Other topics

Gavin L. Simpson

November 25, 2022

Welcome

Today's topics

- PERMANOVA
- Distance-based RDA
- Co-correspondence analysis (Co-CA)
- Principal Response Curves (PRC)
- Diagnostics

PERMANOVA

MANOVA

MANOVA is the multivariate form of ANOVA

- Multivariate response data
- Categorical predictor variables

Decompose variation in the responses into

1. variation within groups
2. variation between groups

Test to see if two is unusually large relative to H_0

PERMANOVA

Doing that test requires lots of assumptions that rarely hold for ecological data

PERMANOVA: Permutational multivariate analysis of variance

Avoids most of these issues through the use of permutation tests

Directly decomposes a dissimilarity matrix into

1. variation within groups
2. variation between groups

PERMANOVA *sensu stricto*

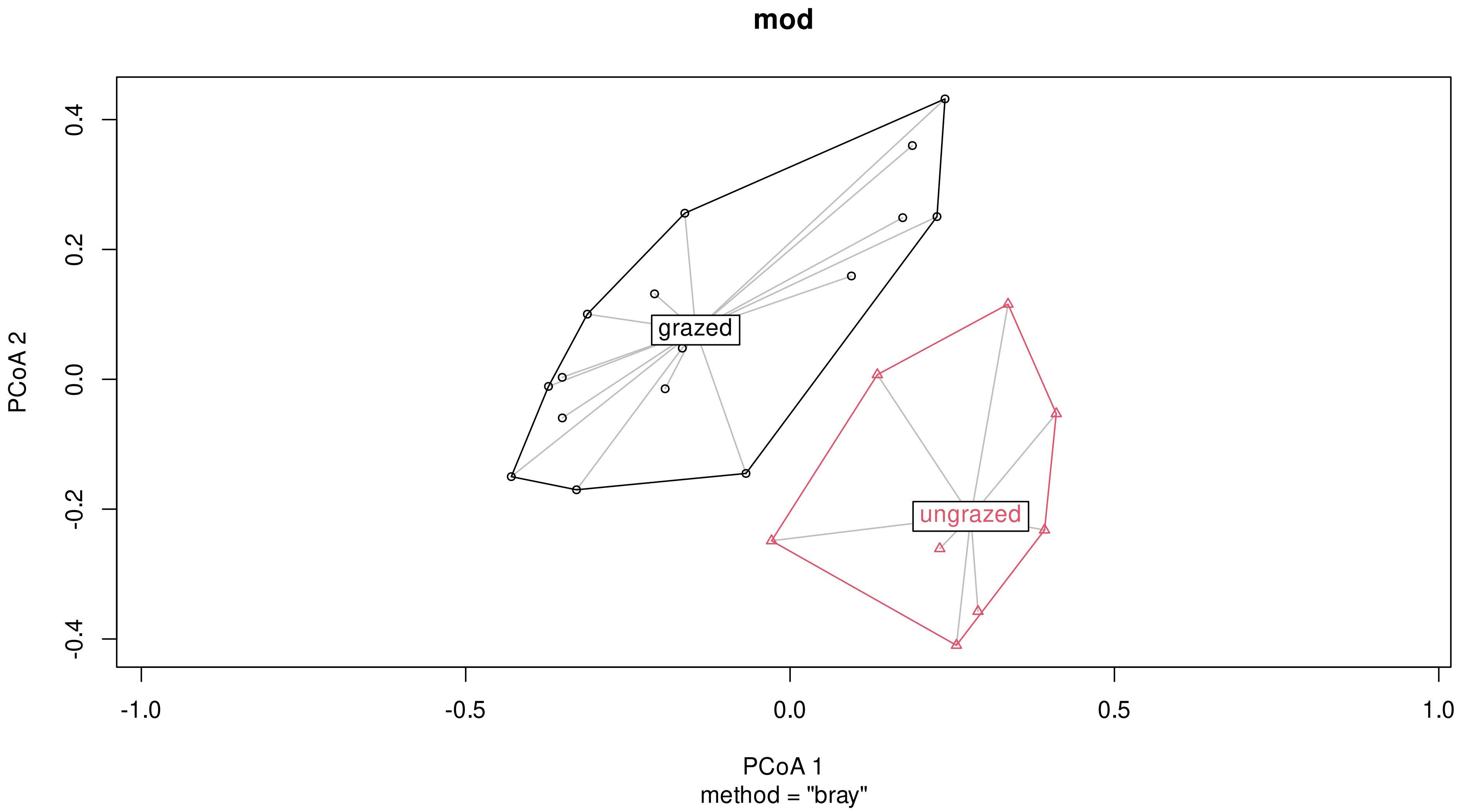
vegan has four different ways to do essentially do this kind of analysis

1. `adonis()` — implements Anderson (2001)
2. `adonis2()` — implements McArdle & Anderson (2001)
3. `dbrda()` — implementation based on McArdle & Anderson (2001)
4. `capscale()` — implements Legendre & Anderson (1999)

Be careful with `adonis()` as it allows only sequential tests

A difference between the functions is how they treat negative eigenvalues

The PERMANOVA idea



PERMANOA — adonis2()

```r
data(dune, dune.env)
adonis2(dune ~ Management*A1, data = dune.env, by = "terms")
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = dune ~ Management * A1, data = dune.env, by = "terms")
##               Df SumOfSqs      R2      F Pr(>F)
## Management     3   1.4686 0.34161 3.2629  0.002 **
## A1             1   0.4409 0.10256 2.9387  0.017 *
## Management:A1  3   0.5892 0.13705 1.3090  0.195
## Residual      12   1.8004 0.41878
## Total         19   4.2990 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

PERMANOA — adonis2()

```r
data(dune, dune.env)
adonis2(dune ~ A1 * Management, data = dune.env, by = "terms")
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = dune ~ A1 * Management, data = dune.env, by = "terms")
##               Df SumOfSqs      R2      F Pr(>F)
## A1             1   0.7230 0.16817 4.8187  0.003 **
## Management     3   1.1865 0.27600 2.6362  0.007 **
## A1:Management  3   0.5892 0.13705 1.3090  0.191
## Residual      12   1.8004 0.41878
## Total         19   4.2990 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

