PCA & tb-PCA (linear unconstrained ordination)

Theory R functions **Examples**

Example 1: PCA on species composition data

Grasslands dataset contains relatively homogenous species composition, since the sampled grassland patches all belong to the same or very closely related vegetation types, which makes this dataset suitable for PCA. To ensure that it passes also the "length of the first DCA axis" rule, let's first calculate DCA on it:

```
grasslands.spe <- read.delim
('http://www.davidzeleny.net/anadat-r/data-download/grasslands-spe.txt', row.names
= 1)
grasslands.env <- read.delim
('http://www.davidzeleny.net/anadat-r/data-download/grasslands-env.txt')

grasslands.spe.log <- log1p (grasslands.spe)

library (vegan)
decorana (grasslands.spe.log)
```

```
Call:
decorana(veg = grasslands.spe.log)

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

                  DCA1   DCA2   DCA3   DCA4
Eigenvalues     0.2652 0.2285 0.1485 0.1545
Decorana values 0.2887 0.2228 0.1676 0.1383
Axis lengths    2.6961 2.1441 2.5704 1.8207
```

The length of the first DCA axis is 2.7 S.D. (i.e. < 3. S.D.), and data are thus suitable for linear ordination methods.

```
PCA <- rda (grasslands.spe.log)  # if rda is used without explanatory variabels, it
calculates PCA
PCA
```

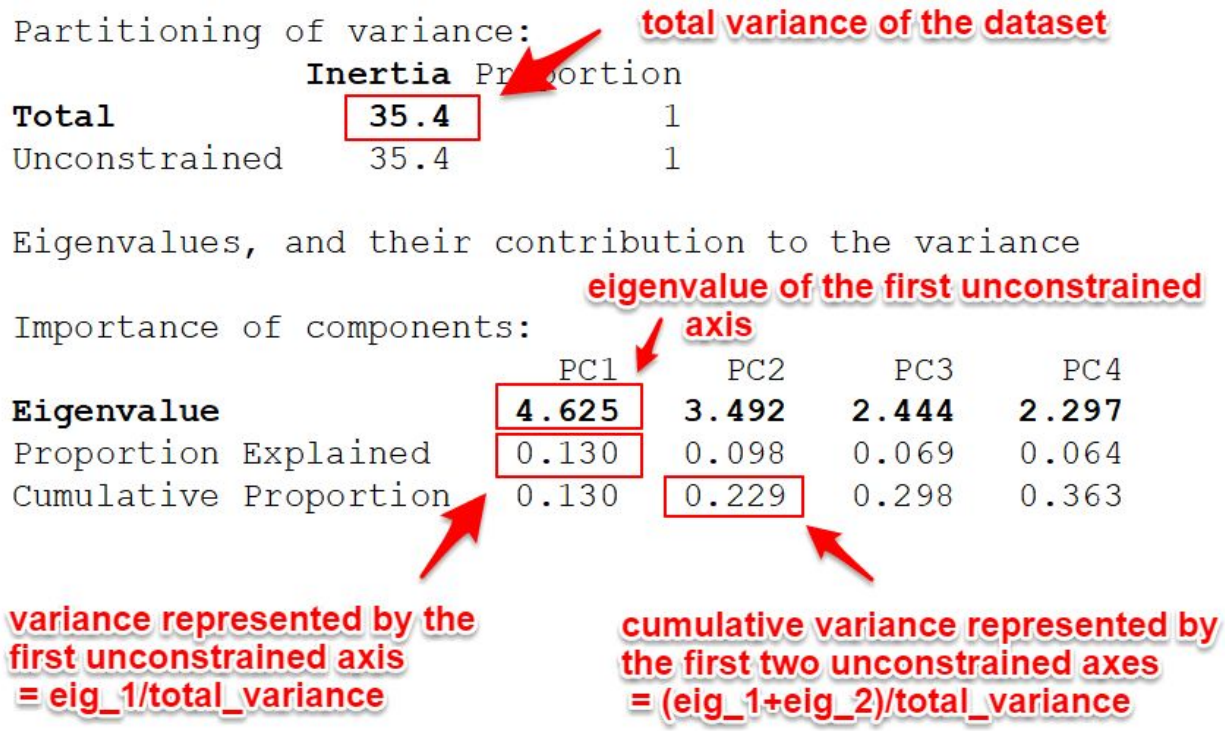
```
Call: rda(X = grasslands.spe.log)

              Inertia Rank
Total            35.4
Unconstrained    35.4   47
Inertia is variance

Eigenvalues for unconstrained axes:
  PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8
4.625 3.492 2.445 2.297 1.648 1.543 1.479 1.331
(Showed only 8 of all 47 unconstrained eigenvalues)
```

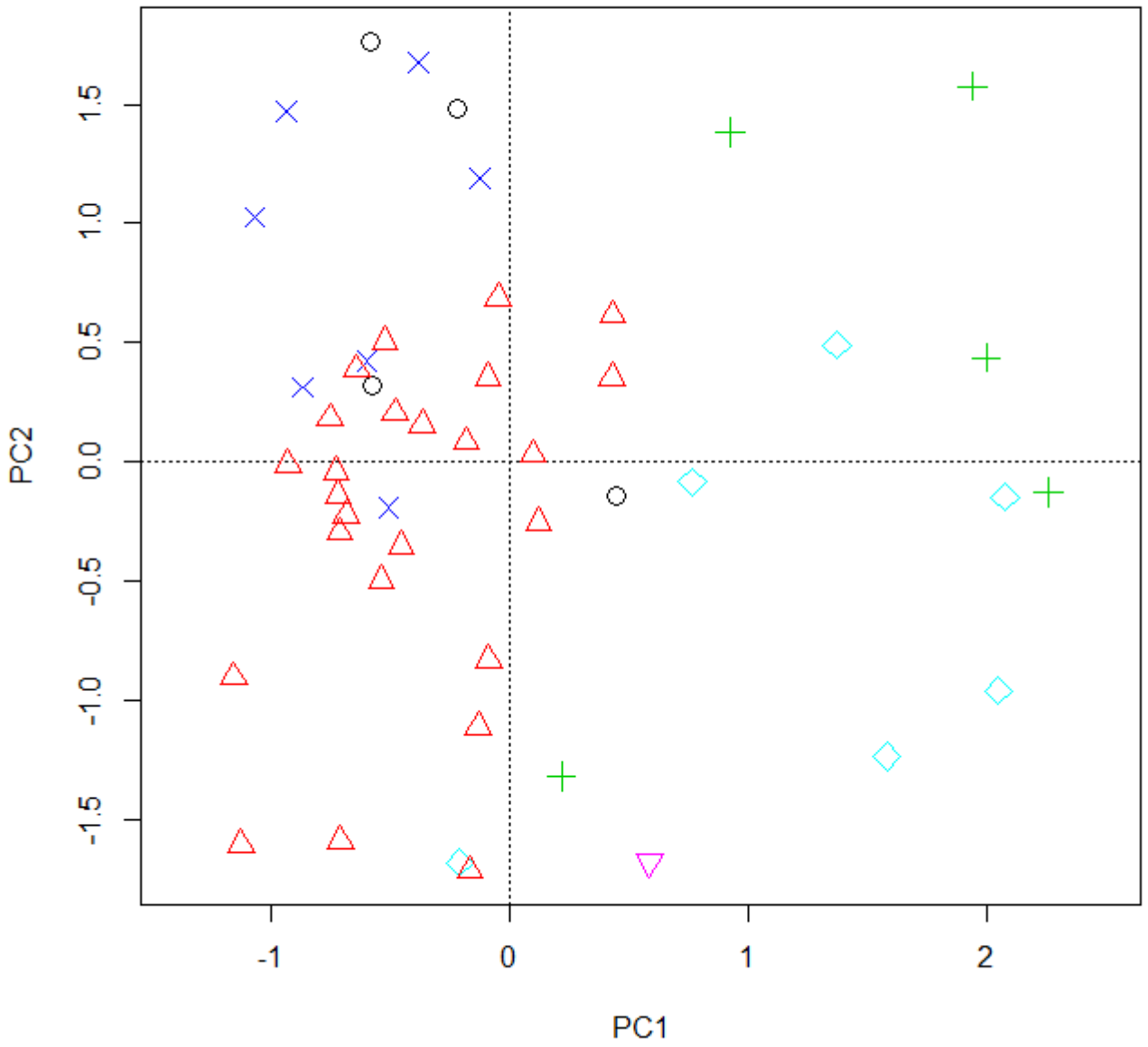
Alternatively, we can display the `summary` of the PCA ordination results (note that the output of the `summary` function is rather talkative, and it may be useful to display only few lines of it by wrapping it into the function `head`):

```
head (summary (PCA))
```



