RDA, tb-RDA, CCA & db-RDA (constrained ordination)

Theory R functions **Examples**

Example 1: How much variation explain soil pH and soil depth in the Vltava valley vegetation? (tb-RDA)

In this example, we will apply constrained ordination (tb-RDA) on Vltava river valley dataset. We will ask how much variance in species composition can be explained by two variables, soil pH and soil depth. Both are important factors for plant growth, and moreover, in the study area, they are somewhat correlated (shallower soils have lower pH since the prevailing geological substrate is acid).

First, upload the Vltava river valley data:

```
vltava.spe <- read.delim
('https://raw.githubusercontent.com/zdealveindy/anadat-r/master/data/vltava-spe.txt
', row.names = 1)
vltava.env <- read.delim
('https://raw.githubusercontent.com/zdealveindy/anadat-r/master/data/vltava-env.txt
')

spe <- vltava.spe  # rename variables to make them shorter
env <- vltava.env[, c('pH', 'SOILDPT')]  # select only two explanatory variables
```

Upload library `vegan` and calculate tb-RDA based on Hellinger pre-transformed species composition data. Note that since the original data represent estimates of percentage cover, it is better to log transform these values first before Hellinger transformation is done (using function `log1p`, which calculates *log (x+1)* to avoid *log (0)*):

```
library (vegan)
spe.log <- log1p (spe)  # species data are in percentage scale which is strongly
rightskewed, better to transform them
spe.hell <- decostand (spe.log, 'hell')  # we are planning to do tb-RDA, this is
Hellinger pre-transformation
tbRDA <- rda (spe.hell ~ pH + SOILDPT, data = env)  # calculate tb-RDA with two
explanatory variables
tbRDA
```

The result printed by `rda` function is the following:

```
Call: rda(formula = spe.hell ~ pH + SOILDPT, data = env)

              Inertia Proportion Rank
Total         0.70476    1.00000
Constrained   0.06250    0.08869    2
Unconstrained 0.64226    0.91131   94
Inertia is variance

Eigenvalues for constrained axes:
   RDA1    RDA2
0.04023 0.02227
```

```
Eigenvalues for unconstrained axes:
    PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8
0.07321 0.04857 0.04074 0.03144 0.02604 0.02152 0.01917 0.01715
(Showed only 8 of all 94 unconstrained eigenvalues)
```

and, the same with comments:

Method: RDA *(tb-RDA with Hell. stand + log-transf.)*

	Inertia	Proportion	Rank
Total	0.70476	1.00000	
Constrained	0.06250	0.08869	2
Unconstrained	0.64226	0.91131	94

Inertia is variance

Eigenvalues for **constrained** axes:

RDA1	RDA2
0.04023	0.02227
5.7%	*3.2%*

← Two quantitative explanatory variables = two constrained axes

Eigenvalues for **unconstrained** axes:

PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
0.07321	0.04857	0.04074	0.03144	0.02604	0.02152	0.01917	0.01715
10.4%	*6.9%*	*5.8%*	*4.5%*	*3.7%*	*3.1%*	*2.7%*	*2.4%*

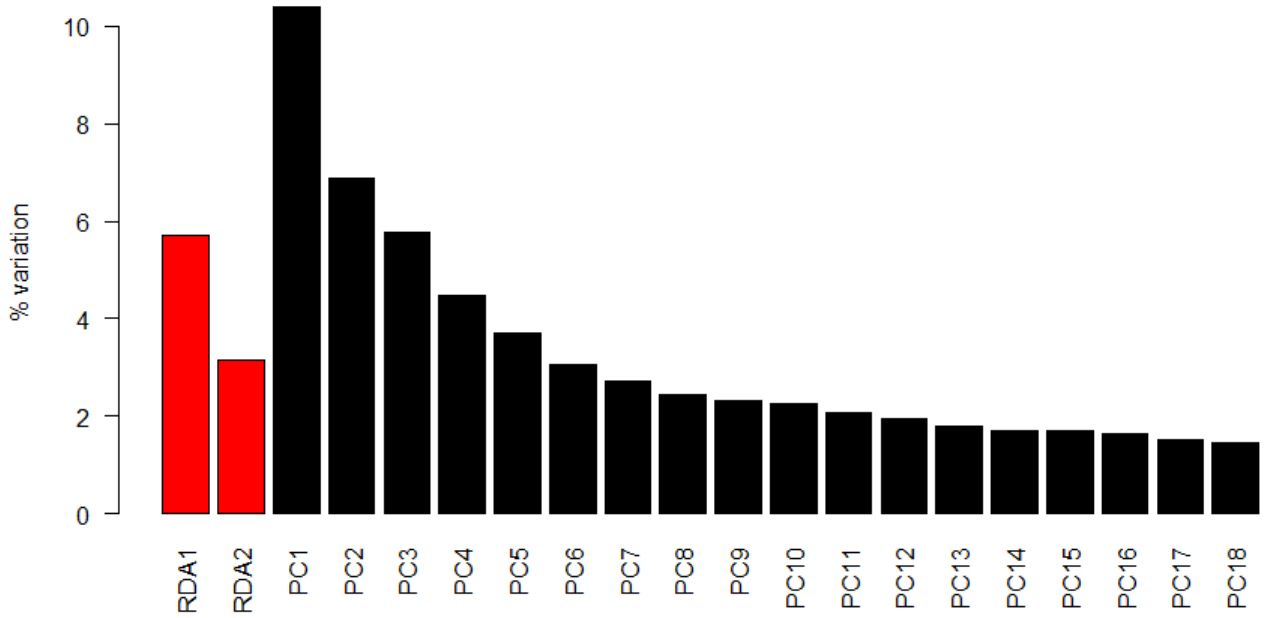
Vltava dataset has 97 samples and 274 species; 97 is the lower number - > ordination has 97-1=96 axes; two of these axes are constrained (because there are two quantitative explanatory variables), the rest (94) are unconstrained

The two variables explain 8.9% of the variance (the row `Constrained` and column `Proportion` in the table above, can be calculated also as the sum of eigenvalues for the constrained axes divided by total variance (inertia): (0.04023+0.02227) /0.70476=0.08869. The first constrained axis (RDA1) explains 0.04023/0.70476=5.7% of the variance, while the second (RDA2) explains 0.02227/0.70476=3.2%. Note that the first unconstrained axis (PC1) represents 0.07321/0.70476=10.4% of the total variance, which is more than the variance explained by both explanatory variables together; the first two unconstrained explain (0.07321+0.04857)/0.70476=17.3%. This means that the dataset may be structured by some strong environmental variable(s) different from pH and soil depth (we will check this below).