PERMANOA — adonis2()

```
data(dune, dune.env)
adonis2(dune ~ Management * A1, data = dune.env, by = "margin")
```

```
## Permutation test for adonis under reduced model
## Marginal effects of terms
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = dune ~ Management * A1, data = dune.env, by = "margin")
##               Df SumOfSqs      R2     F Pr(>F)
## Management:A1  3   0.5892 0.13705 1.309  0.222
## Residual      12   1.8004 0.41878
## Total         19   4.2990 1.00000
```

The interaction is the only term that isn't *marginal* to other terms; not significant

PERMANOA — adonis2()

```r
adonis2(dune ~ Management + A1, data = dune.env, by = "margin")
```

```
## Permutation test for adonis under reduced model
## Marginal effects of terms
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = dune ~ Management + A1, data = dune.env, by = "margin")
##            Df SumOfSqs      R2      F Pr(>F)
## Management  3   1.1865 0.27600 2.4828  0.006 **
## A1          1   0.4409 0.10256 2.7676  0.025 *
## Residual   15   2.3895 0.55583
## Total      19   4.2990 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The dispersion problem

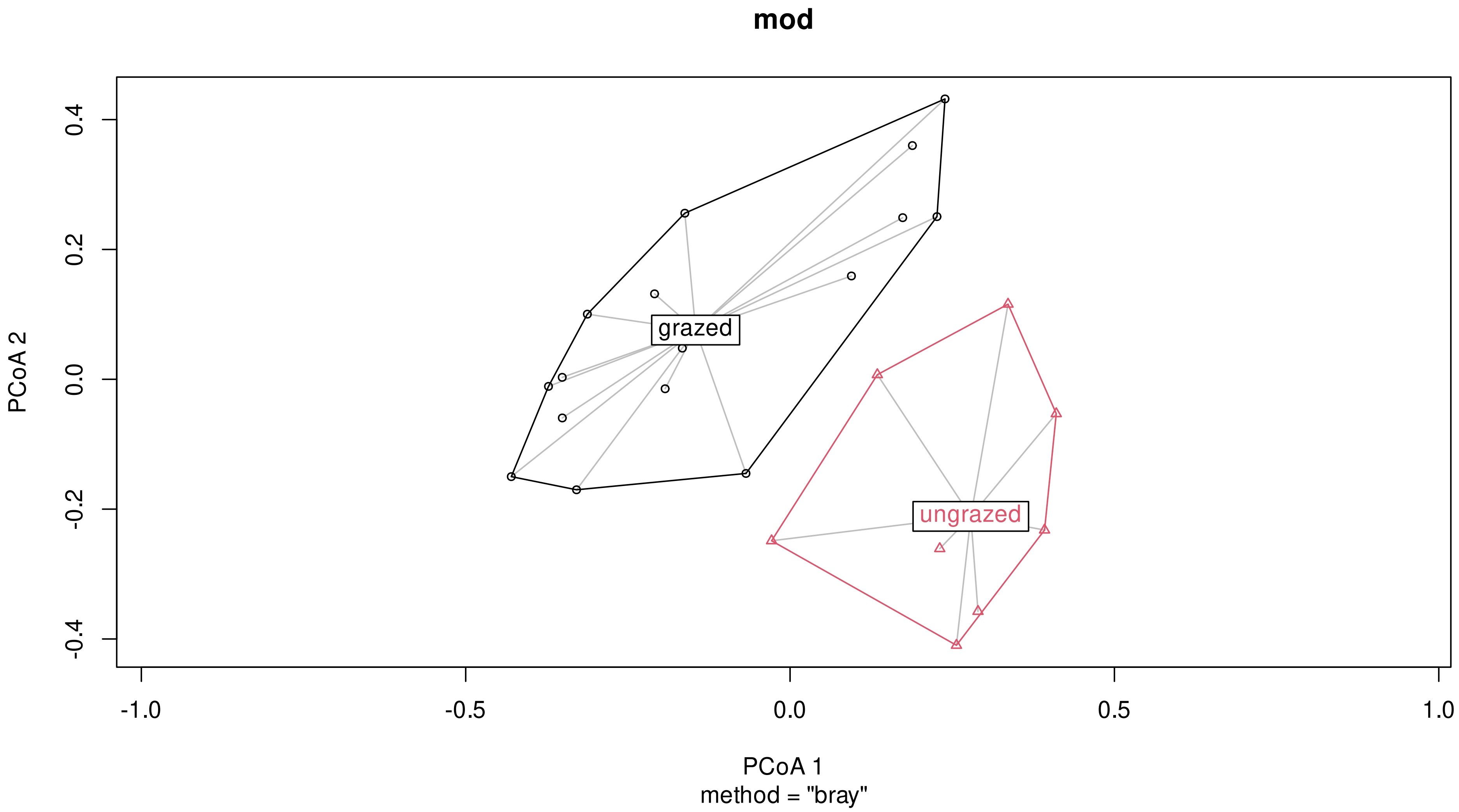
Anderson (2001) noted that PERMANOVA could confound *location* & *dispersion* effects

If one or more groups are more variable — dispersed around the centroid — than the others, this can result in a false detection of a difference of means — a *location* effect

Same problem affects *t* tests, ANOVA

Warton et al (2012) Anderson & Walsh (2013) Anderson *et al* (2017)

Dispersion


mod
grazed
ungrazed
PCoA 2
0.4
0.2
0.0
-0.2
-0.4
-1.0
-0.5
0.0
0.5
1.0
PCoA 1
method = "bray"

Test for dispersion effects

Marti Anderson (2006) developed a test for multivariate dispersions — PERMDISP2

1. Calculate how far each observation is from its group median (or centroid)
2. Take the absolute values of these distances-to-medians
3. Do an ANOVA on the absolute distances with the *groups* as covariates
4. Test the H_0 of equal absolute distances to median among groups using a permutation test

In *vegan* this is `betadisper()`

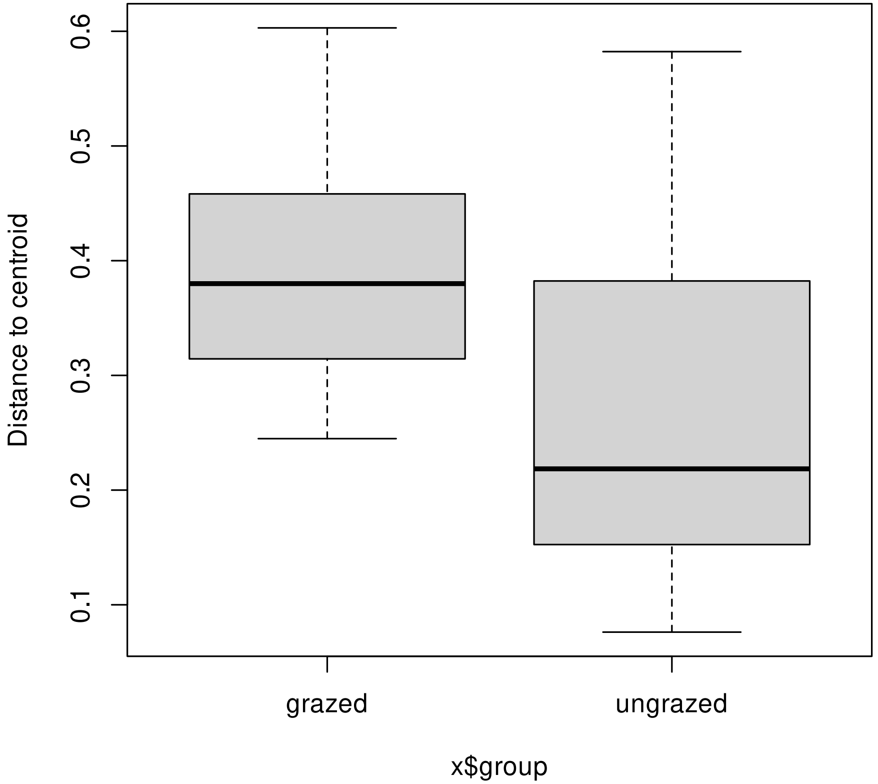
Test for dispersion effects

