SAS®
Initiation et manipulation de données
Etiquettes, sélection de variables et d'observations, formats

Nicolas Poulin

IRMA
Institut de Recherche Mathématique Avancée

UFR Mathématique Informatique Strasbourg

Université de Strasbourg

poulin@math.unistra.fr

Etiquettes

Chaque variable d'une table SAS a

- un nom (utilisé dans les scripts)
- une étiquette (utilisée dans les rapports de sortie)

Généralement les noms des variables sont extraits de la première ligne du jeu de données original lors de l'utilisation de la PROC IMPORT grâce à GETNAMES=YES.

Les étiquettes sont créées en acceptant plus de souplesse que pour le nom des variables.

poulin@math.unistra.fr

PROC IMPORT : exemple

```
PROC IMPORT OUT=UEdata.mais
                DATAFILE="folders/myfolders/UEdata/mais.xlsx"
                DBMS= xlsx;
      GETNAMES=YES;
RUN;
```


poulin@math.unistra.fr

PROC CONTENTS

On peut voir les étiquettes créées grâce à la PROC CONTENTS

```
PROC CONTENTS DATA=libref.filename;

RUN;
```


poulin@math.unistra.fr

Créer des étiquettes

On peut modifier les étiquettes créées grâce à une étape DATA

```
DATA libref.filename2;
        SET libref.filename1;
        LABEL variable1= 'étiquette 1';
        LABEL variable2= 'étiquette 2';
        LABEL variable3= 'étiquette 3';
RUN;
```

Remarque

- *libref.filename1 est le nom de la table d'entrée.*
- *libref.filename2 est la nom de la table sortie qui sera créée.*

poulin@math.unistra.fr

Créer des étiquettes : exemple

```
DATA UEdata.mais1;
        SET UEdata.mais;
        LABEL Hauteur= 'Hauteur en cm';
        LABEL Masse= 'Masse en g';
        LABEL Parcelle= 'Orientation de la parcelle';
RUN;
```

Remarque

On peut garder le même nom pour la table avec les étiquettes modifiées en déclarant le même nom pour la table en sortie que pour la table en entrée.

poulin@math.unistra.fr

PROC PRINT : étiquettes

On peut remplacer les noms de variables par les étiquettes dans l'impression par la PROC PRINT

PROC PRINT DATA=libref.filename LABEL;

RUN;

poulin@math.unistra.fr

Garder des colonnes spécifiées

On peut écarter des variables d'un jeu de données grâce à une étape DATA dans laquelle on va stipuler les variables à garder.

```
DATA libref.filename2;

        SET libref.filename1;

        KEEP liste de variables;

RUN;
```


poulin@math.unistra.fr

KEEP : exemple

```
DATA UEdata.mais2;

        SET UEdata.mais;

        KEEP Hauteur Masse Couleur Parcelle;
RUN;
```

Remarque

Les variables seront ordonnées dans l'ordre de la table d'origine.

poulin@math.unistra.fr

Ordonner les variables

```
DATA libref.filename2;

        RETAIN liste des variables dans l'ordre souhaité;

        SET libref.filename1;

RUN;
```

Remarque

Pour être pris en considération, l'instruction RETAIN doit être placée avant l'instruction SET.

poulin@math.unistra.fr

RETAIN : exemple

```
DATA UEdata.mais2;

        RETAIN Hauteur Masse Couleur Parcelle;

        SET UEdata.mais;

RUN;
```

Remarque

Les variables déclarées dans l'instruction RETAIN sont les premières. Les autres sont ensuite ordonnées dans l'ordre de la table d'origine.

poulin@math.unistra.fr

KEEP et RETAIN : exemple

```
DATA UEdata.mais2;

    RETAIN Hauteur Masse Couleur Parcelle;

    SET UEdata.mais1;

    KEEP Hauteur Masse Couleur Parcelle;

RUN;
```


poulin@math.unistra.fr

Retirer des colonnes spécifiées

On peut écarter des variables d'un jeu de données grâce à une étape DATA dans laquelle on va stipuler les variables à écarter.

```
DATA libref.filename2;

        SET libref.filename1;

        DROP liste de variables;

RUN;
```


poulin@math.unistra.fr

KEEP : exemple

```
DATA UEdata.mais2;
        SET UEdata.mais;
        DROP Hauteur_J7 Masse_grains;
RUN;
```


poulin@math.unistra.fr

Instruction WHERE

L'instruction WHERE crée un sous ensemble de données qui remplissent une condition spécifiée dans le code.

```
DATA libref.filename2;

    SET libref.filename1;

    WHERE instruction;

RUN;
```

Remarque

L'instruction WHERE peut porter sur une variable qualitative (texte) ou quantitative (numérique).

poulin@math.unistra.fr

WHERE : exemple

```
DATA UEdata.mais2;

        SET UEdata.mais;

        WHERE Parcelle="Nord";

RUN;
```


poulin@math.unistra.fr

WHERE : exemple

```
DATA UEdata.mais2;
        SET UEdata.mais;
        WHERE Hauteur>300;
RUN;
```


poulin@math.unistra.fr

Opérateurs de comparaisons

Chaque opérateur de comparaison peut soit être déclaré sous forme de symbole ou sous forme mnémonique.

