Fiche TD avec le logiciel R : tdr603

AFC d'un jeu de données très simple

P^r Jean R. Lobry

Table des matières

1 Introduction

Les méthodes d'analyse multivariées sont une branche à part entière des statistiques. Nous vous proposons ici une introduction, très sommaire, à l'une d'entre elles, connue sous le nom l'analyse factorielle des correspondances (AFC). L'idée est de partir d'un jeu de données ayant une structure très simple.

2 Les données

Nous partons du jeu de données `HairEyeColor` [1] qui est embarqué en standard avec R, il suffit de l'invoquer avec la fonction `data()` pour pouvoir jouer avec lui. Il s'agit d'un cube de contingence dans lequel on a ventilé 592 étudiants en statistique de l'université de Delaware en fonction des modalités observées pour trois variables qualitatives :

1° la couleur des cheveux (*viz.* `Black`, `Brown`, `Red`, `Blond`) ;

2° la couleur des yeux (*viz.* `Brown`, `Blue`, `Hazel`, `Green`) ;

3° le sexe (*viz.* `Male`, `Female`).

C'est bien entendu bien trop compliqué ainsi et notre première tâche va consister à simplifier les choses. Les noms des modalités sont dans une

langue étrangère, il va falloir traduire sans trop trahir [1]. Une autre petite difficulté est que la modalité « `Brown` » est utilisée à la fois pour la couleur des cheveux et des yeux, ce qui risque de porter à confusion. Mais en définitive ce qui est le plus pénible est que nous avons un cube de données, un objet possédant trois dimensions. Nous allons commencer par nous débarrasser d'une dimension en agrégeant les données en oubliant l'information sur le sexe des individus. Avant de procéder à cette opération, regardons par curiosité les données originelles pour voir comment R présente un cube de données :

```
data(HairEyeColor)
HairEyeColor
```

```
, , Sex = Male

       Eye
Hair    Brown Blue Hazel Green
  Black    32   11    10     3
  Brown    53   50    25    15
  Red      10   10     7     7
  Blond     3   30     5     8

, , Sex = Female

       Eye
Hair    Brown Blue Hazel Green
  Black    36    9     5     2
  Brown    66   34    29    14
  Red      16    7     7     7
  Blond     4   64     5     8
```

Le cube est découpé en tranches, une pour chaque sexe. On lit ainsi par exemple qu'il y a 30 étudiants blonds aux yeux bleus et 64 étudiantes blondes aux yeux bleus :

```
HairEyeColor["Blond", "Blue", "Male"]
```

```
[1] 30
```

```
HairEyeColor["Blond", "Blue", "Female"]
```

```
[1] 64
```

Comment faire maintenant pour agréger les données ? La solution est donnée directement par la lecture des exemples de la documentation obtenue par `?HairEyeColor` :

```
x <- apply(HairEyeColor, c(1, 2), sum)
x
```

```
       Eye
Hair    Brown Blue Hazel Green
  Black    68   20    15     5
  Brown   119   84    54    29
  Red      26   17    14    14
  Blond     7   94    10    16
```

On peut vérifier que l'on a bien en tout $30 + 64 = 94$ étudiants aux yeux bleus. Traduisons maintenant les noms des modalités pour nous faciliter la lecture :

```
names(dimnames(x))
```

```
[1] "Hair" "Eye"
```

```
names(dimnames(x)) <- c("Cheveux", "Yeux")
rownames(x)
```

```
[1] "Black" "Brown" "Red"   "Blond"
```

```
rownames(x) <- c("Noir", "Brun", "Roux", "Blond")
colnames(x)
```

1. Essayer de traduire la très belle paronymie italienne « *traduttore, traditore* » est aussi vain que de vouloir chanter « *Bella ciao* » en français.

```
[1] "Brown" "Blue"  "Hazel" "Green"
```

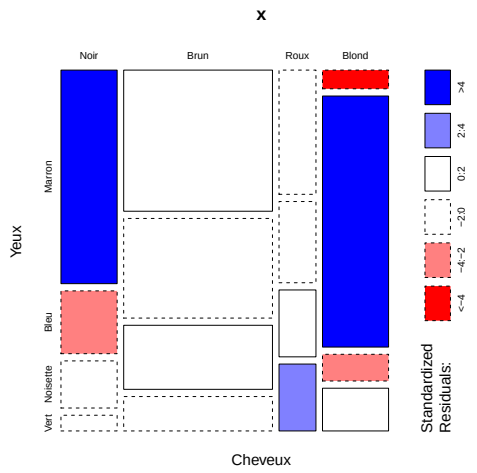
```
colnames(x) <- c("Marron", "Bleu", "Noisette", "Vert")
x
```

```
       Yeux
Cheveux Marron Bleu Noisette Vert
  Noir      68   20       15    5
  Brun     119   84       54   29
  Roux      26   17       14   14
  Blond      7   94       10   16
```

