

Initiation à SAS

par Fabrice MORLAIS ([Fabrice MORLAIS](#))

Date de publication : 24/04/2008

Dernière mise à jour : 24/04/2008

Ce tutoriel a pour objectif de vous initier au logiciel SAS à travers une courte présentation de ces diverses fonctionnalités. Vous trouverez aussi quelques exemples de programmation pour vous permettre de bien débuter tout en comprenant la logique du langage utilisé sur SAS.

Introduction.....	3
I - Un fonctionnement en produits (ou modules).....	4
II - Les interfaces de SAS.....	6
II-A - Les fenêtres d'écriture des programmes	6
II-B - Le "journal" ou "fenêtre des erreurs" ou "LOG".....	6
II-C - La "fenêtre de résultat" ou "OUTPUT".....	8
II-D - L'explorateur.....	8
IV - Le fonctionnement de SAS.....	11
IV-A - Etape DATA	11
IV-B - Procédure PROC.....	12
IV-B-1 - Un premier exemple.....	12
IV-B-2 - Plus de détails.....	14
IV-B-3 - La gestion des sorties.....	16
IV-C - Langage MACRO.....	16
Conclusion.....	17
Ressources utilisées pour la rédaction de l'article.....	18
La communauté SAS de developpez.com.....	18
Internet.....	18
Livre.....	18
Remerciements.....	19

Introduction

SAS est né suite à une idée de son créateur Anthony J. BARR lorsqu'il était étudiant à North Carolina (Etats-Unis). Cette « révélation », très inspirée des travaux du statisticien Maurice KENDALL, lui vint entre 1962 à 1964 lorsqu'il voulu créer un programme informatique réalisant une analyse de variance. A peine sa première idée développée, il rajouta à son outil un programme lui permettant de réaliser des régressions multiples.

A partir de 1969, le projet devenant certainement trop vaste pour un seul homme, BARR s'est associé avec d'autres membres. Ce partenariat a non seulement permis l'ajout de nombreuses autres analyses statistiques, mais aussi la gestion de reporting et la création d'un langage de programmation orienté utilisateur : le macro langage. En 1976, la société SAS Institute Incorporation est créée.

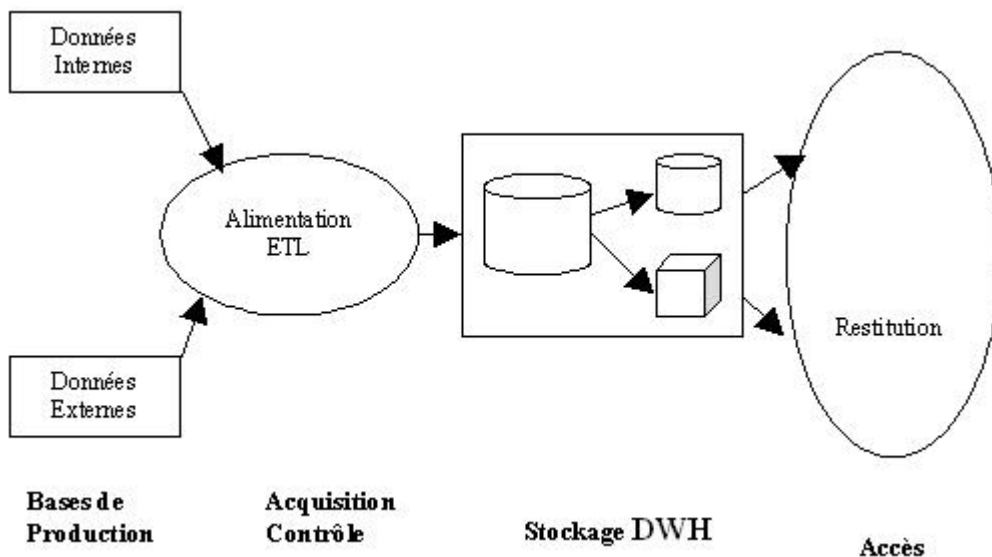
Aujourd'hui acteur majeur du décisionnelle, SAS a su élargir son champ de compétences au cours de son histoire pour s'adapter au marché, aux nouvelles technologies et techniques. Cette stratégie lui a permis de devenir l'un des leaders mondiaux du marché du **Business Intelligence**.

SAS couvre maintenant toute la chaîne du décisionnel.