The relationship between the variation represented by individual (constrained and unconstrained) ordination axes can be displayed using the barplot of percentage variance explained by individual axes (ie their eigenvalue divided by total inertia):

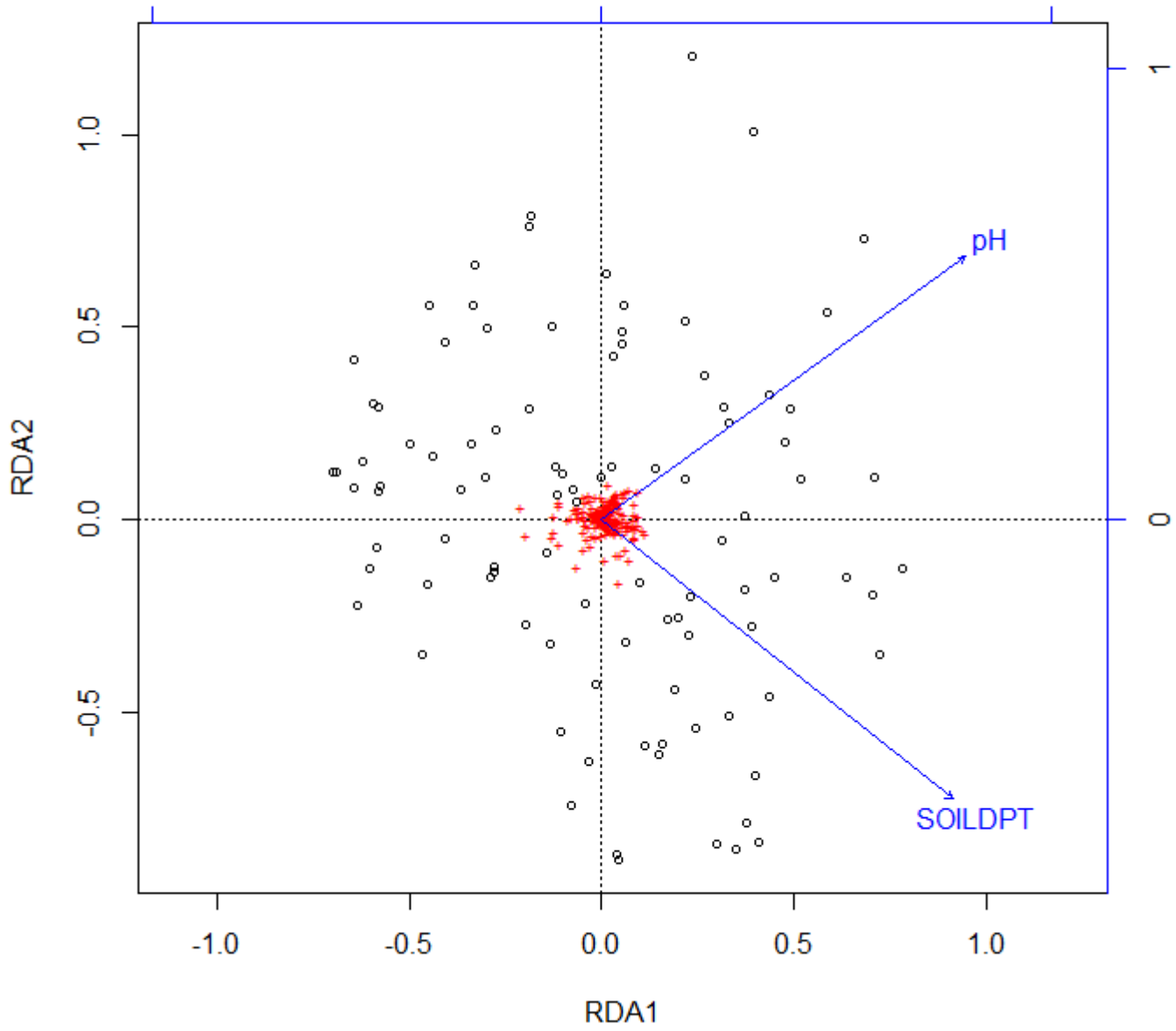
```
constrained_eig <- tbRDA$CCA$eig/tbRDA$tot.chi*100
unconstrained_eig <- tbRDA$CA$eig/tbRDA$tot.chi*100
expl_var <- c(constrained_eig, unconstrained_eig)
barplot (expl_var[1:20], col = c(rep ('red', length (constrained_eig)), rep
('black', length (unconstrained_eig))),
         las = 2, ylab = '% variation')
```

(note that all information about the eigenvalues and total inertia is in the object calculated by `vegan`'s ordination function (`rda` in this case, stored in the list `tbRDA`), you just need to search a bit inside to find it - consider using the function `str` to check the structure of tbRDA first).



Let's see the ordination diagram:

```
ordiplot (tbRDA)
```



Example 2: What may be an ecological meaning of unconstrained axes in constrained ordination? (tb-RDA)

This example is a direct continuation of Example 1 above.

