Consultation statistique avec le logiciel 

Comment remplacer les labels par des icônes sur une carte factorielle ?

D. Chessel & T. Jombart

15 mars 2006

Un message de Kent Löfgren et une réponse de Basille Mathieu invite à placer des icônes à la place des étiquettes sur une carte factorielle. La fonction `addlogo` de Roger Bivand permet de le faire. Il faut régler quelques détails pour que ce soit opérationnel. La fiche détaille ces opérations et donne un exemple.

Table des matières

1 Introduction

Kent Löfgren (Umeå University) a envoyé le 30/11/2005 sur `adelist` un long message dont voici quelques extraits :

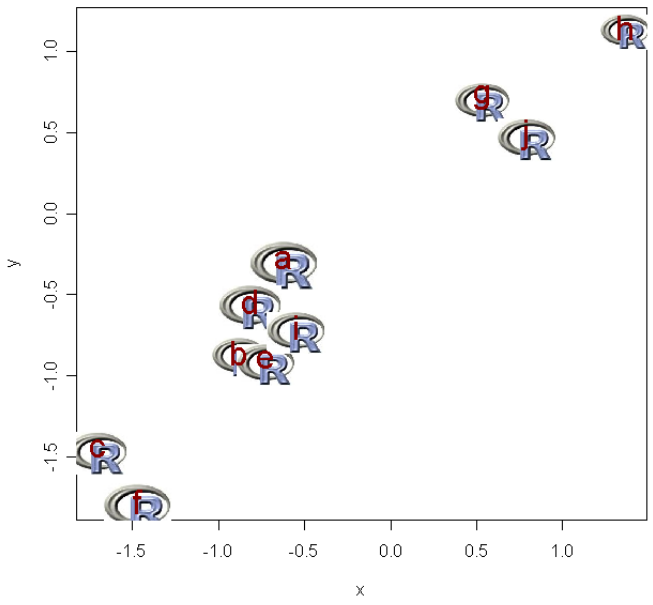
> Things I learned during my first weeks of working with R and the ADE4 package to perform multiple correspondence analysis.
>
> The text is not sorted.
>
> a) There are many packages that perform ca and mca, not just one. Some have the same features and some have unique features. R comes with a number of abilities and built-in packages. It is also possible to download and add more packages. ade4 is one of those add on packages that you need to download and install yourself. For me, ade4 is the best package for my analysis needs.
>
> b) ade4 is developed and maintained in the spirit of the French approach to correspondence analysis in the tradition started by Benzécri (1973) and described in English by, for example, Greenacre (1984).
>
> c) You may of course export coordinates and import them into Microsoft Excel. These xy coordinates are easy to display in an Excel graph. However, it is not possible to plot dots with labels for each dot, e.g. point xy1 with the name "AA", point xy2 with the name "BB" and so on.
>
> To do this, you need to download and install ChartLabeler, a free add-on for Excel.
>
> e) The figures that you export from R can be opened and edited with, for example, Adobe's Illustrator programme. If you want to edit individual dots or labels, do this : a) make sure you export the figure in postscript format (.ps)., b) Open the file in Illustrator, c) Right-click on any dot, and select "Ungroup"., d) Right-click again on any dot, and select "Release Compound path". It is now possible to edit every item in the figure (labels, numbers, lines et cetera).
>
> g) This document describes the ADE4 data-files (the original data that is used in the examples) : "D. Chessel - Biométrie et Biologie Evolutive - Université Lyon1. Data de la librairie ade4." For example the data file "banque", which is a table with qualitative variabler. This data is used in the multiple correspondence analysis examples. 810 rows (individuals) and 21 columns (age group, social group, gender etc).
>
> h) Here is an example with the banque-data (multiple corr. analysis) :
>
> ...
>
> The facts are nothing new to the ade4-list members, I know. I just wanted to make some contribution to the list, and not just ask for help, help and help all the time.
>
> Hopefully it gives some insight to how a complete beginner tries to learn R and ade4.

