

TP: Analyse des correspondances multiple

Résumé

Comparaison entre des analyses des correspondances appliquées à une table de contingence, un tableau disjonctif complet et un tableau de Burt comme généralisation de l'Analyse des correspondances à l'analyse des correspondances multiple. Exemple d'AFCM.

1 Généralisation de l' AFC

Illustration des différentes façons d'obtenir une généralisation de l' AFC. Comparer les résultats (valeurs propres, graphiques) obtenus par AFC simple ou multiple de différents tableaux construits à partir des deux mêmes variables fictives.

1.1 Création des données

```
data afcfc;
input ident $ csp $ sport $ effectif;
cards;
quatre csp3 S 1
un      csp1 R 1
deux   csp2 R 1
trois  csp2 S 1
cinq   csp3 T 1
six    csp4 T 1
;
run;
```

1.2 AFC de la table de contingence

```
proc corresp data=afcfc observed out=resul;
tables csp, sport; /*attention à la virgule~!*/
```

```
weight effectif;
run;
%gafcx;
```

1.3 AFC du tableau de Burt

```
proc corresp data=afcfc observed out=resul;
tables csp sport; /*attention à la virgule~!*/
weight effectif;
run;
%gafcx;
```

1.4 AFC du tableau disjonctif complet

```
proc corresp data=afcfc observed out=resul;
tables ident, csp sport; /*attention à la virgule~!*/
weight effectif;
run;
%gafcx;
```

2 Exemple simple d'AFCM avec SAS

Nous nous proposons de rechercher si les regroupements de races de chiens issus d'une AFCM sont compatibles avec la fonction de ces chiens considérée comme variable supplémentaire.

2.1 Les données

Les données sont extraites de Bréfort (1982). Elles contiennent le descriptif des qualités de 27 races de chien : tailles, poids, vitesse, intelligence, codées sur trois modalités (1 faible, 2 moyen, 3 fort), affection et agressivité sur deux modalités (1 faible, 2 forte), enfin la fonction sur trois modalités (1 compagnie, 2 chasse, 3 utilité). L'obtention de graphes des modalités explicites nécessite un recodage précis des modalités avec des libellés facilement identifiables.

Charger le fichier [chiens.dat](#).

```
data sasuser.chiens;
infile 'chiens.dat';
```

```

input race $ taille $ poids $ velocite $
intellig $ affect $ agress $ fonction $;
select (taille);
when('1')taille='T-';
when('2')taille='T+';
when('3')taille='T++';
otherwise;end;
select (poids);
when('1')poids='P-';
when('2')poids='P+';
when('3')poids='P++';
otherwise;end;
select (velocite);
when('1')velocite='V-';
when('2')velocite='V+';
when('3')velocite='V++';
otherwise;end;
select (intellig);
when('1')intellig='I-';
when('2')intellig='I+';
when('3')intellig='I++';
otherwise;end;
select (affect);
when('1')affect='Af-';
when('2')affect='Af+';
otherwise;end;
select (agress);
when('1')agress='Ag-';
when('2')agress='Ag+';
otherwise;end;
select (fonction);
when('1')fonction='Com';
when('2')fonction='Cha';
when('3')fonction='Uti';
otherwise;end;
run;

```

2.2 Analyse avec ODS

Les graphiques de cette première approche ne sont pas très explicites.

```

ods html;
ods graphics on;
proc corresp data=sasuser.chiens mca out=resul;
tables taille--fonction;
run;
ods graphics off;
ods html close;

```

La même analyse en ajoutant les identificateurs des races des chiens et la variable fonction en supplémentaire.

```

ods html;
ods graphics on;
proc corresp data=sasuser.chiens out=resul;
tables race,taille--fonction;
supplementary fonction;
run;
ods graphics off;
ods html close;

```

L'édition d'un graphique plus explicite représentant chaque race de chien par une couleur dépendant de sa fonction nécessite quelques manipulations. Donner une interprétation.

```

proc sort data=resul out=result;
by _name_;
run;
proc sort data=sasuser.chiens out=chienst;
by race;
run;
data fusion;
merge result chienst (rename=(race=_name_)
keep=race fonction);
by _name_;
run;
%let ident=_name_;

```

```

%let x=1;
%let y=2;
%let nc=4;
data anno;
  set fusion;
  retain xsys ysys '2';
  style='swiss';
  y= dim&y;
  x= dim&x;
  select (fonction);
  when('Cha') color='green';
  when('Com') color='blue';
  when('Uti') color='red';
  otherwise color='black';
end;
text=substr(&ident,1,&nc);
size=0.6;
label y = "Axe &y"
      x = "Axe &x";
keep x y text xsys ysys size color;
proc gplot data=anno;
  title;
  axis1 length=8cm; /* attention taille */
  axis2 length=8cm;
  symbol1 v=none;
  plot y*x=1 / annotate=anno frame href=0 vref=0
      haxis=axis1 vaxis=axis2 ;
run;
goptions reset=all;
quit;
    
```

3 Le même exemple avec FactoMineR

3.1 Transfert des données

Pour éviter de recoder les données dans R, celles-ci sont exportées en l'état.

```

/* Exportation d'un fichier SAS en format .csv */
proc export data=sasuser.chiens
    
```

```

  outfile= "chiens.csv"
  DBMS=CSV REPLACE;
run;
    
```

Rechercher où SAS a "rangé" le fichier `chiens.csv`, sans doute dans le répertoire duquel l'exécution a été lancée. Éventuellement le déplacer dans le répertoire courant de R.

```

# retour à R
chiens=read.csv("chiens.csv")
# Vérifier que tout s'est bien passé
#Attention, la première colonne identifie la race du
#chien et donc chaque ``individu``
dimnames(chiens)[[1]]=as.character(chiens[,1])
chiens=chiens[,-1]
summary(chiens)
    
```

3.2 AFCM avec R

La librairie `FactoMineR` est particulièrement conçue pour exécuter et représenter des analyses factorielles des correspondances.

```

library(FactoMineR)
#afcm avec la fonction en supplémentaire
afcm=MCA(chiens,quali.sup=7,graph=F)
plot(afcm, choix="ind",habillage="quali")
#sans les individus
plot(afcm,habillage="quali",invisible="ind")
    
```

Seule "erreur" dans la représentation : fournir des parts de "variance" ou "chi2" expliquées par les axes qui n'ont pas de signification statistique à cause de la présence de valeurs propres "artificielles" non nulles issues de la construction du tableau de Burt.