When checking results of tb-RDA on Vltava data in Example 1, we may notice that the first and second **unconstrained** ordination axes represent considerably more variation than the two constrained axes:

```
vltava.spe <- read.delim
('https://raw.githubusercontent.com/zdealveindy/anadat-r/master/data/vltava-spe.txt
', row.names = 1)
vltava.env <- read.delim
('https://raw.githubusercontent.com/zdealveindy/anadat-r/master/data/vltava-env.txt
')

spe <- vltava.spe  # rename variables to make them shorter
```

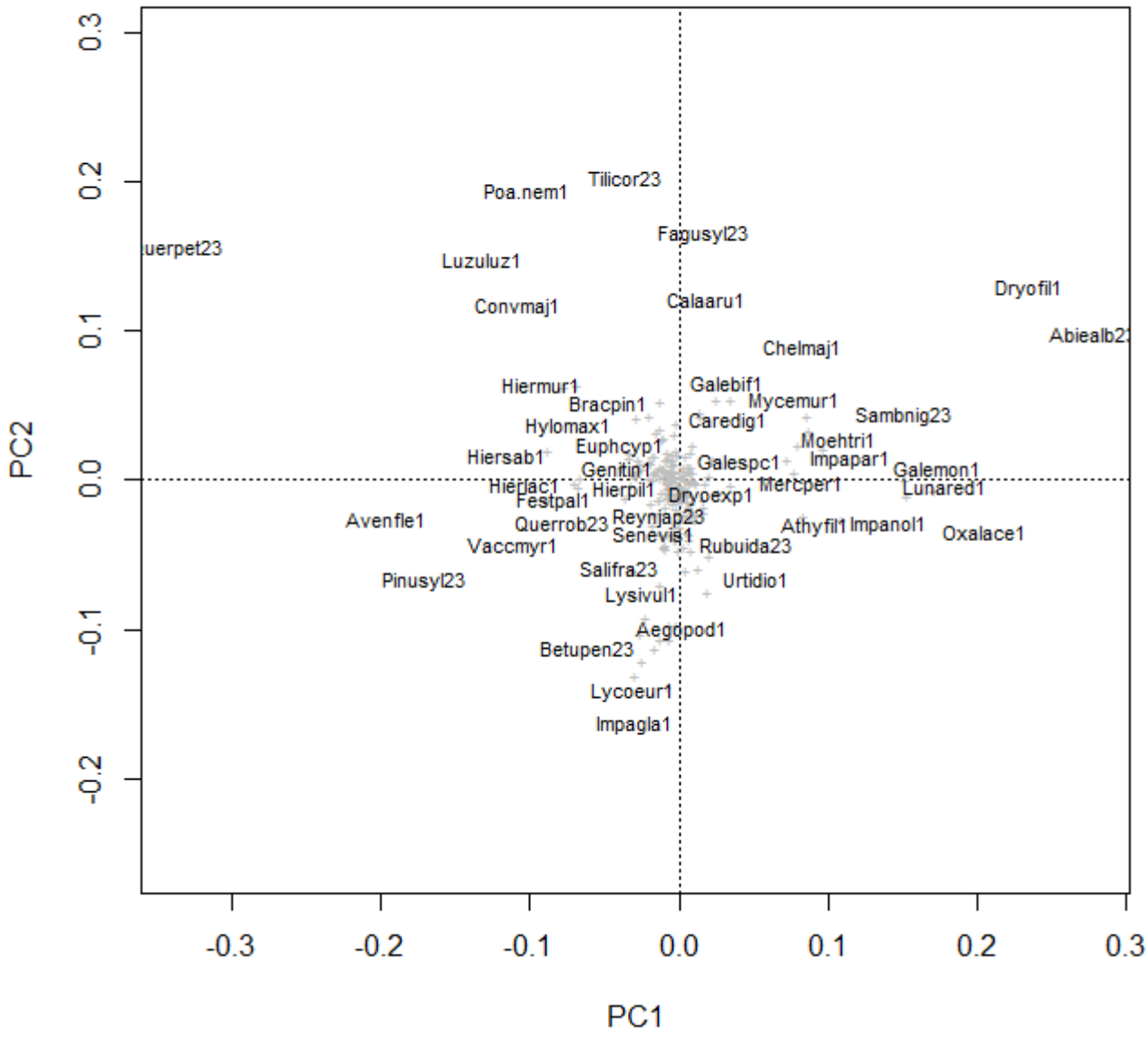
```
env <- vltava.env[, c('pH', 'SOILDPT')]  # select only two explanatory variables

library (vegan)
spe.log <- log1p (spe)
spe.log.hell <- decostand (spe.log, 'hell')
tbRDA <- rda (spe.log.hell ~ pH + SOILDPT, data = env)
head (summary (tbRDA))  # prints first lines of the summary of tbRDA
```

```
Call: rda(formula = spe.log.hell ~ pH + SOILDPT, data = env)

Partitioning of variance:
              Inertia Proportion
Total          0.7048    1.00000
Constrained    0.0625    0.08869
Unconstrained  0.6423    0.91131

Eigenvalues, and their contribution to the variance

Importance of components:
                          RDA1    RDA2     PC1     PC2     PC3     PC4     PC5
PC6
Eigenvalue            0.04023 0.02227 0.07321 0.04857 0.04074 0.03144 0.02604
0.02152
Proportion Explained  0.05709 0.03160 0.10388 0.06891 0.05781 0.04461 0.03695
0.03054
Cumulative Proportion 0.05709 0.08869 0.19257 0.26149 0.31929 0.36391 0.40086
0.43139
...
```

From the output above (rows “Proportion Explained”), you can see that the first and the second constrained axes (RDA1, RDA2) explain 5.7 and 3.1% of the variation, while the first and second unconstrained axes (PC1, PC2) represent 10.4 and 6.9% of the variation, respectively. This indicates that after accounting for explanatory variables (soil pH and soil depth), considerable variation in species composition remains in the data.