Actuellement, SAS en est à la version 9.1.3, sa politique étant de garantir la rétrocompatibilité (les programmes et objets SAS fonctionnent d'une version à l'autre)

Rappel : ETL

Les outils **ETL** (Extract, Transform and Load) sont chargés d'extraire les données de différentes sources, de les nettoyer et de les charger dans un entrepôt de données. Le but étant de faciliter les multiples analyses multidimensionnelles. On peut schématiser le fonctionnement d'un Système d'Information de la façon suivante :



I - Un fonctionnement en produits (ou modules)

Avant toute chose il est intéressant de signaler que le logiciel SAS est prévu pour fonctionner sur de nombreux systèmes d'exploitation tels que Windows, Macintosh, Unix, Mainframe.

SAS est un logiciel fonctionnant en modules : les modules sont en quelques sortes des sous parties du logiciel SAS. Par exemple pour réaliser des opérations de type ETL il est indispensable de posséder le module lié aux ETL (non inclus dans la version basique).

Comme les modules sont généralement ciblés sur des profils de besoins d'utilisateur, il n'est pas rare d'observer une certaine redondance des fonctions. Vous pouvez par exemple tout aussi bien éditer un graphique de type histogramme en programmation SAS qu'à l'aide d'une fenêtre graphique de type click-bouton.

L'offre SAS dans l'ensemble a plus tendance à s'adapter aux besoins de l'utilisateur que l'inverse : le grand nombre de module expliquant certainement cette remarque. En fonction de ses besoins, l'utilisateur choisira les modules les plus adaptés : l'offre de SAS peut en quelque sorte être considérée comme « une offre à la carte ».

Ci-dessous une liste non exhaustive des modules de SAS avec une légère description de leurs propriétés.

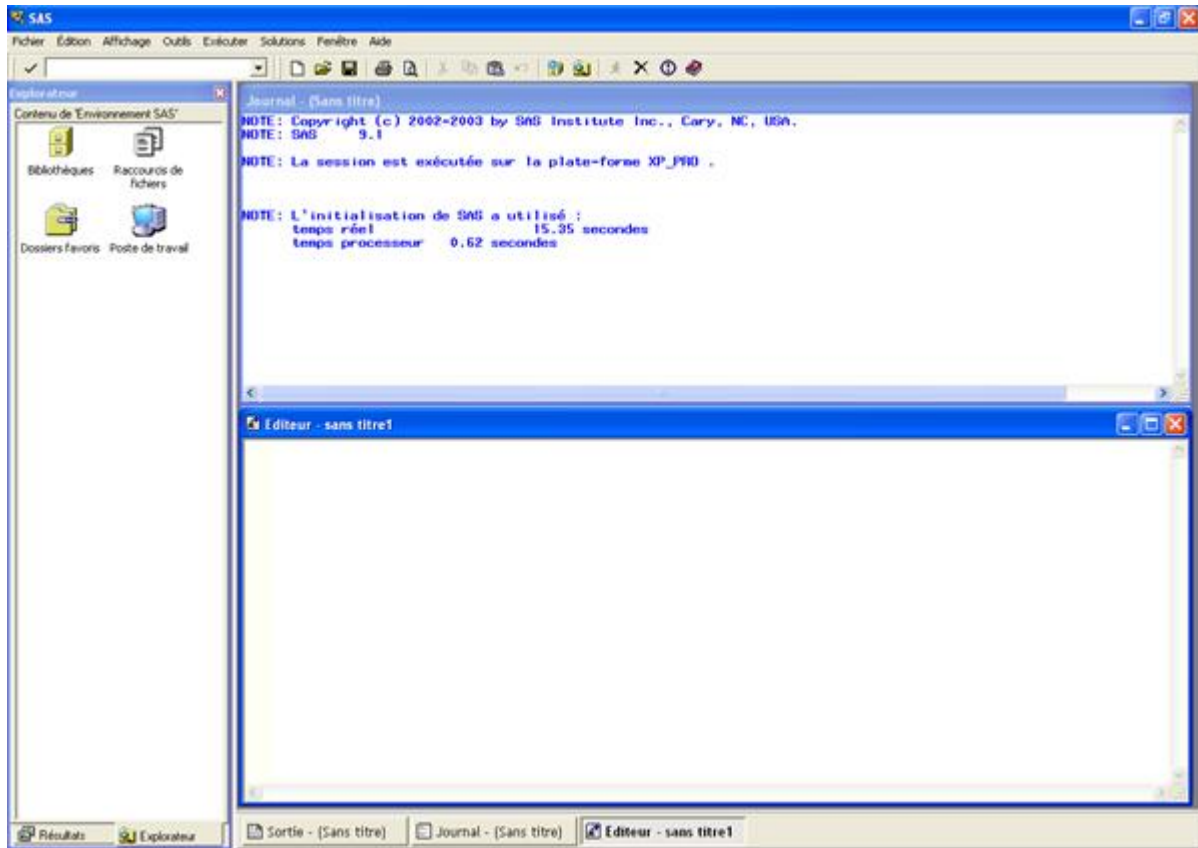
Module	Description
SAS/ACCESS	Module orienté base de données, il permet un accès à de nombreux systèmes de bases de données tels que DB2, INFORMIX, ODBC, ORCALE, SYSDATABASE, SQL SERVER.
SAS/AF	Développement d'interfaces utilisateur (Frame) à l'aide du langage SCL orienté objet.
SAS/ASSIST	Utilise les principales caractéristiques de SAS/BASE en mode fenêtres. A l'avantage d'offrir au débutant une vue du langage SAS