```
data(varespec)
dis ← vegdist(varespec) # Bray-Curtis distances
## First 16 sites grazed, remaining 8 sites ungrazed
groups ← factor(c(rep(1,16), rep(2,8)),
                 labels = c("grazed","ungrazed"))

mod ← betadisper(dis, groups)
mod
```

```
##
##     Homogeneity of multivariate dispersions
##
## Call: betadisper(d = dis, group = groups)
##
## No. of Positive Eigenvalues: 15
## No. of Negative Eigenvalues: 8
##
## Average distance to median:
##   grazed ungrazed
##   0.3926   0.2706
##
## Eigenvalues for PCoA axes:
## (Showing 8 of 23 eigenvalues)
##  PCoA1  PCoA2  PCoA3  PCoA4  PCoA5  PCoA6  PCoA7
PCoA8
## 1.7552 1.1334 0.4429 0.3698 0.2454 0.1961 0.1751
0.1284
```

```
boxplot(mod)
```

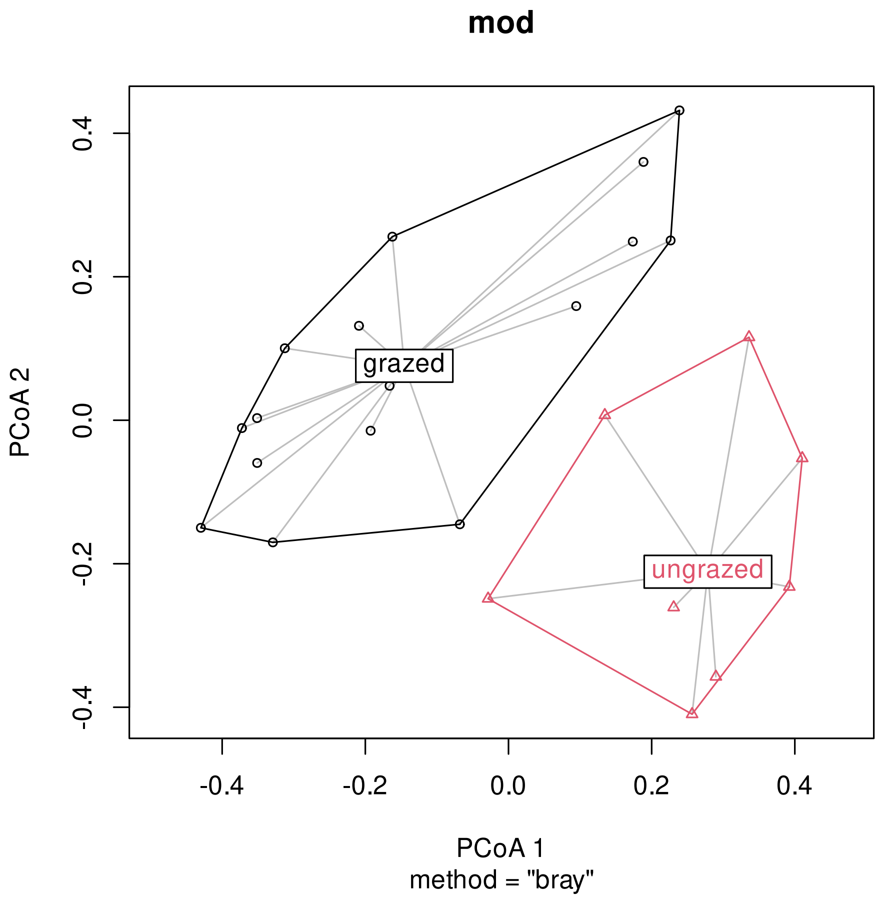


Test for dispersions