One thing we can do is create an ordination diagram using the first and second unconstrained ordination axes (PC1, PC2), project species onto it, and hope that they can help us understand which environmental variables may be behind these gradients. Of course, this assumes that you have at least marginal knowledge about the ecological behaviour of these species (I think I do, since I spent several years working on the vegetation of these valleys, so let's try).

```
ordiplot (tbRDA, display = 'species', choices = c(3,4), type = 'n')
orditorp (tbRDA, display = 'species', choices = c(3,4), pcol = 'grey', pch = '+')
```



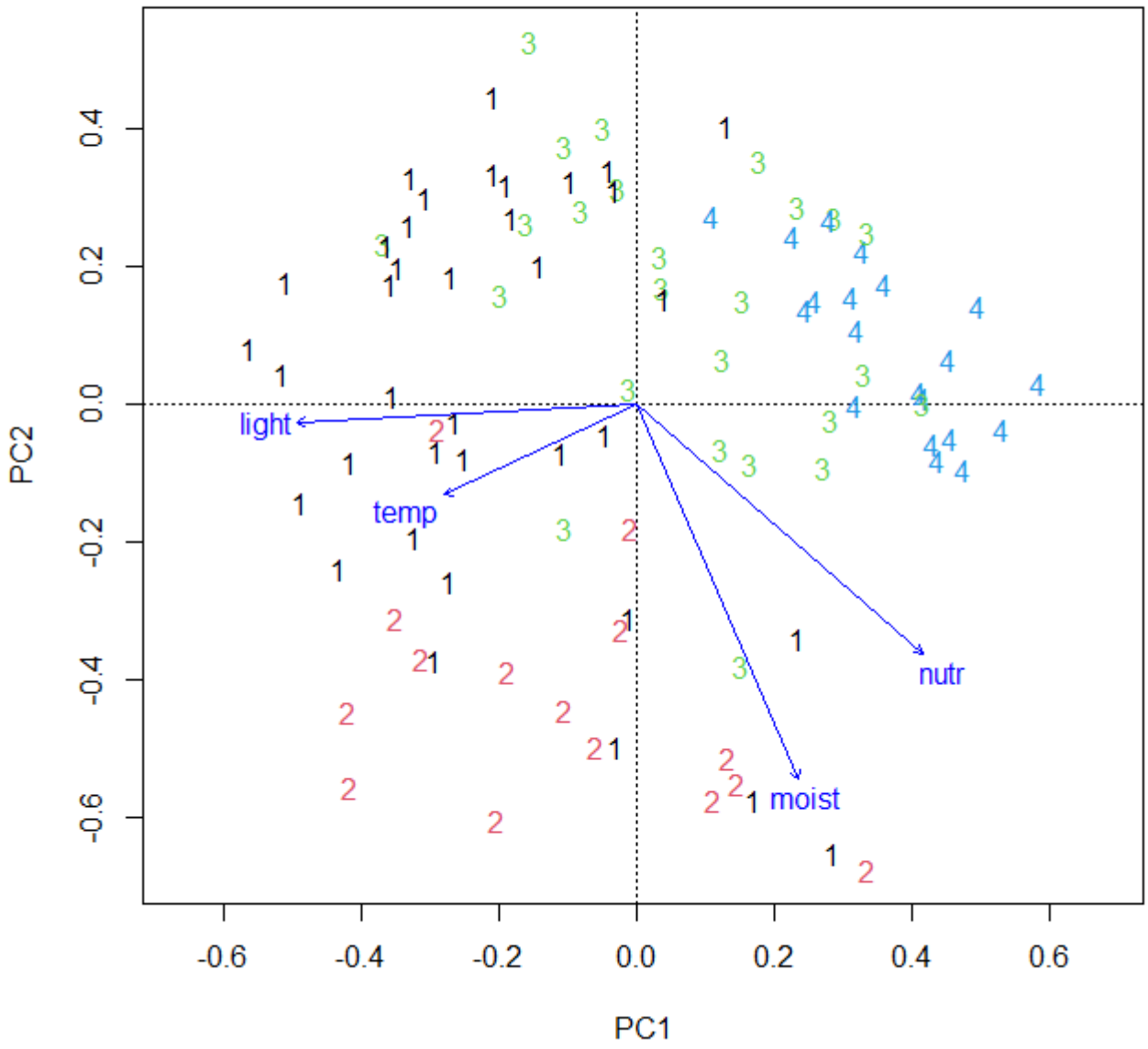
There is 274 species in the Vltava dataset, so I used `orditorp` function to draw only some of them (see Ordination diagrams > Examples for details on how to use this function). Also, since RDA is a linear function, the species should be displayed by vectors, not as centroids, but if we want to display only some of the species, this option in R is not so simple (you may need to use `envfit` function to project vectors of species onto the ordination diagram). Species are displayed by their abbreviation, where the first four letters are from the genus name, the last four letters from the species name, and the numbers indicate the layer (1 for herb layer, 23 for merged shrub and tree layer; for example, `Tilicor23` is an abbreviation of *Tilia cordata* in the shrub and tree layer).

When interpreting the first (horizontal) unconstrained axis (PC1), we can see that the left part (negative scores) is related to high abundances of *Quercus petraea* (`Querpet23`) and *Pinus sylvestris* (`Pinusyl23`) in the shrub and tree layer, and *Avenella flexuosa* in the herb layer (`Avefle1`), while the right part (positive scores) is related to high abundances of *Abies alba* in shrub and tree layer (`Abiealb23`), and *Dryopteris filix-mas* (`Dryofil1`), *Galeobdolon montanum* (`Galemon1`) and *Lunaria rediviva* (`Lunared1`) in the herb layer. These species seem to indicate opposite ends of the gradient of light (light-demanding species on the left, shade-tolerant on the right) and nutrient availability (species tolerating the lack of nutrients on the left, and nutrient-demanding species on the right). When interpreting the second (vertical) unconstrained axis (PC2), the lower part (negative scores) is related to high abundances of *Impatiens glandulifera* (`Impagla1`), *Lycopus europaeus*

(Lycoeur1) and *Aegopodium podagrarium* (Aegopod1) in the herb layer, while the upper part (positive scores) are related to high abundances of *Tilia cordata* (Tilicor23) and *Fagus sylvatica* (Fagusyl23) in the shrub and tree layer, and *Poa nemoralis* (Poa.nem1), *Luzula luzuloides* (Luzuluz1) and *Convalaria majalis* (Convmaj1) in the herb layer; this axis seem to negatively correlate with the gradient of moisture, with species indicating wet soil of the alluvial forest at the lower part of the axis and mesic species in the upper part.

