Fiche TD avec le logiciel R : course7

An introduction to K-table analyses

A.B. Dufour

Contents

1 Introduction

Analysing one table using exploratory methods can be done using principal component analysis on continuous variables or correspondence analysis on contingency tables.

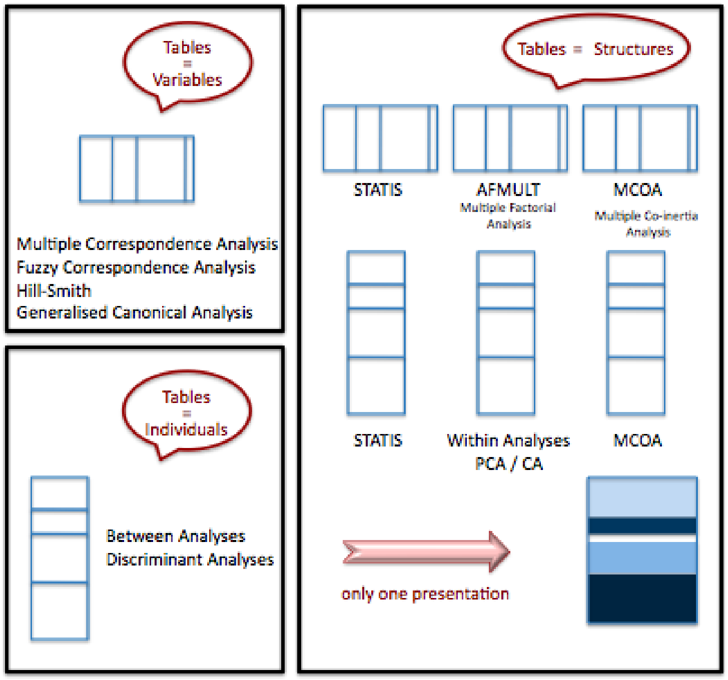
Analysing two tables using exploratory methods can become more difficult, due to the need to specify which relationship one seeks: if we are interested in exploring how one table illustrates the second table (i.e. the stations and the seasons allow to illustrate the physico-chemical information along the Meaudret), we'll go for PCAs and ellipses. If we are interested in exploring how the information contained in the two tables cross (i.e. the abundance frequencies of Species and the physico-chemical variables on the Doubs river), we'll consider the Coinertia Analysis. The ecological question thus leads the choice of methods.

Analysing K tables using exploratory methods increases the difficulty of the problem, and one definitively needs to define which ecological question drives the approach. A K-table analysis is the combination of several separated analyses linked by columns, by rows, or by columns AND rows. Each table can be centered, normed, or organized according to the defined objective.

1. A table may be seen as a 'variable'. This describes a situation where several columns are needed to correctly "recreate" the considered information. One for example needs to know the repartition of clay, silt (alluvium) and sand to correctly define granulometry (three columns, one variable).

2. A table may be seen as an 'individual'. An individual is described by a set of variables. A point in the data analysis shows only one individual (e.g., a table contains the genetic info (0/1) for an individual)

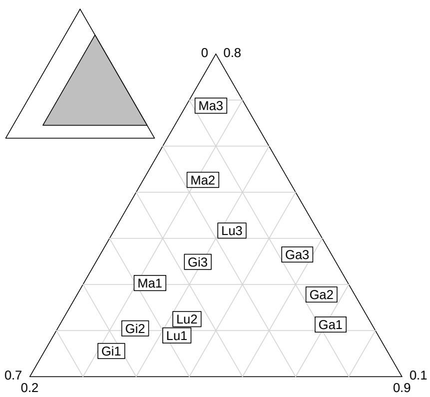
3. A table may be seen as a 'structure'. A faunistic table shows for example the spatial distribution of species between different stations. And one can provide a table per season (e.g., one in spring, one in summer,...)



