

AFC

Présentation des résultats

Le package contient une base de données concernant le choix d'hébergement pour les vacances intitulé "heb".

```
heb
```

```
##          hotel location camping residencesec total
## agriculteur    155      129      239          0    523
## cadresup      1556     1821     1003        1521   5901
## inactif        1944      967      682        1333   4926
## ouvrier        1124     2176     2594        1038   6932
## total          4779     5093     4518        3892  18282
```

Cette base de données est avec totaux. Il y a une autre base de données intitulée "comp" qui est sans totaux.

```
comp
```

```
##   Q10 Q1N Q20 Q2N
## 1   1   0   0   1
## 2   1   0   1   0
## 3   1   0   1   0
## 4   0   1   1   0
## 5   0   1   0   1
## 6   1   0   0   1
```

Les fonctions de ce package ajoutent les totaux automatiquement, donc pour utiliser ces fonctions, il exige une base de données sans totaux. Dans la suite de démonstration de fonction je vais tester sur "comp". Pour ajouter les totaux :

```
total(comp)
```

```
##          Q10 Q1N Q20 Q2N total
## 1          1   0   0   1      2
## 2          1   0   1   0      2
## 3          1   0   1   0      2
## 4          0   1   1   0      2
## 5          0   1   0   1      2
## 6          1   0   0   1      2
## total      4   2   3   3     12
```

Table des profils lignes :

```
tabProLigne(comp)
```

```
##          Q10   Q1N   Q20   Q2N total
## 1      0.500 0.000 0.00 0.50      1
## 2      0.500 0.000 0.50 0.00      1
```

```
## 3      0.500 0.000 0.50 0.00      1
## 4      0.000 0.500 0.50 0.00      1
## 5      0.000 0.500 0.00 0.50      1
## 6      0.500 0.000 0.00 0.50      1
## total 0.333 0.167 0.25 0.25      1
```

Table des profils colonnes :

```
tabProColonne(comp)
```

```
##      Q10 Q1N  Q20  Q2N total
## 1      0.25 0.0 0.000 0.333 0.167
## 2      0.25 0.0 0.333 0.000 0.167
## 3      0.25 0.0 0.333 0.000 0.167
## 4      0.00 0.5 0.333 0.000 0.167
## 5      0.00 0.5 0.000 0.333 0.167
## 6      0.25 0.0 0.000 0.333 0.167
## total 1.00 1.0 1.000 1.000 1.000
```

Table des valeurs théoriques :

```
tabValTheo(comp)
```

```
##      Q10      Q1N Q20 Q2N total
## 1      0.6666667 0.3333333 0.5 0.5      2
## 2      0.6666667 0.3333333 0.5 0.5      2
## 3      0.6666667 0.3333333 0.5 0.5      2
## 4      0.6666667 0.3333333 0.5 0.5      2
## 5      0.6666667 0.3333333 0.5 0.5      2
## 6      0.6666667 0.3333333 0.5 0.5      2
## total 4.0000000 2.0000000 3.0 3.0      12
```

Pour mesurer la liason entre deux variables qualitatives, on utilise un test de chiq2 :

```
d2(comp)
```

```
## $`X-squared`
## [1] 12
##
## $valcritique
## [1] 24.99579
##
## $Summary
## [1] "Les variables sont independantes."
```

Tableau de contributions au khi-2 :

```
contrX2(comp)
```

```
##          Q10   Q1N   Q20   Q2N total
## 1         1.39  2.78  4.17  4.17  12.5
## 2         1.39  2.78  4.17  4.17  12.5
## 3         1.39  2.78  4.17  4.17  12.5
## 4         5.56 11.11  4.17  4.17  25.0
## 5         5.56 11.11  4.17  4.17  25.0
## 6         1.39  2.78  4.17  4.17  12.5
## total 16.67 33.33 25.00 25.00 100.0
```

Table de fréquence :

```
tabFreq(comp)
```

```
##          Q10   Q1N   Q20   Q2N total
## 1         0.083 0.000 0.000 0.083 0.167
## 2         0.083 0.000 0.083 0.000 0.167
## 3         0.083 0.000 0.083 0.000 0.167
## 4         0.000 0.083 0.083 0.000 0.167
## 5         0.000 0.083 0.000 0.083 0.167
## 6         0.083 0.000 0.000 0.083 0.167
## total 0.333 0.167 0.250 0.250 1.000
```

