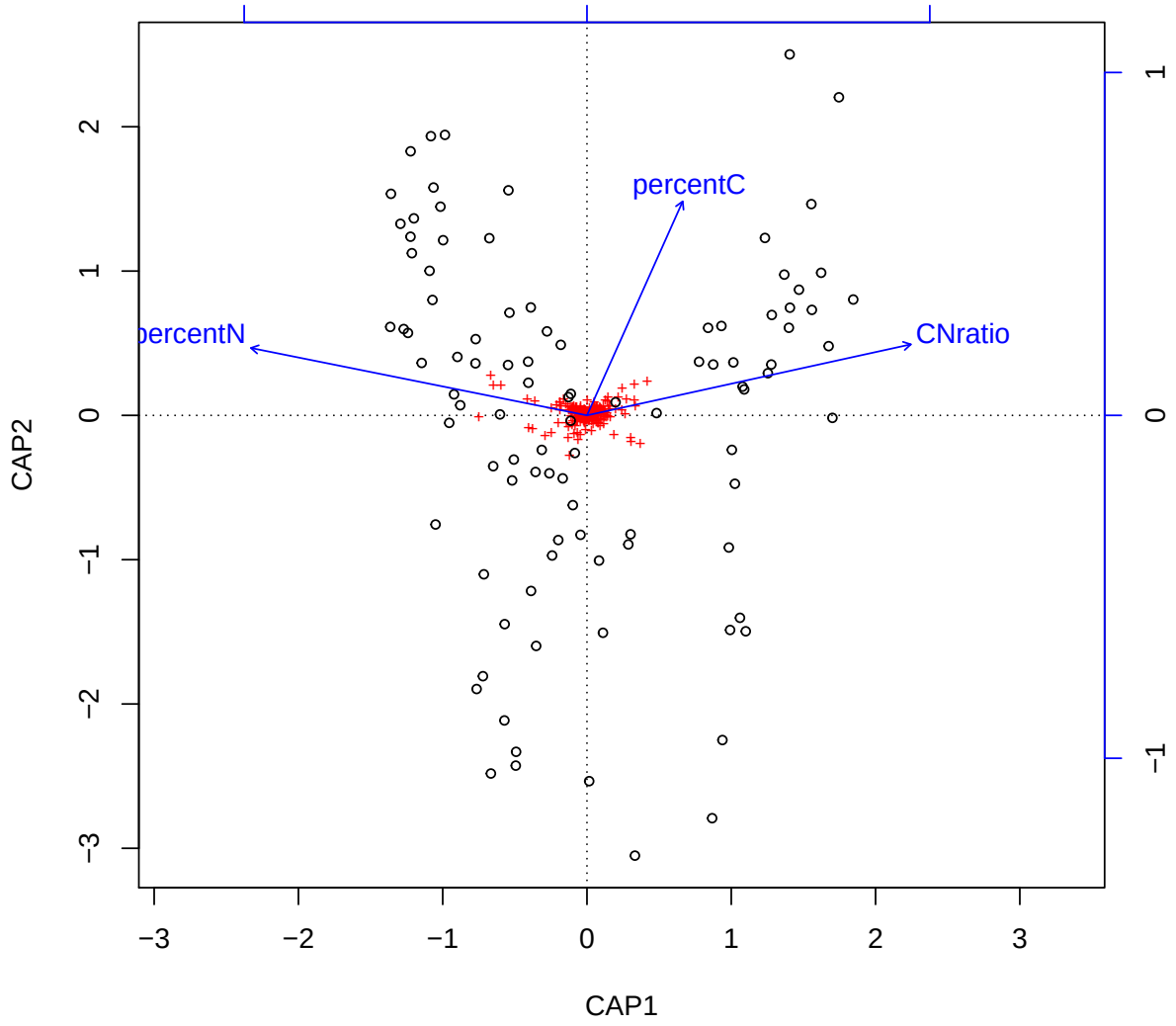
```r
# which environmental variables should be included as predictors?
# first perform PCoA on the community data (input is distance matrix)
endo.pcoa<-wcmdscale(vegdist(endo,method='bray'))
# then use envfit() to see which individual variables are associated with PCoA patterns
# use scale() to account for differences in variance among variables)
envfit(endo.pcoa, scale(endo.env[,3:5]))
```

```
##
## ***VECTORS
##
##              Dim1     Dim2     r2 Pr(>r)
## percentC -0.71763  0.69643 0.0637  0.027 *
## percentN  0.84421 -0.53601 0.5852  0.001 ***
## CNratio  -0.83489  0.55042 0.5352  0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
```

```r
# all three variable are significant, include all three in analysis

# perform db-RDA (aka CAP) using continuous environmental variables as predictors
endo.cap<-capscale(endo~percentC+percentN+CNratio,data=endo.env,distance='bray')
# look at results
endo.cap
```

```
## Call: capscale(formula = endo ~ percentC + percentN + CNratio,
## data = endo.env, distance = "bray")
##
##               Inertia Proportion Eigenvals Rank
## Total         32.6662     1.0000   33.3894
## Constrained    4.1124     0.1259    4.1268    3
## Unconstrained 28.5538     0.8741   29.2626   77
## Imaginary                          -0.7232   20
## Inertia is squared Bray distance
##
## Eigenvalues for constrained axes:
##   CAP1   CAP2   CAP3
## 3.0212 0.6235 0.4820
##
## Eigenvalues for unconstrained axes:
##  MDS1  MDS2  MDS3  MDS4  MDS5  MDS6  MDS7  MDS8
## 3.690 2.251 2.077 1.775 1.492 1.297 1.067 0.941
## (Showed only 8 of all 77 unconstrained eigenvalues)
```

```r
# 12 % of the variation is explained by C, N, and C:N,
# most of that variation is accounted for in one axis (CAP1)

# plot results
plot(endo.cap)
```




```
# N and C:N are strongly collinear,
# C is separated out along the second CAP axis
```

2. ♦ Repeat the analysis in the previous exercise but use the `ordistep` function to determine which variables to include in db-RDA.

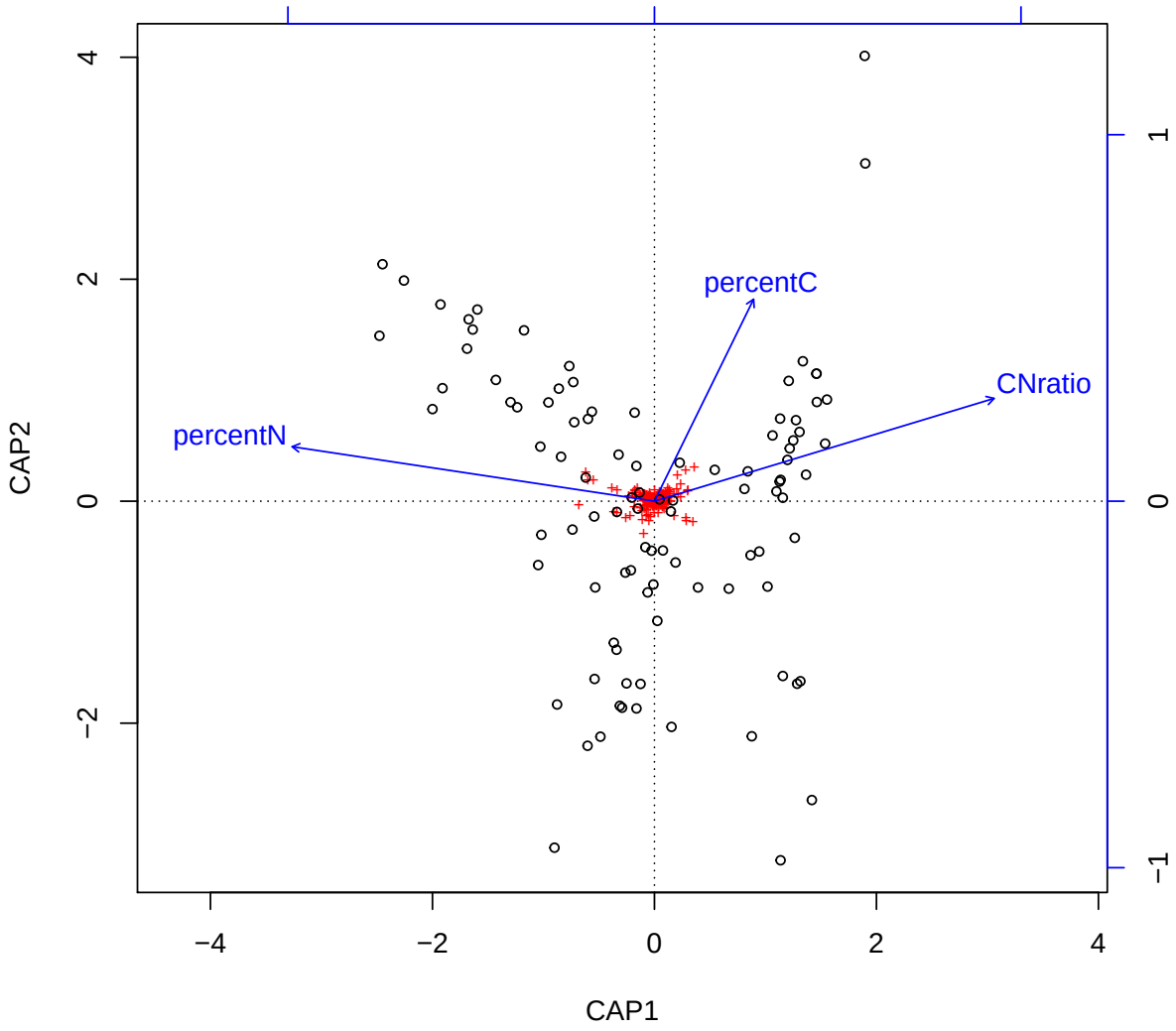
```
# First look at the help page with: ?capscale

# read in endophyte community data
endo<-read.csv('endophytes.csv')
# read in matrix of predictor variables
endo.env<-read.csv('endophytes_env.csv')
# scale continuous variables so variance standardised
endo.env.std <- decostand(endo.env[, 3:5], method='standardize')

# which environmental variables should be included as predictors?
# first perform CAP with each of the environmental variables
```

```r
endo.cap1<-capscale(endo ~ ., data=endo.env.std, method='bray')
# then perform CAP with no predictors (essentially PCoA, but using the 'capscale' function
# use scale() to account for differences in variance among variables)
endo.cap0<-capscale(endo ~ 1, data=endo.env.std, method='bray')


# perform forward and backward selection of explanatory variables
step.env <- ordistep(endo.cap0, scope=formula(endo.cap1))


##
## Start: endo ~ 1
##
##             Df    AIC      F Pr(>F)
## + percentN   1 356.98 7.0670  0.005 **
## + CNratio    1 357.63 6.3844  0.005 **
## + percentC   1 361.90 2.0160  0.020 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo ~ percentN
##
##             Df    AIC     F Pr(>F)
## - percentN   1 361.94 7.067  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##             Df    AIC      F Pr(>F)
## + CNratio    1 356.94 1.9930   0.01 **
## + percentC   1 357.13 1.8092   0.02 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo ~ percentN + CNratio
##
##             Df    AIC      F Pr(>F)
## - CNratio    1 356.98 1.9930  0.020 *
## - percentN   1 357.63 2.6397  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##             Df    AIC      F Pr(>F)
## + percentC   1 357.12 1.7588  0.035 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo ~ percentN + CNratio + percentC
##
##             Df    AIC      F Pr(>F)
## - percentC   1 356.94 1.7588  0.030 *
## - CNratio    1 357.13 1.9407  0.015 *
## - percentN   1 357.86 2.6623  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# look at the significant variables (all are significant)
step.env$anova
```

```
##            Df    AIC      F Pr(>F)
## + percentN  1 356.98 7.0670  0.005 **
## + CNratio   1 356.94 1.9930  0.010 **
## + percentC  1 357.12 1.7588  0.035 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# view ordination
plot(step.env)
```



1.5.3 Analysis of Structure 2: variation partitioning

1.5.3.1 Endophyte data

1. ⧫ Perform variation partitioning to determine whether leaf species, leaf chemistry, or sample type explains the most variation in fungal community composition.

```
# load the vegan library
library(vegan)

# read in tables containing species, and environmental variables
endo.spp <- read.csv('endophytes.csv')  # column names represent OTUs
endo.env <- read.csv('endophytes_env.csv')

dim(endo.spp)
```

```
## [1]  98 874
```

```
str(endo.env)
```

```
## 'data.frame': 98 obs. of  5 variables:
##  $ species : Factor w/ 9 levels "cladocalyx","crebra",..: 1 1 1 1 1 1 1 1 2 2 ...
##  $ type    : Factor w/ 2 levels "fresh","litter": 1 2 1 2 1 2 1 1 1 2 ...
##  $ percentC: num  51.3 53 53.9 54.2 55.4 ...
##  $ percentN: num  2.271 1.212 1.508 0.892 1.916 ...
##  $ CNratio : num  22.6 43.7 35.7 60.8 28.9 ...
```

