Training in ade4 in R - Module I: Basic methods

Principal component analysis

Stéphane Dray

2023-12-03

Data

We will analyze the `doubs` data set (see `?doubs`)

```r
library(ade4)
library(adegraphics)
data(doubs)
names(doubs)
```

```
## [1] "env"     "fish"    "xy"      "species"
```

```r
names(doubs$env)
```

```
##  [1] "dfs" "alt" "slo" "flo" "pH"  "har" "pho" "nit" "amm" "oxy" "bdo"
```

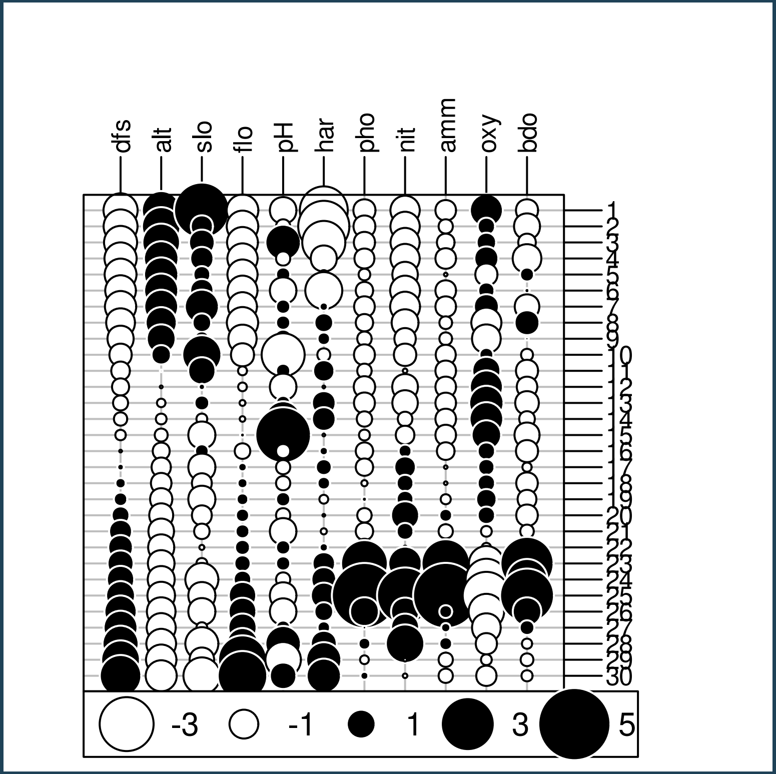
Some general information

```
env <- doubs$env
apply(env, 2, range)
```

```
##       dfs alt   slo  flo pH har pho nit amm oxy bdo
## [1,]    3 172 1.099   84 77  40   1  15   0  41  13
## [2,] 4530 934 6.176 6900 86 110 422 620 180 124 167
```

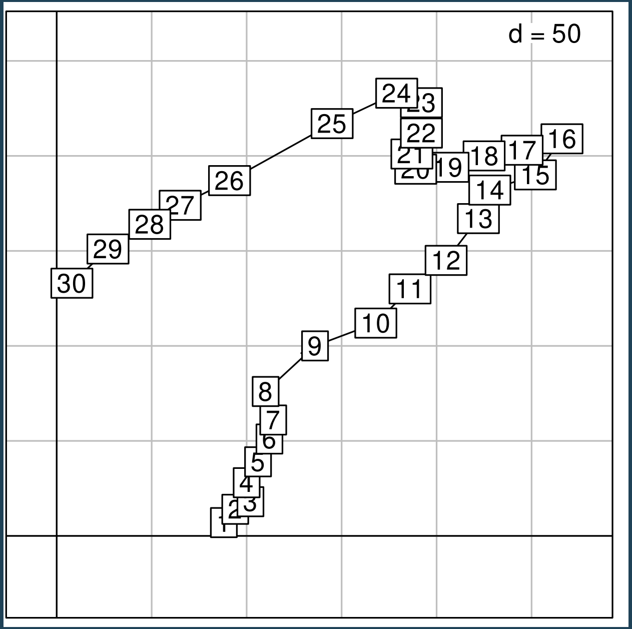
Display the data

```
table.value(scale(env), symbol = "circle")
```