If we are not familiar with the ecological requirements of the species, but we have available external information about the species ecology, we can use this for the interpretation of the ordination axes. For the Vltava dataset, we have available Ellenberg-type indicator values (for soil pH, moisture, nutrients, air temperature, light availability, and continentality), and we can calculate their relationship to *species* scores of selected ordination axes, and project them onto the ordination diagram of the first and second unconstrained ordination axes. For this, we can use the function `envfit`, which, in standard settings, calculates the relationship between supplementary environmental variables (e.g., variables related to *sites*, not *species*) and ordination axes. The same function can also be used to relate species attributes (such as Ellenberg-type indicator values) to species scores along the ordination axes; you need to make sure that the matrix of species attributes has species in the same order as the matrix of species composition, and also that you include the argument `display = "species"`. Also, since for some of the species the Ellenberg indicator value is not available (NA), you need to include the argument `na.rm = TRUE` to remove these species from the calculation.

```
vltava.ell <- read.delim
('https://raw.githubusercontent.com/zdealveindy/anadat-r/master/data/vltava-ell.txt
', row.names = 1)
ef_ell <- envfit (tbRDA, vltava.ell, choices = c(3,4), display = 'species', na.rm =
TRUE)
ordiplot (tbRDA, choices = c(3,4), type = 'n')
points (tbRDA, choices = c(3,4), display = 'sites', pch = as.character
(vltava.env$GROUP), col = vltava.env$GROUP)
plot (ef_ell, p.max = 0.05)  # to display only significant Ell. values
```




```
ef_ell
```

```
***VECTORS

           PC1      PC2     r2 Pr(>r)
light -0.99872 -0.05056 0.2201  0.001 ***
temp  -0.90526 -0.42486 0.0859  0.014 *
cont   0.31591  0.94879 0.0229  0.311
moist  0.39811 -0.91734 0.3164  0.001 ***
react  0.61322 -0.78991 0.0102  0.615
nutr   0.75486 -0.65589 0.2736  0.001 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Permutation: free
Number of permutations: 999

175 observations deleted due to missingness
```

The highest R^2 of the regression with the first two axes has moisture, nutrients, light, and temperature (in this order), with light and temperature associated mostly with the first unconstrained axis and the moisture mostly with the second axis; nutrients are associated with both unconstrained axes. It seems that these ecological factors, not related to soil pH and soil depth, are important for the studied vegetation, but were not included as explanatory variables. This interpretation aligns well with the one based on the relationship between axes and species abundances listed above. Some environmental variables related to those four Ellenberg-type indicator values were actually measured (e.g., HEATLOAD can approximate temperature), and after this analysis we may want to include them as explanatory variables in the model if we plan to reanalyze the data.

Note that an alternative way to interpret ordination axes using Ellenberg-type indicator values would be to calculate mean Ellenberg indicator values for each sample (in a similar way to how community weighted means for traits are calculated, see Analysis of species attributes). These mean values can then be related to site scores. However, since mean Ellenberg indicator values are calculated from species composition, and site scores are also calculated by species composition, it would not be correct to test the significance of their relationship, as they are not independent (this is a type of spurious correlation, about which I write in Zelený 2018, section 2.4). There is a way to solve this problem (modified permutation test), but if species values can be related to species scores directly, it may be a better way to do so.

Example 3: How different are cookies from pastries and pizzas (CCA)?

This example is using Difference between cookies, pastries and pizzas dataset, which I found on reddit.com, posted by author everest4ever. The author attended the Christmas party at his office with a "Christmas cookie competition", which sparked a "huge debate about what are eligible entries for the cookie competition (e.g., are mini-pizzas cookies?)". The author decided to approach the discussion rigorously and did the following: "I scraped 1931 recipes from the Food Network that contain the keywords cookies (my group of interest), pastry, or pizza (two control groups). Next, I extracted the ingredient list and pooled similar ingredients together (e.g.,*salt*, *seasalt*, *Kosher salt*), coming up with a total of 133 unique ingredients. I ended up with a 1931×133 matrix, where each row is one recipe, and each column is whether this recipe contains a certain ingredient (0 or 1)". The author did PCA analysis on the data accompanied by some clustering and predictions, just to prove that "NO IAN AND JOSEPH YOUR FUCKING EGG TARTS AREN'T COOKIES, NO MATTER HOW GOOD THEY WERE!!".

I think we can also use this dataset for a simple constrained ordination exercise.

First, import the data:

```
recipes.ingr <- read.delim
('https://raw.githubusercontent.com/zdealveindy/anadat-r/master/data/cookie_dataset
_everest4ever_composition.txt', row.names = 1)
recipes.type <- read.delim
('https://raw.githubusercontent.com/zdealveindy/anadat-r/master/data/cookie_dataset
_everest4ever_type.txt', row.names = 1)
```

Data represent a matrix of presence/absence of different ingredients in individual recipes (each row of 'recipes.ingr' matrix is a recipe, each column represents one ingredient). To determine how each recipe is classified, we need a variable `type_of_food` in the data frame `recipes.type`.