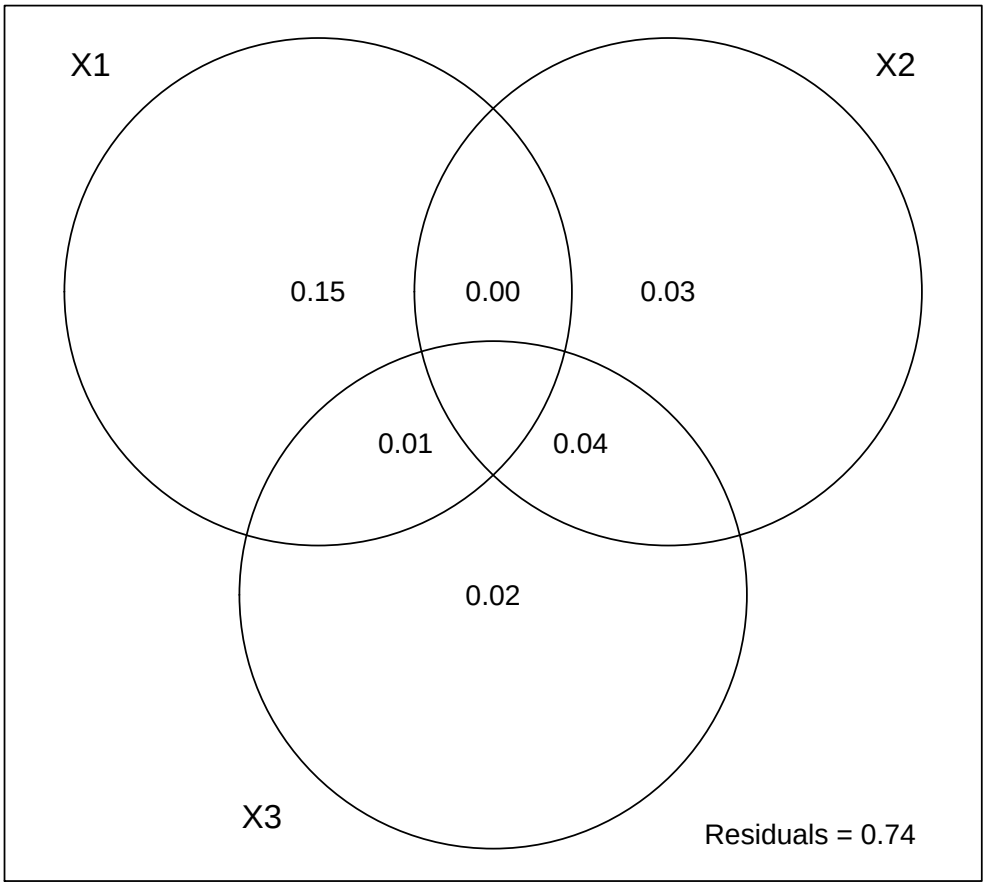
```
# select particular variables to proceed with (here we use both forward and backward selection but

# set up the analysis with all predictors
cap.env <- capscale(endo.spp ~ ., data=endo.env, distance='bray')

# set up the null cases with no predictors
mod0.env <- capscale(endo.spp ~ 1, data=endo.env, distance='bray')

# select variables in each predictor table
step.env <- ordistep(mod0.env, scope=formula(cap.env))
```

```
##
## Start: endo.spp ~ 1
##
##            Df    AIC       F Pr(>F)
## + species   8 331.21  3.5955  0.005 **
## + type      1 333.56 11.5116  0.005 **
## + percentN  1 335.46  9.4418  0.005 **
## + CNratio   1 336.03  8.8342  0.005 **
## + percentC  1 342.44  2.1946  0.010 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo.spp ~ species
##
##           Df    AIC      F Pr(>F)
## - species  8 342.66 3.5955  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
##            Df    AIC       F Pr(>F)
## + type      1 318.67 14.0724  0.005 **
## + percentN  1 322.85  9.8181  0.005 **
## + CNratio   1 323.67  8.9994  0.005 **
## + percentC  1 331.02  1.9932  0.030 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo.spp ~ species + type
##
##           Df    AIC      F Pr(>F)
## - type     1 331.21 14.072  0.005 **
## - species  8 333.56  4.075  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##            Df    AIC      F Pr(>F)
## + CNratio   1 317.88 2.5161  0.005 **
## + percentN  1 318.22 2.2050  0.005 **
## + percentC  1 319.80 0.7775  0.800
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo.spp ~ species + type + CNratio
##
##           Df    AIC      F Pr(>F)
## - CNratio  1 318.67 2.5161  0.005 **
## - type     1 323.67 7.1978  0.005 **
## - species  8 331.89 3.8967  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##            Df    AIC      F Pr(>F)
## + percentN  1 317.85 1.8011  0.015 *
## + percentC  1 319.01 0.7667  0.825
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo.spp ~ species + type + CNratio + percentN
##
##            Df    AIC      F Pr(>F)
## - percentN  1 317.88 1.8011  0.020 *
## - CNratio   1 318.22 2.1073  0.010 **
## - type      1 322.59 6.1214  0.005 **
## - species   8 331.83 3.8476  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##            Df    AIC      F Pr(>F)
## + percentC  1 318.97 0.7669  0.865

# species, tissue type, tissue CN ratio and N concentration predict variation in community composit
step.env
```

```
## Call: capscale(formula = endo.spp ~ species + type + CNratio +
## percentN, data = endo.env, distance = "bray")
##
##               Inertia Proportion Eigenvals Rank
## Total         32.6662     1.0000   33.3894
## Constrained   12.4050     0.3797   12.4449   11
## Unconstrained 20.2612     0.6203   20.9445   77
## Imaginary                          -0.7232   20
## Inertia is squared Bray distance
##
## Eigenvalues for constrained axes:
##  CAP1  CAP2  CAP3  CAP4  CAP5  CAP6  CAP7  CAP8  CAP9 CAP10 CAP11
## 4.071 2.249 1.698 0.923 0.847 0.746 0.594 0.452 0.373 0.305 0.186
##
## Eigenvalues for unconstrained axes:
##   MDS1   MDS2   MDS3   MDS4   MDS5   MDS6   MDS7   MDS8
## 2.3616 1.6386 1.2429 1.0608 1.0216 0.8897 0.7465 0.6718
## (Showed only 8 of all 77 unconstrained eigenvalues)

step.env$anova  # presents results in an ANOVA-like table

##            Df    AIC       F Pr(>F)
## + species   8 331.21  3.5955  0.005 **
## + type      1 318.67 14.0724  0.005 **
## + CNratio   1 317.88  2.5161  0.005 **
## + percentN  1 317.85  1.8011  0.015 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# partition variation among four predictor tables:
#   1) leaf species
#   2) leaf type (canopy/litter)
#   3) leaf chemistry
endo.var <- varpart(endo.spp,
                                      ~ species,
                                      ~ type,
                                      ~ CNratio + percentN, data=endo.env)
endo.var

##
## Partition of variance in RDA
##
## Call: varpart(Y = endo.spp, X = ~species, ~type, ~CNratio +
## percentN, data = endo.env)
##
## Explanatory tables:
## X1:  ~species
## X2:  ~type
## X3:  ~CNratio + percentN
##
## No. of explanatory tables: 3
## Total variation (SS): 39.768
##             Variance: 0.40998
## No. of observations: 98
##
```

```
## Partition table:
##                         Df R.square Adj.R.square Testable
## [a+d+f+g] = X1           8  0.22506      0.15540     TRUE
## [b+d+e+g] = X2           1  0.07982      0.07023     TRUE
## [c+e+f+g] = X3           2  0.08771      0.06850     TRUE
## [a+b+d+e+f+g] = X1+X2    9  0.30449      0.23335     TRUE
## [a+c+d+e+f+g] = X1+X3   10  0.30288      0.22276     TRUE
## [b+c+d+e+f+g] = X2+X3    3  0.13133      0.10360     TRUE
## [a+b+c+d+e+f+g] = All   11  0.34138      0.25714     TRUE
## Individual fractions
## [a] = X1 | X2+X3         8               0.15354     TRUE
## [b] = X2 | X1+X3         1               0.03438     TRUE
## [c] = X3 | X1+X2         2               0.02379     TRUE
## [d]                      0               0.00072    FALSE
## [e]                      0               0.04357    FALSE
## [f]                      0               0.00959    FALSE
## [g]                      0              -0.00844    FALSE
## [h] = Residuals                         0.74286    FALSE
## Controlling 1 table X
## [a+d] = X1 | X3          8               0.15426     TRUE
## [a+f] = X1 | X2          8               0.16312     TRUE
## [b+d] = X2 | X3          1               0.03510     TRUE
## [b+e] = X2 | X1          1               0.07795     TRUE
## [c+e] = X3 | X1          2               0.06736     TRUE
## [c+f] = X3 | X2          2               0.03337     TRUE
## ---
## Use function 'rda' to test significance of fractions of interest

plot(endo.var)
```



