

LES MACRO-EXPRESSIONS AVEC ARGUMENT

Reprenons la macro expression simple sans argument :

```
%MACRO micmac ;  
PROC PRINT;  
%MEND;
```

Si ce texte figure dans la fenêtre PROG d'une session SAS, alors vous pouvez écrire après la dernière ligne du texte (après le %MEND) :

```
%micmac;  
RUN;
```

ce qui aura pour effet d'afficher le contenu du dernier tableau (DATA) créé.

Si vous voulez créer maintenant une macro-expression qui imprime un tableau (DATA) SAS quelconque, il faut alors paramétrer le nom du tableau (DATA) à imprimer, c'est-à-dire lui donner un nom symbolique variable auquel on fera référence dans le texte de la macro-expression. Si, par exemple, vous choisissez le nom *madata* pour désigner le tableau (DATA) à imprimer, le texte de la macro-expression sera alors de la forme :

```
%MACRO micmac (madata);  
PROC PRINT DATA=&madata;  
%MEND;
```

Dans le texte ci-avant, l'expression *&madata* reprend le texte contenu dans l'argument *madata*. Ainsi, si vous écrivez après la macro-expression précédente les commandes

```
%micmac (lecture);  
RUN;
```

vous donnez alors l'ordre d'impression du tableau (DATA) *lecture* dans la macro-expression qui contient implicitement en seconde ligne la procédure

```
PROC PRINT DATA=lecture;
```

Voici un autre exemple de macro-expression avec un argument. Elle calcule le carré (*car*) et le cube (*cube*) d'un nombre *x* quelconque avant de les afficher à l'écran. Faites un essai en utilisant les bonnes commandes.

```
%MACRO calcul (x);  
car=&x*&x;  
cube=&x*&x*&x  
PROC PRINT;  
%MEND;
```