C'est bien plus clair ainsi. Notez au passage que nous avons traduit le « `Brown` » des cheveux par « `Brun` » et le « `Brown` » des yeux par « `Marron` ». Nous pouvons maintenant nous approprier les données.

3 Exploration des données

```
mosaicplot(x, shade = TRUE)
```



L'examen de la sortie de `mosaicplot()` nous permet de repérer les couples de modalités dont les effectifs sont en excès ou défaut par rapport à ce qui est attendu sous l'hypothèse nulle d'indépendance entre la couleur des yeux et des cheveux :

1° un excès d'individus aux cheveux « `Noir` » et yeux « `Marron` », aux cheveux « `Blond` » et yeux « `Bleu` » et, dans une moindre mesure, aux cheveux « `Roux` » et yeux « `Vert` » ;

2° un défaut d'individus aux cheveux « `Blond` » et yeux « `Marron` » et, dans une moindre mesure, aux cheveux « `Noir` » et yeux « `Bleu` » ainsi qu'aux aux cheveux « `Blond` » et yeux « `Noisette` ».

4 Construction d'un test d'indépendance

Notre hypothèse nulle est qu'il y a indépendance entre la couleur des cheveux et la couleur des yeux, notre hypothèse alternative est le rebours d'icelle.

Comme il n'y a pas de gros enjeux ici, nous nous contenterons d'utiliser un risque de première espèce classique $\alpha = 0.05$, nous acceptons donc de rejeter à tort l'hypothèse nulle dans 5 % des cas.

Il nous définir une statistique pour mesurer l'écart entre la distribution observée et la distribution théorique sous l'hypothèse d'indépendance. Nous utilisons la statistique du χ^2 qui dans notre cas d'espèce vaut :

```
chisq.test(x)$statistic
```

```
X-squared
 138.2898
```

La valeur de l'écart entre notre distribution observée et la distribution théorique vaut donc 138.2898. Comment faire pour savoir si cette valeur est anormalement élevée ? La fonction `r2dtable()` permet de tirer au hasard des tables de contingence ayant des totaux marginaux donnés. Tirons au hasard 1000 tables ayant les mêmes totaux marginaux que pour nos donnés :

```
x.sim <- r2dtable(n = 1000, r = rowSums(x), c = colSums(x))
x.sim[[1]]
```

```
     [,1] [,2] [,3] [,4]
[1,]   43   39   16   10
[2,]  100  114   40   32
[3,]   31   19   16    5
[4,]   46   43   21   17
```

```
rowSums(x.sim[[1]]) == rowSums(x)
```

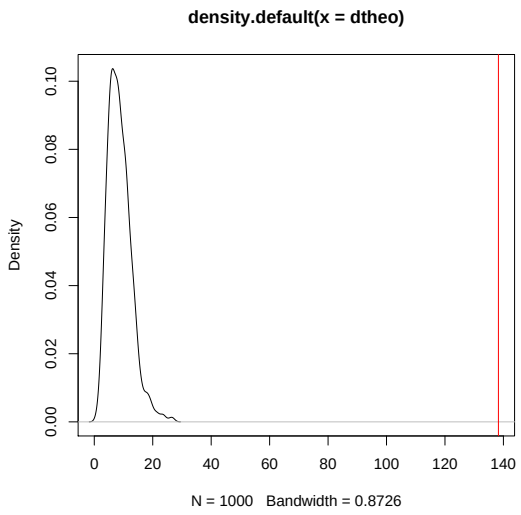
```
Noir  Brun  Roux Blond
TRUE  TRUE  TRUE  TRUE
```

```
colSums(x.sim[[1]]) == colSums(x)
```

```
 Marron      Bleu Noisette     Vert
   TRUE      TRUE      TRUE     TRUE
```

À partir de ces données simulés, il suffit de calculer la valeur de la statistique pour avoir une idée de sa distribution sous l'hypothèse nulle :