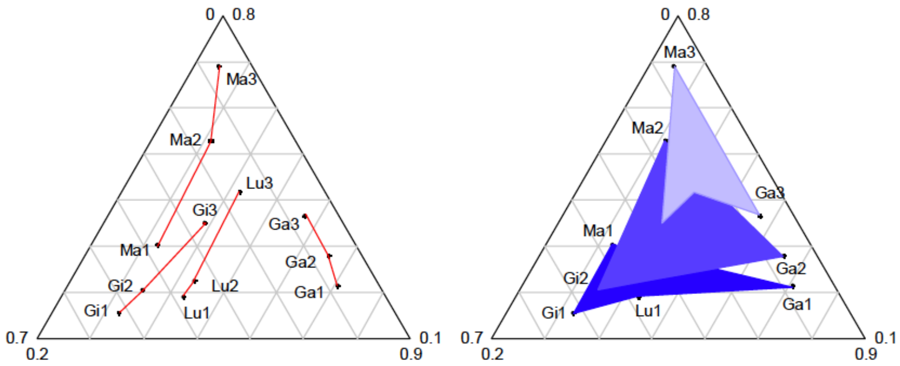
2 Illustration to understand the K-table complexity

The small following dataset gives the proportion of people employed in the primary, secondary and tertiary sectors for 4 town councils in the South of France: Gignac (Gi), Ganges (Ga), Matellas (Ma) and Lunel (Lu) during 3 census (1968, 1975, 1982).

```
c4v3 <- read.table("c4v3.txt", h = F, sep = ",")
ref <- c("Gi1", "Gi2", "Gi3", "Ga1", "Ga2", "Ga3", "Ma1", "Ma2",
    "Ma3", "Lu1", "Lu2", "Lu3")
triangle.plot(c4v3, label = ref, clab = 1, labeltriangle = F)
```



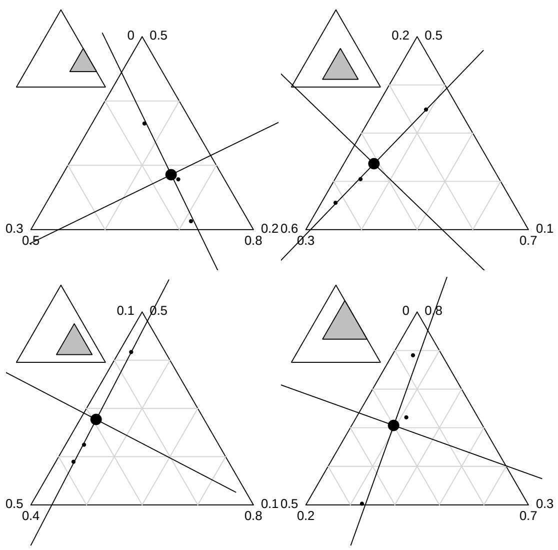
The triangular plots below can be interpreted as displaying the evolution of each town council (on the left) or the evolution of economic typologies (on the right).




```
towncoun <- factor(rep(c("Gi", "Ga", "Ma", "Lu"), c(3, 3, 3, 3)))
dat <- factor(rep(c("D1", "D2", "D3"), 4))
plan <- cbind.data.frame(towncoun, dat)
```

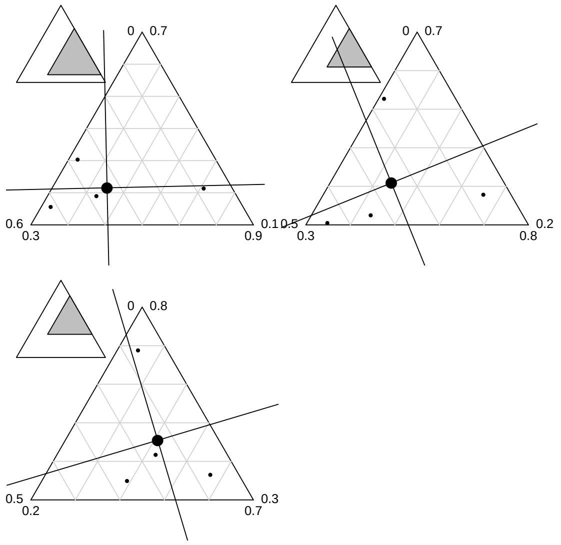
The plots below then show the four separated analyses linked to the first question (i.e, evolution of each town council). For each plot (i.e. each town council), a point represents a date. The big point is the average point (means of the three sectors - a vector of 3 elements - for the corresponding town council). The axes are the two axes of the centred principal component analysis.

```
par(mfrow = c(2, 2))
for (i in 1:4) triangle.plot(c4v3[towncoun == levels(towncoun)[i],
    ], sub = as.character(levels(towncoun)[i]), addax = T, labeltriangle = F)
```



The plots below, on the other hand, show the three separated analyses linked to the second question (i.e., evolution of economic typologies). For each plot (i.e. each date), a point represents a town council. The big point is the average point (means of the three sectors - a vector of 3 elements - for the corresponding date). The axes are the two axes of the centred principal component analysis.

```
par(mfrow = c(2, 2))
for (i in 1:3) triangle.plot(c4v3[dat == levels(dat)[i], ], sub = as.character(levels(dat)[i]),
    addax = T, labeltriangle = F)
```



The two questions (1- what's the evolution of the councils, 2-what's the evolution of the economic typologies) lead to different methods and results.