2. ♦ Use the geographic coordinates of each plot to estimate the contribution of space to variation in fungal community composition. Is this estimate greater than the variation partitioned to the measured leaf variables?

```
# read in table containing geographic distances
endo.dist <- read.csv('endophytes_dist.csv')

str(endo.dist)
```

```
## 'data.frame': 98 obs. of  2 variables:
##  $ x_coord: int  5 5 7 7 9 9 2 4 15 15 ...
##  $ y_coord: int  1 1 3 3 5 5 10 12 1 1 ...
```

```
# represent spatial patterns through PCNMs
endo.pcnm <- pcnm(dist(endo.dist))
# loadings for each PCNM axis can be extracted using scores()
str(scores(endo.pcnm))
```

```
##  num [1:98, 1:19] -0.0454 -0.0454 -0.051 -0.051 -0.0588 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:98] "1" "2" "3" "4" ...
##   ..$ : chr [1:19] "PCNM1" "PCNM2" "PCNM3" "PCNM4" ...

# select particular variables to proceed with (here we use both forward and backward selection but

# set up the analysis with all predictors
cap.pcnm <- capscale(endo.spp ~ ., data=as.data.frame(scores(endo.pcnm)), distance='bray')

# set up the null cases with no predictors
mod0.pcnm <- capscale(endo.spp ~ 1, data=as.data.frame(scores(endo.pcnm)), distance='bray')

# select variables in each predictor table
step.pcnm <- ordistep(mod0.pcnm, scope=formula(cap.pcnm))

##
## Start: endo.spp ~ 1
##
##          Df    AIC      F Pr(>F)
## + PCNM1   1 341.78 2.8617  0.005 **
## + PCNM6   1 342.31 2.3292  0.010 **
## + PCNM2   1 342.65 1.9885  0.015 *
## + PCNM14  1 342.84 1.7996  0.025 *
## + PCNM4   1 342.79 1.8443  0.030 *
## + PCNM3   1 342.93 1.7093  0.040 *
## + PCNM11  1 343.23 1.4094  0.095 .
## + PCNM16  1 343.47 1.1677  0.240
## + PCNM8   1 343.71 0.9339  0.515
## + PCNM15  1 343.70 0.9419  0.580
## + PCNM5   1 343.80 0.8421  0.645
## + PCNM9   1 343.80 0.8439  0.695
## + PCNM12  1 343.81 0.8326  0.720
## + PCNM17  1 343.92 0.7277  0.845
## + PCNM10  1 343.96 0.6878  0.865
## + PCNM19  1 343.95 0.6901  0.885
## + PCNM7   1 344.07 0.5798  0.950
## + PCNM13  1 344.15 0.4980  0.995
## + PCNM18  1 344.35 0.2981  1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo.spp ~ PCNM1
##
##         Df    AIC      F Pr(>F)
## - PCNM1  1 342.66 2.8617  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##          Df    AIC      F Pr(>F)
## + PCNM6   1 341.36 2.3753  0.005 **
## + PCNM2   1 341.71 2.0277  0.010 **
## + PCNM4   1 341.86 1.8805  0.015 *
## + PCNM14  1 341.90 1.8350  0.015 *
```

```
## + PCNM3   1 342.00 1.7428  0.050 *
## + PCNM11  1 342.31 1.4370  0.075 .
## + PCNM16  1 342.56 1.1904  0.210
## + PCNM15  1 342.79 0.9601  0.455
## + PCNM8   1 342.80 0.9520  0.490
## + PCNM9   1 342.89 0.8603  0.625
## + PCNM5   1 342.90 0.8584  0.655
## + PCNM12  1 342.91 0.8487  0.665
## + PCNM19  1 343.05 0.7034  0.835
## + PCNM17  1 343.02 0.7417  0.855
## + PCNM10  1 343.06 0.7011  0.885
## + PCNM7   1 343.17 0.5910  0.970
## + PCNM13  1 343.26 0.5075  0.990
## + PCNM18  1 343.46 0.3038  0.995
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo.spp ~ PCNM1 + PCNM6
##
##         Df    AIC      F Pr(>F)
## - PCNM6  1 341.78 2.3753  0.005 **
## - PCNM1  1 342.31 2.9027  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##          Df    AIC      F Pr(>F)
## + PCNM2   1 341.24 2.0576  0.010 **
## + PCNM4   1 341.39 1.9082  0.020 *
## + PCNM14  1 341.44 1.8619  0.020 *
## + PCNM3   1 341.53 1.7684  0.045 *
## + PCNM11  1 341.85 1.4579  0.085 .
## + PCNM16  1 342.11 1.2077  0.235
## + PCNM15  1 342.35 0.9740  0.515
## + PCNM8   1 342.36 0.9658  0.530
## + PCNM5   1 342.45 0.8708  0.550
## + PCNM9   1 342.45 0.8727  0.645
## + PCNM12  1 342.46 0.8610  0.705
## + PCNM17  1 342.58 0.7524  0.820
## + PCNM19  1 342.62 0.7135  0.840
## + PCNM10  1 342.62 0.7112  0.890
## + PCNM7   1 342.73 0.5995  0.970
## + PCNM13  1 342.82 0.5148  0.985
## + PCNM18  1 343.04 0.3081  1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo.spp ~ PCNM1 + PCNM6 + PCNM2
##
##         Df    AIC      F Pr(>F)
## - PCNM2  1 341.36 2.0576  0.010 **
## - PCNM6  1 341.71 2.4018  0.005 **
## - PCNM1  1 342.25 2.9350  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
##          Df    AIC      F Pr(>F)
## + PCNM4   1 341.22 1.9301  0.010 **
## + PCNM14  1 341.27 1.8833  0.015 *
## + PCNM3   1 341.37 1.7886  0.030 *
## + PCNM11  1 341.69 1.4745  0.070 .
## + PCNM16  1 341.96 1.2214  0.210
## + PCNM15  1 342.20 0.9850  0.445
## + PCNM8   1 342.21 0.9766  0.465
## + PCNM12  1 342.32 0.8706  0.610
## + PCNM5   1 342.31 0.8806  0.625
## + PCNM9   1 342.31 0.8825  0.645
## + PCNM17  1 342.44 0.7608  0.750
## + PCNM19  1 342.48 0.7215  0.830
## + PCNM10  1 342.48 0.7192  0.865
## + PCNM7   1 342.60 0.6062  0.950
## + PCNM13  1 342.69 0.5206  0.975
## + PCNM18  1 342.91 0.3116  1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo.spp ~ PCNM1 + PCNM6 + PCNM2 + PCNM4
##
##         Df    AIC      F Pr(>F)
## - PCNM4  1 341.24 1.9301  0.015 *
## - PCNM2  1 341.39 2.0779  0.005 **
## - PCNM6  1 341.75 2.4255  0.005 **
## - PCNM1  1 342.30 2.9640  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##          Df    AIC      F Pr(>F)
## + PCNM3   1 341.32 1.8068  0.015 *
## + PCNM14  1 341.22 1.9025  0.020 *
## + PCNM11  1 341.65 1.4894  0.085 .
## + PCNM16  1 341.92 1.2337  0.150
## + PCNM8   1 342.18 0.9864  0.380
## + PCNM15  1 342.17 0.9948  0.385
## + PCNM12  1 342.29 0.8793  0.535
## + PCNM9   1 342.28 0.8913  0.585
## + PCNM5   1 342.28 0.8893  0.590
## + PCNM17  1 342.41 0.7684  0.765
## + PCNM19  1 342.45 0.7287  0.845
## + PCNM10  1 342.45 0.7263  0.875
## + PCNM7   1 342.57 0.6122  0.940
## + PCNM13  1 342.66 0.5257  0.970
## + PCNM18  1 342.89 0.3146  1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo.spp ~ PCNM1 + PCNM6 + PCNM2 + PCNM4 + PCNM3
##
##         Df    AIC      F Pr(>F)
## - PCNM3  1 341.22 1.8068  0.015 *
```

```
## - PCNM2  1 341.52 2.0960  0.015 *
## - PCNM4  1 341.37 1.9468  0.010 **
## - PCNM6  1 341.89 2.4466  0.005 **
## - PCNM1  1 342.45 2.9898  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##          Df    AIC      F Pr(>F)
## + PCNM14  1 341.27 1.9195  0.010 **
## + PCNM11  1 341.71 1.5026  0.100 .
## + PCNM16  1 341.99 1.2445  0.185
## + PCNM8   1 342.25 0.9950  0.395
## + PCNM15  1 342.24 1.0035  0.480
## + PCNM9   1 342.35 0.8991  0.520
## + PCNM5   1 342.35 0.8971  0.615
## + PCNM12  1 342.37 0.8870  0.620
## + PCNM17  1 342.49 0.7751  0.755
## + PCNM10  1 342.53 0.7326  0.800
## + PCNM19  1 342.53 0.7350  0.850
## + PCNM7   1 342.65 0.6175  0.925
## + PCNM13  1 342.75 0.5303  0.960
## + PCNM18  1 342.98 0.3173  1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo.spp ~ PCNM1 + PCNM6 + PCNM2 + PCNM4 + PCNM3 + PCNM14
##
##          Df    AIC      F Pr(>F)
## - PCNM3   1 341.22 1.8249  0.030 *
## - PCNM4   1 341.37 1.9663  0.020 *
## - PCNM14  1 341.32 1.9195  0.010 **
## - PCNM2   1 341.52 2.1169  0.010 **
## - PCNM6   1 341.90 2.4710  0.005 **
## - PCNM1   1 342.47 3.0196  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##          Df    AIC      F Pr(>F)
## + PCNM11  1 341.63 1.5180  0.055 .
## + PCNM16  1 341.91 1.2572  0.180
## + PCNM15  1 342.17 1.0137  0.380
## + PCNM8   1 342.18 1.0051  0.465
## + PCNM5   1 342.29 0.9062  0.530
## + PCNM12  1 342.30 0.8960  0.580
## + PCNM9   1 342.29 0.9082  0.605
## + PCNM17  1 342.42 0.7829  0.755
## + PCNM10  1 342.47 0.7400  0.785
## + PCNM19  1 342.47 0.7424  0.855
## + PCNM7   1 342.59 0.6237  0.920
## + PCNM13  1 342.69 0.5356  0.985
## + PCNM18  1 342.92 0.3205  1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# only six/seven of the PCNM axes appear to predict variation in community composition
# significance of PCNM11 varies each time because it is based on permutations
step.pcnm
```

```
## Call: capscale(formula = endo.spp ~ PCNM1 + PCNM6 + PCNM2 + PCNM4
## + PCNM3 + PCNM14, data = as.data.frame(scores(endo.pcnm)),
## distance = "bray")
##
##               Inertia Proportion Eigenvals Rank
## Total         32.6662     1.0000   33.3894
## Constrained    4.1705     0.1277    4.2013    6
## Unconstrained 28.4957     0.8723   29.1881   77
## Imaginary                          -0.7232   20
## Inertia is squared Bray distance
##
## Eigenvalues for constrained axes:
##   CAP1   CAP2   CAP3   CAP4   CAP5   CAP6
## 1.4622 1.0641 0.6444 0.4342 0.3301 0.2663
##
## Eigenvalues for unconstrained axes:
##  MDS1  MDS2  MDS3  MDS4  MDS5  MDS6  MDS7  MDS8
## 4.633 2.863 1.970 1.664 1.479 1.184 0.972 0.906
## (Showed only 8 of all 77 unconstrained eigenvalues)
```

```
step.pcnm$anova  # presents results in an ANOVA-like table
```

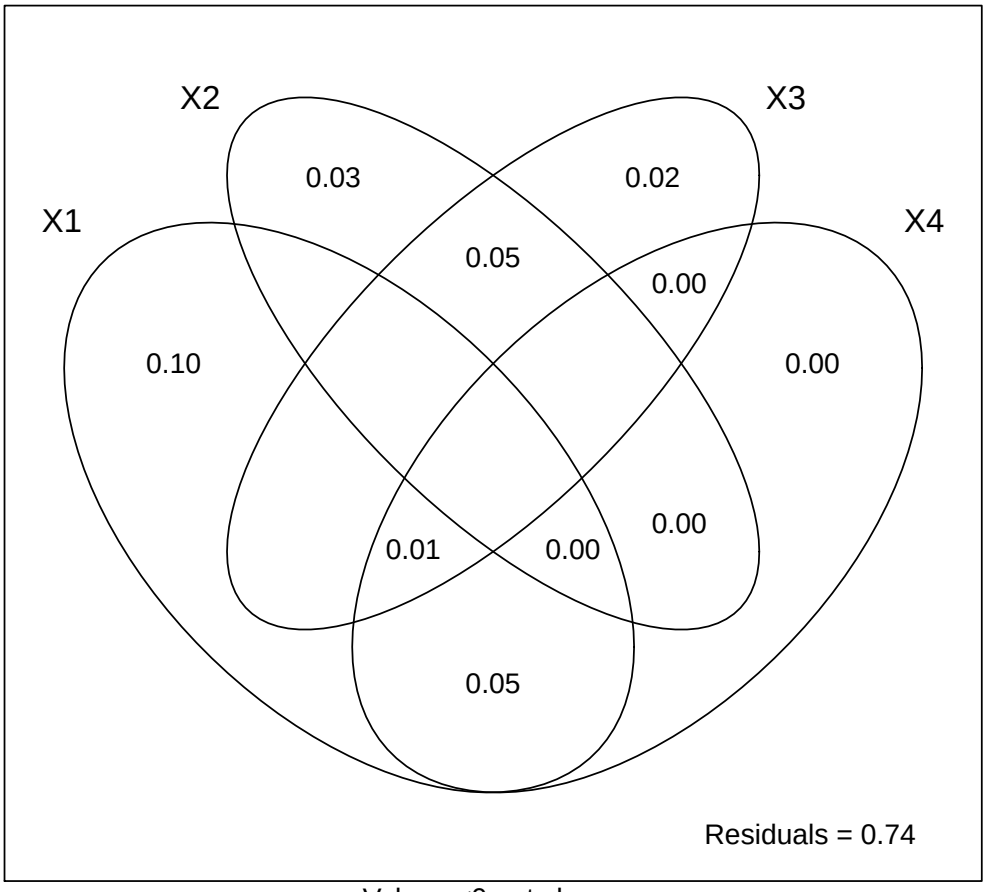
```
##          Df    AIC      F Pr(>F)
## + PCNM1   1 341.78 2.8617  0.005 **
## + PCNM6   1 341.36 2.3753  0.005 **
## + PCNM2   1 341.24 2.0576  0.010 **
## + PCNM4   1 341.22 1.9301  0.010 **
## + PCNM3   1 341.32 1.8068  0.015 *
## + PCNM14  1 341.27 1.9195  0.010 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# create pcnm table with only significant axes
endo.pcnm.sub <- scores(endo.pcnm,
                                      choices=c(1:4, 6, 11, 14))

# partition variation among four predictor tables:
#   1) leaf species
#   2) leaf type (canopy/litter)
#   3) leaf chemistry
#   4) spatial gradients
endo.var <- varpart(endo.spp,
                                      ~ species,
                                      ~ type,
                                      ~ CNratio + percentN,
                                      endo.pcnm.sub, data=endo.env)
endo.var
```

```
##
## Partition of variance in RDA
##
## Call: varpart(Y = endo.spp, X = ~species, ~type, ~CNratio +
```

```
## percentN, endo.pcnm.sub, data = endo.env)
##
## Explanatory tables:
## X1:  ~species
## X2:  ~type
## X3:  ~CNratio + percentN
## X4:  endo.pcnm.sub
##
## No. of explanatory tables: 4
## Total variation (SS): 39.768
##             Variance: 0.40998
## No. of observations: 98
##
## Partition table:
##                               Df R.square Adj.R.square Testable
## [aeghklno] = X1                8  0.22506      0.15540     TRUE
## [befiklmo] = X2                1  0.07982      0.07023     TRUE
## [cfgjlmno] = X3                2  0.08771      0.06850     TRUE
## [dhijkmno] = X4                7  0.12999      0.06232     TRUE
## [abefghiklmno] = X1+X2         9  0.30449      0.23335     TRUE
## [acefghjklmno] = X1+X3        10  0.30288      0.22276     TRUE
## [adeghijklmno] = X1+X4        15  0.28494      0.15414     TRUE
## [bcefgijklmno] = X2+X3         3  0.13133      0.10360     TRUE
## [bdefhijklmno] = X2+X4         8  0.20846      0.13732     TRUE
## [cdfghijklmno] = X3+X4         9  0.21086      0.13016     TRUE
## [abcefghijklmno] = X1+X2+X3 11  0.34138      0.25714     TRUE
## [abdefghijklmno] = X1+X2+X4 16  0.36195      0.23592     TRUE
## [acdefghijklmno] = X1+X3+X4 17  0.36150      0.22582     TRUE
## [bcdefghijklmno] = X2+X3+X4 10  0.24551      0.15878     TRUE
## [abcdefghijklmno] = All     18  0.39623      0.25866     TRUE
## Individual fractions
## [a] = X1 | X2+X3+X4            8               0.09988     TRUE
## [b] = X2 | X1+X3+X4            1               0.03284     TRUE
## [c] = X3 | X1+X2+X4            2               0.02274     TRUE
## [d] = X4 | X1+X2+X3            7               0.00152     TRUE
## [e]                            0              -0.00422    FALSE
## [f]                            0               0.04894    FALSE
## [g]                            0              -0.00128    FALSE
## [h]                            0               0.05366    FALSE
## [i]                            0               0.00154    FALSE
## [j]                            0               0.00104    FALSE
## [k]                            0               0.00493    FALSE
## [l]                            0              -0.00257    FALSE
## [m]                            0              -0.00537    FALSE
## [n]                            0               0.01086    FALSE
## [o]                            0              -0.00587    FALSE
## [p] = Residuals                0               0.74134    FALSE
## Controlling 2 tables X
## [ae] = X1 | X3+X4              8               0.09566     TRUE
## [ag] = X1 | X2+X4              8               0.09860     TRUE
## [ah] = X1 | X2+X3              8               0.15354     TRUE
## [be] = X2 | X3+X4              1               0.02862     TRUE
## [bf] = X2 | X1+X4              1               0.08178     TRUE
```

```
## [bi] = X2 | X1+X3            1               0.03438     TRUE
## [cf] = X3 | X1+X4            2               0.07168     TRUE
## [cg] = X3 | X2+X4            2               0.02147     TRUE
## [cj] = X3 | X1+X2            2               0.02379     TRUE
## [dh] = X4 | X2+X3            7               0.05518     TRUE
## [di] = X4 | X1+X3            7               0.00307     TRUE
## [dj] = X4 | X1+X2            7               0.00257     TRUE
## Controlling 1 table X
## [aghn] = X1 | X2             8               0.16312     TRUE
## [aehk] = X1 | X3             8               0.15426     TRUE
## [aegl] = X1 | X4             8               0.09182     TRUE
## [bfim] = X2 | X1             1               0.07795     TRUE
## [beik] = X2 | X3             1               0.03510     TRUE
## [befl] = X2 | X4             1               0.07500     TRUE
## [cfjm] = X3 | X1             2               0.06736     TRUE
## [cgjn] = X3 | X2             2               0.03337     TRUE
## [cfgl] = X3 | X4             2               0.06784     TRUE
## [dijm] = X4 | X1             7              -0.00126     TRUE
## [dhjn] = X4 | X2             7               0.06708     TRUE
## [dhik] = X4 | X3             7               0.06166     TRUE
## ---
## Use function 'rda' to test significance of fractions of interest

plot(endo.var)
```