We can see that first two axes respresent (4.625+3.492)/35.4 ≈ 23% of variation. To draw ordination diagram with samples of various vegetation types displayed by different color and symbol, use:

```
vegtype <- as.numeric (as.factor (grasslands.env$classification))
ordiplot (PCA, display = 'sites', type = 'n')
points (PCA, pch = vegtype, col = vegtype)
```



Example 2: PCA on environmental matrix

Carpathian wetlands dataset (Hájek et al.) contains information about species composition of vascular plants and mosses, and also extensive information about the environment, mostly water chemistry. In the following example, we will explore the intercorrelated nature of environmental variables. Note that we select PCA since we assume linear correlations between the variables; there is no reason to apply DCA on the data first to decide between linear and unimodal ordination methods (this decision step is reserved only for species composition data).

```
library (vegan) # needed for function rda
chem0 <- read.delim
('https://raw.githubusercontent.com/zdealveindy/anadat-
r/master/data/chemistry.txt', row.names = 1)
chem <- chem0 [, -15] # removes slope, which is not chemical variable
PCA <- rda (chem, scale = TRUE)  # the argument scale standardizes the variables
```

To see the results, you can simply check the object PCA to list the total variance and eigenvalues of individual axes. More informative is the output of the `summary` function applied to this object. Since the summary returns a complete list of sample and species scores, which can be too long, use the `head` to print only the first six rows of each output element:

```
head (summary (PCA))
```

The output is:

```
Call:
rda(X = chem, scale = TRUE)

Partitioning of correlations:
              Inertia Proportion
Total              14          1
Unconstrained      14          1

Eigenvalues, and their contribution to the correlations

Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8
PC9    PC10    PC11    PC12    PC13    PC14
Eigenvalue            4.3861 1.8311 1.6322 1.21612 0.92221 0.89790 0.72924 0.63060
0.53379 0.40476 0.29093 0.22711 0.18121 0.11672
Proportion Explained  0.3133 0.1308 0.1166 0.08687 0.06587 0.06414 0.05209 0.04504
0.03813 0.02891 0.02078 0.01622 0.01294 0.00834
Cumulative Proportion 0.3133 0.4441 0.5607 0.64754 0.71341 0.77755 0.82963 0.87468
0.91280 0.94172 0.96250 0.97872 0.99166 1.00000

Scaling 2 for species and site scores
* Species are scaled proportional to eigenvalues
* Sites are unscaled: weighted dispersion equal on all dimensions
* General scaling constant of scores:  5.574992


Species scores

         PC1      PC2      PC3      PC4      PC5      PC6
Ca   -1.2395  0.26073 -0.08279  0.07019 -0.21455  0.31631
Mg   -1.2847 -0.07431  0.03779 -0.03741 -0.05047  0.34011
Fe    0.2337 -0.74467 -0.18301  0.91026 -0.36570  0.12707
K    -0.8757 -1.02193  0.02213 -0.23047  0.24166 -0.06626
Na   -0.9686 -0.65514 -0.03874  0.17597  0.40968 -0.29167
Si   -0.8009 -0.94159  0.13567 -0.40204 -0.33330  0.26581
....


Site scores (weighted sums of species scores)

           PC1     PC2      PC3       PC4      PC5     PC6
1    -1.298386 -1.6531  0.32020 -0.410813 -0.27795  1.1076
2     0.331266  0.3436  0.87967 -0.112347  0.05342 -0.5761
3    -0.578705  1.0812 -0.49588 -0.339626  0.30775 -0.4156
4    -0.517608  1.1056  0.04964  0.101172  0.03688  0.1731
5    -0.759527 -0.1386  0.32003 -0.230092 -0.74608  0.2601
6    -0.004093  0.7204  0.44371  0.005363  0.04296 -0.2904
```

```
....
```

The total variation of the whole dataset is 14 in this case, and the first axis explains 31.3% of total variation (see the row `Proportion Explained`, or calculate it as the ratio between eigenvalue of the first PCA axis and total variance, 4.3861/14 = 0.313)[1]. Total variation is a sum of variations of each variable in the analyzed matrix - in this case, all variables have been standardized to zero mean and unit standard deviation (mean = 0, sd = 1), and there are 14 variables, so total variation (inertia) is 14:

```
stand.chem <- scale (chem)
stand.chem.var <- apply (stand.chem, 2, var)
stand.chem.var
```

```
     Ca      Mg      Fe       K      Na      Si     SO4     PO4     NO3     NH3
Cl    Corg      pH conduct
      1       1       1       1       1       1       1       1       1       1
1       1       1       1
```

```
sum (stand.chem.var)
[1] 14
```