	généralisé lors des tâches exécutées en mode fenêtre.
SAS/BASE	Importation, manipulation et exportation des données
SAS DATA INTEGRATOR	ETL de SAS.
SAS/ENTREPRISE BI SERVER	Plateforme décisionnelle de SAS.
SAS/ENTREPRISE MINER	Système de fenêtre pour réaliser du DATA MINING.
SAS/ETS	Fonctionnalités statistiques avancées pour la prévision, modélisation et simulation.
SAS/GIS	Manipulation de données géographiques.
SAS/GRAPH	Gestion des sorties graphiques.
SAS/IML	Programmation en langage matricielle.
SAS/INSIGHT	Manipulation interactive pour explorer et analyser les données.
SAS/INTRNET	Interaction avec Internet pour faciliter la publication des rapports obtenus sur SAS.
SAS/MACO	Module complémentaire à SAS/BASE, Langage permettant d'automatiser certaines tâches.
SAS OLAP CUBE STUDIO	Création de cubes.
SAS/OR	Fonctionnalités de Recherche Opérationnelle.
SAS/QC	Outils statistiques pour le contrôle qualité.
SAS/STAT	Ensemble des fonctionnalités statistiques courantes et poussées.
SAS/TOOLKIT	Outil pour l'écriture d'extensions SAS.
SAS/WAREHOUSE ADMINISTRATOR	Création et manipulation de DATA WAREHOUSE ET DATA MART.

Licence

SAS est un logiciel payant dont la licence doit être renouvelée tous les ans. L'extension d'utilisation se fait par l'intermédiaire d'un fichier (pas besoin de réinstaller le logiciel).

II - Les interfaces de SAS



On distingue principalement quatre types de fenêtre:

II-A - Les fenêtres d'écriture des programmes

Il existe deux fenêtres permettant l'écriture de code SAS : les fenêtres "Program Editor" et "Enhanced Editor". Elles sont toutes les deux très similaires. Par la suite, nous ne nous intéresserons qu'à l'"Enhanced Editor".

II-B - Le "journal" ou "fenêtre des erreurs" ou "LOG"