3 Computing a multiway analysis with ade4

A K-table analysis is the study of K duality diagrams which have in common their rows, their columns, or both. To compute a K-table analysis using ade4, two steps are needed:

⋆ 1- How do we enter the data?

no	function	strategy
1	ktab.list.df	list of data frames
2	ktab.list.dudi	list of duality diagrams
3	ktab.data.frame	one data frame
4	ktab.within	list of data frames

no 1. Let's euro123 be the dataset containing the proportion of people employed in the primary, secondary and tertiary sectors for 12 European countries in 1978, 1986 and 1997. euro123 is a list of three dataframes:

```
data(euro123)
class(euro123)
```

```
[1] "list"
```

```
names(euro123)
```

```
[1] "in78" "in86" "in97" "plan"
```

```
ktab1 <- ktab.list.df(euro123)
class(ktab1)
```

```
[1] "ktab"
```

no 2. Using the previous dataset, one can compute a centred PCA on each table (1978, 1986, 1997) and gather the results in one list of objects `dudi`.

```
pca78 <- dudi.pca(euro123$in78, scale = F, scann = F)
pca86 <- dudi.pca(euro123$in86, scale = F, scann = F)
pca97 <- dudi.pca(euro123$in97, scale = F, scann = F)
listpca <- list(pca78, pca86, pca97)
ktab2 <- ktab.list.dudi(listpca)
class(ktab2)
```

```
[1] "ktab"
```

no 3. Let's `escopage` be a dataset describing 27 characteristics of 21 wines. These characteristics are attributed according to 4 evaluation components: characteristics linked to its first olfactory attributes ("premier nez", when you just opened the bottle; variables 1 to 5), to its visual appearance (variables 6 to 8), to its olfactory attributes ("second nez", olfactory attributes after a while; variables 9 to 18) and to its global appearance (variables 19 to 27). `escopage$blo` contains the number of columns for each of the 4 evaluation components.

```
data(escopage)
names(escopage)
```

```
[1] "tab"       "tab.names" "blo"
```

```
escopage$blo
```

```
[1]  5  3 10  9
```

```
nescopage <- data.frame(scalewt(escopage$tab))
ktab3 <- ktab.data.frame(nescopage, escopage$blo, tabnames = escopage$tab.names)
class(ktab3)
```

```
[1] "ktab"
```

no 4. In course4, we studied the within and between principal component analyses. The dataset `meaudret` contains 9 physico-chemical variables measured on 5 sites during the 4 seasons. We computed a within PCA to remove the seasonal effect.

```
data(meaudret)
names(meaudret)
```

```
[1] "mil"  "plan" "fau"
```

```
pca1 <- dudi.pca(meaudret$mil, scann = F, nf = 3)
wit1 <- within(pca1, meaudret$plan$dat, scan = FALSE)
kta4 <- ktab.within(wit1)
class(kta4)
```

```
[1] "ktab"
```

⋆ Which method should we use?

name	function
`pta`	Partial Triadic Analysis (or pre-STATIS)
`statis`	STATIS
`mcoa`	Multiple co-inertia analysis
`mfa`	Multiple factorial analysis

To plot the results of a K-table analysis, use the `plot` or `kplot` functions.