```
set.seed(25)
permutest(mod)
```

```
##
## Permutation test for homogeneity of multivariate
dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##           Df  Sum Sq  Mean Sq      F N.Perm Pr(>F)
## Groups     1 0.07931 0.079306 4.6156    999  0.045 *
## Residuals 22 0.37801 0.017182
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
```

```
plot(mod)
```



Test for dispersions

```r
set.seed(4)
permutest(mod, pairwise = TRUE)
```

```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##           Df  Sum Sq  Mean Sq      F N.Perm Pr(>F)
## Groups     1 0.07931 0.079306 4.6156    999  0.036 *
## Residuals 22 0.37801 0.017182
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##           grazed ungrazed
## grazed              0.043
## ungrazed 0.04295
```

Test for locations with non-equal dispersion?

Marti Anderson & colleagues (2017) have proposed a solution that is related to the Berens-Fisher problem

This is in Primer but not yet in *vegan*

https://github.com/vegandevs/vegan/issues/344

Distance-based RDA

Distance-based RDA

Multiple models that all do something Similar

1. `adonis()` (deprecated)
2. `adonis2()`
3. `capscale()`
4. `dbrda()`

They all do essentially the same thing, but they do it differently & have slightly different behaviour

Distance-based RDA

Distance-based RDA (db-RDA) is a constrained form of principal coordinates analysis (PCO)

It is similar to RDA but allows for non-Euclidean dissimilarity indices

In *vegan*, db-RDA is implemented in `dbrda()`

Constrained analysis of principal coordinates

`capscale()` is *another* constrained form of PCO due to Legendre & Anderson (1999)

It is *very* similar to `dbrda()`

Constrained analysis of principal coordinates

`capscale()` works by

1. convert the response data into dissimilarities
2. apply PCO on the dissimilarities, take the PCO sample (site) scores as *new* response data
3. fit `rda()` to the *new* response data and predictor variables as constraints

Essentially, we embed the dissimilarities in a Euclidean space using PCO, and then we use RDA on this highly transformed response data

Distance-based RDA

db-RDA foregoes step 2., and directly decomposes the dissimilarities into components explained by each term in the model

Negative eigenvalues resulting from non-metric dissimilarity coefficients are handled via

1. square-root transform of the dissimilarities, or
2. adding a constant to the dissimilarities using methods `"lingoes"` (default, preferred) or `"cailliez"`

db-RDA is based on the ideas in McArdle & Anderson (2001)

Err... isn't that what `adonis2()` was developed to do?

Yes, but...

Distance-based RDA

`adonis2()` was a ground up redevelopment of the `adonis()` implementation and as such it retains many of the arguments and concepts of PERMANOVA, just updated to use the direct decomposition of dissimilarities

`dbrda()` inherits from `rda()` and `cca()` and as a result has expanded set of capability

`dbrda()` can use `Condition()` in the formula to fit partial db-RDA

`Condition()` is often needed to provide correct restricted perumtation tests

Distance-based RDA

The equivalent model to `adonis2()` in `dbrda()`-form is

```
data(dune, dune.env)
dune_dbrda <- dbrda(dune ~ Management * A1, data = dune.env,
    method = "bray")
```

because they have different default `method` values

Ohraz Case Study

Restricted permutations | Ohraz

Analyse the Ohraz data Case study 5 of Leps & Smilauer

Repeated observations of composition from an experiment

- Factorial design (3 replicates)
- Treatments: fertilisation, mowing, *Molinia* removal

Test 1 of the hypotheses

> There are *no* directional changes in species composition in time that are common to all treatments or specific treatments

Restricted permutations | Ohraz

```r
spp ← read_csv(url("https://bit.ly/ohraz-spp")) %>%
    rename(label = "...1") %>%
    janitor::clean_names()

molinia ← spp %>%
    select(label:molicaer)

spp ← spp %>%
    select(-molicaer) %>%
    column_to_rownames("label")

env ← read_csv(url("https://bit.ly/ohraz-env")) %>%
    rename(label = "...1") %>%
    mutate(across(c(mowing:removal, plotid), ~ factor(.x))) %>%
    column_to_rownames("label")
```

Distance-based RDA | Ohraz

```r
ohraz_dbrda ← dbrda(spp ~ year +
    year:mowing + year:fertilizer + year:removal +
    Condition(plotid), data = env, method = "bray", add = "lingoes")
h ← how(within = Within(type = "free"),
    plots = Plots(strata = env$plotid, type = "none"))
set.seed(42)
anova(ohraz_dbrda, permutations = h, model = "reduced")
```

```
## Permutation test for dbrda under reduced model
## Plots: env$plotid, plot permutation: none
## Permutation: free
## Number of permutations: 199
##
## Model: dbrda(formula = spp ~ year + year:mowing + year:fertilizer + year:removal +
Condition(plotid), data = env, add = "lingoes", method = "bray")
##          Df Variance      F Pr(>F)
## Model     4   96.846 5.3777  0.005 **
## Residual 68  306.146
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Co-CA

Co-correspondence analysis

We may be interested in relating two species data sets to one another

Can can't do this using CCA with one data set as the response and the other as the predictors, because this would be too many predictors

We might also not want to treat one data set as the response, we might want to analyse the data symmetrically

Co-correspondence Analysis (Co-CA) is a suitable method

Co-correspondence analysis

There are two types of Co-CA

1. symmetric Co-CA, and
2. predictive Co-CA

Symmetric Co-CA finds axes in both data sets which maximise the covariation of the two data sets

Predictive Co-CA finds directions in one data set which best predict the response data set

Symmetric Co-CA

```
library("cocorresp")
data(beetles)
## log transform the beetle data
beetles <- log1p(beetles)
data(plants)
```

Symmetric Co-CA

```r
bp.sym <- coca(beetles ~ ., data = plants, method = "symmetric")
```