Values <0 not shown

3. ♦ Test the significance of each individual partition.

```
# significance of partition X1
anova(rda(endo.spp      ~ species + Condition(endo.env$type) +
          + Condition(endo.env$CNratio) + Condition(endo.env$percentN)
          + Condition(endo.pcnm.sub),
          data=endo.env))
```

```
## Permutation test for rda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = endo.spp ~ species + Condition(endo.env$type) + +Condition(endo.env$CNratio
##          Df Variance      F Pr(>F)
## Model     8 0.061795 2.4652  0.001 ***
## Residual 79 0.247536
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# significance of partition X2
anova(rda(endo.spp      ~ type + Condition(endo.env$species) +
          + Condition(endo.env$CNratio) + Condition(endo.env$percentN)
          + Condition(endo.pcnm.sub),
          data=endo.env))
```

```
## Permutation test for rda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = endo.spp ~ type + Condition(endo.env$species) + +Condition(endo.env$CNratio
##          Df Variance      F Pr(>F)
## Model     1 0.014238 4.5438  0.001 ***
## Residual 79 0.247536
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# significance of partition X3
anova(rda(endo.spp      ~ CNratio + percentN
          + Condition(endo.env$species) + Condition(endo.env$type)
          + Condition(endo.pcnm.sub),
          data=endo.env))
```

```
## Permutation test for rda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = endo.spp ~ CNratio + percentN + Condition(endo.env$species) + Condition(end
##          Df Variance      F Pr(>F)
## Model     2 0.014053 2.2425  0.001 ***
## Residual 79 0.247536
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# significance of partition X4
anova(rda(endo.spp      ~ endo.pcnm.sub
          + Condition(endo.env$species) + Condition(endo.env$type)
          + Condition(endo.env$CNratio) + Condition(endo.env$percentN)))
```

```
## Permutation test for rda under reduced model
## Permutation: free
## Number of permutations: 999
##
## Model: rda(formula = endo.spp ~ endo.pcnm.sub + Condition(endo.env$species) + Condition(endo.env
##          Df Variance      F Pr(>F)
## Model     7 0.022487 1.0252  0.417
## Residual 79 0.247536
```

4. ♦ Generate dummy variables (using 'dudi.hillsmith') for each of the levels of 'species' and check whether there are particular leaf species that explain variation in fungal community composition.

```
# load library
library(ade4)

# generate new table containing one column for each species
endo.leafspp <- dudi.hillsmith(endo.env[['species']], scannf=F, nf=2)$tab
```

```r
# set up the analysis with all predictors
cap.spp <- capscale(endo.spp ~ ., data=endo.leafspp, distance='bray')

# set up the null cases with no predictors
mod0.spp <- capscale(endo.spp ~ 1, data=endo.leafspp, distance='bray')

# select variables in each predictor table
step.env <- ordistep(mod0.spp, scope=formula(cap.spp))
```

```
##
## Start: endo.spp ~ 1
##
##                   Df    AIC      F Pr(>F)
## + df.grandis       1 340.66 3.9924  0.005 **
## + df.globulus      1 340.68 3.9748  0.005 **
## + df.melliodora    1 341.14 3.5052  0.005 **
## + df.tereticornis  1 341.50 3.1431  0.005 **
## + df.dunnii        1 341.63 3.0123  0.005 **
## + df.cladocalyx    1 341.84 2.8031  0.005 **
## + df.crebra        1 342.17 2.4652  0.005 **
## + df.sideroxylon   1 342.36 2.2733  0.010 **
## + df.saligna       1 342.53 2.1060  0.020 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo.spp ~ df.grandis
##
##              Df    AIC      F Pr(>F)
## - df.grandis  1 342.66 3.9924  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                   Df    AIC      F Pr(>F)
## + df.globulus      1 338.76 3.8604  0.005 **
## + df.dunnii        1 339.02 3.5966  0.005 **
## + df.melliodora    1 339.50 3.1189  0.005 **
## + df.tereticornis  1 339.70 2.9197  0.005 **
## + df.cladocalyx    1 339.86 2.7602  0.005 **
## + df.crebra        1 340.08 2.5343  0.005 **
## + df.saligna       1 339.92 2.6988  0.010 **
## + df.sideroxylon   1 340.24 2.3739  0.015 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo.spp ~ df.grandis + df.globulus
##
##                Df    AIC      F Pr(>F)
## - df.globulus   1 340.66 3.8604  0.005 **
## - df.grandis    1 340.68 3.8778  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                   Df    AIC      F Pr(>F)
## + df.melliodora    1 337.25 3.4244  0.005 **
```

```
## + df.dunnii        1 337.26 3.4222  0.005 **
## + df.cladocalyx    1 337.61 3.0734  0.005 **
## + df.tereticornis  1 337.89 2.7890  0.005 **
## + df.crebra        1 337.96 2.7260  0.005 **
## + df.saligna       1 337.99 2.6975  0.005 **
## + df.sideroxylon   1 338.36 2.3325  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo.spp ~ df.grandis + df.globulus + df.melliodora
##
##                 Df    AIC      F Pr(>F)
## - df.grandis     1 338.74 3.4006  0.005 **
## - df.melliodora  1 338.76 3.4244  0.005 **
## - df.globulus    1 339.50 4.1606  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                   Df    AIC      F Pr(>F)
## + df.tereticornis  1 335.73 3.4078  0.005 **
## + df.cladocalyx    1 335.97 3.1703  0.005 **
## + df.dunnii        1 336.13 3.0155  0.005 **
## + df.crebra        1 336.32 2.8289  0.005 **
## + df.saligna       1 336.65 2.5051  0.005 **
## + df.sideroxylon   1 336.71 2.4470  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo.spp ~ df.grandis + df.globulus + df.melliodora + df.tereticornis
##
##                   Df    AIC      F Pr(>F)
## - df.grandis       1 336.79 2.9487  0.005 **
## - df.tereticornis  1 337.25 3.4078  0.005 **
## - df.melliodora    1 337.89 4.0407  0.005 **
## - df.globulus      1 337.99 4.1322  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                  Df    AIC      F Pr(>F)
## + df.cladocalyx   1 334.14 3.4248  0.005 **
## + df.dunnii       1 334.55 3.0311  0.005 **
## + df.crebra       1 334.78 2.8051  0.005 **
## + df.sideroxylon  1 335.00 2.5934  0.005 **
## + df.saligna      1 335.28 2.3284  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo.spp ~ df.grandis + df.globulus + df.melliodora + df.tereticornis +      df.cladocalyx
##
##                   Df    AIC      F Pr(>F)
## - df.grandis       1 334.97 2.6868  0.005 **
## - df.cladocalyx    1 335.73 3.4248  0.005 **
## - df.tereticornis  1 335.97 3.6605  0.005 **
## - df.melliodora    1 336.60 4.2766  0.005 **
```