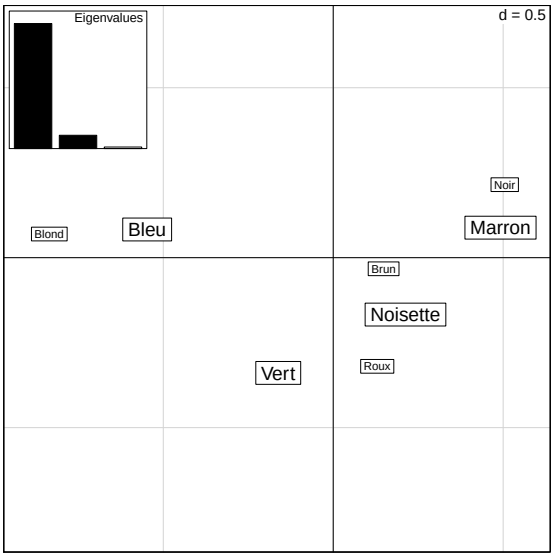
```
dtheo <- sapply(x.sim, function(x) chisq.test(x)$statistic)
plot(density(dtheo), xlim = c(0, chisq.test(x)$statistic))
abline(v = chisq.test(x)$statistic, col = "red")
```



La valeur observée de la statistique est donnée par la ligne verticale rouge, vu sa position par rapport à la distribution il est extrêmement peu probable que d'observer une valeur aussi élevée par hasard, on peut rejeter l'hypothèse nulle les yeux fermés.

5 AFC

```
library(ade4)
scatter(acp <- dudi.coa(x, scan = FALSE, nf = 2))
```



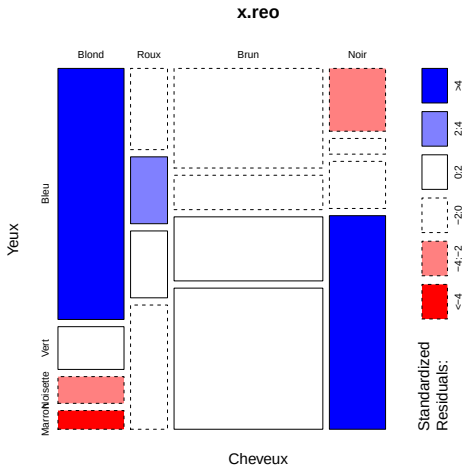
Les modalités des colonnes, ici la couleur des yeux, sont en plus gros caractères que les modalités des lignes, ici la couleur des cheveux. Le premier facteur est un gradient clair-foncé qui oppose les modalités claires (`Blond`, `Bleu`) aux modalités foncées (`Noir`, `Marron`). Les cheveux clairs vont avec les yeux clairs, les cheveux foncés avec les yeux foncés.

On peut utiliser les coordonnées des modalités sur le premier facteur pour ré-ordonner le tableau de départ, ceci permet de bien mettre en évidence le gradient clair-foncé dans les données originelles :

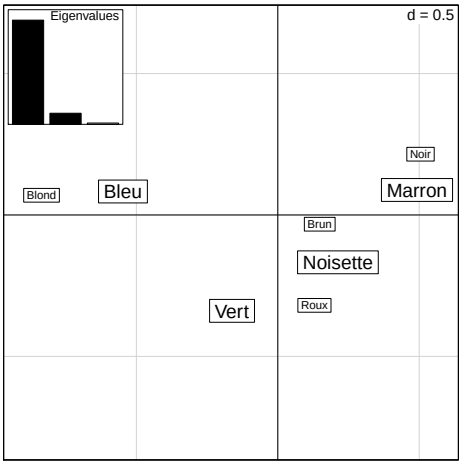
```
x.reo <- x[order(acp$li[, 1]), order(acp$co[ , 1])]
x.reo
```

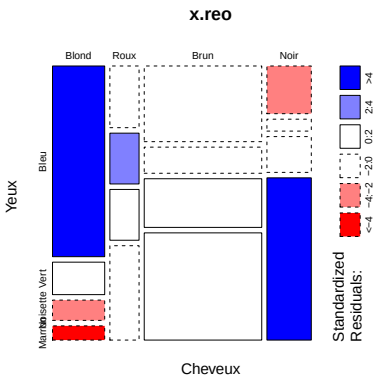
```
      Yeux
Cheveux Bleu Vert Noisette Marron
  Blond   94   16       10      7
  Roux    17   14       14     26
  Brun    84   29       54    119
  Noir    20    5       15     68
```

```
mosaicplot(x.reo, shade = TRUE)
```