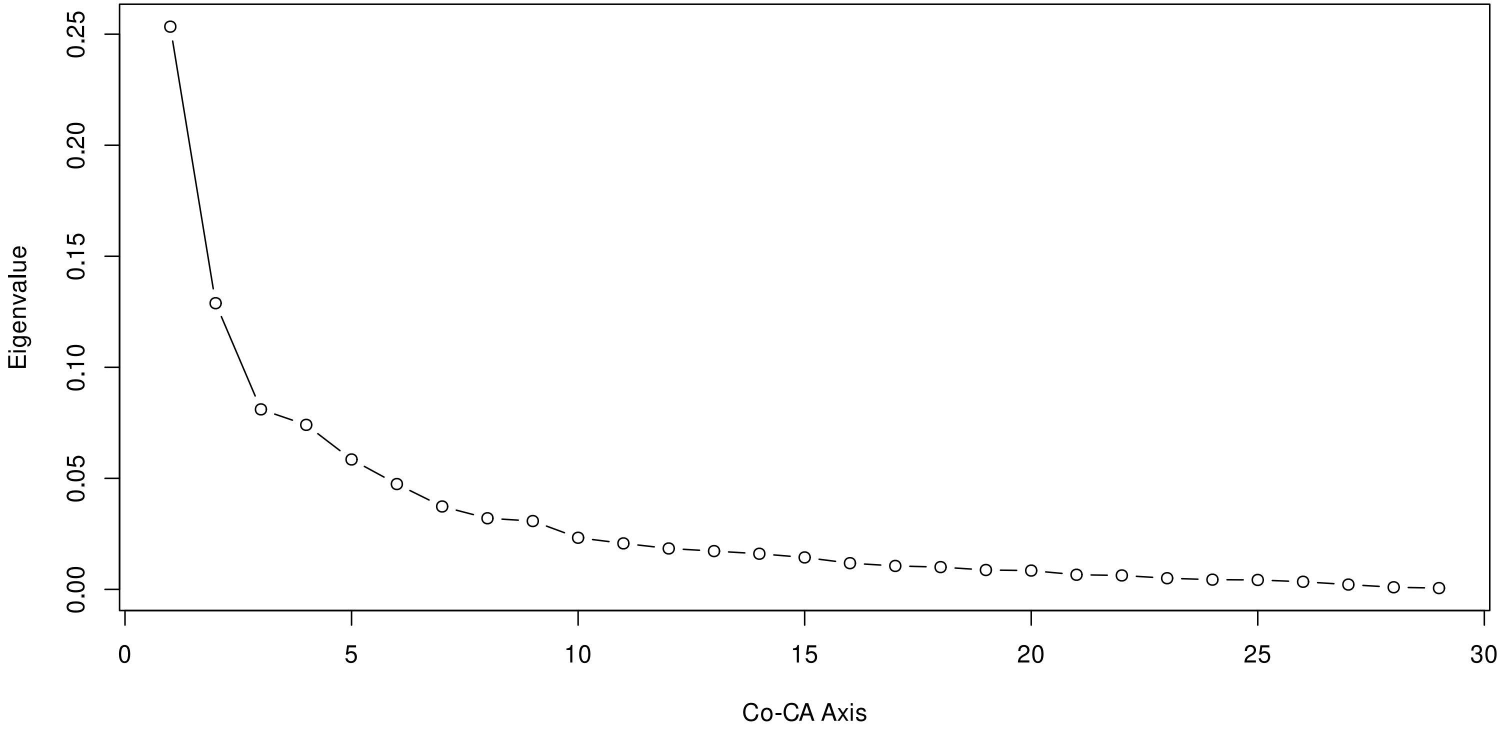
```
##
## Removed some species that contained no data in: beetles, plants
```

```r
bp.sym
```

```
##
## Symmetric Co-Correspondence Analysis
##
## Call: symcoca(y = y, x = x, n.axes = n.axes, R0 = weights, symmetric =
## symmetric, nam.dat = nam.dat)
##
## Inertia:
##          Total Explained Residual
## beetles: 3.988     3.971    0.018
## plants:  5.757     5.740    0.018
```

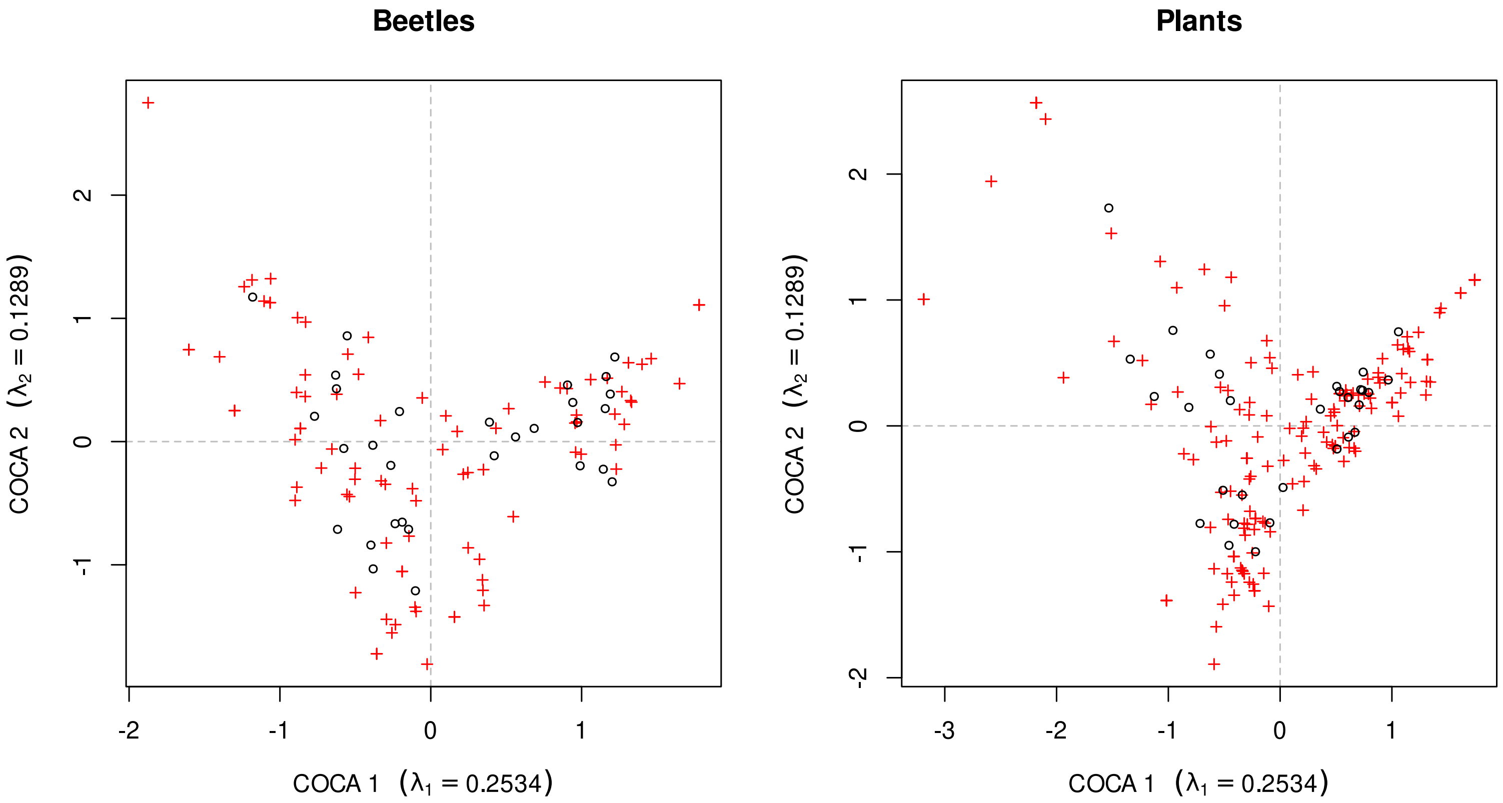
Symmetric Co-CA

```
screeplot(bp.sym)
```