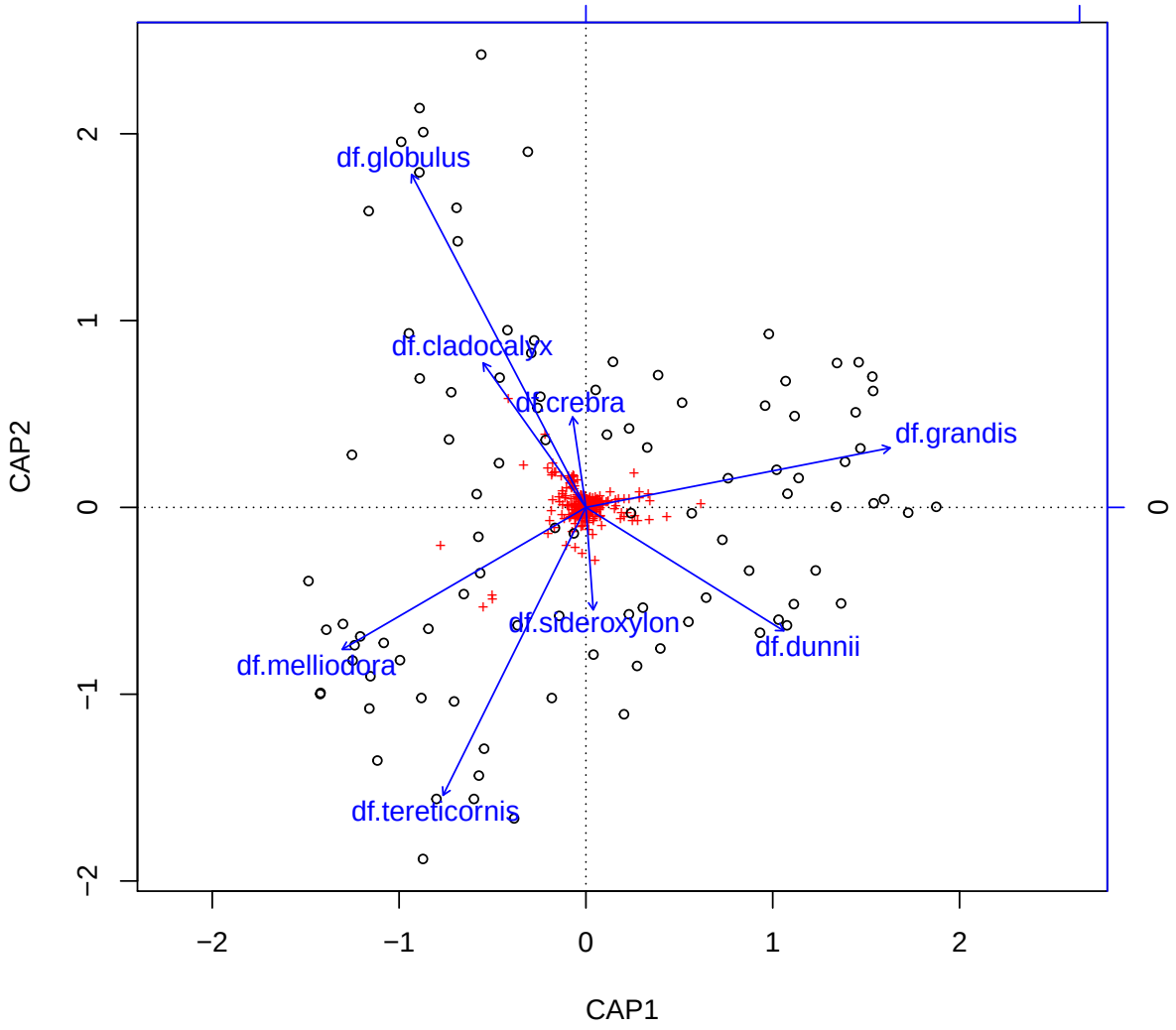
```
## - df.globulus      1 336.94 4.6178  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                  Df    AIC      F Pr(>F)
## + df.crebra       1 332.85 3.1067  0.005 **
## + df.sideroxylon  1 333.19 2.7889  0.005 **
## + df.dunnii       1 333.34 2.6381  0.005 **
## + df.saligna      1 333.63 2.3657  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo.spp ~ df.grandis + df.globulus + df.melliodora + df.tereticornis +      df.cladocalyx
##
##                   Df    AIC      F Pr(>F)
## - df.grandis       1 333.32 2.3140  0.005 **
## - df.crebra        1 334.14 3.1067  0.005 **
## - df.cladocalyx    1 334.78 3.7219  0.005 **
## - df.tereticornis  1 334.79 3.7296  0.005 **
## - df.melliodora    1 335.73 4.6400  0.005 **
## - df.globulus      1 336.20 5.0973  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                  Df    AIC      F Pr(>F)
## + df.sideroxylon  1 331.94 2.7177  0.005 **
## + df.dunnii       1 332.08 2.5858  0.005 **
## + df.saligna      1 332.21 2.4620  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo.spp ~ df.grandis + df.globulus + df.melliodora + df.tereticornis +      df.cladocalyx
##
##                   Df    AIC      F Pr(>F)
## - df.grandis       1 331.85 1.7755  0.025 *
## - df.sideroxylon   1 332.85 2.7177  0.015 *
## - df.crebra        1 333.19 3.0319  0.005 **
## - df.tereticornis  1 334.18 3.9783  0.005 **
## - df.cladocalyx    1 334.21 4.0052  0.005 **
## - df.melliodora    1 335.39 5.1467  0.005 **
## - df.globulus      1 335.45 5.2104  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##               Df    AIC      F Pr(>F)
## + df.dunnii    1 331.21 2.5117  0.005 **
## + df.saligna   1 331.21 2.5117  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: endo.spp ~ df.grandis + df.globulus + df.melliodora + df.tereticornis +      df.cladocalyx
##
##                   Df    AIC      F Pr(>F)
## - df.grandis       1 330.99 1.6267  0.015 *
```

```
## - df.sideroxylon   1 332.08 2.6421  0.010 **
## - df.dunnii        1 331.94 2.5117  0.005 **
## - df.crebra        1 332.24 2.7915  0.005 **
## - df.cladocalyx    1 332.67 3.1971  0.005 **
## - df.tereticornis  1 333.20 3.7001  0.005 **
## - df.globulus      1 333.56 4.0377  0.005 **
## - df.melliodora    1 333.57 4.0435  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##              Df    AIC F Pr(>F)
## + df.saligna  0 331.21
```

```
# look at ordistep result
step.env$anova  # presents results in an ANOVA-like table
```

```
##                   Df    AIC      F Pr(>F)
## + df.grandis       1 340.66 3.9924  0.005 **
## + df.globulus      1 338.76 3.8604  0.005 **
## + df.melliodora    1 337.25 3.4244  0.005 **
## + df.tereticornis  1 335.73 3.4078  0.005 **
## + df.cladocalyx    1 334.14 3.4248  0.005 **
## + df.crebra        1 332.85 3.1067  0.005 **
## + df.sideroxylon   1 331.94 2.7177  0.005 **
## + df.dunnii        1 331.21 2.5117  0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(step.env)
```



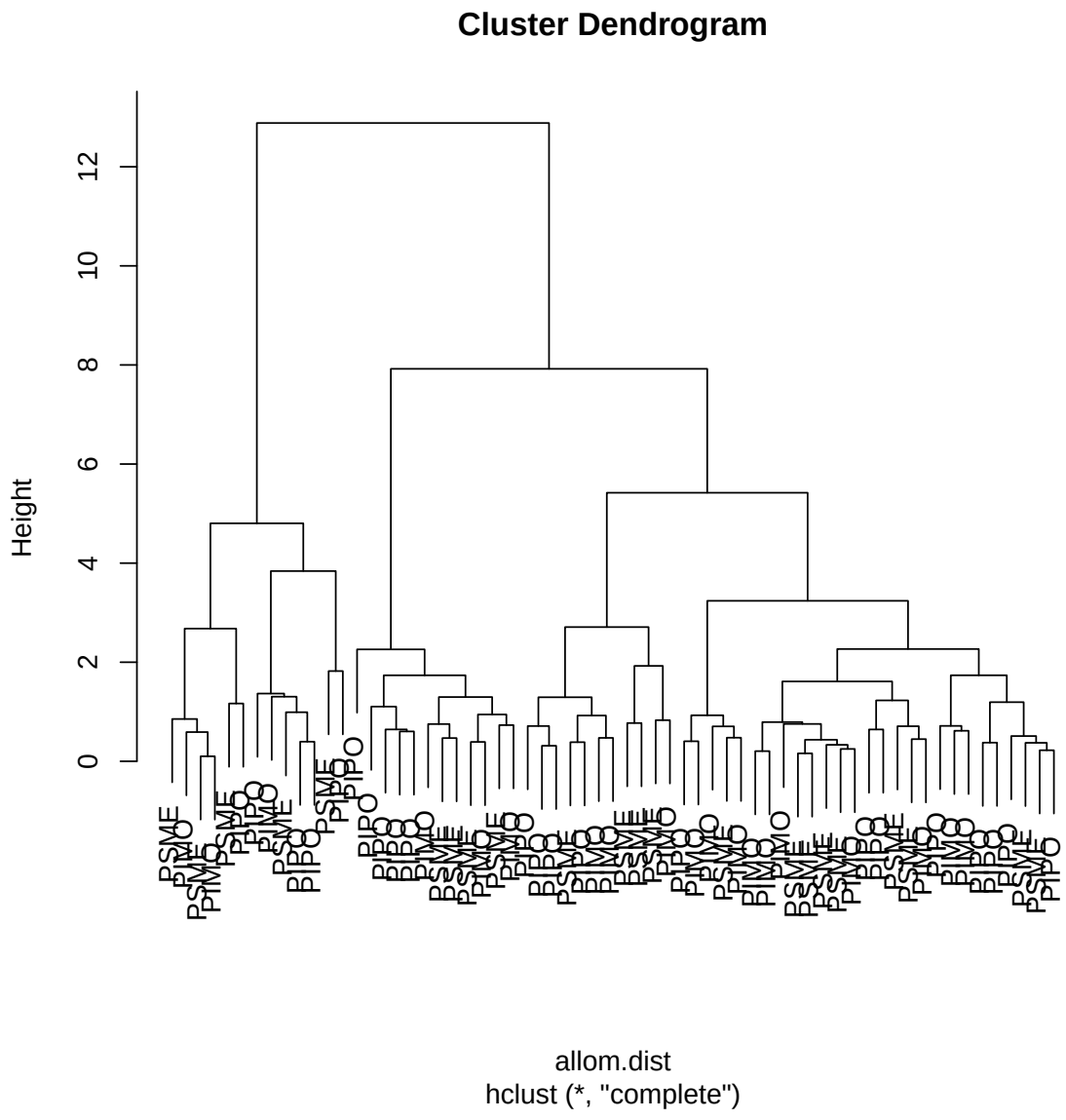
1.5.4 Analysis of Structure 3: 'experimental' systems

1.5.4.1 Allometry data

1. ♦ For the allometry data, plot a dendrogram of multivariate distances (euclidean) among individual trees based on the four growth parameters, labelling the tips of the dendrogram with the species level. Use ANOSIM and PERMANOVA to test the hypothesis that clusters can be explained by interspecific variation. See Section **??** for help, if necessary.

```
library(vegan)

allom<-read.csv('Allometry.csv')

# log-transform the data, then generate distance matrix based on
# euclidean (geometric) distances
```

```
allom.dist <- vegdist(decostand(allom[,2:5],'log'),method='euclidean')
## Warning:  non-integer data:  divided by smallest positive value
# use hierarchical clustering to determine different levels of similarity among individuals
allom.clust<-hclust(allom.dist)

# If the plotting window is too small, open one like this: windows()
# (Or click 'Zoom')

# plot the hierarchical clustering result, specifying species with
# the 'labels' argument.
plot(allom.clust,labels=allom[,1])
```




```
# estimate the significance of tree species as a predictor of multivariate growth response
# using ANOSIM
allom.ano<-anosim(allom.dist,allom[,1])
```

```
summary(allom.ano)  # p-value is nonsignificant
```

```
##
## Call:
## anosim(dat = allom.dist, grouping = allom[, 1])
## Dissimilarity: euclidean
##
## ANOSIM statistic R: -0.01231
##       Significance: 0.665
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0343 0.0462 0.0612 0.0797
##
## Dissimilarity ranks between and within classes:
##         0%   25%    50%     75% 100%    N
## Between  1 488.0  967.5 1450.25 1952 1320
## PIMO     3 432.0  895.0 1377.50 1925  171
## PIPO     8 577.5 1074.0 1486.00 1953  231
## PSME     2 470.0  980.0 1502.50 1938  231
```

```
# using PERMANOVA
adonis(allom.dist~allom[,1])  # p-value is nonsignificant
```