To get familiar with data, let's first calculate DCA:

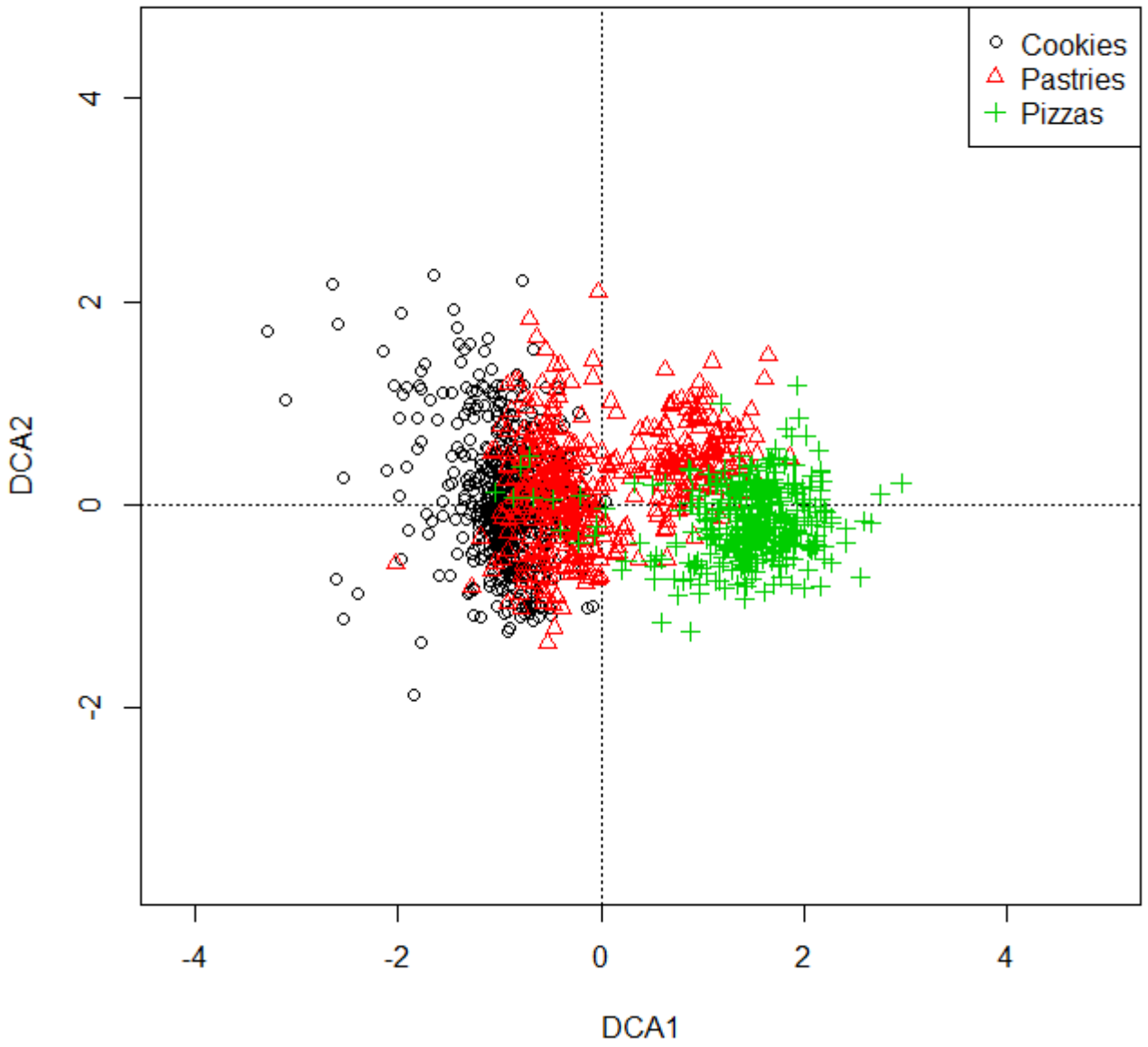
```
library (vegan)
DCA <- decorana (recipes.ingr)
DCA
```

```
Call:
decorana(veg = recipes.ingr)
```

```
Detrended correspondence analysis with 26 segments.
Rescaling of axes with 4 iterations.

                  DCA1   DCA2   DCA3   DCA4
Eigenvalues     0.5633 0.2598 0.2224 0.2185
Decorana values 0.5918 0.2622 0.2352 0.2138
Axis lengths    6.2276 4.1302 5.6396 3.5449
```

The output shows that the length of the first axis is 6.2 S.D. units, so unimodal ordination methods are advisable. The ordination diagram, which displays recipes of cookies, pastries, and pizzas by different symbols and colours, is more informative:

```
type_num <- as.numeric (as.factor (recipes.type$type_of_food))

ordiplot (DCA, type = 'n')
points (DCA, display = 'sites', col = type_num, pch = type_num)
legend ('topright', col = 1:3, pch = 1:3, legend = levels (as.factor
(recipes.type$type_of_food)))
```



(the variable `type_num` contains numerical values 1, 2 and 3 in place of Cookies, Pastries and Pizzas from the original `type_of_food` variable in `recipes.type` data frame, so as we can use these values as colors and symbols in ordination diagram).

It seems that pizzas are somewhat different from the rest (although a part of the pastries is somewhat close), while pastries and cookies form a cloud with a large overlap. Let's try to ask the following question: can the classification of a recipe into cookies/pastries/pizzas (done largely subjectively by authors of those recipes based on their opinion of what each category item should look like) explain the difference in "ingredients composition" of individual recipes? This is a task for constrained ordination. We use CCA for constrained ordination, with recipes as the dependent variable and assignment into the type as the explanatory variables. Note that explanatory variable is categorical with three levels (`pastry`, `cookie`, `pizza`):

```
type <- recipes.type$type_of_food
CCA <- cca (recipes.ingr ~ type)
CCA
```