Additionally to eigenvalues and species/site scores, we may want to know the factor (or component) loadings of individual variables and axes, i.e. the standardized correlation of each variable to each axis. This can be achieved by function `scores` with `scaling = 0`:

```
loadings <- scores (PCA, display = 'species', scaling = 0)
loadings
<code>
<code>
                 PC1         PC2
Ca       -0.397207108  0.12931763
Mg       -0.411695240 -0.03685437
Fe        0.074879310 -0.36934346
K        -0.280639035 -0.50686184
Na       -0.310399857 -0.32493807
Si       -0.256661125 -0.46701175
SO4      -0.217276338  0.09023013
PO4       0.115086359 -0.25838198
NO3       0.034618703  0.03814265
NH3       0.174227393 -0.02750712
Cl        0.007920445  0.02129442
Corg      0.322058906 -0.21255537
pH       -0.307202160  0.28944051
conduct  -0.368754666  0.24163441
attr(,"const")
[1] 5.574992
```

A quick sorting reveals which variables have the highest absolute correlation to the first and second axis:

```
sort (abs (loadings[,1]), decreasing = TRUE)
```

```
        Mg          Ca     conduct        Corg          Na          pH           K
Si         SO4         NH3         PO4          Fe         NO3          Cl
0.411695240 0.397207108 0.368754666 0.322058906 0.310399857 0.307202160 0.280639035
0.256661125 0.217276338 0.174227393 0.115086359 0.074879310 0.034618703 0.007920445
```

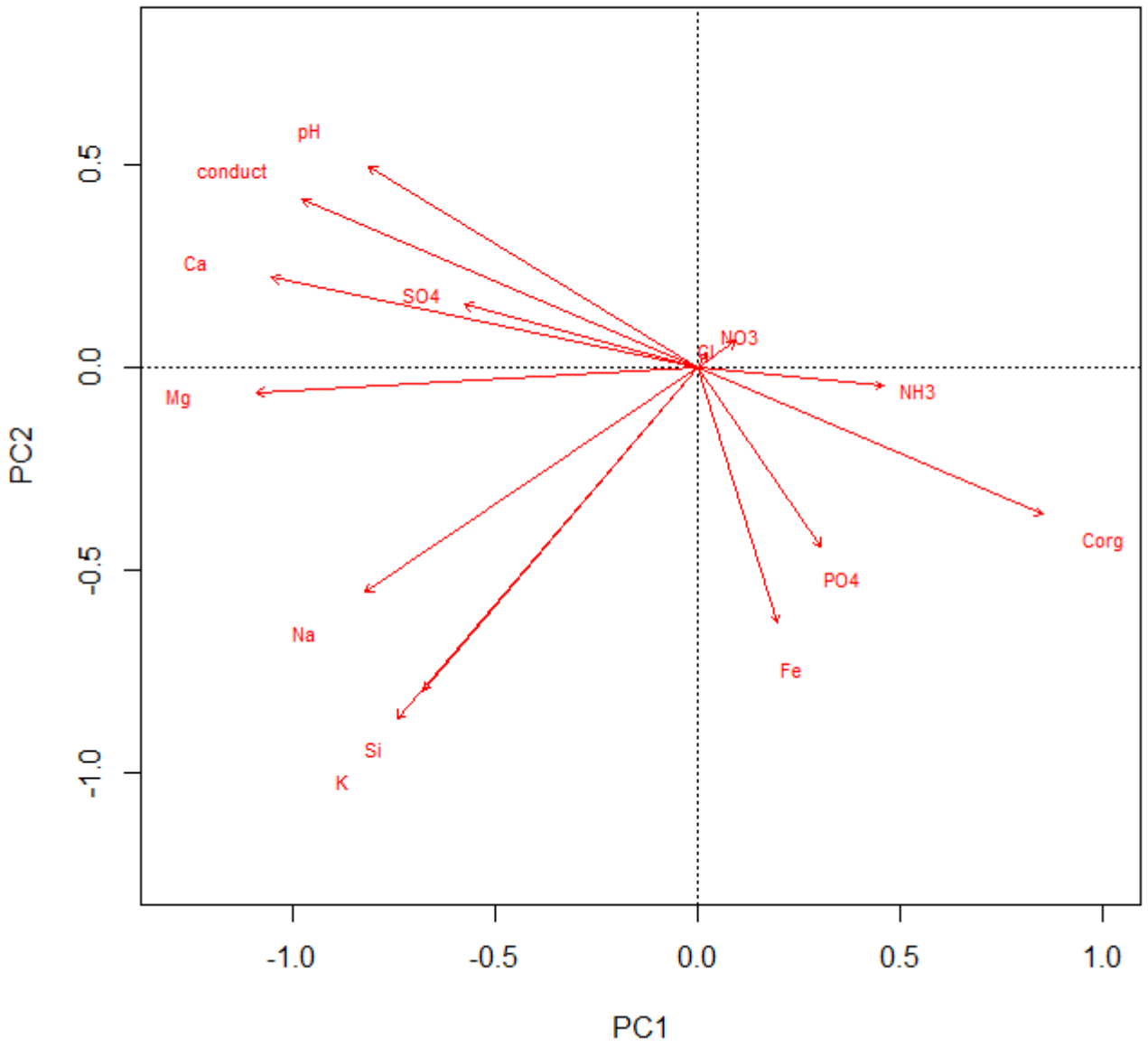
```
sort (abs (loadings[,2]), decreasing = TRUE)
```

```
         K         Si         Fe         Na         pH        PO4    conduct
Corg         Ca        SO4        NO3         Mg        NH3         Cl
0.50686184 0.46701175 0.36934346 0.32493807 0.28944051 0.25838198 0.24163441
0.21255537 0.12931763 0.09023013 0.03814265 0.03685437 0.02750712 0.02129442
```

Magnesium, calcium and conductivity have high loadings to the first axis, while potassium (K), silica (Si) and iron (Fe) to the second).