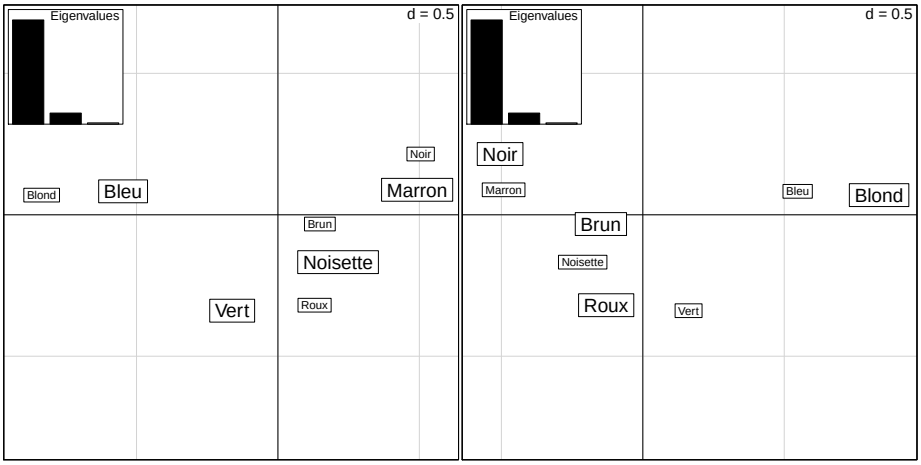
QUAND deux modalités sont associées, comme le « `Bleu` » des yeux et le « `Blond` » des cheveux, alors elles sont voisines sur un plan factoriel. Mais attention la réciproque est fausse puisqu'il s'agit de projections. Dans le plan factoriel ci-dessous, cerclez les couples de modalités qui apparaissent en bleu dans la sortie de `mosaicplot()`.





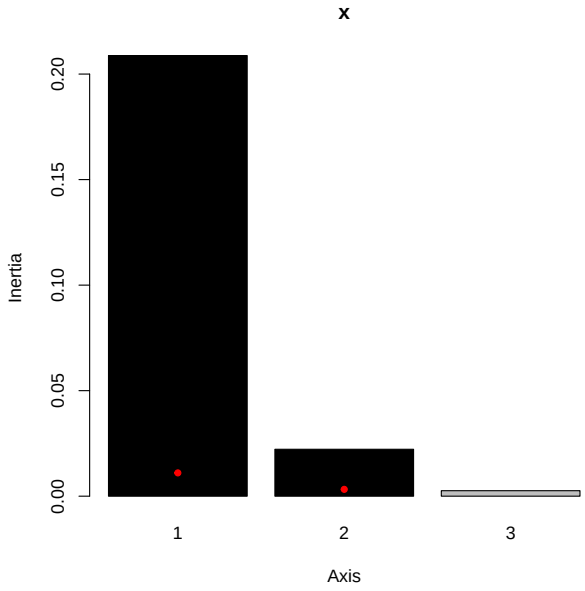
L'ANALYSE est parfaitement symétrique : on obtient exactement les mêmes résultats avec les données transposées :

```
par(mfrow = c(1,2))
scatter(acp)
scatter(dudi.coa(t(x), scan = FALSE, nf = 2))
```



On peut avoir une idée de la significativité des facteurs en faisant rouler l'AFC sur notre jeu de 1000 tables simulées sous l'hypothèse nulle.

```
sapply(x.sim, function(x) dudi.coa(x, scan = FALSE, nf = 2)$eig[1:2]) -> res
moy <- rowMeans(res)
myscreeplot <- function(x, y, npcs = 15, nf = x$nf,
                        p.pch = 19, p.col = "red", p.cex = 0.75, ...){
  screeplot(x, npcs = npcs, ...) ; res.barplot <- barplot(x$eig, plot = FALSE)
  imax <- min(npcs, x$rank)
  points(res.barplot[1:imax, 1], y[1:imax], pch = p.pch, col = p.col, cex = p.cex)
}
myscreeplot(acp, moy, 3)
```



Il n'est donc pas déraisonnable d'essayer d'interpréter le deuxième facteur. Il oppose les individus ayant des cheveux `Roux` et des yeux `Vert` aux autres. C'est le facteur qui oppose les moldus aux sorciers.

Références

[1] R.D. Snee. Graphical display of two-way contingency tables. *The American Statistician*, 28 :9–12, 1974.