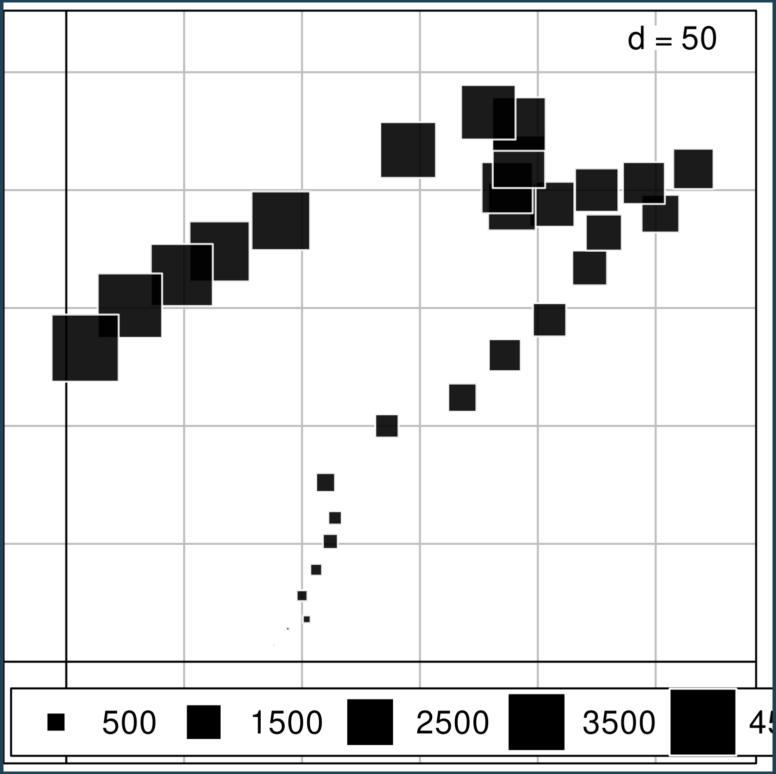
Spatial information

```
xy <- doubs$xy
sl1 <- s.label(xy, ppoints.cex = 0, plot = FALSE)
st1 <- s.traject(xy, ppoints.cex = 0, plabels.cex = 0,
    plot = FALSE)
s1 <- superpose(st1, sl1, plot = TRUE)
```



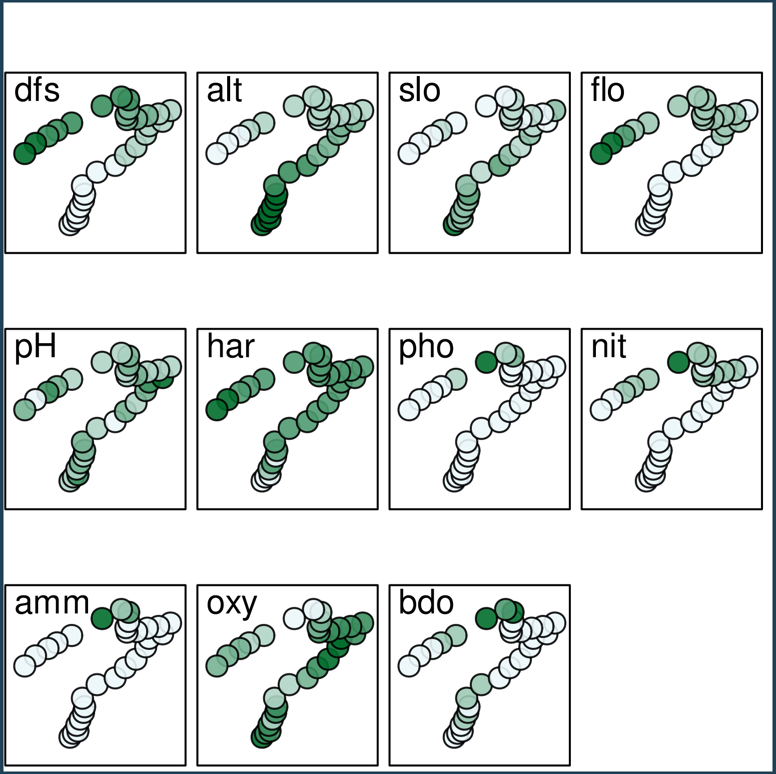
Map one variable

```r
s.value(xy, doubs$env[, 1])
```



