STT1682 – Progiciels en Statistique et Actuariat
Cours 3 - Étape DATA et Fonctions

Programmation en SAS

Étape DATA - Options

Lors de la lecture ainsi que lors de la création de base de données, il est possible de spécifier certaines options à SAS afin d'optimiser le code SAS. Les options seront généralement utilisés pour filtrer et/ou reformatter la base de donnée.

Les options d'une étape DATA (**ou de tout autre étape SAS**) pourront être spécifié à deux endroits selon nos besoins :

Traitement doit être effectuer avant tout traitement sur les données => Options à la lecture (déclaration SET)

Traitement doit être effectuer après tout traitement sur les donnée => Options à l'écriture (déclaration DATA)

La syntaxe pour utiliser les différentes options est de simplement écrire les options à l'intérieur de paranthèses à la fin du nom de la base SAS sur laquelle elles s'appliquent. Par exemple :

Syntaxe :

```
DATA LIBRAIRIE1.NOMBDSORTANTE (...Options à l'écriture...);
SET LIBRAIRIE2.NOMBDENTRANTE(...Options à la lecture...);
RUN;
```

Il sera aussi possible d'utiliser la majorité des options en tant que déclaration SAS à l'intérieur de l'étape DATA. Pour des fins d'efficacité et d'uniformité, il est recommandé des les utiliser en tant qu'options aux déclarations SET/DATA.

Option KEEP

L'option KEEP sert à seulement garder un sous-groupe des variables de la base de donnée. Toutes variables listés dans le KEEP seront conservés et les autres enlevés de la base de donnée .

Syntaxe :

```
DATA LIBRAIRIE1.NOMBDSORTANTE (KEEP= VAR1 VAR2 ...);*Option ecriture;
SET LIBRAIRIE2.NOMBDENTRANTE(KEEP= VAR1 VAR2 ...);*Option lecture;
KEEP VAR1 VAR2...;*Option interieur etape DATA;
RUN;
```

Note : Comme toutes les options, on peut soit l'utiliser à la lecture ou à l'écriture. Il est fortement recommandé d'utiliser cette option à la **lecture** afin de limiter initialement le nombre de variable et ainsi accélérer le temps d'exécution.

Option DROP

L'option DROP est tout simplement l'inverse de l'option KEEP. Toutes variables listés dans le DROP seront enlevés et les autres conservés de la base de donnée .

Syntaxe :

```
DATA LIBRAIRIE1.NOMBDSORTANTE(DROP= VAR1 VAR2 ...);*Option ecriture;
SET LIBRAIRIE2.NOMBDENTRANTE(DROP= VAR1 VAR2 ...);*Option lecture;
DROP VAR1 VAR2...;*Option interieur etape DATA;
RUN;
```

Note : Encore une fois, il est plus efficace d'utiliser le DROP à la lecture des données

Option OBS

L'option OBS est un filtre sur le nombre d'observation conservé dans la base de donnée. À l'aide de cette option, on peut conserver les X premières observations d'une base de donnée.

Syntaxe :

```
DATA LIBRAIRIE1.NOMBDSORTANTE(OBS= 1000);*Option ecriture;
SET LIBRAIRIE2.NOMBDENTRANTE(OBS= 1000);*Option lecture;
RUN;
```

Note : Cette option sera généralement utilisé pour tester un programme afin de ne pas exécuter le code sur toutes les observations à chaque itération. Encore une fois, plus efficace d'utiliser cette option à la lecture des données

Option RENAME

L'option RENAME sert à renommer une variable soit à la lecture d'une base de donnée ou lors de l'écriture.

Syntaxe :

```
DATA LIBRAIRIE1.NOMBDSORTANTE(RENAME= (NOM_PREC1=NOM_NOUV1
NOM_PREC2=NOM_NOUV2...) );*Option ecriture;
SET LIBRAIRIE2.NOMBDENTRANTE(RENAME= (NOM_PREC1=NOM_NOUV1
NOM_PREC2=NOM_NOUV2...) );*Option lecture;
RENAME NOM_PREC1=NOM_NOUV1 NOM_PREC2=NOM_NOUV2...;*Option interieur etape DATA;
RUN;
```

Option WHERE

L'option WHERE sert à appliquer des filtres aux bases SAS directement lors de l'écriture/lecture. Un filtre sert à limiter la quantité d'observation en gardant seulement celles respectant une ou plusieurs conditions.

Les conditions doivent êtres des égalités/inégalités pouvant être soit VRAI ou FAUSSE

Syntaxe :

```
DATA LIBRAIRIE1.NOMBDSORTANTE(WHERE= (...FILTRES... ) );*Option ecriture;
SET LIBRAIRIE2.NOMBDENTRANTE(WHERE= (...FILTRES... ) );*Option lecture;
WHERE ...FILTRES...;*Option interieur etape DATA;
RUN;
```

Opérateurs logiques

Syntaxe	**Description**
= ou EQ	Égale à
< ou LT	Strictement plus petit que
<= ou LE	Plus petit ou égale que
> ou GT	Strictement plus grand que
>= ou GE	Plus grand ou égale
^= ou NE	Différent de
IN (...)	Parmis la liste cité entre paranthèses
NOT ...	N'est pas (pour inverser une condition)
IS ...	Est un état spécial(MISSING/NULL...)
AND	ET (Pour lier plusieurs filtres)
OR	OU (Pour lier plusieurs filtres)