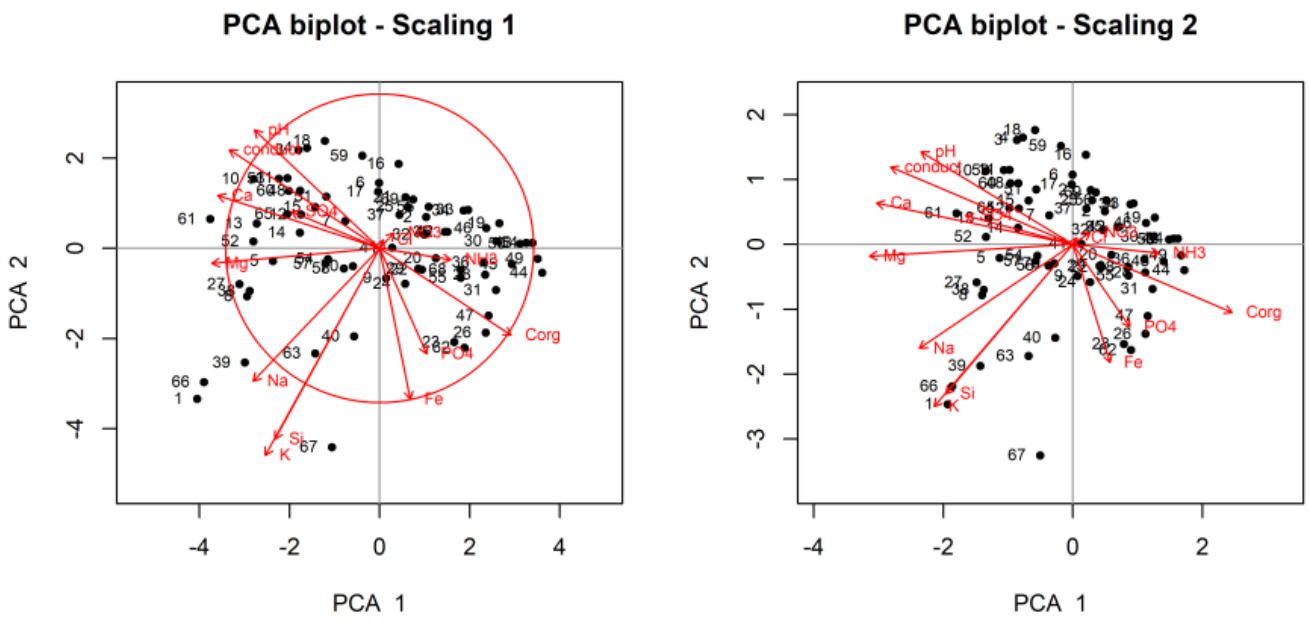
Note that in this specific case, when we are analyzing a dataset of environmental variables, data had to be standardized, either ahead of analysis (e.g. by applying `scale (chem)` or `decostand (chem, method = 'standardize')`), or by setting the argument `scale = TRUE` in the function `rda`. In this way, all variables have the same units and variance; otherwise, the variables with large values will have too high an influence on the analysis. To draw the diagrams, you can use function `biplot`, which is drawing arrows for species (note that function `ordiplot` draws both species/sample scores as centroids):

```
biplot (PCA, display = 'species', scaling = 'species')
```



Alternatively, you may use Legendre's function `cleanplot.pca` (definition here) instead:

```
source
('https://raw.githubusercontent.com/zdealveindy/anadat-r/master/scripts/NumEcolR2/cleanplot.pca.R')
par (mfrow = c(1,2))
cleanplot.pca (PCA, scaling = 1)
cleanplot.pca (PCA, scaling = 2)
```



The left figure is for scaling 1 (focus on distances among plots), the right one for scaling 2 (focus on the correlation among species/variables, which is reflected in the angle of particular vectors). The circle in the left figure is a so-called circle of equilibrium contribution - the variables with vectors longer than the radius of the circle could be interpreted with confidence as important for a given combination of axes [2)]. In this case, the first axis represents so-called poor-rich gradient[3)].

** What if we did not standardise the variables?**

An important step in this example was to standardize the variables to zero mean and unit variance (in the example done by adding argument `scale = TRUE` into the `rda` function, but could be done also by applying the function `scale` on the original matrix before PCA is done on it). This standardisation ensures that each variable brings the same amount of variance into the analysis (as we verified in that example - each variable has variance 1, and 14 variables have thus variance 14 - the total variance of the PCA analysis). This is important in the case that variables are each in very different units and have therefore very different variances. So what will happen if we did not standardise?

First, let's calculate variance of each variable (without standardising them) and use it to sort variables decreasingly according to their variance:

```
chem.var <- apply (chem, 2, var)
sort (chem.var, decreasing = T)
```

```
     conduct         Corg           Ca           pH           Fe           Si
Na           SO4           Mg           Cl          NH3            K          NO3
4.710531e+04 5.525604e+01 8.441988e+00 9.644493e-01 4.197189e-01 3.424972e-01
1.130149e-01 7.279456e-02 6.111429e-02 3.672466e-02 1.353447e-02 3.713112e-03
2.130561e-04
         PO4
1.910736e-06
```

You can see that conductivity has far the highest variance (47105.3), while the second-highest variable, organic carbon (Corg) has only 5.5. The sum of variance is:

```
sum (chem.var)
[1] 47171.04
```

which means that `conduct` represents more than 99% of the total variance in the dataset (47105/47171=99.9%).

PCA calculated on these data will look like this:

```
PCA.nst <- rda (chem, scale = FALSE)
head (summary (PCA.nst))
```

```
Call:
rda(X = chem, scale = FALSE)

Partitioning of variance:
              Inertia Proportion
Total           47171          1
Unconstrained   47171          1

Eigenvalues, and their contribution to the variance

Importance of components:
                            PC1      PC2     PC3     PC4     PC5     PC6     PC7
PC8     PC9    PC10    PC11     PC12      PC13
Eigenvalue            47122.760 42.70274 4.23955 0.47327 0.41291 0.26169 0.07179
0.05107 0.03319 0.01619 0.01107 0.001055 0.0001616
Proportion Explained      0.999  0.00091 0.00009 0.00001 0.00001 0.00001 0.00000
0.00000 0.00000 0.00000 0.00000 0.000000 0.0000000
Cumulative Proportion     0.999  0.99988 0.99997 0.99998 0.99999 1.00000 1.00000
1.00000 1.00000 1.00000 1.00000 1.000000 1.0000000

...
...
```