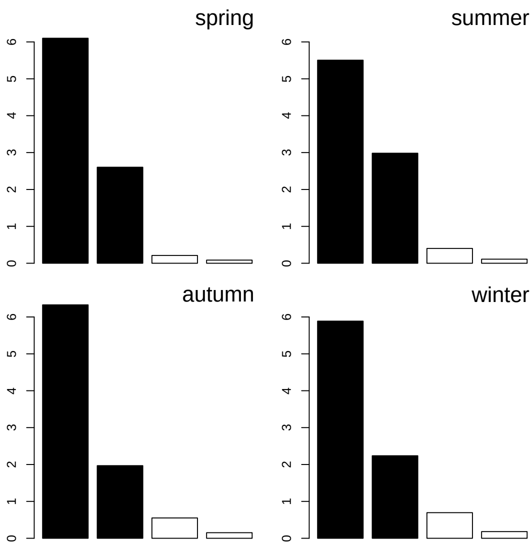
4 An ecological illustration

To illustrate the principle of the multiway analysis, we can have a look a dataset we already know and analysed: the Meaudret dataset. To remove the seasonal effect, we previously computed a within principal component analysis. We'll differentiate two ways to compute the K-table analysis on this new dataset (output from the within PCA):the first way with `ktab` and the second way with `partial` (PS: `partial` means that each sub-table is centred and normed)

```
pcadat <- withinpca(meaudret$mil, meaudret$plan$dat, scal = "partial",
    scann = F)
ktadat <- ktab.within(pcadat)
```

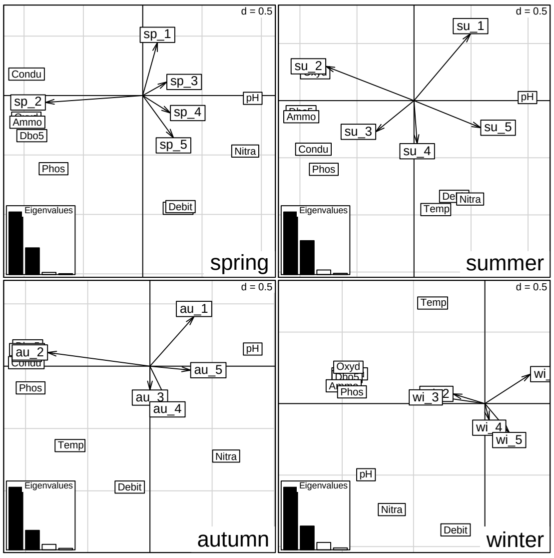
One can compute a PCA per season using one single command, and then represent all the screeplots.

```
plot(sepan(ktadat))
```



One can represent the biplots (i.e. putting both the variables and the individuals) of each PCA.

```
kplot(sepan(ktadat))
```



The partial triadic analysis looks for the common part of all separated analyses. This common part is called compromise. These four tables can be compared because they have the same rows (the sites) and the same columns (the physico-chemical variables).

```
col.names(ktadat) <- rep(paste("sta", 1:5, sep = ""), 4)
ptadat <- pta(ktadat, scannf = F)
names(ptadat)
```

```
 [1] "RV"         "RV.eig"     "RV.coo"     "tabw"       "rank"       "nf"
 [7] "tab"        "lw"         "cw"         "eig"        "li"         "co"
[13] "l1"         "c1"         "Tcomp"      "Tax"        "Tli"        "Tco"
[19] "cos2"       "TL"         "TC"         "T4"         "blo"        "tab.names"
[25] "call"
```

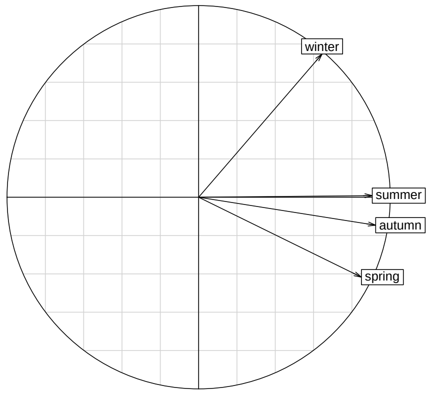
To analyse the relationships between these tables, one can look at the RV-coefficients.

```
ptadat$RV
```

```
          spring    summer    autumn    winter
spring 1.0000000 0.6934558 0.7886185 0.2834592
summer 0.6934558 1.0000000 0.7671756 0.5340456
autumn 0.7886185 0.7671756 1.0000000 0.4794976
winter 0.2834592 0.5340456 0.4794976 1.0000000
```

