

Les dates SAS expliquées à ma fille

Eternel sujet de questionnements, les dates SAS ont un fonctionnement qui a tout pour surprendre. Petite série d'explications avant qu'il vous prenne l'envie de passer toutes vos dates au goudron et aux plumes.

Comment sont stockées les dates SAS ?

Il existe deux types de variables qui contiennent des dates dans SAS : toutes deux sont de type numérique. On distingue les simples dates et les « datetime ».

- Les simples dates sont stockées comme des nombres de jours depuis le 1^{er} janvier 1960. Les dates antérieures à 1960 sont stockées sous forme d'entiers négatifs. On peut stocker dans SAS des dates comprises entre le 1^{er} janvier 1582 (- 138 061) et le 31 décembre 20 000 (6 589 335). Il est cependant déconseillé de travailler sur des dates antérieures à octobre 1582, car le passage du calendrier julien au calendrier grégorien¹ n'est pas très bien géré par SAS et peut produire des résultats erronés.
- Les datetime intègrent, comme leur nom l'indique, une dimension temporelle plus fine. Ils sont stockés en nombre de secondes depuis le 1^{er} janvier 1960 à minuit. Comme pour les simples dates, tout ce qui est antérieur à 1960 est stocké comme un nombre négatif. Les datetimes sont compris entre le 1^{er} janvier 1582 minuit (- 11 928 470 400) et le 31 décembre 20 000 à 23h59 et 59 secondes (569 318 630 399).

On trouve également des variables numériques « time » qui ne contiennent que les informations en secondes.

Stockage d'une date et d'un datetime

Pour le stockage d'une simple date, on peut utiliser une longueur (LENGTH) de seulement 4 octets (contre 8 par défaut). Pour cela, il faut qu'avant la création de la variable on insère l'instruction

```
LENGTH nomVariableDate 4 ;
```

En revanche, pour un datetime, il convient de conserver la longueur par défaut de 8 octets.

Conversion entre date et datetime

Les variables datetime sont particulièrement courantes dans les tables créées à partir d'un logiciel hébergeant une base de données relationnelle (SGBDR) comme Oracle, DB2, Access, Sybase, ... car c'est ainsi que le module SAS/ACCESS récupère les données dates. En revanche, les datetime sont pénibles à manipuler dans SAS, en particulier parce qu'il y a peu de formats et surtout de fonctions qui s'y appliquent, comparativement aux simples dates. Il est donc intéressant de savoir convertir ou éclater un datetime.

La fonction DATEPART permet d'extraire la date (en nombre de jours) d'un datetime. De manière similaire, la fonction TIMEPART récupère la partie temps, en nombre de secondes depuis minuit.

¹ L'adoption du calendrier grégorien ne s'est pas faite à date harmonisée de par le monde (voir par exemple http://fr.wikipedia.org/wiki/Passage_au_calendrier_gr%C3%A9gorien) ; en France, elle s'est traduite par le passage immédiate du jeudi 4 octobre 1582 au vendredi 15 octobre, sans aucun jour intermédiaire.

```
variableSimpleDate = DATEPART(variableDatetime) ;
variableTemps      = TIMEPART(variableDatetime) ;
```

A l'inverse, la conversion d'une simple date en datetime, ou la fusion d'une simple date et d'une variable temps en une seule variable datetime nécessite l'emploi de la fonction DHMS.

```
variableDatetime = DHMS(variableSimpleDate,0,0,0) ;
variableDatetime = DHMS(variableSimpleDate,
                        HOUR (variableTemps),
                        MINUTE(variableTemps),
                        SECOND(variableTemps)) ;
```

Quels formats utiliser sur des dates SAS ?

Les dates et datetime étant stockés en nombres de jours ou de secondes, leur affichage par défaut n'est pas très informatif. Il convient donc de leur associer un format afin de les rendre plus lisibles.

Les formats permettent également de choisir un niveau d'agrégat lors d'un traitement statistique (procédures Means, Freq, Tabulate, mais aussi instructions CLASS des procédures de modélisation comme GLM, Logistic, Genmod, ...) ; on peut ainsi, sans modifier ni créer de variable, agréger à l'année, au trimestre, au mois ou au jour de la semaine.

Dans les exemples suivants, la valeur 15 000 correspond, en simple date, au jeudi 25 janvier 2001. Considérée comme un datetime, 150 000 correspond au samedi 2 janvier 1960 à 17h40.

Tableau 1 - Principaux formats pour les simples dates

Nom du format	Exemple de valeur	Remarques
DDMMYY10.	25/01/2001	-
DDMMYYB10.	25 01 2001	-
DDMMYYD10.	25-01-2001	-
MMYY5.	01/2001	-
YEAR4.	2001	-
FRADFWKX.	Jeudi 25 janvier 2001	-
FRADFWDX.	25 janvier 2001	-
FRADFMN.	janvier	-
FRADFDWN.	Jeudi	-
DATE9.	25JAN2001	Le mois est sur trois lettres en anglais (JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC)
DAY2.	25	-
MONTH2.	1	-
QTR3.	I	Trimestre en chiffres romains
WEEKDAY1.	5	Numéro du jour de la semaine (1=dimanche, 2=lundi, ...)
WEEKV.	2001-W04-04	Année 2001, semaine 4, 4 ^e jour (les conventions sont celles des numéros de semaines françaises, et le 1 ^{er} jour de la semaine est ici le lundi)
YYQRD9.	2001QI	Année et trimestre en chiffres romains

Tableau 2 - Principaux formats pour les datetime

Nom du format	Exemple de valeur	Remarques
DATETIME.	02JAN60:17:40:00	
DATEAMP.	02JAN60:05:40:00 PM	
DTDATE9.	02JAN1960	
DTMONYY7.	JAN1960	
DTYEAR4.	1960	
NLDATMW.	samedi 02 janvier 1960 17 h 40	Avec l'option système LOCALE=FRENCH
TOD10.1	17:40:00.0	TOD=Time Of Day

Quelles fonctions utiliser sur des dates SAS ?