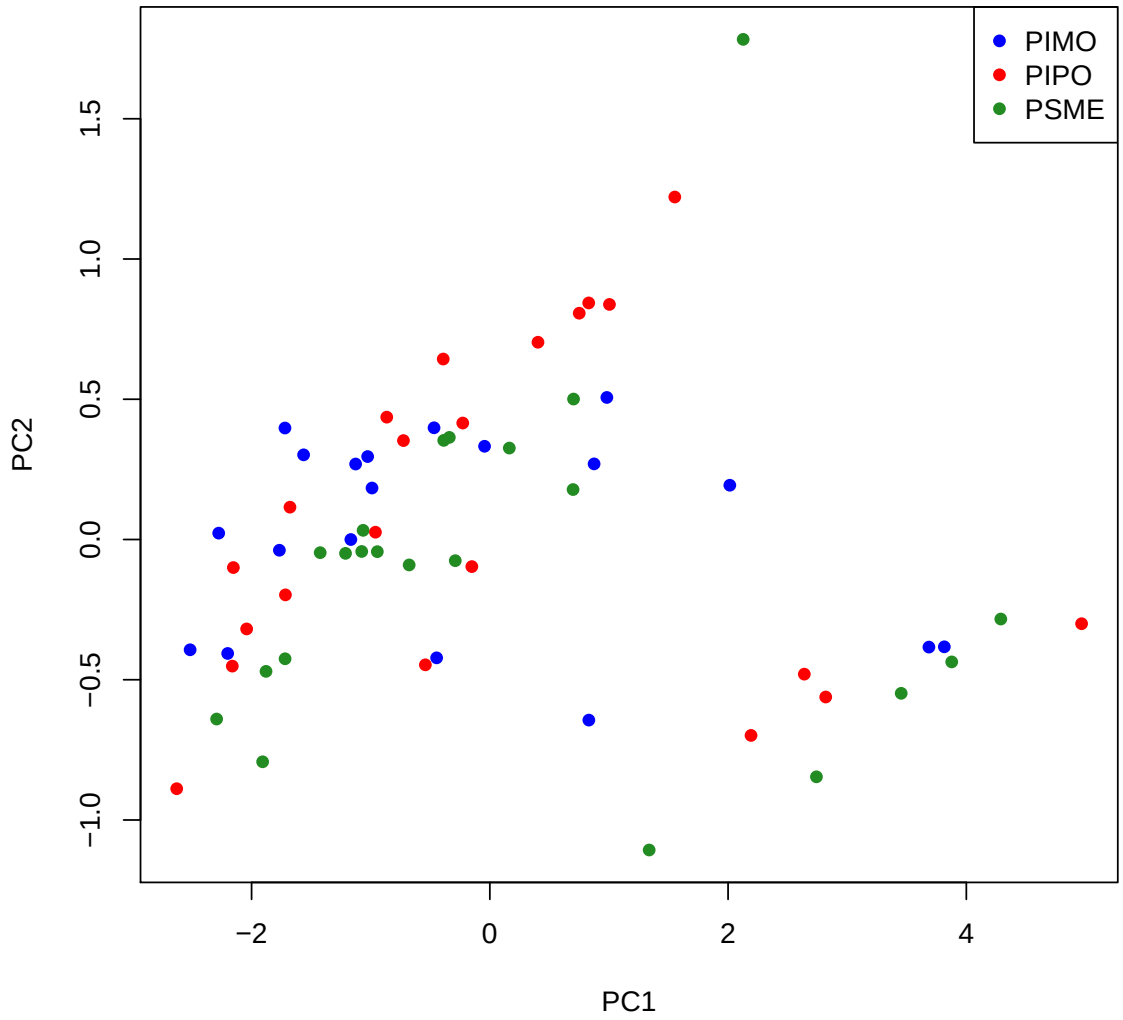
```
##
## Call:
## adonis(formula = allom.dist ~ allom[, 1])
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##            Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## allom[, 1]  2      9.06  4.5302 0.42656 0.01402  0.681
## Residuals  60    637.22 10.6204         0.98598
## Total      62    646.29                1.00000
```

```
# variation between species is similar to variation within species
```

2. ♦ Using your knowledge from Chapter **??** and Sections **??** and **??**, plot the ordination results using coloured circles to represent the different tree species and include a legend.

```
library(vegan)

allom<-read.csv('Allometry.csv')

#log-transform the data prior to PCA
allom.pca.log<-prcomp(log(allom[,2:5]),scale=T)

#use the scores argument to extract the site loadings;
# we want the first two columns (PC1, PC2)
allom.scores<-scores(allom.pca.log)[,1:2]
```

```
#use 'plot' to plot the data and index the 'col' argument by 'species'
palette(c("blue","red","forestgreen"))  # set the colour palette to these three colours
plot(allom.scores, pch=16, col=allom$species)  # 'pch=16' results in closed circles

#add a legend
legend('topright', legend=levels(allom$species), pch=16, col=palette())
```



3. ▲ Overlay the plot with the centroid (i.e., average) for each species, using a different symbol than for the individual points. Modify the axes to reflect the percentage of inertia (i.e., variance) explained by each axis. Refer to Chapter **??** for help tabulating mean values, if necessary.

```
library(vegan)

allom<-read.csv('Allometry.csv')

# log-transform the data prior to PCA
```

```r
allom.pca.log<-prcomp(log(allom[,2:5]),scale=T)

# use the scores argument to extract the site loadings;
# we want the first two columns (PC1, PC2)
allom.scores<-scores(allom.pca.log)[,1:2]

#estimate mean associated with each species, using aggregate on the PCA result
allom.agg<-aggregate(allom.scores, by=list(species=allom$species), FUN=mean)

# proportion of inertia explained by each axis can be found using the 'summary' argument
summary(allom.pca.log)
```

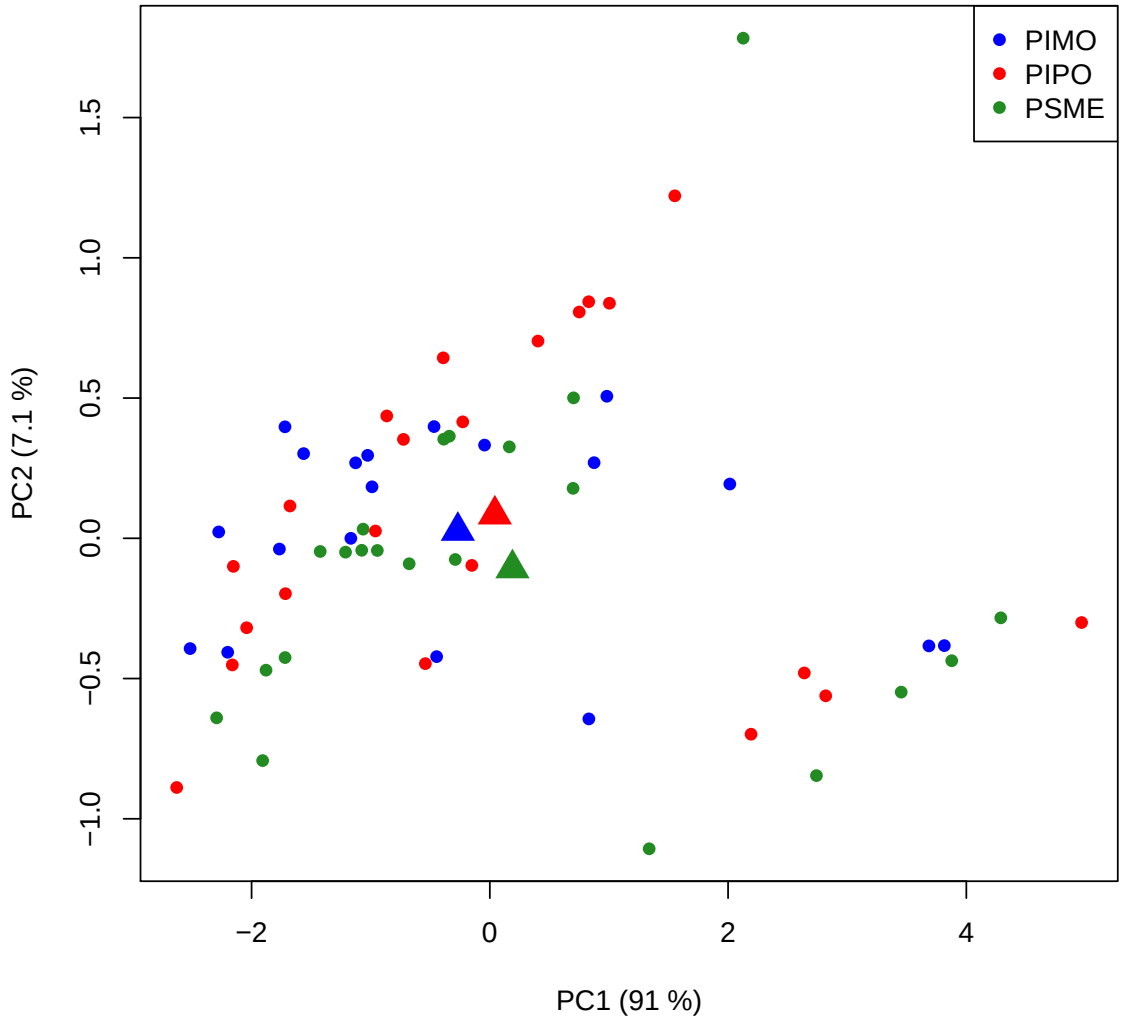
```
## Importance of components%s:
##                           PC1     PC2     PC3     PC4
## Standard deviation     1.9081 0.53172 0.24722 0.12366
## Proportion of Variance 0.9102 0.07068 0.01528 0.00382
## Cumulative Proportion  0.9102 0.98090 0.99618 1.00000
```

```r
# using 'str()', we see that the summary object is a named list,
# we need to extract the 'importance' element
str(summary(allom.pca.log))
```

```
## List of 6
##  $ sdev      : num [1:4] 1.908 0.532 0.247 0.124
##  $ rotation  : num [1:4, 1:4] -0.515 -0.483 -0.496 -0.506 0.243 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "diameter" "height" "leafarea" "branchmass"
##   .. ..$ : chr [1:4] "PC1" "PC2" "PC3" "PC4"
##  $ center    : Named num [1:4] 3.37 3.09 4.16 4.01
##   ..- attr(*, "names")= chr [1:4] "diameter" "height" "leafarea" "branchmass"
##  $ scale     : Named num [1:4] 0.714 0.673 1.25 1.576
##   ..- attr(*, "names")= chr [1:4] "diameter" "height" "leafarea" "branchmass"
##  $ x         : num [1:63, 1:4] -1.907 -0.677 2.127 -0.29 -0.387 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:4] "PC1" "PC2" "PC3" "PC4"
##  $ importance: num [1:3, 1:4] 1.9081 0.9102 0.9102 0.5317 0.0707 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:3] "Standard deviation" "Proportion of Variance" "Cumulative Proportion"
##   .. ..$ : chr [1:4] "PC1" "PC2" "PC3" "PC4"
##  - attr(*, "class")= chr "summary.prcomp"
```

```r
# the relevant proportions for each axis can be found in the second row;
# we want the first two columns
allom.pcvar<-summary(allom.pca.log)$importance[2,1:2]

# use 'plot' to plot the data and index the 'col' argument by 'species',
# customise the axis labels using 'paste'
palette(c("blue","red","forestgreen"))
plot(allom.scores, pch=16, col=allom$species,
         xlab=paste('PC1 (', 100*round(allom.pcvar[1], 3), ' %)', sep=''),
         ylab=paste('PC2 (', 100*round(allom.pcvar[2], 3), ' %)', sep='')
         )

# add a legend
```

```
legend('topright', legend=levels(allom$species), pch=16, col=palette())

# use 'points' to overlay the centroids over the plot
points(allom.agg[,2:3], pch=17, cex=2, col=palette())
```