Symbole	Mnémonique	Définition
=	EQ	égal à
¬ = ~= ∧ =	NE	pas égal à
>	GT	plus grand que
<	LT	plus petit que
>=	GE	plus grand ou égal à
<=	LE	plus petit ou égal à
·	IN	égal à une valeur de la liste

poulin@math.unistra.fr

WHERE : exemple

```
DATA UEdata.mais2;

        SET UEdata.mais;

        WHERE Parcelle IN ("Nord","Sud");

RUN;
```


poulin@math.unistra.fr

WHERE : données manquantes

L'instruction WHERE peut être utilisée pour obtenir les lignes avec des données manquantes pour une variable spécifiée

```
DATA UEdata.mais2;

        SET UEdata.mais;

        WHERE Parcelle =" ";

RUN;
```


poulin@math.unistra.fr

WHERE : données manquantes

```
DATA UEdata.mais2;

        SET UEdata.mais;

        WHERE Hauteur= .;

RUN;
```


poulin@math.unistra.fr

Opérateurs arithmétiques

Pour les variables quantitatives, des opérateurs arithmétiques peuvent être utilisés dans la condition du WHERE.

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

Remarque

Des parenthèses peuvent être utilisées pour marquer les priorités de calcul.

poulin@math.unistra.fr

Opérateurs arithmétiques : exemple

```
DATA UEdata.mais2;
        SET UEdata.mais;
        WHERE (Hauteur/10 + Masse/100)/10 > 5;
RUN;
```


poulin@math.unistra.fr

Opérateurs logiques

Les opérateurs logiques peuvent être utilisés pour définir la condition spécifiée dans le WHERE.

Symbole	Mnémonique	Définition
&	AND	logique et
|	OR	logique ou
¬ ~ ∧	NOT	pas logique

Remarque

Si plusieurs instructions WHERE sont déclarées séparément, seule la dernière sera prise en compte.

poulin@math.unistra.fr

Opérateurs logiques : exemple

```
DATA UEdata.mais2;

        SET UEdata.mais;

        WHERE Hauteur > 300 & Masse > 2000;

RUN;
```


poulin@math.unistra.fr

Opérateurs logiques : exemple

```
DATA UEdata.mais2;

        SET UEdata.mais;

        WHERE Hauteur > 300 OR Masse > 2000;

RUN;
```


poulin@math.unistra.fr

Opérateurs logiques : exemple

```
DATA UEdata.mais2;

        SET UEdata.mais;

        WHERE Parcelle NOT IN ("Nord","Sud");

RUN;
```


poulin@math.unistra.fr

Opérateurs spéciaux

Symbole	Mnémonique	Définition
	BETWEEN-AND	intervalle fermé
	IS NULL	valeur manquante
	IS MISSING	valeur manquante
?	CONTAINS	une partie de chaîne de caractère
	LIKE	une forme de caractère

Remarque

Ces opérateurs spéciaux ne peuvent être utilisés que dans une instruction WHERE.
IS MISSING et IS NULL ont le même effet.

poulin@math.unistra.fr

Opérateurs spéciaux : exemple

```
DATA UEdata.mais2;
        SET UEdata.mais;
        WHERE Hauteur BETWEEN 250 AND 300;
RUN;
```


poulin@math.unistra.fr

Opérateurs spéciaux : exemple

```
DATA UEdata.mais2;
        SET UEdata.mais;
        WHERE Hauteur BETWEEN 250 AND 300;
RUN;
```

```
DATA UEdata.mais2;
        SET UEdata.mais;
        WHERE 250 <= Hauteur <= 300;
RUN;
```


poulin@math.unistra.fr

Opérateurs spéciaux : exemple

```
DATA UEdata.mais2;

    SET UEdata.mais;

    WHERE Hauteur NOT BETWEEN 250 AND 300;

RUN;
```

Remarque

Le code ci-dessous sélectionnera aussi les individus pour lesquels la valeur de Hauteur est manquante.