Map the different variables

```
mypal <- colorRampPalette(c("#EDF8FB", "#006D2C"))
s.value(xy, doubs$env, pgrid.draw = FALSE, porigin.draw = FALSE,
    plegend.drawKey = FALSE, psub.cex = 2, method = "color",
    symbol = "circle", ppalette.quanti = mypal, ppoints.cex = 0.5)
```



Perform the PCA

Must we scale or not the data?

```
pca1 <- dudi.pca(env, scale = TRUE, scannf = FALSE,
    nf = 2)
```

How many axes?

Choosing the number of dimensions that can be interpreted is an open question. Several methods have been proposed in the literature. Using your eyes to detect breaks in eigenvalue decrease is probably the best method.

However, `ade4` proposes one method developed for the case of PCA on scaled data. You can try it:

```
testdim(pca1)$nb.cor
```

```
## [1] 2
```

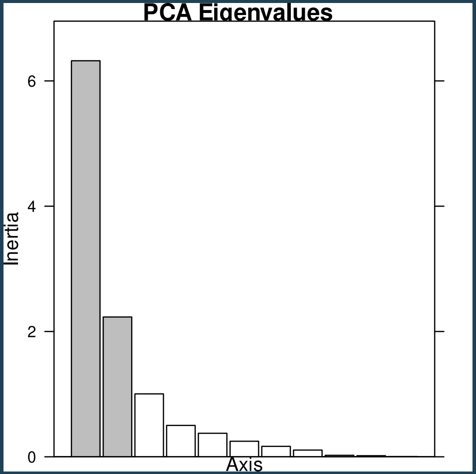
Display eigenvalues

- Make a barplot of eigenvalues
- Compute the percentage of variation explained by each dimension
- Compare to the results obtained by the function `summary`

```
round(pca1$eig/sum(pca1$eig) * 100, 2)
```

```
##  [1] 57.47 20.29  9.13  4.55  3.41  2.25  1.51  0.97  0.21  0.16  0.04
```

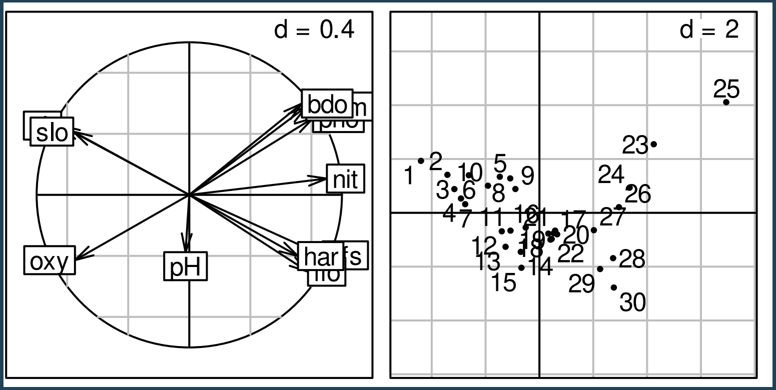
```
screeplot(pca1, main = "PCA Eigenvalues")
```



Graphical representation of PCA results

- Plot the variable scores with a correlation circle
- Plot the site scores on a separate plot
- Interpret

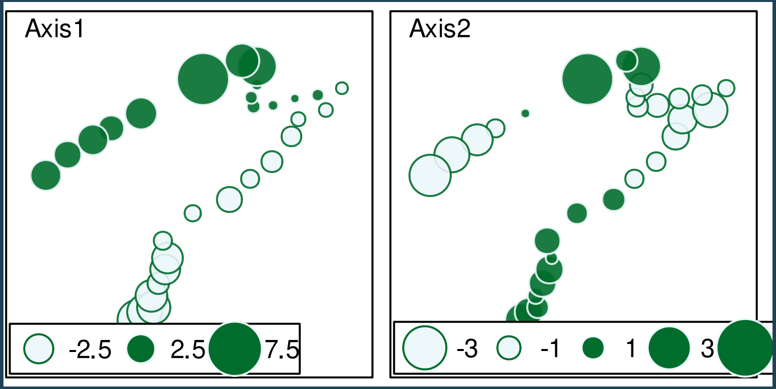
```
sc1 <- s.corcircle(pca1$co, plot = FALSE)
sl1 <- s.label(pca1$li, plabels.optim = TRUE, ppoints.cex = 0.5,
    plot = FALSE)
ADEgS(list(sc1, sl1))
```