Symmetric Co-CA

```r
layout(matrix(1:2, ncol = 2))
biplot(bp.sym, which = "y1", main = "Beetles")
biplot(bp.sym, which = "y2", main = "Plants")
layout(1)
```



PRC

Principal Response Curves

PRC is a special form of redundancy analysis (RDA) that is useful when you want to compare the development of a biological community in time, under different conditions (treatments)

The different conditions don't have to be experimental; could be used for monitoring relative to a control site or control period

Principal Response Curves

PRC focuses on the temporally structured effects of different levels of a factor

In standard RDA (etc) it is difficult to compare the temporal trajectories of plots as the time direction is usually a complex path through the ordination space

PRC is designed to focus specifically on the temporal effects and their differences among sites

Allows for a PRC diagram that displays these time-structured effects optimally

Principal Response Curves

Assume k treatment levels in a factor F

We observed community composition (abundance) at the same set of samples at (the same) multiple time points

The time points are coded as t

PRC fits a *partial* RDA of the form

```
rda(comm ~ F:t + Condition(t), data = df)
```

As t is partialled out, the PRC represents the overall **differences** among treatment level and how these difference change through time

PRC — pyrifos example

Data are log transformed abundances of aquatic invertebrate in twelve ditches studied in eleven times before and after an insecticide treatment

12 mesocosms were allocated randomly to treatments, with 4 controls, while the remaining 8 mesocosms were treated with a dose of an insecticide, *chloropyrifos* (0.1, 0.9, 6, 44 µg/ L)

Invertebrates were samples 11 times, from 4 weeks *prior* to treatment through 24 weeks post-treatment (132 samples total)

van den Brink, P.J. & ter Braak, C.J.F. (1999). Principal response curves: Analysis of time-dependent multivariate responses of biological community to stress. Environmental Toxicology and Chemistry, 18, 138–148.

PRC — pyrifos example

```r
data(pyrifos)
dim(pyrifos)
```

```
## [1] 132 178
```

```r
ditch <- gl(12, 1, length = 132)
week <- gl(11, 12, labels = c(-4, -1, 0.1, 1, 2, 4, 8, 12, 15, 19, 24))
dose <- factor(rep(c(0.1, 0, 0, 0.9, 0, 44, 6, 0.1, 44, 0.9, 0, 6), 11))
```

Important that the control (or reference) site is the reference (first) level of the "treatment" factor — use `relevel()`

PRC — pyrifos example

```
mod ← prc(pyrifos, dose, week)
```

```
mod
```

```
## Call: prc(response = pyrifos, treatment = dose, time = week)
## 
##                Inertia Proportion Rank
## Total         288.9920     1.0000     
## Conditional    63.3493     0.2192   10
## Constrained    96.6837     0.3346   44
## Unconstrained 128.9589     0.4462   77
## Inertia is variance 
## 
## Eigenvalues for constrained axes:
##   RDA1   RDA2   RDA3   RDA4   RDA5   RDA6   RDA7   RDA8   RDA9  RDA10  RDA11 
## 25.282  8.297  6.044  4.766  4.148  3.857  3.587  3.334  3.087  2.551  2.466 
##  RDA12  RDA13  RDA14  RDA15  RDA16  RDA17  RDA18  RDA19  RDA20  RDA21  RDA22 
##  2.209  2.129  1.941  1.799  1.622  1.579  1.440  1.398  1.284  1.211  1.133 
##  RDA23  RDA24  RDA25  RDA26  RDA27  RDA28  RDA29  RDA30  RDA31  RDA32  RDA33 
##  1.001  0.923  0.862  0.788  0.750  0.712  0.685  0.611  0.584  0.537  0.516 
##  RDA34  RDA35  RDA36  RDA37  RDA38  RDA39  RDA40  RDA41  RDA42  RDA43  RDA44 
##  0.442  0.417  0.404  0.368  0.340  0.339  0.306  0.279  0.271  0.205  0.179 
## 
## Eigenvalues for unconstrained axes:
##    PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8 
## 17.156  9.189  7.585  6.064  5.730  4.843  4.518  4.105 
## (Showing 8 of 77 unconstrained eigenvalues)
```

