T.P. SAS numéro 07

Cette feuille de T.P. est consacrée à l'étude des macro-commandes de SAS, appelées tout simplement **macros**. On se reportera au chapitre 12 du cours polycopié pour des compléments sur ces macros. Le contexte choisi pour illustrer cette notion est celui de l'A.C.P. et certaines macros ci-dessous reprennent donc des procédures étudiées dans la feuille précédente.

Avertissements :
- Les macros SAS ne peuvent fonctionner que si l'on dispose d'un environnement correct sous UNIX. Cet environnement est mis en œuvre par les 2 fichiers `.login` et `.tcshrc`. Si on ne les a pas copiés lors de la séance de TP 01, on doit faire maintenant :
  `cp ~besse/.login ~`
  `cp ~besse/.tcshrc ~`
- Il faut noter que lorsqu'on a fait une erreur dans une macro, une fois cette erreur détectée et corrigée, on doit sortir de SAS puis y revenir pour activer la macro corrigée.

Préambule

Les macro-commandes de SAS permettent d'enchaîner l'exécution de plusieurs commandes SAS au moyen d'une seule instruction (qui sera appelée macro), comportant, éventuellement, un ou plusieurs paramètres.

Afin de les conserver, et pour que SAS les reconnaisse automatiquement, il faut créer dans son répertoire principal le répertoire "macros" (utiliser la commande UNIX : `mkdir macros`).

Chaque fois que l'on crée une nouvelle macro, on doit l'enregistrer dans un fichier placé dans le répertoire "macros". Ce fichier doit avoir pour nom l'identificateur de cette macro, suivi de l'extension **.sas**. Par exemple, si l'on crée la macro appelée **impres**, on doit l'enregistrer dans le fichier **impres.sas** du repertoire "macros". Son exécution sera alors lancée par la commande `%impres` au sein d'un programme SAS.

Exercice 07.1

On réalise ici une première macro-commande élémentaire.

Sans entrer dans SAS, dans le répertoire "macros", créer le fichier `impres.sas` contenant les instructions suivantes :

```
%macro impres;
%* cette macro affiche dans la fenetre OUTPUT la derniere table SAS creee;
proc print; /* voici un autre commentaire */
run;
%mend impres;
```

Commentaires. Le caractère `%` est, dans SAS, le symbole des macros. Dans sa première ligne, une macro doit toujours comporter l'instruction `%macro` suivie du nom (identificateur) de la macro (ici `impres`). De même, une macro se termine toujours par une ligne comportant l'instruction `%mend`, éventuellement suivie du nom de la macro (`impres` est donc facultatif ici). Noter qu'une ligne de commentaire débute toujours par `%*` (au lieu de `*`) et se termine par le caractère de ponctuation `;` (les commentaires sur une partie de ligne ayant toujours la même syntaxe).

Une fois le fichier `impres.sas` sauvegardé, pour voir ce que produit cette macro, rentrer dans SAS (dans votre répertoire de T.P.) et taper les instructions suivantes :

```
data essai;
set sasuser.notes;
%impres;
```

Noter que l'exécution d'une macro s'obtient en faisant précéder son nom (identificateur) de `%`.

Exercice 07.2

On introduit maintenant la notion de paramètre à l'intérieur d'une macro.

Modifier la macro précédente pour qu'elle prenne en paramètre le nom de la table à afficher ; pour cela, créer le fichier `impres2.sas` (toujours dans le répertoire `macros`) contenant la séquence d'instructions suivante :

```
%macro impres2(table);
proc print data=&table;
run;
%mend impres2;
```

Cette macro affiche dans la fenetre OUTPUT de SAS le contenu de "table" ; ainsi, "table" est un paramètre de la macro `impres2`. Pour utiliser ce paramètre, il suffit de faire précéder son nom de `&` dans la suite du programme, le symbole "`&`" (*et commercial*, ou *eperluette*, se dit "et") étant celui des paramètres dans les macros.