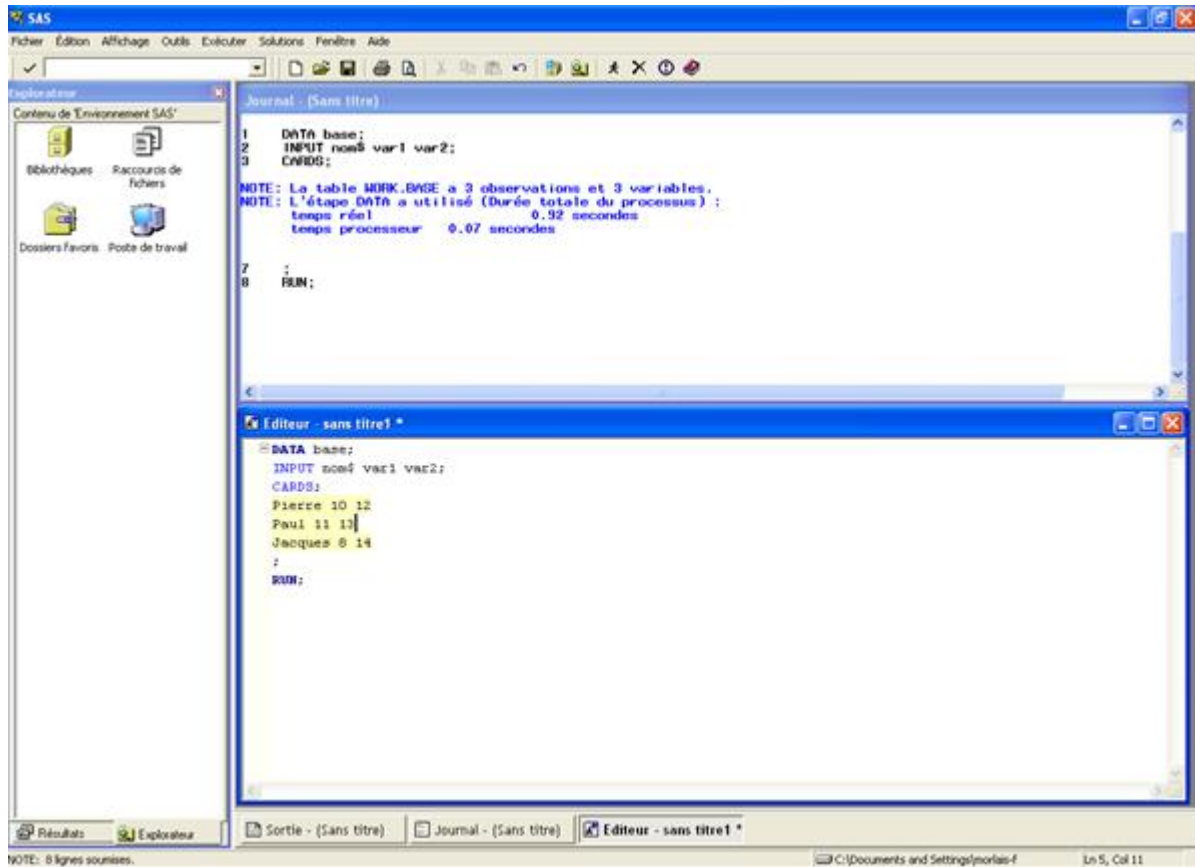
Cette fenêtre est très certainement la plus importante de l'environnement de programmation de SAS. C'est avec cette fenêtre que vous percevrez la réaction du logiciel au code soumis.

L'aspect visuel du texte généré dans cette fenêtre suite à une soumission de code permet d'interpréter le comportement de SAS vis-à-vis de cette exécution :