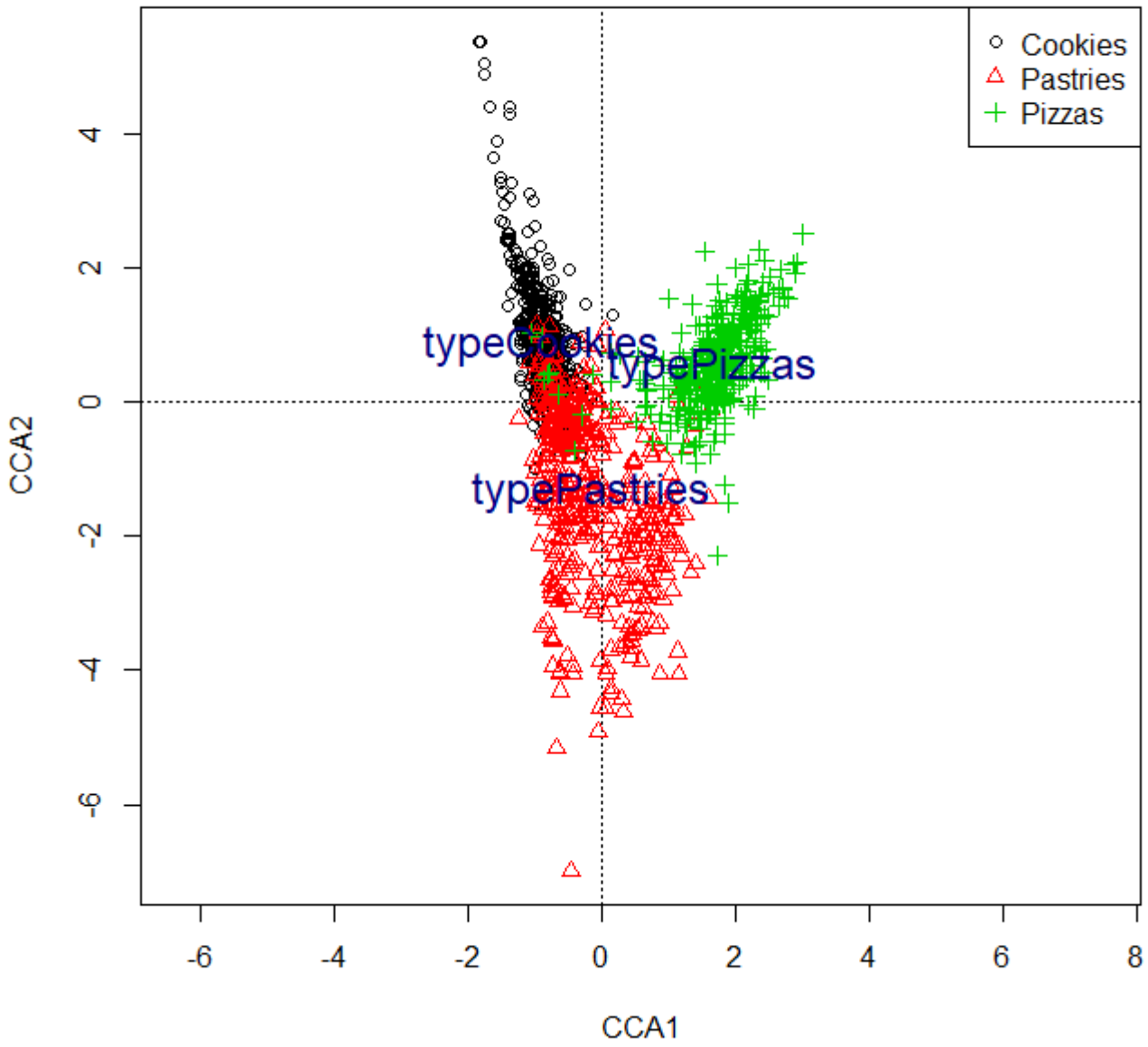
```
Call: cca(formula = recipes.ingr ~ type)
```

```
               Inertia Proportion Rank
Total         14.28961    1.00000
Constrained    0.60311    0.04221    2
Unconstrained 13.68650    0.95779  132
Inertia is scaled Chi-square

Eigenvalues for constrained axes:
  CCA1   CCA2
0.4649 0.1382

Eigenvalues for unconstrained axes:
    CA1     CA2     CA3     CA4     CA5     CA6     CA7     CA8
0.30447 0.28409 0.26249 0.25278 0.23942 0.21521 0.21069 0.20674
(Showed only 8 of all 132 unconstrained eigenvalues)
```

(Note that I saved the column `type_of_food` from the data frame `recipes.type` into a variable `type`, because this will make it simple to display individual factor levels onto an ordination diagram (the levels are displayed as the Name_of_variableName_of_category, which would be too long with the original names).

We got two constrained axes (explanatory variable is qualitative with three factor levels -> number of contrained axes = number of levels - 1), the first exlaining more than three time more than the second (eig_{CCA1} = 0.4649, eig_{CCA2} = 0.1382, which means that the first axis represents 0.4649/14.28961 = 3.3 % of variance (eigenvalue/total inertia), while the second 0.1382/14.28961 = 1.0%. The ordination diagram shows that the first axis mostly separates pizzas (right) and cookies+pastries (left), while the second axis mostly separates cookies (up) from pastries (bottom):

```
ordiplot (CCA, display = c('si', 'cn'), type = 'n')
points (CCA, display = 'si', col = type_num, pch = type_num)
text (CCA, display = 'cn', col = 'navy', cex = 1.5)
legend ('topright', col = 1:3, pch = 1:3, legend = levels (as.factor
(recipes.type$type_of_food)))
```