De même que pour les formats, il est souvent nécessaire de recourir à des fonctions permettant d'extraire des parties d'une date, ou de construire de nouvelles dates à partir d'éléments existants.

Les valeurs utilisées dans les exemples sont toujours varDate=15 000 pour les simples dates et varDateTime=150 000 pour les datetime.

Tableau 3 - Principales fonctions pour les simples dates

Fonction	Exemple d'utilisation	Résultat de l'exemple	Remarques
YEAR	YEAR(varDate)	2001	-
QTR	QTR(varDate)	1	1 ^{er} trimestre
MONTH	MONTH(varDate)	1	-
DAY	DAY(varDate)	25	-
WEEKDAY	WEEKDAY(varDate)	5	5 ^e jour de la semaine (début de la semaine = dimanche)
WEEK	WEEK(varDate, "V")	4	4 ^e semaine de l'année
TODAY	TODAY()	17203	Date du jour (date système)
MDY	MDY(1, 25, 2001)	15000	-
INTCK	INTCK("YEAR", 15000, TODAY())	6	Nombre de 1 ^{er} jour de la période ¹ existant entre 2 dates
INTNX			-

Notons également les quelques « recettes » usuelles pour les calculs sur les dates...

- Nombre de jours entre deux dates : simplement soustraire l'une à l'autre.
- Age en années révolues : `YEAR(TODAY())-YEAR(dateNaissance)`
- Age au jour d'aujourd'hui : `FLOOR((TODAY()-dateNaissance)/365.25)`
- Anniversaire des 25 ans :

```
MDY(MONTH(dateNaissance),
    DAY(dateNaissance),
    YEAR(dateNaissance)+25)
```

Sauf si la personne est née un 29 février. Sinon, une formule universelle :

```
INTNX("MONTH", dateNaissance, 25*12)+DAY(dateNaissance)-1
```

¹ Par exemple : YEAR, YEAR2 (période de deux ans), SEMIYEAR (semestre), QUARTER, MONTH, SEMIMONTH (quinzaine commençant le 1^{er} ou le 15 du mois), WEEK (semaine commençant le dimanche), WEEK . 2 (semaine commençant le lundi)

Tableau 4 - Principales fonctions pour les datetime

Fonction	Exemple d'utilisation	Résultat de l'exemple	Remarques
DATETIME	DATETIME()	1486395963.304	Date et heure système
HOURL	HOURL(varDateTime)	17	-
MINUTE	MINUTE(varDateTime)	40	-
SECOND	SECOND(varDateTime)	0	-
INTNX	Cf. Tableau 3		
INTCK			

Comment évoquer une date SAS dans un programme

Proposer une date ou un temps dans un programme est quelque peu contraignant : on peut utiliser différentes écritures, pour comparer une variable date avec le 25 janvier 2001 par exemple...

```
WHERE varDate > 15000 ;
WHERE varDate > "25jan2001"d ;
WHERE varDate > MDY(1,25,2001) ;
```

Le premier exemple est difficilement utilisable, à moins de connaître le nombre de jours depuis 1960 correspondant à la date à tester ! Pour le second, la norme est la suivante :

- le nom du mois est sur 3 lettres, en anglais (JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC) ;
- la chaîne jour mois année est entre guillemets ;
- un D est collé au guillemet fermant pour demander à SAS la conversion en nombre de jours depuis 1960 ;
- majuscules et minuscules (tant sur le D que sur le nom du mois) sont équivalents.

Pour un datetime, on peut procéder à partir des mêmes principes...

```
WHERE varDate > 150000 ;
WHERE varDate > "01jan1960:17:40:00"dt ;
WHERE varDate > DHMS(MDY(1,1,1960),HMS(17,40,0)) ;
WHERE varDate > "01jan1960"d * 24*60*60 + "17:40:00"t ;
```

Comment importer une date SAS

L'import de données dates dans SAS à partir d'un autre logiciel ne pose généralement pas de problèmes, à condition d'être attentif aux points suivants :

- l'import de feuilles Excel dans Enterprise Guide génère souvent des datetime là où Excel stocke des dates ;
- l'import de tables Oracle, DB2, Sybase via les modules SAS/ACCESS correspondants (avec un LIBNAME ou du SQL *pass-through*) génère systématiquement des datetime quand la table d'origine contient des dates ;
- l'import d'une date dans un fichier plat (texte, tabulé, colonné, csv, etc.) doit utiliser un informat pour lire les données dates. Les principaux informats pour les dates sont DDMMYY, YYMMDD, ANYDTDTE (n'importe quel type de date), DATETIME, ANYDTDTM (n'importe quel type de datetime).

La partie la plus piègeuse de l'import de dates concerne le traitement par SAS des années sur 2 chiffres. Il est piloté par l'option système YEARCUTOFF qui indique quelle est l'année pivot (voir exemple ci-dessous pour la définition de l'année pivot).

```
OPTION YEARCUTOFF = 1950 ;
```

Les années 50 à 99 sont traitées comme appartenant au même siècle que 1950. Donc 50 sera compris comme 1950, 70 comme 1970, 99 comme 1999.

Les années 0 à 49 sont traitées comme appartenant au siècle suivant 1950. Donc 49 sera compris comme 2049, 0 comme 2000.