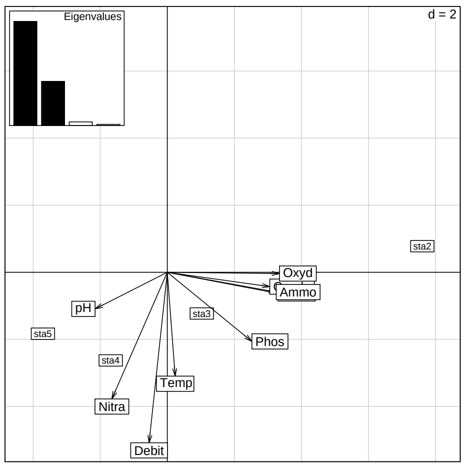
To plot the relationships between these 4 tables, one can compute a correlation circle using the RV-coefficients.

```
s.corcircle(ptadat$RV.coo)
```



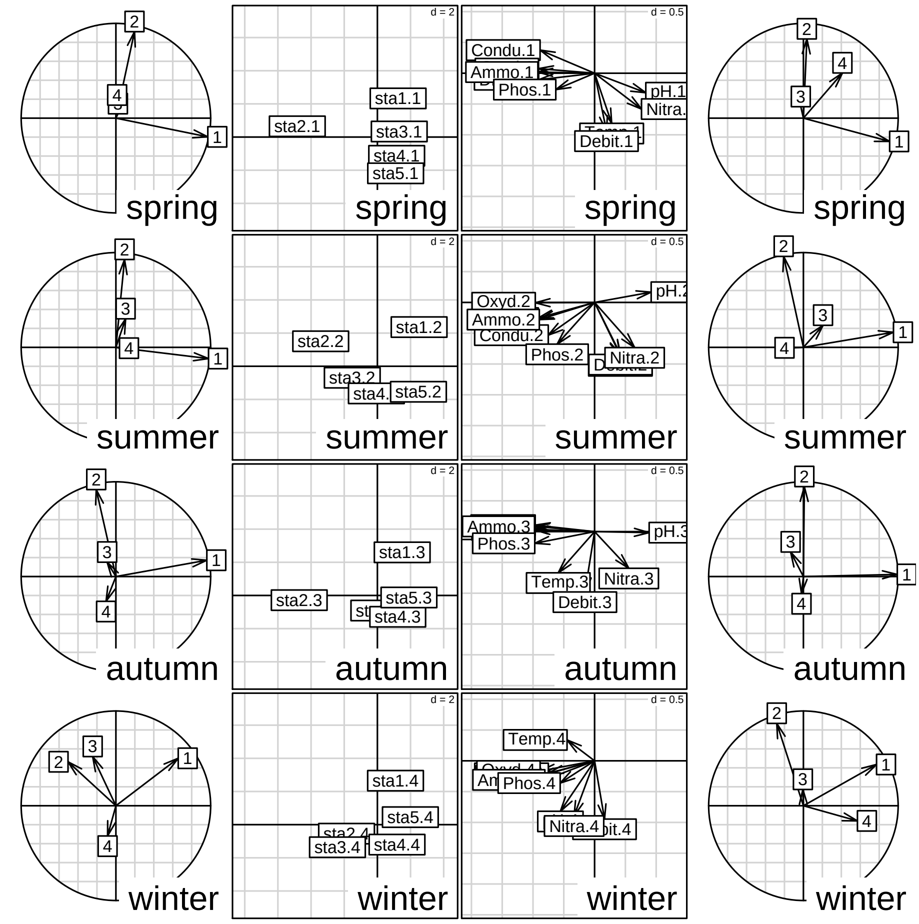
The compromise (which is a summary of the 4 tables) can be plotted using a simultaneous representation of the physico-chemical variables and the sites.

```
scatter.dudi(ptadat, permute = T)
```



Finally, the last representation shows each season in the same geometrical space (i.e. the compromise).

```
kplot(pta(t(ktadat), scan = F), clab = 1.5, csub = 3)
```


spring
summer
autumn
winter
sta1.1
sta2.1
sta3.1
sta4.1
sta5.1
Condu.1
Ammo.1
Phos.1
pH.1
Nitra.
Debit.1
sta1.2
sta2.2
sta3.2
sta5.2
Oxyd.2
Ammo.2
Condu.2
Phos.2
Nitra.2
pH.2
sta1.3
sta2.3
sta5.3
sta4.3
Ammo.3
Phos.3
pH.3
Temp.3
Nitra.3
Debit.3
sta1.4
sta5.4
sta3.4
sta4.4
Temp.4
Phos.4
Nitra.4
d = 2
d = 0.5