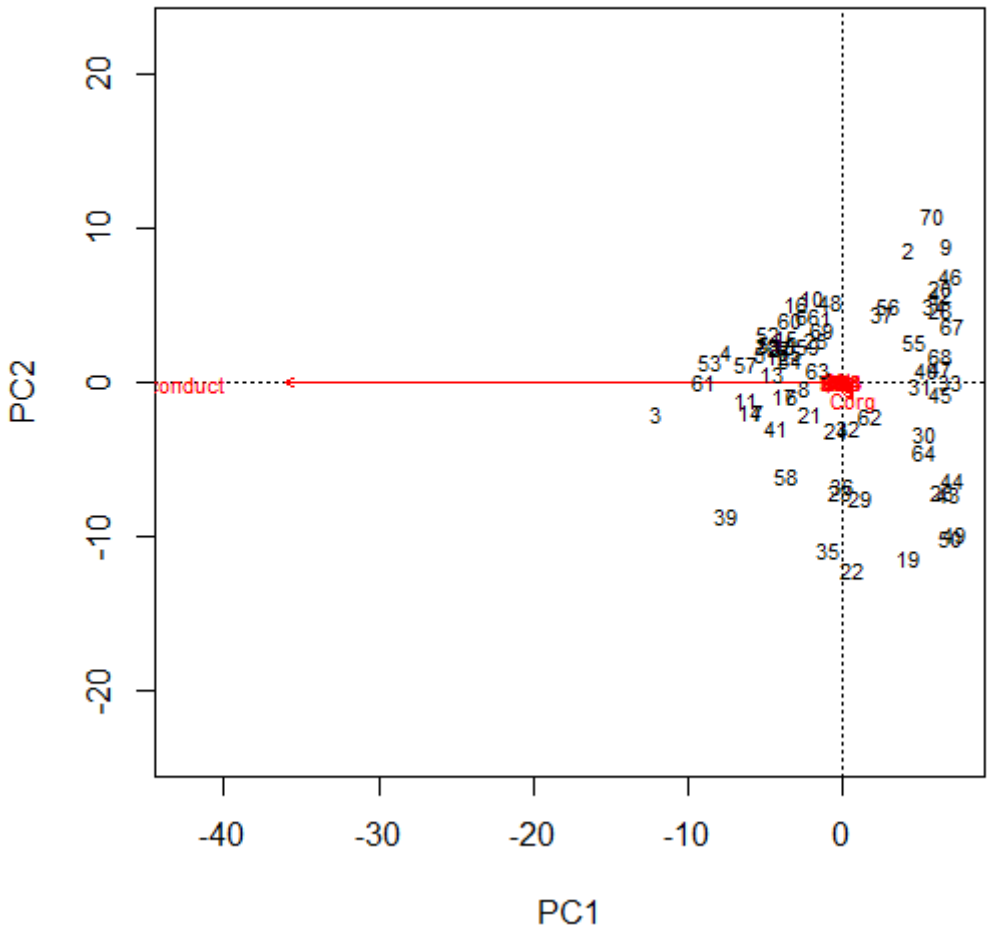
Total inertia (variance) is equal to the sum of variances for individual variables (47171), and the first PCA axis represents 99.9% of the variance. The variable with the highest loading on the first axis is, not surprisingly, conductivity, and the variable with the highest loading on the second axis is the second one with the highest variance, organic carbon (`Corg`):

```
scores (PCA.nst, display = 'species', scaling = 0)
```

```
                  PC1           PC2
Ca      -8.560316e-03  1.445838e-01
Mg      -6.516418e-04  1.171036e-02
Fe       4.716507e-04 -2.592277e-03
K       -5.596122e-05  8.104458e-04
Na      -7.230009e-04  3.278259e-04
Si      -3.201003e-04  1.916717e-02
SO4     -4.681383e-04  2.532191e-03
PO4      5.796201e-07 -6.815188e-05
NO3     -2.658280e-06 -3.835122e-04
NH3      1.147267e-04 -4.051857e-03
Cl      -1.024590e-04 -2.768432e-03
Corg     1.686178e-02 -9.889817e-01
pH      -3.393022e-03  1.204411e-02
conduct -9.998147e-01 -1.797451e-02
```

Loadings (correlation) of `conduct` with first axis is 0.999, that of `Corg` with second axis is -0.989. Next is `Ca` with 0.145 with second axis. Ordination diagram reveals the same pattern:

```
biplot (PCA.nst)
```



This example illustrates that, in the case of environmental variables, the question of whether to standardise or not has a rather simple answer - yes, in most cases. In the case of species composition data (where all variables are in the same units, species abundances or covers) the answer is not that straightforward: by standardising the species you remove differences in their absolute importance (i.e. highly abundant species become the same important as species with low abundances), which makes sense if you are focused on changes in relative abundances of species, but not if you are interested in changes in absolute abundances.

Example 3: Evaluation of importance of ordination axes in PCA

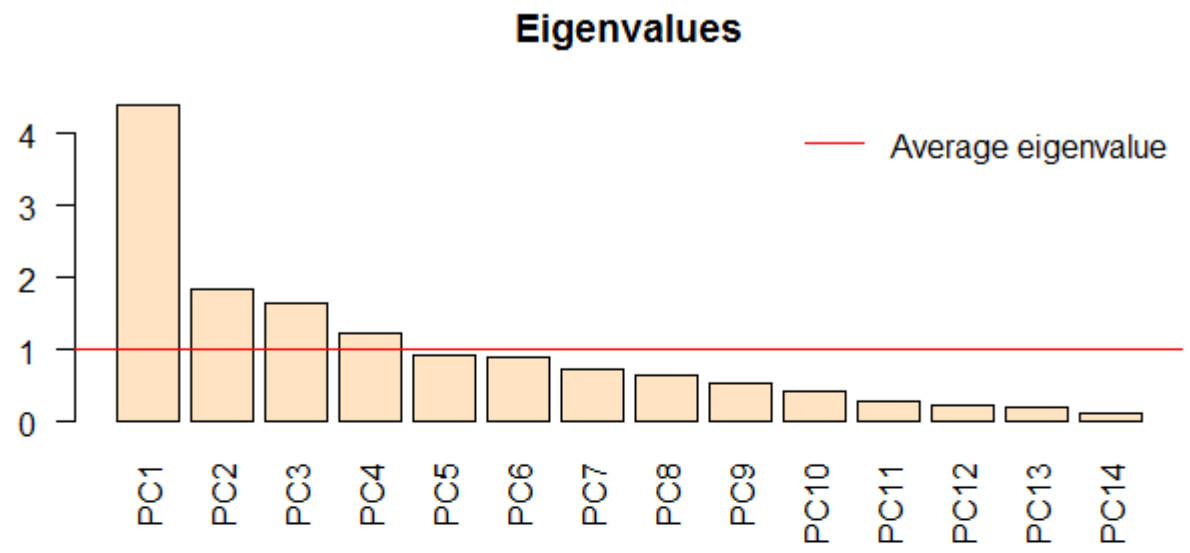
This example uses environmental variables from Carpathians wetlands as above. It illustrates how to decide which PCA axis or axes should be used for the interpretation of results. You need the define the function `evplot` first (written by Francois Gillet, the definition here, by using the `source` method directly from this website).

