

SAS®

Initiation et manipulation de données Création de variables

Nicolas Poulin



Création de variable simple

- Il suffit de rajouter une instruction à l'étape **DATA**.
- Syntaxe générale : *variable = expression*;
 - ▶ *variable* : nouveau nom de variable ou variable pré-existante qui sera écrasée.
 - ▶ *expression* : constante ou formule (pouvant utiliser une fonction).

Constante de date et date du jour.

Remarque

- Une constante de date sera donnée sous la forme d'une date entre apostrophes suivi d'un d.

Newdate = '27SEP2016' d

- Pour obtenir la date du jour, la fonction **TODAY()** peut être utilisée.

Opérateurs arithmétiques

Pour les variables quantitatives, des opérateurs arithmétiques peuvent être utilisés lors de la création d'une variable.

Symbole	Définition
+	addition
-	soustraction
/	division
*	multiplication
**	puissance

Remarque

- *Des parenthèses peuvent être utilisées pour marquer les priorités de calcul.*
- *Une valeur manquante sera produite si une valeur manquante est utilisée dans un calcul.*



Exemple : IMC du maïs

```
DATA work.mais2;
```

```
SET work.mais;
```

```
Massekg=Masse/1000;
```

```
Hauteurm=Hauteur/100;
```

```
IMC=Massekg/Hauteurm**2;
```

```
RUN;
```

Exemple : IMC du maïs

```
DATA work.mais2;
```

```
SET work.mais;
```

```
IMC=(Masse/10)/Hauteur**2;
```

```
RUN;
```

Utilisation de fonctions

- Une fonction SAS est une routine qui retourne une valeur spécifiée à partir d'arguments spécifiques.
- Selon la fonction, les arguments peuvent être des variables de type caractère et numériques (même dates) mais aussi des paramètres spécifiques.
- Certaines fonctions peuvent être employées sans aucun argument. Ex : **TODAY()**
- Les arguments sont séparés par des virgules.

Nom_de_fonction(arg1, arg2, ...)

Exemple d'utilisation de fonctions numériques

```
DATA work.mais2;
```

```
SET work.mais;
```

```
dummy=SUM(Hauteur,Hauteur_J7);
```

```
Hauteur_mean=MEAN(Hauteur,Hauteur_J7);
```

```
RUN;
```


Exemple d'utilisation de fonctions texte

```
DATA work.mais2;
```

```
SET work.mais;
```

```
Parcelle=UPCASE(Parcelle);
```

```
RUN;
```

Remarque

Cette fonction peut s'avérer très utile lorsque les notations du fichier de données ne sont pas uniformes (par ex Nord, nord et NORD utilisés pour définir un même niveau de la variable Parcelle).

Fonctions de dates

Des fonctions peuvent être utilisées pour extraire des parties de dates SAS.

Fonction	Résultat
YEAR (date-SAS)	4 chiffres de l'année
QTR (date-SAS)	nombre de 1 à 4 (numéro du trimestre)
MONTH (date-SAS)	nombre de 1 à 12 (numéro du mois)
DAY (date-SAS)	nombre de 1 à 31 (numéro du jour dans le mois)
WEEKDAY (date-SAS)	nombre de 1 à 7 (numéro du jour : 1 = dimanche)

Exemple d'utilisation de fonctions texte

```
DATA work.sales2;
```

```
SET work.sales;
```

```
Birth_Year=YEAR(Birth_Date);
```

```
Birth_QTR=QTR(Birth_Date);
```

```
Birth_month=MONTH(Birth_Date);
```

```
Birth_day=DAY(Birth_Date);
```

```
Birth_wday=WEEKDAY(Birth_Date);
```

```
RUN;
```

Création de variable sous condition

- Il s'agit ici de créer une variable qui prendra des valeurs différentes selon des conditions spécifiques.
- Au lieu de mettre simplement le nom de la variable à créer ou modifier, on va utiliser :

IF condition THEN expression;

Remarque

Si la condition de la clause IF est vérifiée pour une observation, l'expression sera utilisée pour cette observation.