1.5.4.2 Endophyte data

1. ♦ Use the `adonis` function to test for main and interactive effects of tree species and tissue type (fresh vs litter) on fungal community composition (see Section **??**). The predictor variables can be found in 'endophytes_env.csv'. What terms were significant? Which term explained the most variation?

```
# read in endophyte community data
endo<-read.csv('endophytes.csv')
# read in matrix of predictor variables
```

```
endo.env<-read.csv('endophytes_env.csv')

# use PERMANOVA to test statistical significance of tree species and
# tissue type, interaction.
# Use the formula interface, response is a distance matrix
adonis(vegdist(endo,method='bray') ~ type * species, data=endo.env)
```

```
##
## Call:
## adonis(formula = vegdist(endo, method = "bray") ~ type * species,      data = endo.env)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## type          1     3.498  3.4977 18.9119 0.10707  0.001 ***
## species       8     7.885  0.9856  5.3291 0.24137  0.001 ***
## type:species  8     6.488  0.8110  4.3852 0.19862  0.001 ***
## Residuals    80    14.796  0.1849         0.45293
## Total        97    32.666                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

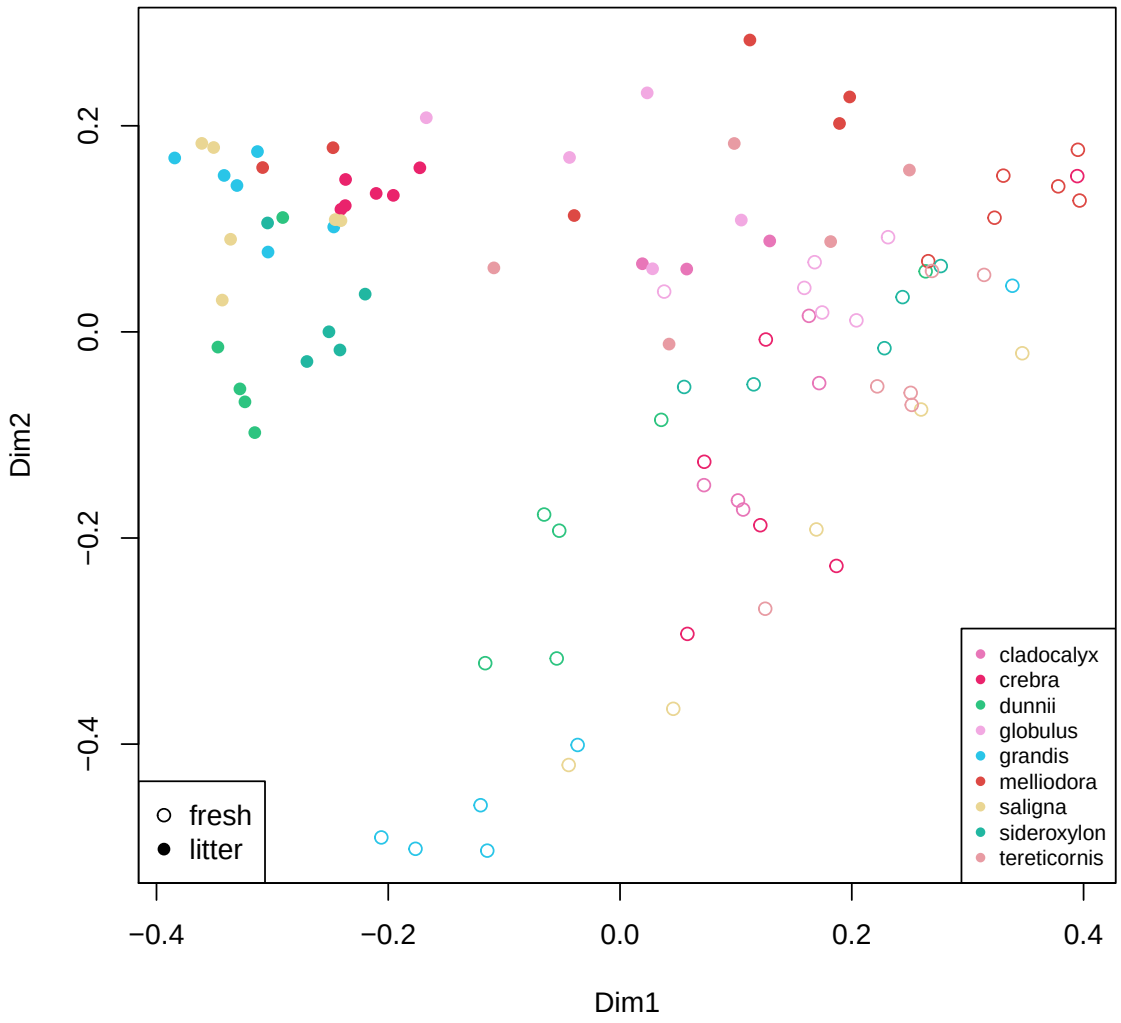
```
# all terms were significant, species had the largest R2
```

2. ▲ Plot the PCoA results and use different symbols and colours to reflect the identities of tree species and tissue types. Add a legend to the plot. Use information from Chapter **??** and Sections **??** and **??** for help, if necessary. Hint: automatic functions for generating vectors of colours, such as `rainbow`, can lead to very similar colours with so many treatment levels. Check out the `randomcoloR` package for alternative approaches.

```
# read in endophyte community data
endo<-read.csv('endophytes.csv')
# read in matrix of predictor variables
endo.env<-read.csv('endophytes_env.csv')

# perform PCoA, input is a distance matrix
endo.pcoa<-wcmdscale(vegdist(endo,method='bray'))

# set up palette
library(randomcoloR)
```

```
## Warning:  package 'randomcoloR' was built under R version 3.4.3
```

```
palette(randomColor(length(levels(endo.env$species))))

# plot PCoA results using scores() to extract the site loadings
# eight colours indexed by species, two symbols indexed by tissue type
plot(scores(endo.pcoa,display='sites'),col=endo.env$species,
     pch=c(1,16)[endo.env$type])
# add a legend indicating tissue type
legend('bottomleft',levels(endo.env$type),pch=c(1,16))
# add a legend indicating tree species
legend('bottomright',levels(endo.env$species),col=palette(),pch=16,cex=0.75)
```



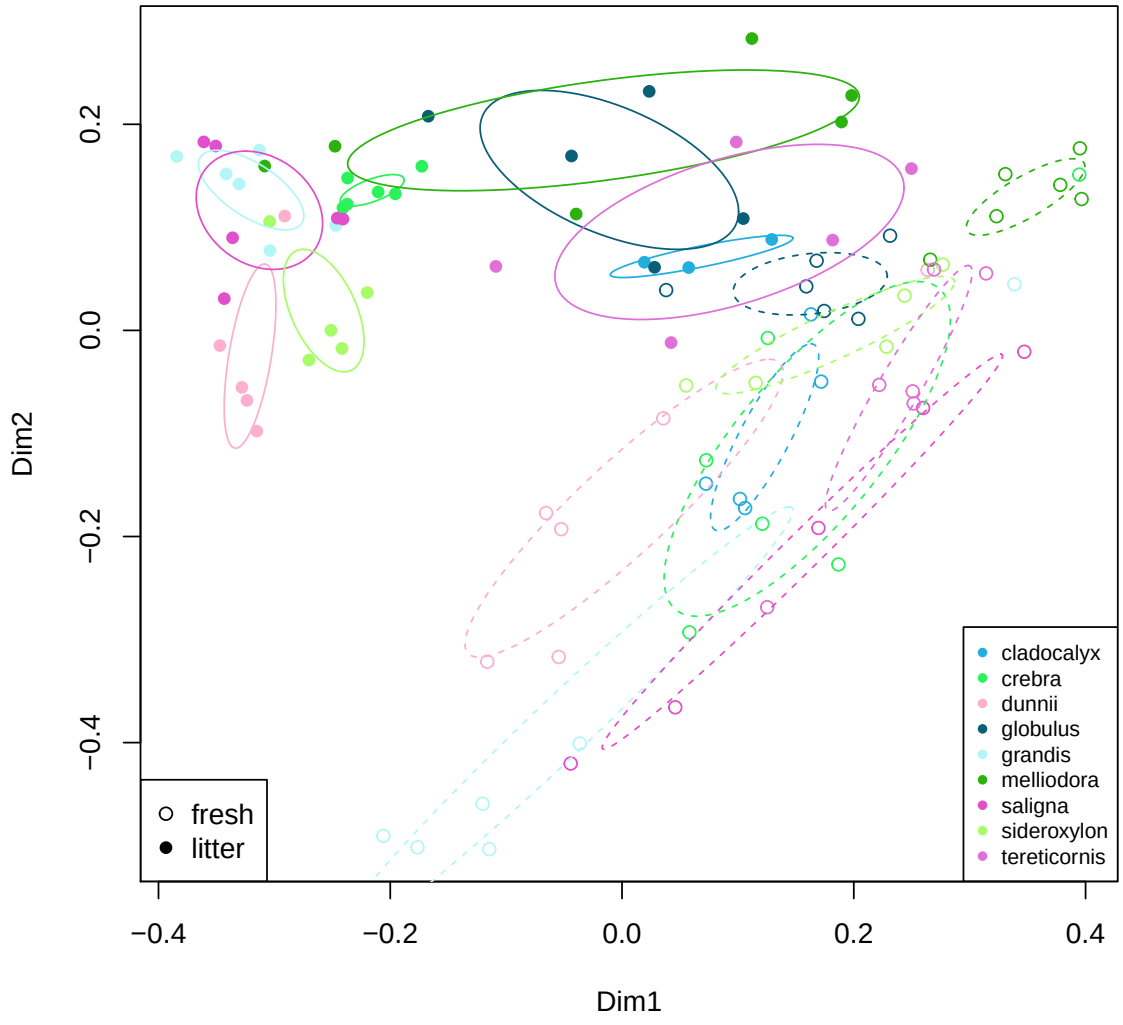
3. ▲ Overlay the plot with ellipses representing 95% confidence intervals for each species and sample type using functions seen in section **??**.

```
# create vector identifying unique treatment combinations
trts <- with(endo.env, interaction(type, species, sep='-'))
levels(trts)
```

```
##  [1] "fresh-cladocalyx"    "litter-cladocalyx"   "fresh-crebra"
##  [4] "litter-crebra"       "fresh-dunnii"        "litter-dunnii"
##  [7] "fresh-globulus"      "litter-globulus"     "fresh-grandis"
## [10] "litter-grandis"      "fresh-melliodora"    "litter-melliodora"
## [13] "fresh-saligna"       "litter-saligna"      "fresh-sideroxylon"
## [16] "litter-sideroxylon"  "fresh-tereticornis"  "litter-tereticornis"
```

```
# set up palette
library(randomcoloR)
palette(randomColor(length(levels(endo.env$species))))
```

```
# set up vectors for colours and line types based on treatment combination levels
col.spp <- rep(palette(), each=2)
lty.type <- rep(c('dashed', 'solid'), 9)

# plot PCoA results using scores() to extract the site loadings
# eight colours indexed by species, two symbols indexed by tissue type
plot(scores(endo.pcoa,display='sites'),col=endo.env$species,
     pch=c(1,16)[endo.env$type])
# add a legend indicating tissue type
legend('bottomleft',levels(endo.env$type),pch=c(1,16))
# add a legend indicating tree species
legend('bottomright',levels(endo.env$species),col=palette(),pch=16,cex=0.75)

# plot ellipses, indexing colours by species and line types by sample type
ordiellipse(endo.pcoa, trts, kind='se', conf=0.95,
            col=col.spp, lty=lty.type)
```



1.5.4.3 Mite data

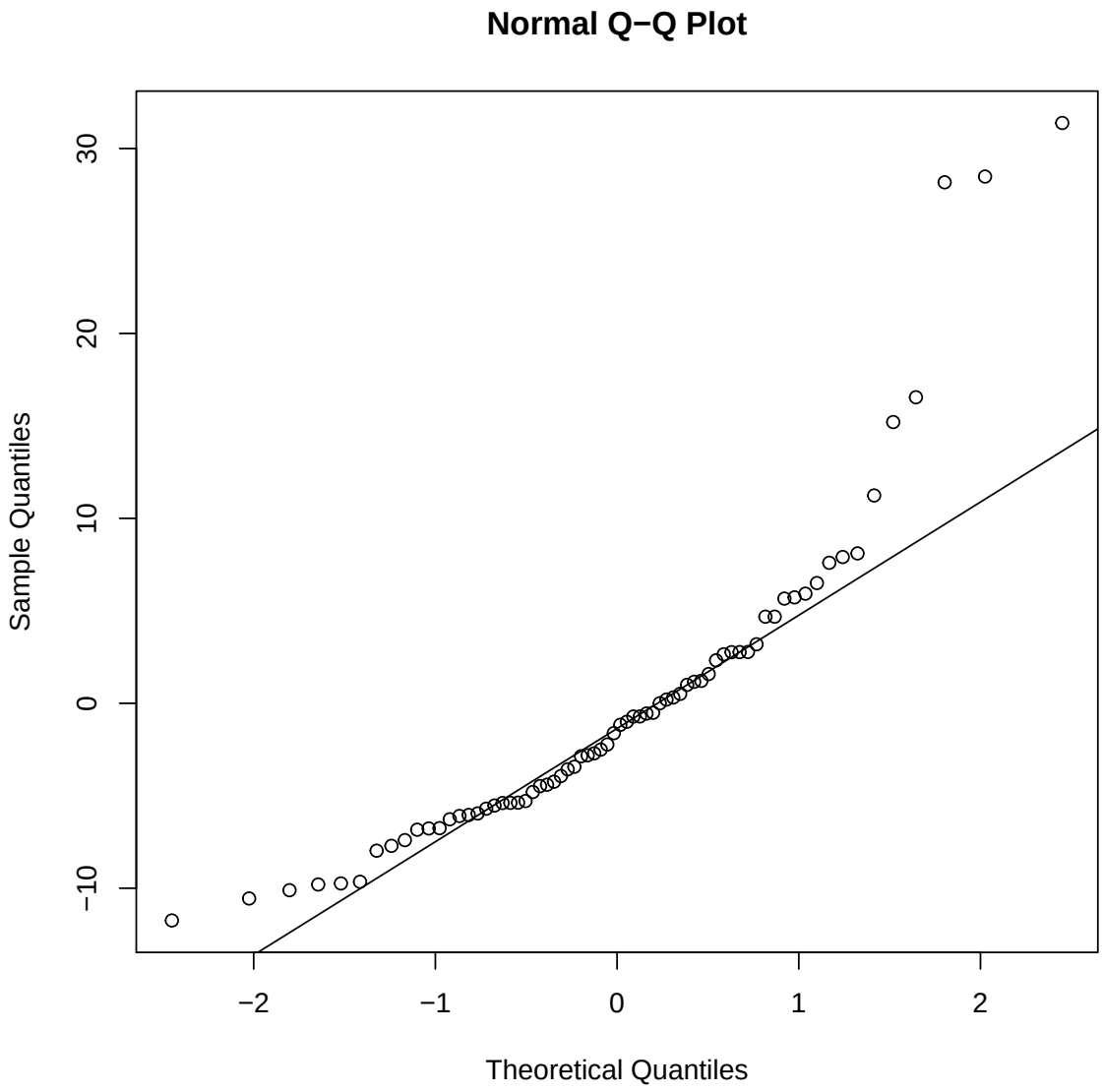
1. ■ Use `manova` to estimate the responses of mite community composition to the environmental variables associated with the `mite` data.