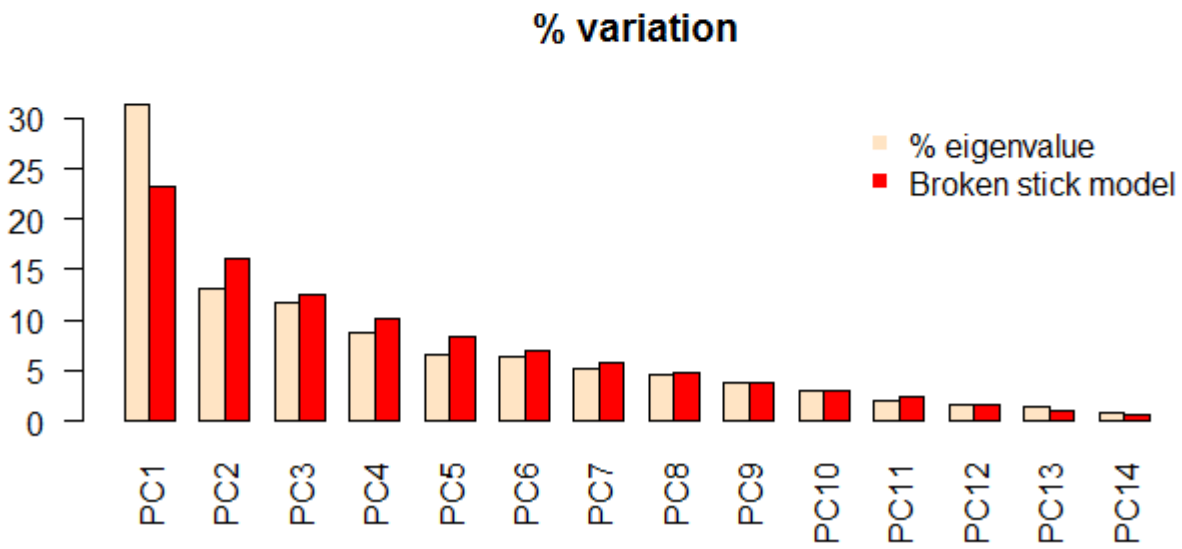
```
# define "evplot" function first:
source
('https://raw.githubusercontent.com/zdealveindy/anadat-r/master/scripts/NumEcolR1/evplot.R')
```

```r
# then load the original data and calculate the PCA as in the Exercise 1:
library (vegan) # needed for function rda
chem0 <- read.delim
('https://raw.githubusercontent.com/zdealveindy/anadat-
r/master/data/chemistry.txt', row.names = 1)
chem <- chem0 [, -15] # removes slope, which is not chemical variable
stand.chem <- scale (chem) #standardize the variables - alternatively, use
decostand (chem, 'stand')
PCA <- rda (stand.chem)

# Finally, in the PCA object select the component $eig with the vector of
eigenvalues:
ev <- PCA$CA$eig

# calculate axis-importance and draw the barplots:
evplot (ev)
```





Note that recommendation based on Keiser-Guttman criterion and broken stick model differs - the earlier points out that the first four axes are important, while the latter indicates that only the first axis is important[4)].

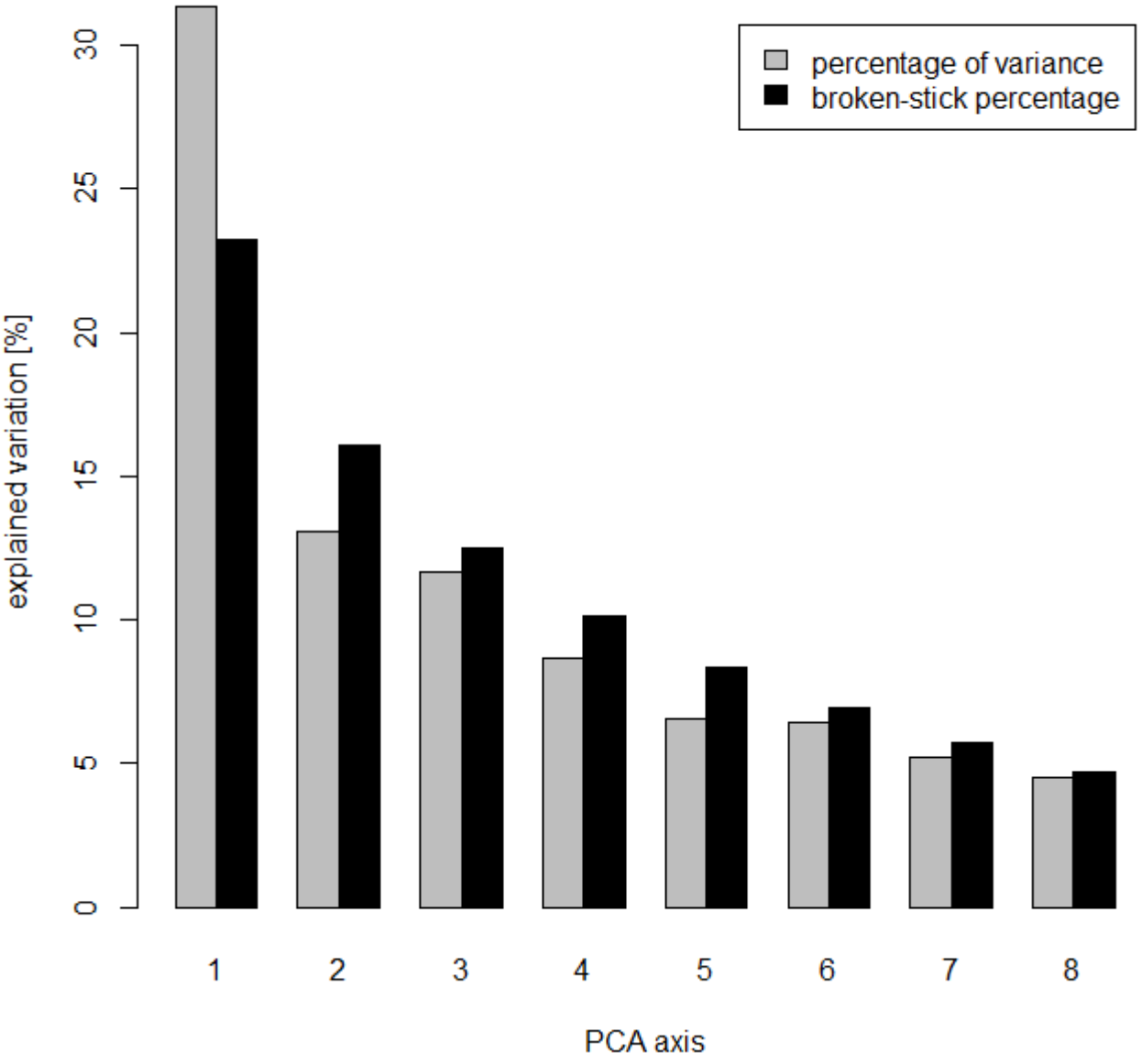
Alternatively, you may use function `PCAsignificance` from library `BiodiversityR`, applied directly on object `PCA` returned by `vegan`'s function `rda`:

```
library (BiodiversityR)
sig <- PCAsignificance (PCA, axes = 14)
sig
```

```
                                       1         2         3         4
5
eigenvalue                          4.386129  1.831062  1.632211  1.216122
0.9222093
percentage of variance             31.329493 13.079014 11.658650  8.686589
6.5872096
cumulative percentage of variance 31.329493 44.408507 56.067158 64.753747
71.3409561
broken-stick percentage            23.225445 16.082588 12.511159 10.130207
8.3444928
broken-stick cumulative %          23.225445 39.308033 51.819193 61.949400
70.2938926
% > bs%                             1.000000  0.000000  0.000000  0.000000
0.0000000
cum% > bs cum%                      1.000000  1.000000  1.000000  1.000000
1.0000000
                                         6          7          8          9
10
eigenvalue                          0.8979007  0.7292376  0.6305977  0.5337917
0.404761
percentage of variance              6.4135766  5.2088400  4.5042690  3.8127980
2.891150
cumulative percentage of variance 77.7545327 82.9633727 87.4676417 91.2804396
94.171590
broken-stick percentage             6.9159214  5.7254452  4.7050370  3.8121799
3.018529
broken-stick cumulative %          77.2098140 82.9352592 87.6402962 91.4524761
94.471005
% > bs%                             0.0000000  0.0000000  0.0000000  1.0000000
0.000000
cum% > bs cum%                      1.0000000  1.0000000  0.0000000  0.0000000
0.000000
                                        11         12         13          14
eigenvalue                          0.2909347  0.227107  0.1812111   0.1167245
percentage of variance              2.0781050  1.622193  1.2943653   0.8337467
cumulative percentage of variance 96.2496948 97.871888 99.1662533 100.0000000
broken-stick percentage             2.3042434  1.654893  1.0596546   0.5102041
broken-stick cumulative %          96.7752486 98.430141 99.4897959 100.0000000
% > bs%                             0.0000000  0.000000  1.0000000   1.0000000
cum% > bs cum%                      0.0000000  0.000000  0.0000000   0.0000000
```