poulin@math.unistra.fr

Opérateurs spéciaux : exemple

```
DATA UEdata.mais2;
        SET UEdata.mais;
        WHERE Hauteur IS NULL;
RUN;
```


poulin@math.unistra.fr

Opérateurs spéciaux : exemple

```
DATA UEdata.mais2;
        SET UEdata.mais;
        WHERE Hauteur IS NULL;
RUN;
```

```
DATA UEdata.mais2;
        SET UEdata.mais;
        WHERE Hauteur IS MISSING;
RUN;
```


poulin@math.unistra.fr

Opérateurs spéciaux : exemple

```
DATA UEdata.mais2;
        SET UEdata.mais;
        WHERE Couleur IS NULL;
RUN;
```


poulin@math.unistra.fr

Opérateurs spéciaux : exemple

```
DATA UEdata.mais2;
        SET UEdata.mais;
        WHERE Couleur IS NULL;
RUN;
```

```
DATA UEdata.mais2;
        SET UEdata.mais;
        WHERE Couleur IS MISSING;
RUN;
```


poulin@math.unistra.fr

Opérateurs spéciaux : exemple

```
DATA UEdata.mais2;
        SET UEdata.mais;
        WHERE Parcelle CONTAINS "st";
RUN;
```


poulin@math.unistra.fr

Opérateurs spéciaux : exemple

```
DATA UEdata.mais2;
        SET UEdata.mais;
        WHERE Parcelle CONTAINS "st";
RUN;
```

```
DATA UEdata.mais2;
        SET UEdata.mais;
        WHERE Parcelle ? "st";
RUN;
```

Remarque

Cet opérateur est sensible à la casse : "st" et "ST" ne sont pas la même chose.

poulin@math.unistra.fr

Opérateurs spéciaux : LIKE

L'opérateur LIKE sélectionne des observations en comparant des valeurs de type caractère aux modèles spécifiés.

- Un symbole % remplace n'importe quel nombre de caractères
- Un symbole _ remplace un caractère
- Ces 2 symboles peuvent être utilisés plusieurs fois dans la même commande

Remarque

L'opérateur LIKE est sensible à la casse.

poulin@math.unistra.fr

Opérateurs spéciaux : exemple

```
DATA UEdata.mais2;

        SET UEdata.mais;

        WHERE Parcelle LIKE "%t";

RUN;
```

Remarque

Un warning est reporté dans le journal. Cela est dû au fait que le symbole % est aussi utilisé pour les noms de macros.

poulin@math.unistra.fr

Opérateurs spéciaux : exemple

```
DATA UEdata.mais2;
        SET UEdata.mais;
        WHERE Parcelle LIKE "O%";
RUN;
```


poulin@math.unistra.fr

Opérateurs spéciaux : exemple

```
DATA UEdata.mais2;
        SET UEdata.mais;
        WHERE Parcelle LIKE "O%";
RUN;
```

```
DATA UEdata.mais2;
        SET UEdata.mais;
        WHERE Parcelle LIKE "O_e%";
RUN;
```


poulin@math.unistra.fr

Opérateurs spéciaux : LIKE

Lorsqu'on veut rechercher une chaîne de caractère comportant les symboles % ou _

- il faut spécifier à SAS qu'il s'agit du caractère et non d'un marqueur de nombres de caractères. 'opérateur LIKE sélectionne des observations en comparant des valeurs de type caractère aux modèles spécifiés
- Exemple : valeurs finissant par le symbole %
  WHERE variable LIKE "%/%" ESCAPE "/";

poulin@math.unistra.fr

Instruction FORMAT

L'instruction FORMAT permet de changer le format dans lequel est stockée une variable, notamment pour l'affichage.

```
DATA libref.filename2;
        SET libref.filename1;
        FORMAT liste de variables suivie du format;
RUN;
```

Remarque

L'utilisation de cette instruction assigne de façon permanente de nouveaux formats aux variables. Ceci seront stockés dans le bloc descripteur de la table SAS.

poulin@math.unistra.fr

formats SAS

Les formats SAS se présentent sous la forme :

$$< \$ > \textit{format} < w > . < d >$$

$	présent uniquement pour les variables de type caractère
format	format SAS
w	longueur totale du format (incluant les décimales et éventuels caractères spéciaux)
.	délimiteur obligatoire
d	nombre de décimales pour les formats numériques

Remarque

La notation $< \cdot >$ signifie qu'il s'agit d'un argument facultatif.

poulin@math.unistra.fr

formats SAS

Format	Valeur stockée	Valeur affichée
$4.	Ouest	Oues
12.	12345.6789	12345
12.2	12345.6789	12345.68
COMMA12.2	12345.6789	12,345.68
COMMAX12.2	12345.6789	12.345,68
DOLLAR12.2	12345.6789	$ 12,345.68
EURO12.2	12345.6789	€ 12,345.68

poulin@math.unistra.fr

PROC IMPORT

```
PROC IMPORT OUT=UEdata.maisok
                DATAFILE="folders/myfolders/UEdata/maisok.xlsx"
                DBMS= xlsx;
        GETNAMES=YES;
RUN;
```


poulin@math.unistra.fr

Formats : exemple

```
DATA UEdata.maisok2;

    SET UEdata.maisok;

    FORMAT Parcelle $3. masse_grains 12.2

    Masse comma12.2 hauteur commax12.2

    hauteur_j7 dollar12.2 nb_grains eurox12.2;

RUN;
```


poulin@math.unistra.fr

formats SAS

Format	Valeur stockée	Valeur affichée
DOLLAR12.2	12345.6789	$ 12,345.68
DOLLAR9.2	12345.6789	$ 12345.68
DOLLAR8.2	12345.6789	12345.68
DOLLAR5.2	12345.6789	12345
DOLLAR4.2	12345.6789	12E3
DOLLARX12.2	12345.6789	$ 12.345,68