Opérateurs mathématiques & textuels

Syntaxe	**Description**
+	Addition
-	Soustraction
*	Multiplication
/	Division
**	Exposant
!!	Concaténation

Exemple :

```
DATA Exemple1(WHERE= (POLICE IN ("01","02","03","04" ) AND  PRIME/1000 ^=1 ));
INPUT POLICE $ NOM $ PRIME;
DATALINES;
01 Isabelle 500
02 Luc 100
03 Jean 1000
04 Charles 2500
05 Pierre 800
;
RUN;

DATA Exemple1_2(KEEP= POLICE NOM WHERE= (NOM NOT EQ "Luc" ));
SET Exemple1(WHERE= (PRIME >= 800 OR PRIME=100) );
RUN;
```

Fonctions SAS

Il existe une multitude de fonctions prédéfinis en SAS et vous pouvez trouver une liste exhaustive sur le site de support officiel de SAS. Pour ce cours, nous utiliserons les fonctions suivantes :

Fonctions arithmétiques/mathématiques

Syntaxe	**Description**
ABS(VAR1)	Valeur absolu
MIN(VAR1,VAR2...)	Minimum
MAX(VAR1,VAR2...)	Maximum
SUM(VAR1,VAR2...)	Somme
MEAN(VAR1,VAR2...)	Moyenne
LOG(VAR1)	Logarithme naturel
SQRT(VAR1)	Racine Carrée
SIGN(VAR1)	Signe de la variable
MOD(VAR1,modulo)	Modulo
STD(VAR1)	L'écart type d'une variable
VAR(VAR1)	La variance d'une variable

Note : Les arguments des fonctions ci-dessous peuvent être soit des variables, des constantes, des expressions ou d'autres fonctions

Fonctions de traitement de texte simples

Syntaxe	Description
LENGTH(VAR1)	Longueur (Length) de la chaîne de caratère
MISSING(VAR1)	Retourne VRAI si la valeur de la variable est manquante sinon FAUX
COMPRESS(VAR1)	Élimine tous les espaces d'une chaîne de caratères
UPCASE(VAR1)	Renvoie la chaîne de texte en majuscule seulement
LOWCASE(VAR1)	Renvoie la chaîne de texte en miniscule seulement

Note : Les arguments des fonctions ci-dessous peuvent être soit des variables, des constantes, des expressions ou d'autres fonctions

Fonctions de traitement de texte complexes

Fonction INPUT
La fonction INPUT lie une variable texte (chaîne de caratères) selon un format spécifié et retourne la valeur SAS correspondate. Elle est souvent utilisé pour convertir des dates textes en date SAS ou des variables textes représentant des nombres en variables numériques.

Syntaxe :

```
INPUT(VAR1,FORMAT1);
```

Exemple :
```
VAR1="10";
NEW_VAR1=INPUT(VAR1,8.);
```

Fonction PUT
La fonction PUT lie une variable texte (chaîne de caratères) ou numérique et l'écrit selon format spécifié. Elle est souvent utilisé pour convertir des variables numériques en variables de chaîne de caratères.

Syntaxe :

```
PUT(VAR1,FORMAT1);
```

Exemple :
```
VAR1=10;
NEW_VAR1=PUT(VAR1,$2.);
```

Fonction SUBSTR
La fonction SUBSTR sert à extraire une sous-chaîne de caratère d'une autre chaîne de caratère. Elle sera frequement utilisé pour définir des variables textes à partir d'autres variables textes.

Syntaxe :

SUBSTR(VAR1,X,Y);*Extrait la chaine de caratere Y de long a partir de la Xe position;

Exemple :
VAR1="abcdefg";
NEW_VAR1=SUBSTR(VAR1,1,4);*La reponse serait "abcd";

Fonction FIND
La fonction FIND trouve la position d'une chaîne de caratère à l'intérieur d'une autre châine de caratère.

Syntaxe :

FIND(VAR1,"texte");

Exemple :
VAR1="abcdefg";
NEW_VAR1=FIND(VAR1,"d");*La reponse serait 4;

Fonctions pour dates

Syntaxe	Description
MDY(MONTH, DAY, YEAR)	Crée une date SAS à partir d'une valeure numérique de mois, jour, et année
MONTH(VAR1)	Renvoie la valeure numérique du mois d'une variable en format date
DAY(VAR1)	Renvoie la valeure numérique de la journée d'une variable en format date
YEAR(VAR1)	Renvoie la valeure numérique de l'année d'une variable en format date

Fonction INTCK
La fonction INTCK calcule l'interval de temps entre deux dates SAS. L'interval peut être calculer en nombre d'année, mois ou jours.

Syntaxe :

INTCK("interval",DATE1,DATE2);*Calcul l interval entre DATE1 et DATE2;

Exemple :
VAR1=MDY(01,01,2011);
VAR2=MDY(01,01,2012);
NEW_VAR=INTCK("month",VAR1,VAR2);*La reponse serait 12 pour 12 mois;