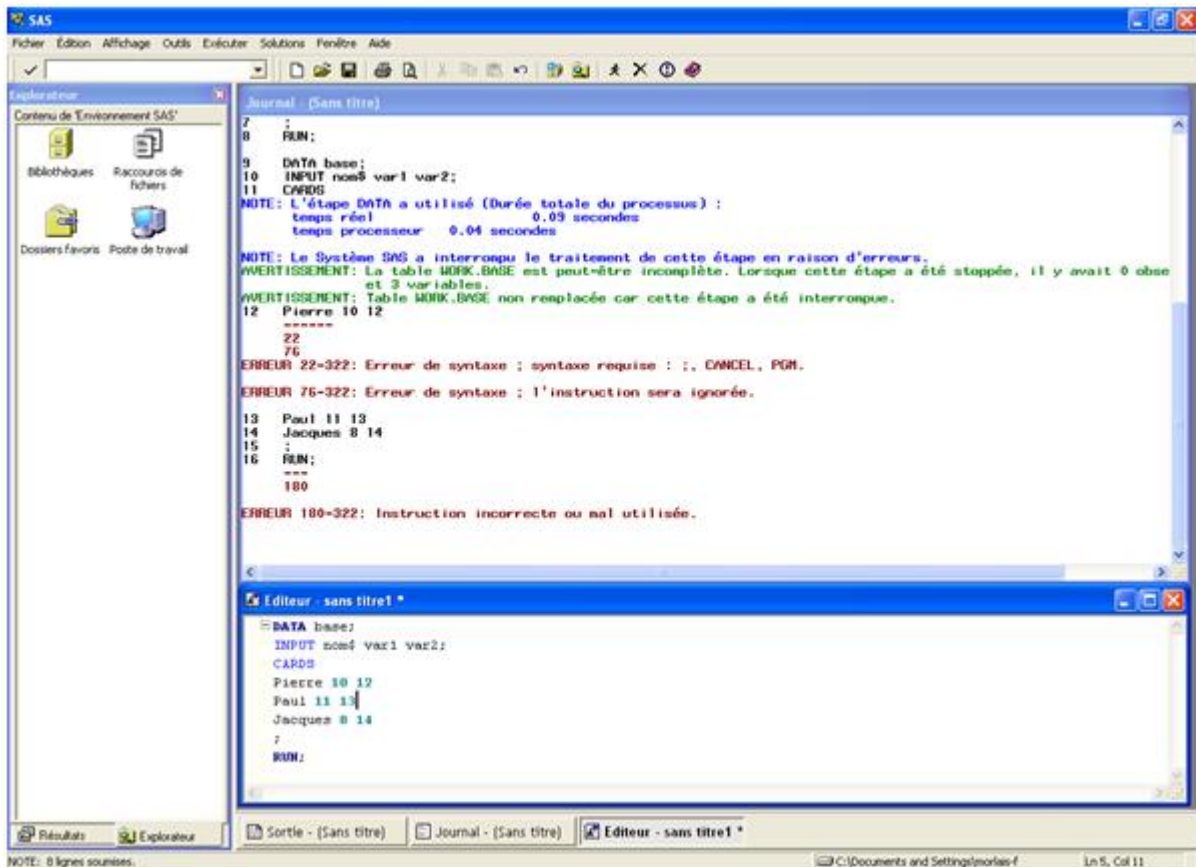
- Si la couleur du texte est bleue : SAS n'a détecté aucune erreur.
- Si la couleur est verte cela signifie que certaines erreurs « non fatales » à l'exécution ont été détectées.
- Si la couleur est rouge : des erreurs graves ont été détectées, empêchant l'exécution du programme.

Ci-dessous quelques exemples visuels de code générés :

Un programme correctement exécuté :



Un programme avec des erreurs :



Nous verrons plus en détails dans la prochaine partie un exemple explicitant ces erreurs lors d'exécution.

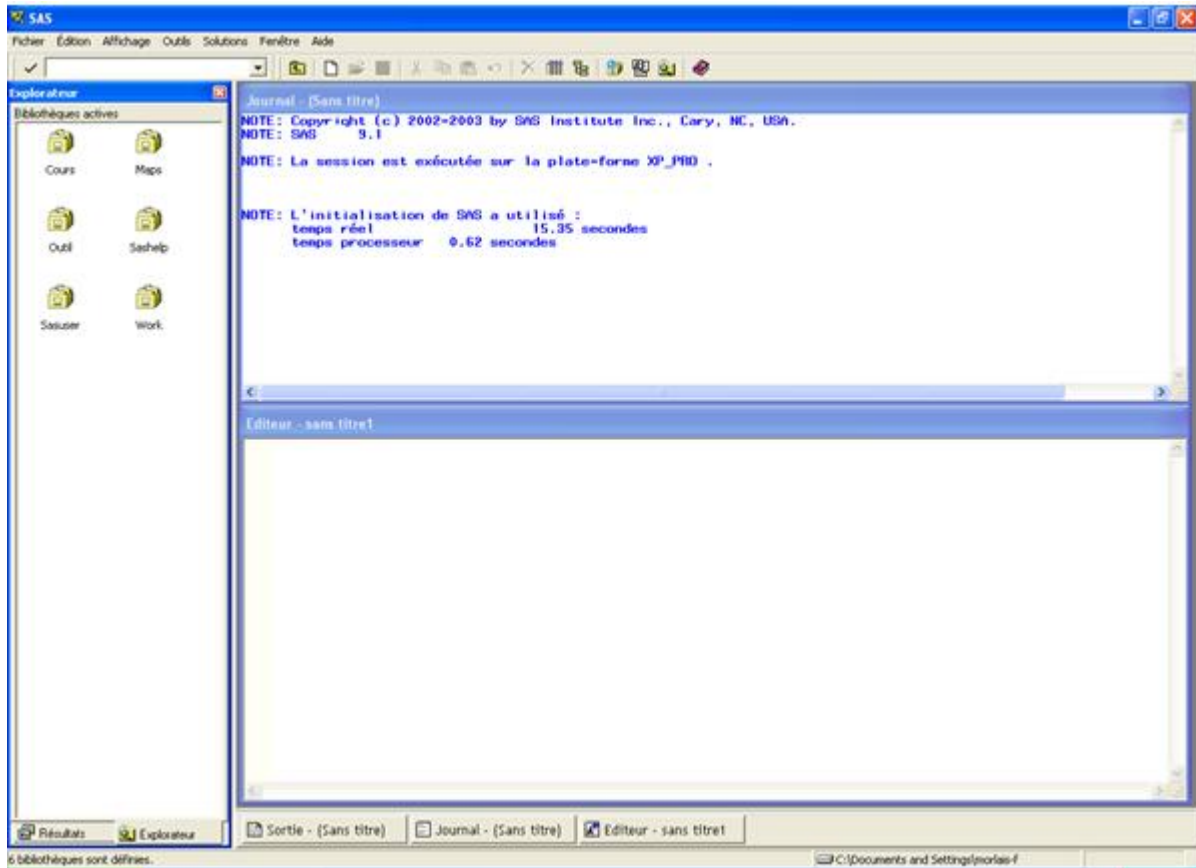
II-C - La "fenêtre de résultat" ou "OUTPUT"