Ce message très sympathique soulève par deux fois la question de l'import-export des figures de R dans un logiciel commercial pour améliorer le résultat. Les utilisateurs avertis ne se posent plus cette question tant la qualité graphique

de R est établie et celle d'Excel problématique. Échanger des données avec Excel pour faire un graphique ne pouvait que chatouiller certains membres de la liste et c'est Basille Mathieu qui a été le plus rapide. Au point c) il répond :

> I'm wondering WHY you want to export things from R to MS Excel, especially for graphs ! (Eventually, from R to OOo Calc, but I still can't see the expected advantages of the export). R has very good graphic abilities.
>
> A nice example for your labels problem :

```
require(pixmap)
logo <- read.pnm(system.file("pictures/logo.ppm", package = "pixmap")[1])
x <- rnorm(10)
y <- rnorm(10, sd = 0.4) + x
plot(x, y, type = "n")
for (i in 1:10) {
    u <- runif(1)/10 + 0.1
    addlogo(logo, x[i] + c(-u, u), y[i] + c(-u, u), asp = 1)
}
points(x, y, pch = letters[1:10], col = "darkred", cex = 2)
```



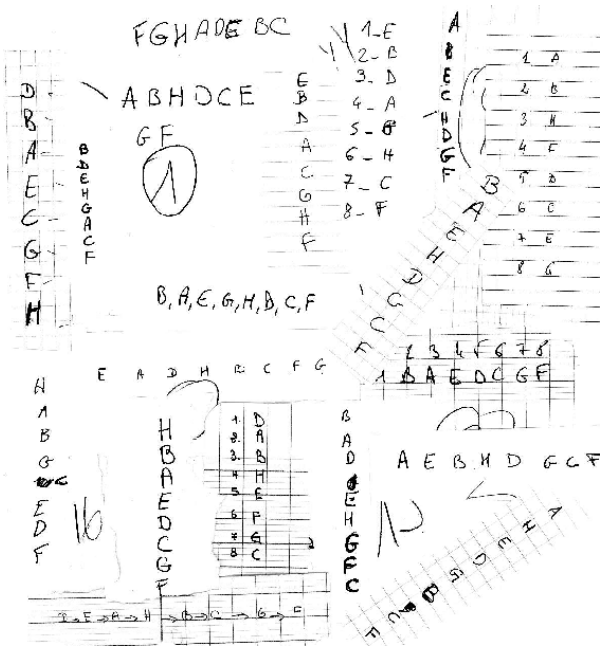
Le message de Kent Löfgren, même si ce n'était pas son objectif, en provoquant une réponse de volée, était utile. Celui de Basille Mahieu, même si c'était plus ou moins une réponse, soulevait un lièvre. Il semble possible de remplacer des étiquettes par des icônes dans des graphes de R. Comment faire ceci en obtenant un bon résultat ? Cette fiche examine cette question.

2 Un exemple

Il est toujours intéressant pour les étudiants de traiter des données en direct. Les étudiants du M1 `aMIV` ont joué le jeu. Ils acceptent de citer 8 groupes de musique qui les intéressent. Arrivent dans l'ordre :

Code	Nom	Code	Nom	Code	Nom
a	Pink Floyd	**d**	Tryo	**g**	Stevo's teen
b	Muse	**e**	Louise Attack	**h**	Louis Armstrong
c	Slipknot	**f**	Lorie		

Chacun transcrit sur un petit papier une chaîne de 8 caractères qui donne son choix en commençant par celui qu'il préfère pour finir par celui qui l'attire le moins. On ramasse les papiers :



et on dicte le résultat. On obtient le texte :

abhdcegf	dabhefgc	hbaedcgf	habgcedf
abechdgf	deahbcgf	abhfdceg	fghadebc
hbaedcgf	dbaecgfh	bdehgacf	baehdgcf
eadhbcfg	ebdacghf	ebdaghcf	badehgfc
aebhdgcf	ahedgbcf	baeghdcf	

C'est un exercice sur l'ACP pas si simple que ça. Ce qu'on veut c'est manipuler des figures où les groupes, présents en image sur des centaines de milliers de page de la toile, seraient représenter par des icônes.