Centre gravite de colonnes :

```
gcol(comp)
```

```
##          Q10          Q1N   Q20   Q2N
## total 0.33333333 0.1666667 0.25 0.25
```

Valeurs propres servent à choisir le nombre des axes pour présenter le graphique. Vecteurs propres servent à calculer les condonnées.

```
valeurp(comp)
```

```
## [1] 5.000000e-01 5.000000e-01 4.440892e-16
```

```
vecteurp(comp)
```

```
##          [,1]          [,2]          [,3]
## [1,] -5.773503e-01  0.000000e+00  0.5773503
## [2,]  8.164966e-01  1.110223e-16  0.4082483
## [3,] -5.551115e-16 -7.071068e-01 -0.5000000
## [4,] -4.440892e-16  7.071068e-01 -0.5000000
```

Coordonnees des profils-lignes :

```
coProLigne(comp)
```

```
##      [,1]      [,2]      [,3]
## 1 -0.5  0.7071068  4.440892e-16
## 2 -0.5 -0.7071068  4.440892e-16
## 3 -0.5 -0.7071068  4.440892e-16
## 4  1.0 -0.7071068 -1.498801e-15
## 5  1.0  0.7071068 -1.498801e-15
## 6 -0.5  0.7071068  4.440892e-16
```

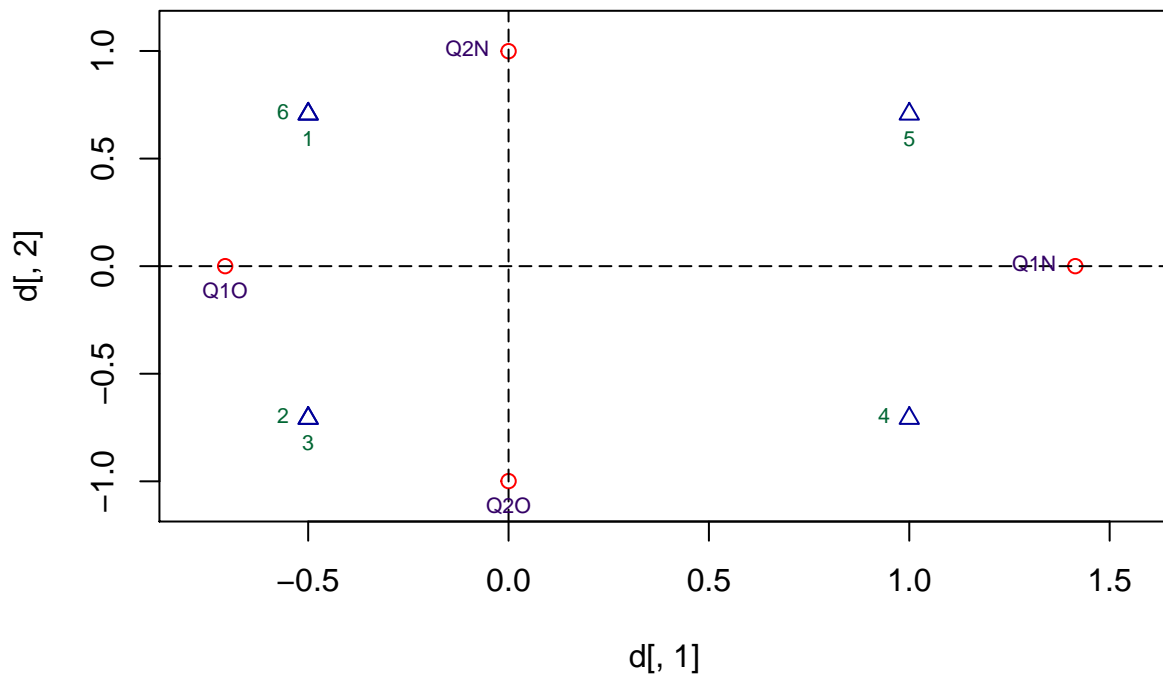
Coordonnees des profils-colonnes :

```
coProColonne(comp)
```

```
##      [,1]      [,2]      [,3]
## Q10 -7.071068e-01  7.850462e-17  2.107342e-08
## Q1N  1.414214e+00  2.355139e-16 -7.112281e-08
## Q20  2.355139e-16 -9.990000e-01 -9.648994e-09
## Q2N  4.710277e-16  9.990000e-01 -9.648994e-09
```

Graphiques finales :

```
plotAFC(comp)
```



Une autre demonstration avec la base de donnée heb :

```
h <- heb[1:4, 1:4]  
plotAFC(h)
```