La procédure MEANS					
Variable	Nb	Moyenne	Écart-t=type	Minimum	Maximum
var1	3	9.666667	1.5275252	8.0000000	11.0000000
var2	3	13.000000	1.0000000	12.0000000	14.0000000

Cette fenêtre affiche les résultats des analyses statistiques lancées par l'intermédiaire des procédures ou fonctions SAS.

II-D - L'explorateur

Il permet la navigation parmi les objets SAS (données et catalogues) et parmi votre espace de travail. Pour naviguer dans vos données SAS vous devez cliquer sur Bibliothèque (LIBNAME). Les bibliothèques sont en quelques sortes un raccourci pointant vers un répertoire. En créant vos propres bibliothèques vous pouvez ainsi sauvegarder vos données et/ou vos macro pour ainsi pouvoir les réutiliser facilement. La création de bibliothèques est très importante car il faut savoir que par défaut SAS ne stocke pas en mémoire les manipulations d'une session à l'autre. En effet au cours d'une même session toutes vos données sont par défaut sauvegarder dans une bibliothèque temporaire (WORK) qui est vidée à chaque fermeture du logiciel SAS.



Pour créer vos propres bibliothèque de travail vous devez cliquer sur « Nouvelle Bibliothèque » sur la barre de commande SAS.

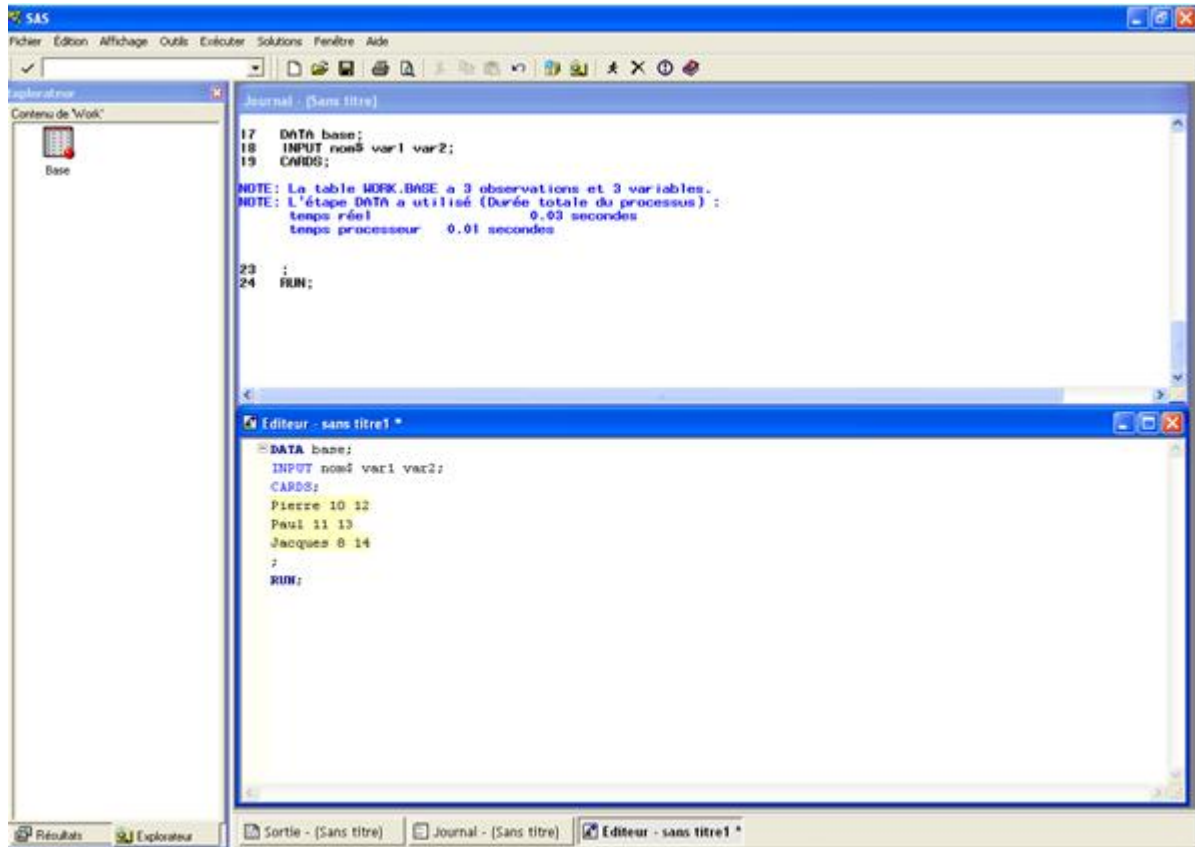
Vous pouvez aussi en créer une par le code suivant :

```
LIBNAME MonEspace « c:\MesDocs\SAS » ;
```