Remarque

La même chose existe avec

- *EURO*
- *EUROX* .

poulin@math.unistra.fr

formats de date SAS

Format	Valeur stockée	Valeur affichée
MMDDYY6.	365	123160
DDMMYY6.	365	311260
MMDDYY8.	365	12/31/60
DDMMYY8.	365	31/12/60
MMDDYY10.	365	12/31/1960
DDMMYY10.	365	31/12/1960

poulin@math.unistra.fr

Formats : exemple

```
DATA UEdata.maisok2;

    SET UEdata.maisok;

    FORMAT nb_grains MMDDYY6.

    Masse DDMMYY6. hauteur MMDDYY8.

    hauteur_j7 DDMMYY10. nb_grains MMDDYY10.;

RUN;
```


poulin@math.unistra.fr

formats de date SAS

Format	Valeur stockée	Valeur affichée
DATE7.	365	31DEC60
DATE9.	365	31DEC1960
WORDDATE.	365	December 31, 1960
WEEKDATE.	365	Saturday, December 31, 1960
MONYY7.	365	DEC1960
YEAR4.	365	1960

poulin@math.unistra.fr

Formats : exemple

```
DATA UEdata.maisok2;

    SET UEdata.maisok;

    FORMAT Individu DATE9.

    Masse WORDDATE. hauteur WEEKDATE.

    hauteur_j7 MONYY7. nb_grains YEAR4.;

RUN;
```


poulin@math.unistra.fr

formats de date NLS SAS

- Les formats de date NLS (National Language Support) affichent les dates selon les usages locaux.
- Il faudra éventuellement, avant d'utiliser les formats NLS, stipuler à SAS le langage souhaité en utilisant l'option LOCALE.

formats de date NLS SAS

Format	Locale	Exemple
NLDATEw.	English_UnitedStates French_France German_Germany	January 01, 1960 01 Janvier 1960 01. Januar 1960
NLDATEMN10.	English_UnitedStates French_France German_Germany	January Janvier Januar
NLDATEW10.	English_UnitedStates French_France German_Germany	Fri, Jan 01, 60 01 Janvier 1960 Fr, 01. Jan 60
NLDATEWN10.	English_UnitedStates French_France German_Germany	Friday Vendredi Freitag

poulin@math.unistra.fr

Formats : exemple

```
DATA UEdata.maisok2;

    SET UEdata.maisok;

    FORMAT Masse NLDATEw. hauteur NLDATEMN10.

    hauteur_j7 NLDATEW10. nb_grains NLDATEWN10. ;

RUN;
```

Remarque

Nous pouvons voir que la session est en NLS French_France.

poulin@math.unistra.fr

Formats : exemple

```
options locale=German_Germany;
DATA UEdata.maisok2;
        SET UEdata.maisok;
        FORMAT Masse NLDATEw. hauteur NLDATEMN10.
        hauteur_j7 NLDATEW10. nb_grains NLDATEWN10. ;
RUN;
```


poulin@math.unistra.fr

Formats : exemple

```
options locale=English_UnitedStates;
DATA UEdata.maisok2;
        SET UEdata.maisok;
        FORMAT Masse NLDATEw. hauteur NLDATEMN10.
        hauteur_j7 NLDATEW10. nb_grains NLDATEWN10. ;
RUN;
```


poulin@math.unistra.fr

Formats : exemple

```
options locale=German_Germany;
DATA UEdata.maisok2;
        SET UEdata.maisok;
        FORMAT Masse NLDATEw. hauteur NLDATEMN10.
        hauteur_j7 NLDATEW10. nb_grains NLDATEWN10. ;
RUN;
```


poulin@math.unistra.fr

Astuce

Lorsque l'on veut, par exemple, définir le même format pour des variables commençant par la même lettre (par exemple H), au lieu de mettre la liste des variables, on peut utiliser la notation "H:"

```
DATA UEdata.maisok2;

    SET UEdata.maisok;

    FORMAT H: 12.2;

RUN;
```

Remarque

Cela marche aussi avec différentes instructions comme KEEP par exemple.

poulin@math.unistra.fr