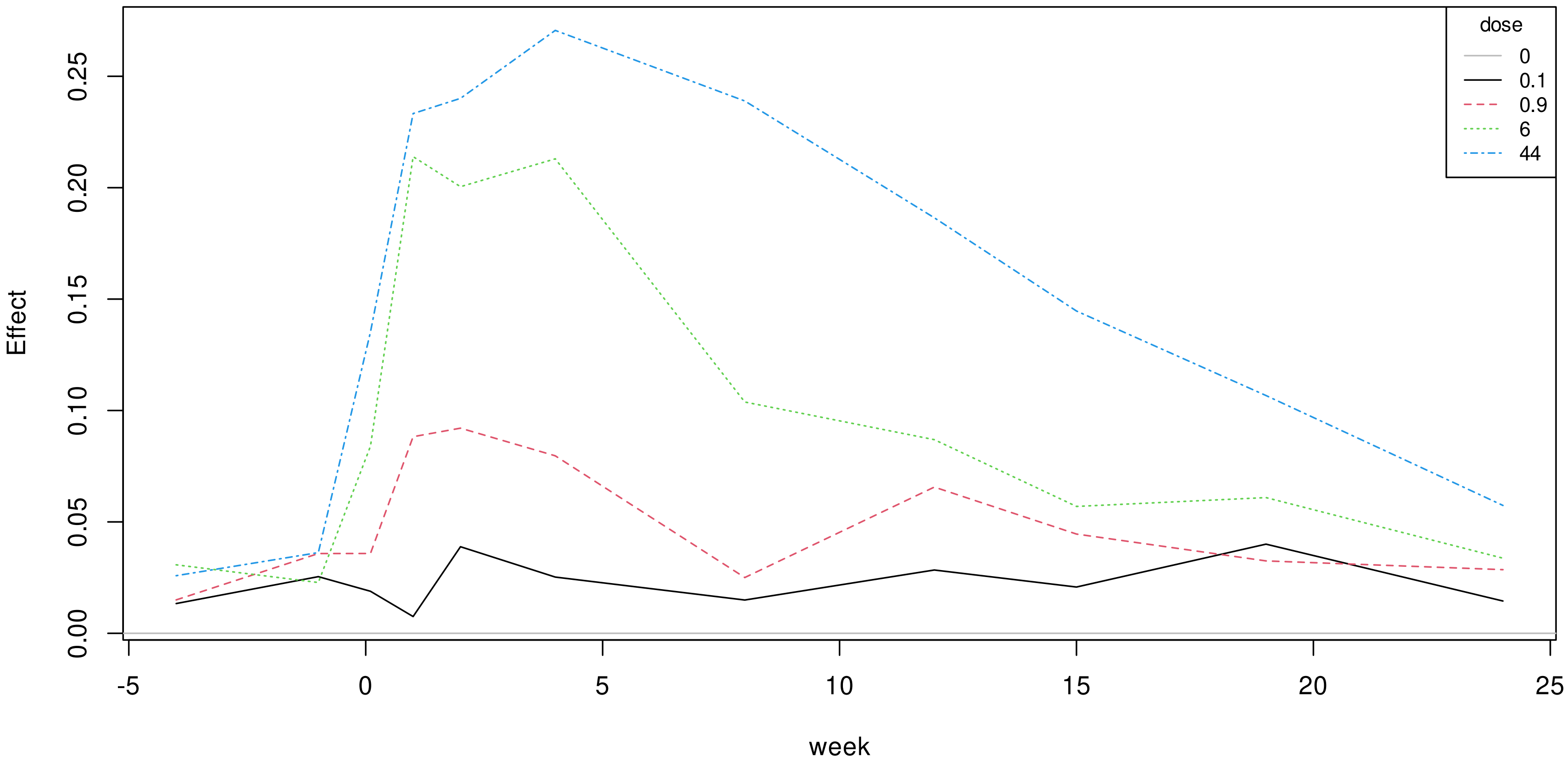
PRC — pyrifos example

```r
ctrl <- how(plots = Plots(strata = ditch,type = "free"),
    within = Within(type = "series"), nperm = 99)
anova(mod, permutations = ctrl, first = TRUE)
```

```
## Permutation test for rda under reduced model
## Plots: ditch, plot permutation: free
## Permutation: series
## Number of permutations: 99
##
## Model: prc(response = pyrifos, treatment = dose, time = week)
##          Df Variance      F Pr(>F)
## RDA1      1   25.282 15.096   0.01 **
## Residual 77  128.959
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

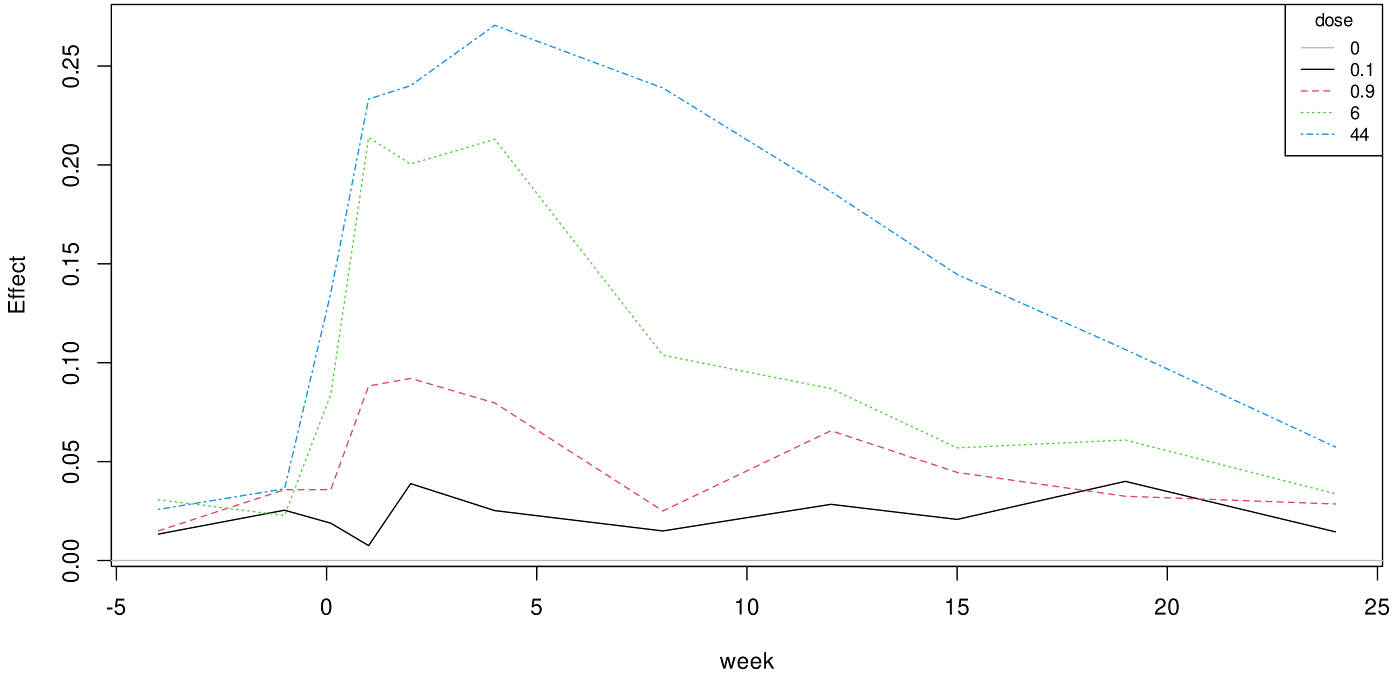
PRC — pyrifos example