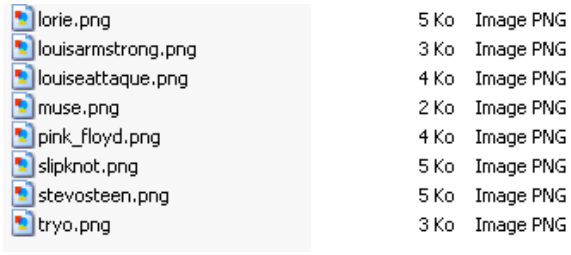
3 Faire une liste d'icônes

Créer dans le dossier de travail un dossier pour capturer, transformer et éditer les images qui vont former une liste d'icônes dans R. On l'appelle ici ce dossier `prepfigs`. Utiliser un logiciel de capture d'écran : il en existe des dizaines, par exemple à :

`http://www.01net.com/telecharger/windows/Multimedia/capture_ecran/`

Plusieurs d'entre eux sont excellents. Repérer la combinaison de touches clavier qui lance la sélection d'une portion d'écran et faire une sélection de taille constante, par exemples 40 pixels x 40 pixels. Sauvegarder chaque image dans le même format. Utiliser la recherche d'images de Google pour choisir ces illustrations. On trouvera la résultat d'une telle opération sous la forme de 8 fichiers `.png` par :

```
dir.create("prepfigs")
lf <- c("lorie.png", "louisarmstrong.png", "louiseattaque.png",
    "muse.png")
lf <- c(lf, "pink_floyd.png", "slipknot.png", "stevosteen.png",
    "tryo.png")
lf
"f1" <- function(x) {
    input <- paste("http://pbil.univ-lyon1.fr/R/donnees/qrefigs/",
        x, sep = "")
    output <- paste("./prepfigs/", x, sep = "")
    download.file(input, output, mode = "wb")
}
lapply(lf, f1)
```

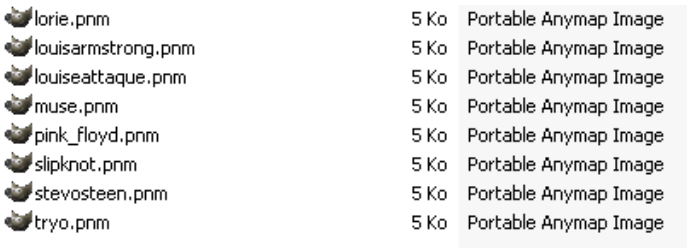


Il faut ensuite, pour lire les figures dans R, les passer au format `pnm`. Utiliser un programme de conversion et d'édition d'images en ligne de commande comme Image Magick :

`http://www.imagemagick.org/`

Après installation et inscription du dossier d'installation dans la variable d'environnement `PATH` on peut utiliser une seule ligne de commande :

```
\texttt{for \%f in (*.png) do convert  \%~nf.png \%~nf.pnm}
```



Pour simuler la réussite de cette opération :

```
"f2" <- function(x) {
    input <- paste("http://pbil.univ-lyon1.fr/R/donnees/qrefigs/",
        x, sep = "")
    output <- paste("./prepfigs/", x, sep = "")
    download.file(input, output, mode = "wb")
}
lapply(gsub(".png", ".pnm", lf), f1)
```

Il reste enfin à lire les fichiers pnm et à les ranger dans une liste dont chaque élément est un objet `pixmapRGB` de la librairie `pixmap` :

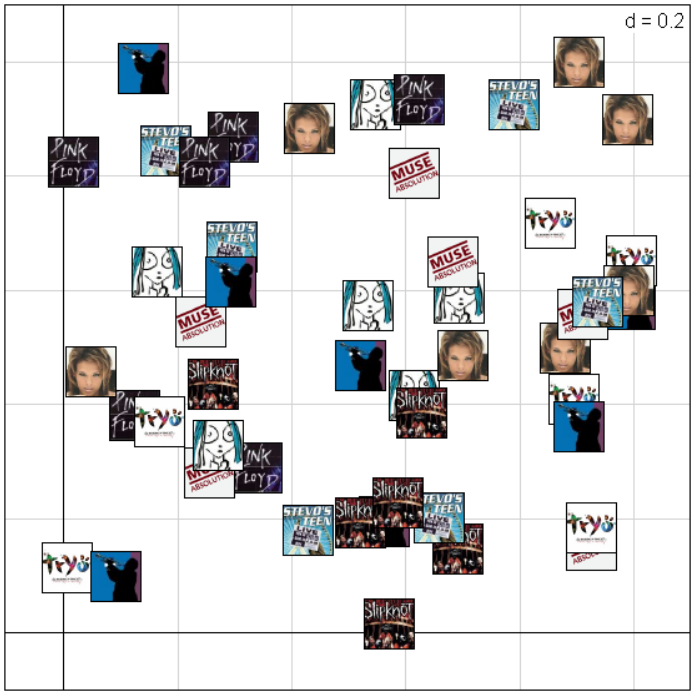
```
lpix <- list()
for (nomfic in list.files("./prepfigs", pattern = ".pnm")) {
    nomobj <- strsplit(nomfic, "[.]")[[1]][1]
    toto <- read.pnm(paste("./prepfigs/", nomfic, sep = ""))
    toto <- list(toto)
    names(toto) <- nomobj
    lpix = c(lpix, toto)
}
lpix
```