The last two rows show the comparison between real variation represented by individual PCA axes and relevant variation calculated by the broken-stick model (1 in the column with a particular axis means that this axis explains more than would explain the axis of the same order in a null model). With a bit of effort, the barplot visualizing the relationship between variation represented by individual PCA axis and variation explained by broken-stick model (the line `% > bs%` from the table above) can be drawn:

```
barplot (sig[c('percentage of variance', 'broken-stick percentage'), ], beside = T,
  xlab = 'PCA axis', ylab = 'explained variation [%]', col = c('grey', 'black'),
  legend = TRUE)
```



To see why it is important to evaluate whether given PCA axes are actually worth being interpreted, let's try the following situation: **calculate PCA on the dataset in which variables (columns) are not correlated to each other**, i.e., there is no redundant information, and PCA cannot simplify the information stored in many variables into few main ordination axes. We can use the `chem` dataset again, but first, randomize individual chemical variables in the dataset so that the variables are not correlated to each other:

```
# For sure, read data again if you haven't done it above:
library (vegan)
chem0 <- read.delim
('https://raw.githubusercontent.com/zdealveindy/anadat-
r/master/data/chemistry.txt', row.names = 1)
chem <- chem0 [, -15] # removes slope, which is not chemical variable
```

```
# randomize values in each column independently; function "apply" takes each column
(MARGIN = 2), assigns it into
# variable "x", and applies "sample (x)" to randomize it; then the columns are
stacked back into data frame:
set.seed (1234) # this is here just to make sure that you will get the same random
values as me
chem.rand <- apply (chem, MARGIN = 2, FUN = function (x) sample (x)) # or simply
chem.rand <- apply (chem, 2, sample)

# standardize the random variables, and calculate PCA:
stand.chem.rand <- scale (chem.rand) #alternatively, use "decostand (chem, method =
'stand')"
PCA.rand <- rda (stand.chem.rand)
PCA.rand
```

```
Call: rda(X = stand.chem.rand)

              Inertia Rank
Total              14
Unconstrained      14   14
Inertia is variance

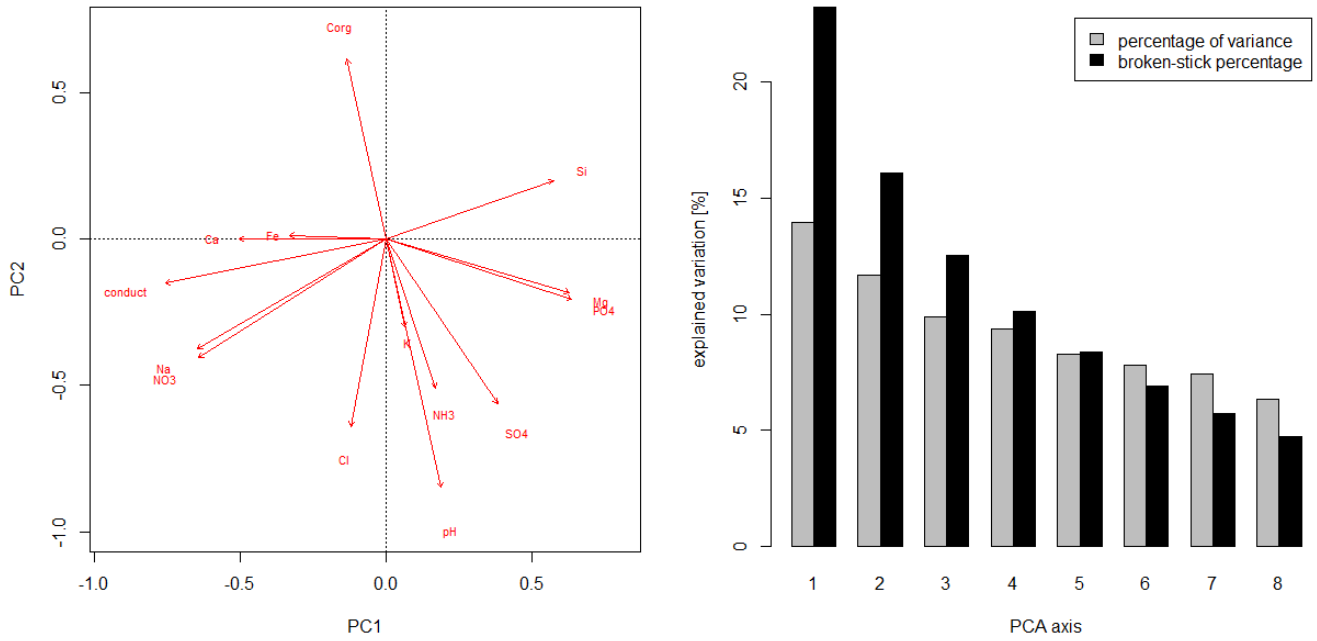
Eigenvalues for unconstrained axes:
   PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8    PC9   PC10   PC11   PC12
PC13   PC14
1.9558 1.6330 1.3839 1.3083 1.1606 1.0917 1.0361 0.8862 0.8277 0.6736 0.6431 0.5876
0.4781 0.3344
```