```
plot(mod, species = FALSE, legpos = "topright")
```



PRC — pyrifos example

```
plot(mod, species = FALSE, legpos = "topright")
logabu <- colSums(pyrifos)
scrs <- scores(mod, display = "species", choices = 1)
linestack(scrs[logabu > 150, , drop = FALSE]); axis(side = 2)
```



Other stuff

Diagnostics for constrained ordinations

vegan provides a series of diagnostics to help assess the model fit

- `goodness()`
- `inertcomp()`
- `spenvcor()`
- `intersetcor()`
- `vif.caa()`

Diagnostics | goodness of fit

`goodness()` computes a goodness of fit statistic for species or sites, controlled by argument `display`

Gives the cumulative proportion of variance explained by each axis

```
upr ← cca(varespec ~ ., data = varechem)
lwr ← cca(varespec ~ 1, data = varechem)
set.seed(1)

mods ← ordistep(lwr, scope = formula(upr), trace = 0)
head(goodness(mods))
```

```
##                  CCA1        CCA2      CCA3
## Callvulg 0.0062471656 0.318907619 0.8254657
## Empenigr 0.1164701677 0.137604904 0.1953245
## Rhodtome 0.0999089739 0.169697909 0.1824153
## Vaccmyrt 0.2361482843 0.240516323 0.2406730
## Vaccviti 0.1523704591 0.156502301 0.2110550
## Pinusylv 0.0009244423 0.004802076 0.0060096
```

Diagnostics | inertia decomposition

`inertcomp()` decomposes the variance in samples or species in partial, constrained, and unconstrained components

- `statistic = "explained` (default) gives the decomposition in terms of variance
- `statistic = "distance"` gives decomposition in terms of the the residual distance

```
head(inertcomp(mods, proportional = TRUE))
```

```
##                CCA        CA
## Callvulg 0.8254657 0.1745343
## Empenigr 0.1953245 0.8046755
## Rhodtome 0.1824153 0.8175847
## Vaccmyrt 0.2406730 0.7593270
## Vaccviti 0.2110550 0.7889450
## Pinusylv 0.0060096 0.9939904
```

Diagnostics | species-environment correlations

`spenvcor()` returns the (weighted) correlation between the weighted average-based and the linear combination-based sets of site scores

A *poor* measure of goodness of fit. Sensitive to

- outliers (like all correlations)
- overfitting (using too many constraints)

Better models can have poorer species-environment correlations

```
spenvcor(mods)
```

```
##      CCA1      CCA2      CCA3 
## 0.8554793 0.8131627 0.8792221
```

Diagnostics | interset correlations

`intersetcor()` returns the (weighted) correlation between the weighted average-based site scores and each constraint variable

Another *poor* diagnostic

- correlation based
- focuses on a single constraint--axis combination at a time

```
intersetcor(mods)
```

```
##           CCA1       CCA2      CCA3
## Al  0.7356445 -0.1304293 0.4260453
## P  -0.3588931 -0.6109601 0.4478786
## K  -0.3767902 -0.1339051 0.7759566
```

Vector fitting (`envfit()`) or biplot scores (`scores(model, display = "bp")`) are better alternatives

References

- Anderson, M.J., 2001. A new method for non-parametric multivariate analysis of variance. Austral Ecol. 26, 32–46
- Anderson, M.J., 2006. Distance-based tests for homogeneity of multivariate dispersions. Biometrics 62, 245–253
- Anderson, M.J., Walsh, D.C.I., 2013. PERMANOVA, ANOSIM, and the Mantel test in the face of heterogeneous dispersions: What null hypothesis are you testing? Ecol. Monogr. 83, 557–574
- Anderson, M.J., Walsh, D.C.I., Robert Clarke, K., Gorley, R.N., Guerra-Castro, E., 2017. Some solutions to the multivariate Behrens-Fisher problem for dissimilarity-based analyses. Aust. N. Z. J. Stat. 59, 57–79
- Blanchet, F.G., Legendre, P., Borcard, D., 2008. Forward selection of explanatory variables. Ecology 89, 2623–2632
- Legendre, P., Anderson, M.J., 1999. Distance-based redundancy analysis: testing multispecies responses in multifactorial ecological experiments. Ecol. Monogr. 69, 1–24
- McArdle, B.H., Anderson, M.J., 2001. Fitting Multivariate Models to Community Data: A Comment on Distance-Based Redundancy Analysis. Ecology 82, 290–297
- Warton, D.I., Wright, S.T., Wang, Y., 2012. Distance-based multivariate analyses confound location and dispersion effects. Methods Ecol. Evol. 3, 89–101