L'objet est prêt à l'emploi. On observera que c'est une liste de figures 40 pixels x 40 pixels.

4 Afficher des icônes

Une fonction s.logo complètera la collection dans la prochaine version. Pour essayer :

```
source("http://pbil.univ-lyon1.fr/R/donnees/qrefigs/logo.r")
w <- cbind.data.frame(x = runif(50), y = runif(50))
s.logo(w, lpix)
```



On veut maintenant admettre des icônes rectangulaires quelconques.

```
load(url("http://pbil.univ-lyon1.fr/R/donnees/qrefigs/capitales.rda"))
w1 <- capitales$area
area.plot(w1)
rect(min(w1$x), min(w1$y), max(w1$x), max(w1$y), col = "lightblue")
invisible(lapply(split(w1, w1$id), function(x) polygon(x[, -1],
    col = "white")))
s.logo(capitales$xy, capitales$logo, add.plot = T)
```

Il y a une erreur ? Les paramètres de la fonction sont :

dfxy data frame contenant les coordonnées des points (les icônes seront centrées en ce point).

listlogo liste d'icônes au format pixmap.

klogo numérique donnant pour chaque point le numéro de l'icône dans la liste à tracer en ce point. Le vecteur est éventuellement recyclé si sa longueur est inférieure au nombre de points. Par défaut, le paramètre vaut NULL et les icônes sont affichées dans l'ordre naturel, éventuellement répétés.

clogo numérique donnant la taille des icônes sur le graphe. Le vecteur est éventuellement recyclé si sa longueur est inférieure au nombre d'icônes. Par defaut, le paramètre vaut 1 et les icônes sont affichées sans modification.

rectlogo logique indiquant si un rectangle doit être tracé autour de l'icône. Le vecteur est éventuellement recyclé si sa longueur est inférieure au nombre d'icônes. Par défaut, le paramètre vaut TRUE et les icônes sont affichées dans un rectangle.

... comme dans tous les programmes du type `s.--`.

L'ordre des icônes est :

```
ordlogo <- names(capitales$logo)
ordlogo
```

```
 [1] "amsterdam"  "athenes"    "berlin"     "bruxelles"  "copenhague" "dublin"
 [7] "helsinki"   "lisbonne"   "londres"    "luxembourg" "madrid"     "paris"
[13] "rome"       "stokholm"   "vienne"
```

Les coordonnées sont dans l'ordre :

```
ordxy <- row.names(capitales$df)
ordxy
```

```
 [1] "Madrid"     "Paris"      "Londres"    "Dublin"     "Rome"       "Bruxelles"
 [7] "Amsterdam"  "Berlin"     "Copenhague" "Stokholm"   "Luxembourg" "Helsinki"
[13] "Vienne"     "Athenes"    "Lisbonne"
```

On utilisera donc :

```
index <- unlist(lapply(1:15, function(k) which(ordlogo == tolower(ordxy[k]))))
index
```

```
[1] 11 12  9  6 13  4  1  3  5 14 10  7 15  2  8
```

```
w1 <- capitales$area
area.plot(w1)
rect(min(w1$x), min(w1$y), max(w1$x), max(w1$y), col = "lightblue")
invisible(lapply(split(w1, w1$id), function(x) polygon(x[, -1],
    col = "white")))
s.logo(capitales$xy, capitales$logo, klogo = index, add.plot = TRUE)
```

Dans un rapport, ça peut servir.