You can see that, in this case, the eigenvalues still decrease from the first to the last axis (they must; they do so by definition), but the rate of decrease is rather slow. We may draw the ordination diagram to see the result, and also the comparison with the broken-stick model:

```
par (from = c(1,2))
biplot (PCA.rand)

sig.rand <- BiodiversityR::PCAsignificance (PCA.rand)  # calculates broken stick
model comparison with observed
# eigenvalues; I typed directly the name of the package and ''::'', which means to
apply the function from the package
#without the need to upload the whole package using the function ''library''.

barplot (sig.rand[c('percentage of variance', 'broken-stick percentage'), ], beside
= T,
         xlab = 'PCA axis', ylab = 'explained variation [%]', col = c('grey',
'black'),
         legend = TRUE)
```



The ordination diagram looks ok; we may start to interpret it happily. But the broken stick comparison shows that there is actually no axis worth interpreting - their eigenvalues are smaller than those of the null broken stick solution. Note that the broken stick interpretation applies only to the first axes, which have a higher eigenvalue than the broken stick (axes like 6, 7, and 8 seem like they can be interpreted, but they are not interpretable).

Example 4: tb-PCA on species data pre-transformed using Hellinger transformation

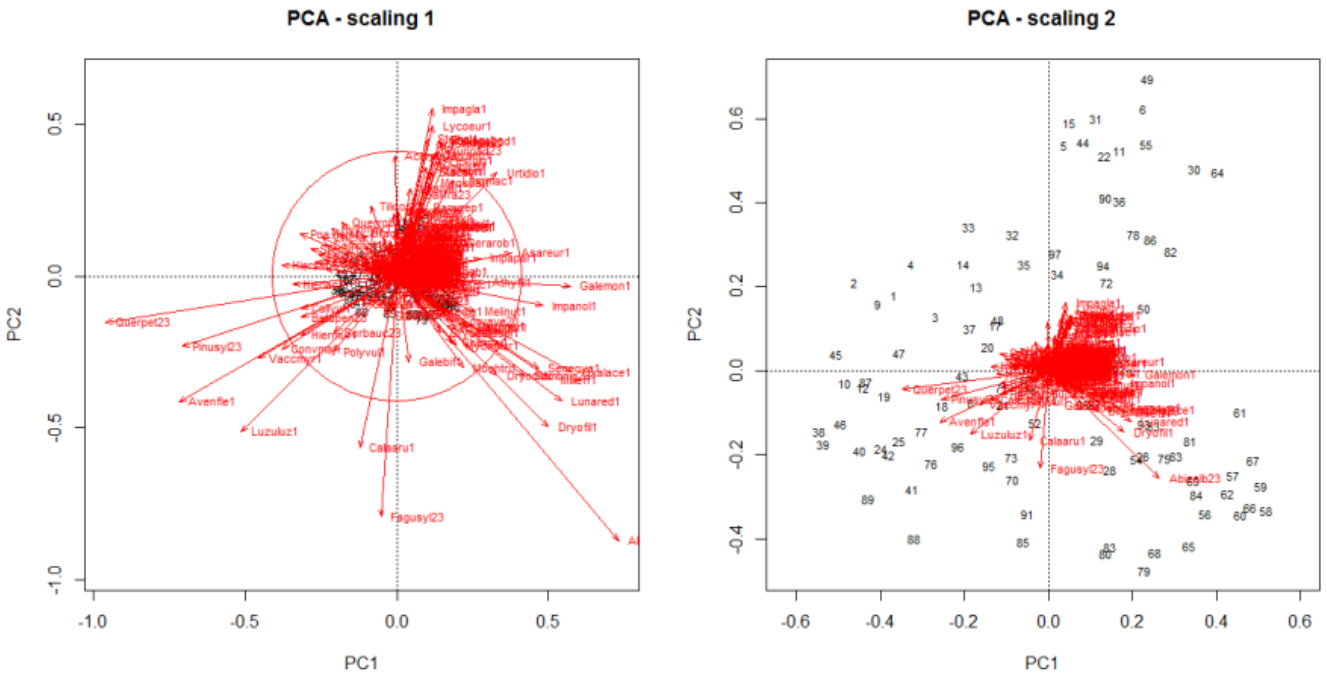
In this example, we will use vegetation data from Vltava river valley dataset and analyse them by the transformation-based version of principal component analysis, meaning after pre-transformation by Hellinger transformation.

```
library (vegan)

vltava.spe <- read.delim
('https://raw.githubusercontent.com/zdealveindy/anadat-r/master/data/vltava-spe.txt
', row.names = 1)
vltava.spe.hel <- decostand (log1p (vltava.spe), 'hellinger') # the species data
(percentage scale) are first log transformed, and then transformed using Hellinger
transformation

tbPCA <- rda (vltava.spe.hel)

source
('https://raw.githubusercontent.com/zdealveindy/anadat-r/master/scripts/NumEcolR1/c
leanplot.pca.R') # define the cleanplot.pca function
cleanplot.pca (tbPCA)
```



These ordination diagrams do not look too helpful (you need to click to enlarge them to see more details) - but we will see later how to visualize the results of ordination more effectively.

1)

Note the difference here between vegan and CANOCO 5: while vegan reports unscaled eigenvalues, CANOCO 5 directly reports eigenvalues scaled in the way that their sum equals to one, not total variation; in case of CANOCO 5 you may directly read the percentage variation explained by individual axes from the eigenvalues reported in Summary by multiplying them by 100.

2)

Daniel Borcard explains (p. 6) what does equilibrium contribution circle means:*... it is possible to draw, on a plane made of two principal components, a circle representing the equilibrium contribution of the variables. Equilibrium contribution is the length that a descriptor-vector would have if it contributed equally to all the dimensions (principal axes) of the PCA. Variables that contribute little to a given reduced space (say, the 1×2 plane) have vectors that are shorter than the radius of the equilibrium contribution circle. Variables that contribute more have vectors whose lengths exceed the radius of that circle. The circle has a radius equal to √(d/p), where d equals the number of dimensions of the reduced space considered (usually d=2) and p equals the total number of descriptors (and hence of principal components) in the analysis.*

3)

Hájek et al. 2002 comment it in their paper in the following way: *Water calcium and magnesium concentrations, pH and conductivity as well as the soil organic carbon content were the properties measured that showed the strongest correlation with the main vegetation gradient (the poor-rich gradient).*

4)

There are some other methods of selecting important axes in PCA, reviewed by Peres-Neto et al. (2005). The broken-stick model performs well in a case that the variables are at least partly correlated.

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

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

Last update: **2026/04/09 06:39**