PCA scores on the geographical map

- Draw maps of PCA scores on the first two axes
- Interpret the maps to describe the environmental structure of the river

```
s.value(xy, pca1$li[, 1:2], pgrid.draw = FALSE, porigin.draw = FALSE,
    method = "size", symbol = "circle", col = mypal(2),
    ppoints.cex = 1)
```



A look to variables

- Compute the correlations between original variables and PCA axes
- In which element of PCA outputs are stored these values

```
cor(pca1$l1, doubs$env)
```

```
##              dfs        alt        slo        flo          pH        har
## RS1  0.8733503 -0.8390535 -0.7602047  0.7777617 -0.02457895  0.7122973 0.8
## RS2 -0.3963445  0.4537709  0.4145874 -0.5037445 -0.37560694 -0.4011204 0.5
```

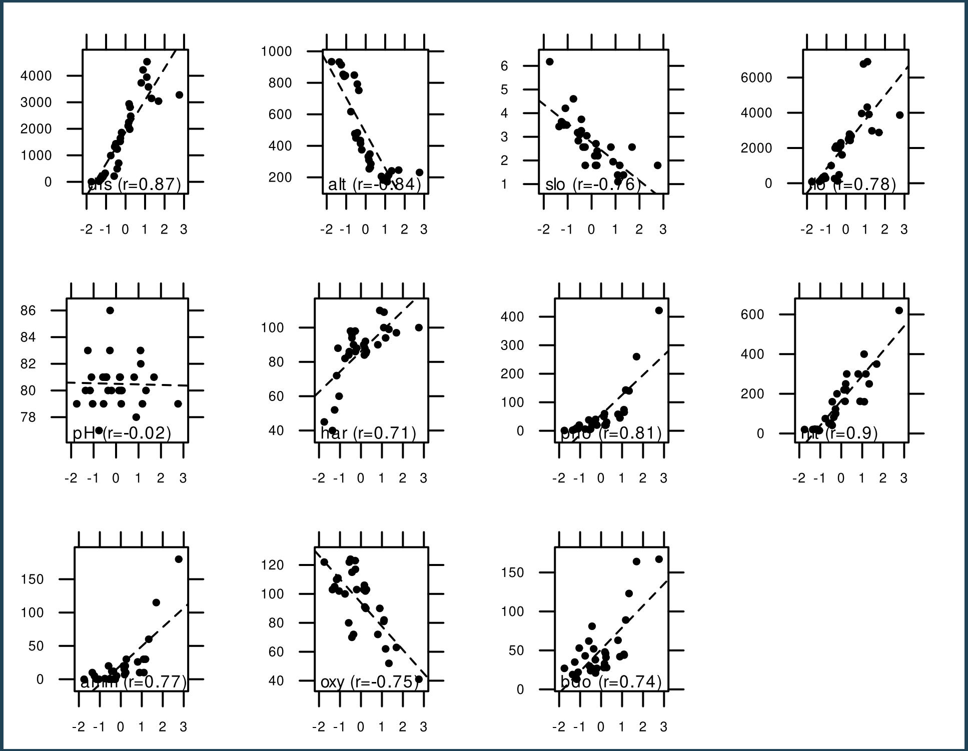
```
head(pca1$co)
```

```
##            Comp1      Comp2
## dfs  0.87335029 -0.3963445
## alt -0.83905354  0.4537709
## slo -0.76020468  0.4145874
## flo  0.77776175 -0.5037445
## pH  -0.02457895 -0.3756069
## har  0.71229727 -0.4011204
```

A look to variables

The generic function `score` provides an optimal representation of the maximized criteria

```
score(pca1)
```


dis (r=0.87)
alt (r=-0.84)
slo (r=-0.76)
flo (r=0.78)
pH (r=-0.02)
har (r=0.71)
pho (r=0.81)
nit (r=0.9)
amm (r=0.77)
oxy (r=-0.75)
bdo (r=0.74)