Exemple IF-THEN

```
DATA work.sales2;
```

```
SET work.sales;
```

```
IF Country='US' THEN Bonus=500;
```

```
RUN;
```

Remarque

*Les opérateurs spéciaux **WHERE**, **BETWEEN-AND**, **IS NULL**, **IS MISSING**, **CONTAINS** et **LIKE** ne peuvent pas être utilisés dans une condition de **IF**.*

IF-THEN/ELSE

- On peut utiliser un ELSE pour donner une seconde expression à utiliser lorsque la condition de la clause IF n'est pas vérifiée.
- L'instruction devient alors :

IF condition THEN expression1;

ELSE expression2;

Exemple IF-THEN/ELSE

```
DATA work.sales2;
```

```
    SET work.sales;
```

```
    IF Country='US' THEN Bonus=500;
```

```
    ELSE Bonus=300;
```

```
RUN;
```

Exemple IF-THEN/ELSE

On peut retrouver un IF-THEN/ELSE après un ELSE :

```
DATA work.mais2;
```

```
    SET work.mais;
```

```
    IF Hauteur>300 THEN Taille='Grande';
```

```
    ELSE IF Hauteur<200 THEN Taille='Petite';
```

```
    ELSE Taille='Normale';
```

```
RUN;
```


IF-THEN/ELSE

- Il est possible d'utiliser plusieurs instructions **ELSE IF**.
- SAS va vérifier, dans l'ordre de déclaration, les conditions des **ELSE IF**. Les vérifications s'arrêteront à la première condition vérifiée.
- Il vaut donc mieux ordonner les conditions par probabilité décroissante.
- Pour la création d'une variable texte, SAS prend la première valeur déclarée pour définir la taille nécessaire à cette variable.
- Il faut utiliser une instruction **LENGTH** si nécessaire.

Exemple IF-THEN/ELSE

DATA work.mais;

SET work.mais;

LENGTH Taille \$7.;

IF Hauteur>300 THEN Taille='Grande';

ELSE IF Hauteur<200 THEN Taille='Petite';

ELSE Taille='Normale';

RUN;

Exemple IF-THEN/ELSE

```
DATA work.mais2;
```

```
SET work.mais;
```

```
IF Hauteur>300 THEN Taille='Grande';
```

```
ELSE IF Hauteur<200 THEN Taille='Petite';
```

```
ELSE IF Hauteur=. THEN Taille='Manquante';
```

```
ELSE Taille='Normale';
```

```
RUN;
```

Exemple IF-THEN/ELSE

```
DATA work.mais2;
```

```
SET work.mais;
```

```
IF Hauteur=. THEN Taille='Manquante';
```

```
ELSE IF Hauteur<200 THEN Taille='Petite';
```

```
ELSE IF Hauteur>300 THEN Taille='Grande';
```

```
ELSE Taille='Normale';
```

```
RUN;
```

IF-THEN/ELSE

- Une seule expression exécutable est autorisée après un **THEN**.
- Pour exécuter plusieurs expressions il faut utiliser un **DO**.
- Chaque groupe **DO** se termine par un **END**.

IF-THEN/ELSE

```
IF condition1 THEN DO;  
  
    expression1;  
  
    expression2;  
  
END;  
ELSE IF condition2 THEN DO;  
  
    expression3;  
  
    expression4;  
  
END;
```

Exemple DO/END

```
DATA work.mais2;  
  SET work.mais;  
  IF Hauteur=. THEN Taille='Manquante';  
  ELSE IF Hauteur<200 THEN DO  
    Taille='Petite';  
    Hauteur=200;  
  END;  
  ELSE IF Hauteur>300 THEN DO  
    Taille='Grande';  
    Hauteur=300;  
  END;  
  ELSE Taille='Normale';  
RUN;
```

Expression DELETE

L'expression DELETE peut être utilisée après un THEN pour supprimer les observations vérifiant la condition du IF correspondant.

```
DATA work.mais2;  
  SET work.mais;  
  IF Hauteur=. THEN DELETE;  
  ELSE IF Hauteur<200 THEN Taille='Petite';  
  ELSE IF Hauteur>300 THEN Taille='Grande';  
  ELSE Taille='Normale';  
RUN;
```