A few more things. First, we should ask whether the CCA ordination is significant and whether it is worth interpreting it:

```
anova (CCA)
```

```
Permutation test for cca under reduced model
Permutation: free
Number of permutations: 999

Model: cca(formula = recipes.ingr ~ type)
          Df ChiSquare      F Pr(>F)
Model      2    0.6031 42.479  0.001 ***
Residual 1928   13.6865
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Indeed, it is. And how about individual axes? Are they both significant? We will use the argument `by = “axis”`

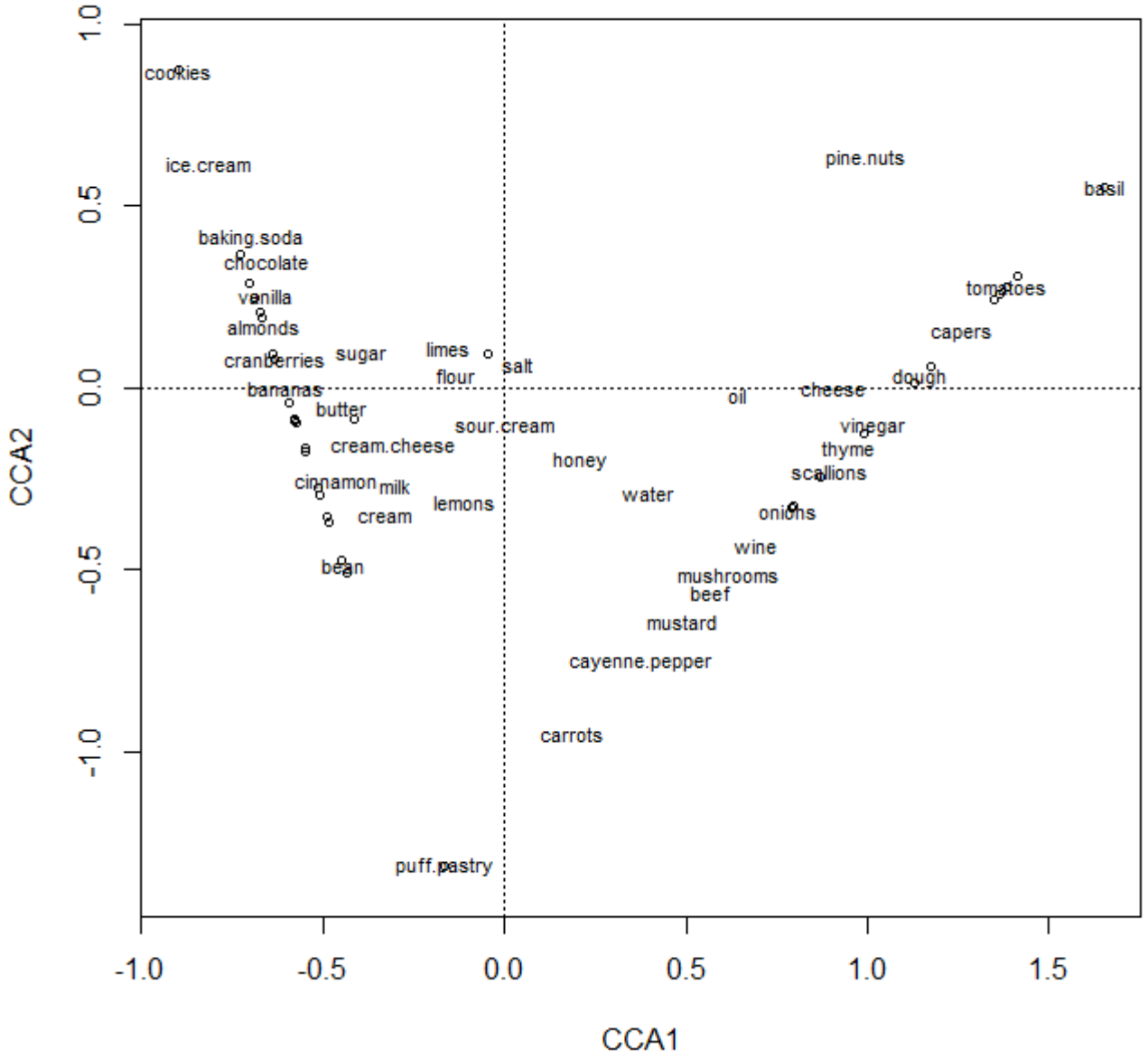
in the function anova - see this explanation what it means:

```
anova (CCA, by = 'axis')
```

```
Permutation test for cca under reduced model
Forward tests for axes
Permutation: free
Number of permutations: 999

Model: cca(formula = recipes.ingr ~ type)
          Df ChiSquare      F Pr(>F)
CCA1       1    0.4649 65.493  0.001 ***
CCA2       1    0.1382 19.465  0.001 ***
Residual 1928   13.6865
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Both axes are significant, which means that even the distinction between cookies and pastries along the second axis is important. So I would say, without further testing, that not only is pizza (quite obviously) different from cookies, but also the much more ambiguous category of pastries differs in the ingredients they use. Btw, let's see these ingredients (display species in CCA ordination diagram):

```
ordiplot (CCA, display = c('sp', 'cn'), type = 'n')
orditorp (CCA, display = 'sp', priority = colSums (recipes.ingr))
```



Note that I did not display all 133 ingredients ("species"); otherwise, the diagram gets too cluttered. I used the low-level graphical function `orditorp`, which adds only some labels and draws others as symbols. It has an argument `priority` (the species with the highest priority will be more likely plotted as text, with lower as symbols, if there is not enough space); the priority here is the overall frequency of ingredients in the dataset (`colSums` applied on the `recipes.ingr` data frame). The diagram shows a clear triangle, with each corner representing one type of food. There is a gradient of ingredients connecting pizzas with pastries and pastries with cookies, but almost no ingredients connecting pizzas and cookies (except `pine nuts`, which could perhaps make it into both pizzas and cookies, but I have no idea). Some ingredients are shared among all three (obviously *water* and seems that also *honey*), some are only for that type of food (*basil* for pizza, *puff pastry* for pastries, and *cookies*(??) and *ice cream* for cookies. Please, see the diagram and guess which item in your opinion should be where (*carrots* in pastry? not sure...).

From:
https://anadat-r.davidzeleny.net/ - **Analysis of community ecology data in R**

Permanent link:
https://anadat-r.davidzeleny.net/doku.php/en:rda_cca_examples

Last update: **2026/04/06 18:54**