Pour tester cette macro, faire dans SAS
`%impres2(sasuser.crime);`
puis
`%impres2(sasuser.temp);`

Exercice 07.3

De la même façon, créer le fichier `graph.sas` (toujours dans le répertoire `macros`) contenant la séquence d'instructions suivante :

```
%macro graph(table,xvar,yvar);
proc gplot data=&table;
plot &yvar*&xvar;
run;
quit;
%mend;
```

Cette macro réalise le graphique haute résolution de la variable "yvar" en fonction de la variable "xvar", ces deux variables étant prises dans "table".

Tester la macro en faisant :
`%graph(sasuser.crime,LARCENY,RAPE);`
puis
`%graph(sasuser.notes,MECA,STAT);`

Exercice 07.4

Créer le fichier `pca.sas` contenant une macro paramétrée, réalisant l'A.C.P. réduite et créant les tables SAS `comprin` et `eltpr`. La tester en faisant :

```
%pca(sasuser.crime);
%impres2(comprin);
%impres2(eltpr);
```

Exercice 07.5

On aborde maintenant des macros un peu plus complexes.

Créer le fichier `pcaind.sas` suivant (toujours dans le répertoire `macros`) :

```
%macro pcaind(table,name,xx=1,yy=2);
proc princomp data=&table vardef=n out=comprin;
run;
data anno;
set comprin;
```

```
x=prin&xx;
y=prin&yy;
xsys='2';
ysys='2';
text=&name;
size=1;
keep x y text xsys ysys size;
run;
proc gplot data=anno;
label x="axe &xx";
label y="axe &yy";
symbol v='none';
plot y*x / annotate=anno href=0 vref=0;
run;
quit;
goptions reset=all;
%mend;
```

Cette macro réalise l'A.C.P. réduite de "table", puis représente les individus à partir des composantes principales fournies par l'A.C.P. Les paramètres de cette macro sont :

- `table` : table SAS contenant les données ;
- `name` : nom de la variable contenant les identificateurs des individus ;
- `xx` et `yy` ; à propos de ces derniers paramètres, il convient de noter diverses choses : chacun de ces 2 paramètres représente une partie seulement du nom de la variable considérée (on peut donc paramétrer seulement une partie de nom) ; par défaut, ces paramètres valent `xx=1` et `yy=2` ; dans l'appel de la macro, ou bien on ne met pas ces paramètres, et ils prennent alors la valeur par défaut, ou bien on les y met, de la façon suivante : `xx=5, yy=8`.

Tester cette macro en faisant (successivement) :

```
%pcaind(sasuser.crime,staten);           /*  premier plan factoriel de "crime" */
%pcaind(sasuser.crime,staten,xx=2,yy=3); /* deuxieme plan factoriel de "crime" */
%pcaind(sasuser.notes,ind);              /*  premier plan factoriel de "notes" */
%pcaind(sasuser.notes,ind,xx=2,yy=3);    /* deuxieme plan factoriel de "notes" */
```

Exercice 07.6

Créer maintenant le fichier `pcavalp.sas` (répertoire `macros`) contenant la macro réalisant ce qui a déjà été fait à l'exercice 06.4. La table SAS utilisée dans cette macro sera paramétrée et on ne fera rien afficher dans la fenêtre OUTPUT de SAS.

Tester cette macro en faisant :

```
%pcavalp(sasuser.crime);
%impres2(lambda);
```

Ensuite, pour représenter l'éboulis des valeurs propres, faire :

```
data lambda1;
set lambda;
k = _n_;  /*  (1)  */
run;
%impres2(lambda1);
%graph(lambda1,k,valpr);
```

(1) La variable `_n_` de SAS prend les valeurs de 1 à n si n est le nombre d'observations de la table SAS en cours de traitement. Cette variable existe de manière implicite (elle n'apparaît pas à l'affichage) dans toute table SAS.

Exercice 07.7

Réaliser maintenant une macro contenant elle-même des appels à des macros et faisant la même chose que ce qui a été fait dans l'exercice précédent. La tester sur la table `sasuser.notes`.