```
# load library
library(vegan)

# load 'mite' data
data(mite)
data(mite.env)

# convert response table to matrix
Y <- as.matrix(mite)

# fit model
mite.manova <- manova(Y ~ SubsDens + WatrCont + Substrate + Shrub + Topo,
                      data = mite.env)


# view model summary
summary(mite.manova)
```

```
##            Df Pillai approx F num Df den Df    Pr(>F)
## SubsDens    1 0.7041   1.6320     35     24   0.10618
## WatrCont    1 0.9146   7.3425     35     24 1.559e-06 ***
## Substrate   6 3.1676   0.9266    210    174   0.70203
## Shrub       2 1.3755   1.5732     70     50   0.04644 *
## Topo        1 0.8105   2.9319     35     24   0.00373 **
## Residuals 58
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2. ♦ There is not a built in function to view diagnostic plots for `manova` output. Use the `resid` function to obtain the residuals for each response variable and use the `qqnorm` and `qqline` functions to produce quantile-quantile plots for a few of the response variables to determine whether it is appropriate to model the responses using a normal error distribution.

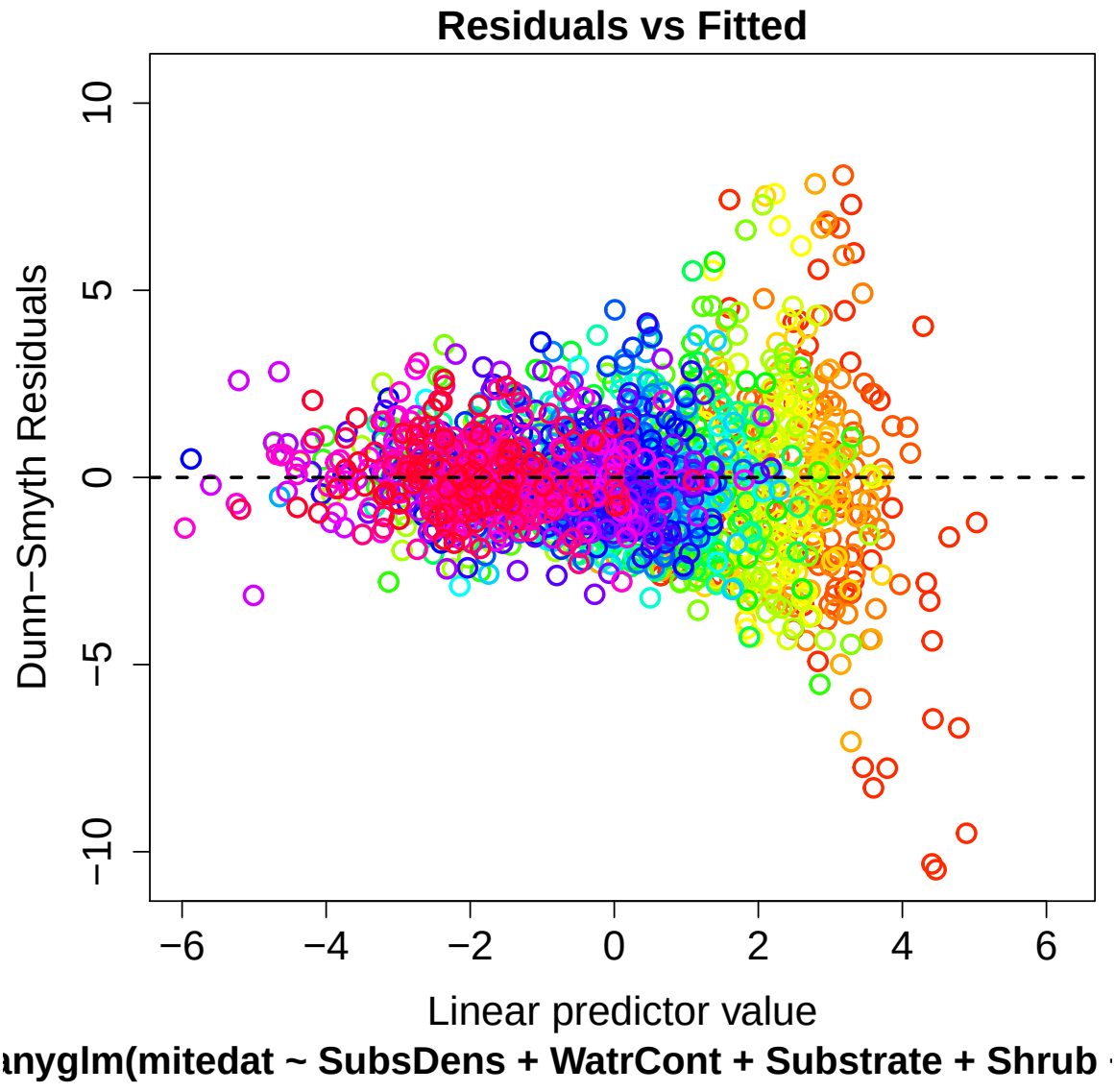
```
# plot residuals for first response variable
qqnorm(resid(mite.manova)[, 1])
# add line
qqline(resid(mite.manova)[, 1])
```



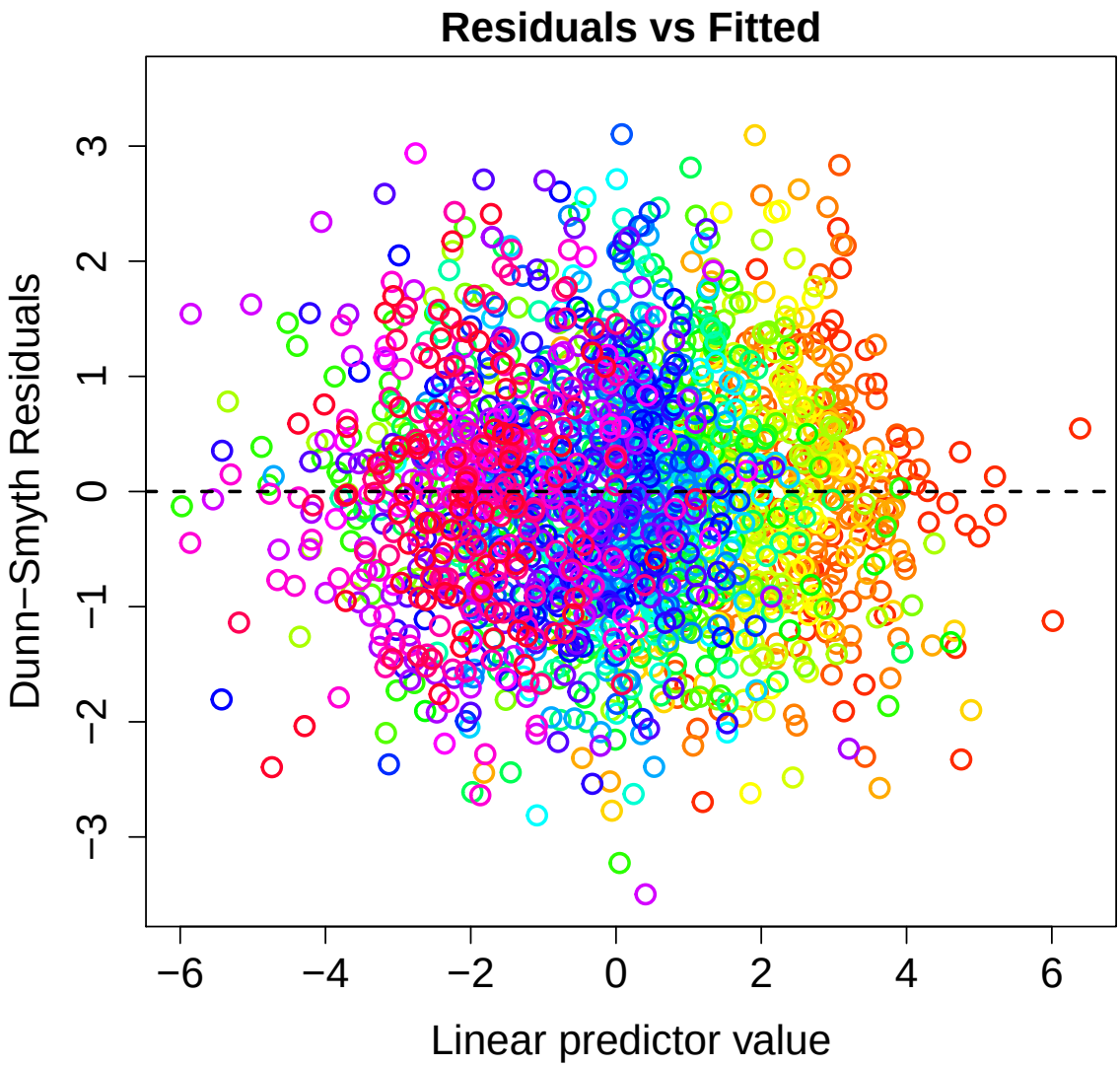
3. ♦ Use `manyglm` to estimate the responses of mite community composition to the environmental variables associated with the `mite` data. Which error family, poisson or negative binomial, provides the best fit to the data? Look at the results of the best fitting model.

```
# load libraries
library(vegan)
library(mvabund)

# load 'mite' data
data(mite)
data(mite.env)

# convert response table to an mvabund object
mitedat <- mvabund(mite)

# fit multivariate GLM model with poisson error distribution
mite.pois <- manyglm(mitedat ~ SubsDens + WatrCont + Substrate + Shrub + Topo,
```

```
                    data = mite.env, family='poisson')

# fit multivariate GLM model with negative binomial error distribution
mite.nbin <- manyglm(mitedat ~ SubsDens + WatrCont + Substrate + Shrub + Topo,
                    data = mite.env, family='negative.binomial')

# view model diagnostics
plot(mite.pois)
```




```
plot(mite.nbin)
```



anyglm(mitedat ~ SubsDens + WatrCont + Substrate + Shrub

```
# view model summary and ANOVA table for 'negative binomial' model
summary(mite.nbin)

##
## Test statistics:
##                     wald value Pr(>wald)
## (Intercept)             18.821  0.000999 ***
## SubsDens                15.983  0.000999 ***
## WatrCont                19.411  0.000999 ***
## SubstrateSphagn2         8.689  0.003996 **
## SubstrateSphagn3         6.791  0.042957 *
## SubstrateSphagn4         4.018  0.624376
## SubstrateLitter          6.247  0.147852
## SubstrateBarepeat        4.590  0.009990 **
## SubstrateInterface       7.232  0.287712
## Shrub.L                  8.511  0.000999 ***
## Shrub.Q                  9.943  0.000999 ***
```

```
## TopoHummock            12.950  0.000999 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Test statistic:  36.88, p-value: 0.000999
## Arguments:
##  Test statistics calculated assuming response assumed to be uncorrelated
##  P-value calculated using 1000 resampling iterations via pit.trap resampling (to account for cor
```

```
anova(mite.nbin)
```

```
## Time elapsed: 0 hr 3 min 32 sec
## Analysis of Deviance Table
##
## Model: manyglm(formula = mitedat ~ SubsDens + WatrCont + Substrate +
## Model:     Shrub + Topo, family = "negative.binomial", data = mite.env)
##
## Multivariate test:
##             Res.Df Df.diff   Dev Pr(>Dev)
## (Intercept)     69
## SubsDens        68       1  73.5    0.013 *
## WatrCont        67       1 600.3    0.001 ***
## Substrate       61       6 306.9    0.015 *
## Shrub           59       2 247.7    0.001 ***
## Topo            58       1 146.2    0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Arguments:
##  Test statistics calculated assuming uncorrelated response (for faster computation)
##  P-value calculated using 999 resampling iterations via PIT-trap resampling (to account for corr
```