Le gros avantage de l'utilisation des bibliothèques provient du fait qu'elles permettent le stockage des bases de données : utilisées, modifiées ou créées lors de vos sessions SAS. Un autre avantage non négligeable est le stockage des macros compilées.

Ainsi vous pouvez rappeler ces marcos sans forcément avoir besoin de le recompiler.

On fera remarquer que par défaut les tables sont enregistrées dans la librairie Work. Cette librairie à l'avantage ou l'inconvénient, cela dépend de l'utilisation, de supprimer toutes les tables créées lors d'une session.



IV - Le fonctionnement de SAS

Le logiciel SAS fonctionne par l'intermédiaire du langage SAS. Ce code visible lorsque l'on développe des programmes sur console est invisible lorsque l'on utilise le mode interactif.

La programmation SAS peut être découpée en trois parties :

- Les étapes DATA : elles permettent au mettre titre que SQL une manipulation des données.
- Les procédures PROC : elles permettent les analyses statistiques de bases de données.
- Le langage macro : il sert à automatiser les programmes et réaliser des manipulations plus sophistiquées sur les données.

Le langage SAS doit être écrit dans l'« Enhanced Editor » et exécuter en cliquant sur le bouton soumettre de la barre de menu SAS (symbolisé par un petit bonhomme qui court).

IV-A - Etape DATA

Une étape DATA commence toujours par le mot clé DATA, suivi du nom d'une ou de plusieurs bases que l'on désire créer, et finit par le mot clé RUN. Dans la suite de ce tutorial les mots clés seront écrits en majuscule.

```
DATA nomBase ;
  .
  Jacques 8 14
RUN ;
```

Nous allons maintenant créer une table en saisissant les données directement sur SAS :

```
DATA base ; /*La nouvelle table se nommera base*/
INPUT nom$ var1 var2 ; /*On déclare 3 variable : une qualitative (suivi d'un $) nom et deux
numériques v1 et v2*/
CARDS ; /*Mode saisie des données*/
Pierre 10 12
Paul 11 13
Jacques 8 14
;
RUN ; /*Fin de l'étape DATA*/
```

Il est important de signaler que le respect de la ponctuation est de première importance dans SAS, l'oubli ou le mauvais placement de ceux-ci peuvent générer des erreurs.

Après exécution du code, vous devez aller dans la fenêtre « Journal » pour voir s'il des erreurs sont apparues.

Vous devez normalement avoir le texte ci-dessous dans votre journal, si ce n'est pas le cas vérifiez bien la ponctuation « ; » :

```
1 DATA base ; /*La nouvelle table se nommera base*/
2 INPUT nom$ var1 var2 ; /*On déclare 3 variable : une qualitative ($) nom et deux
numériques v1 et v2*/
3 CARDS ;
NOTE: La table WORK.BASE a 3 observations et 3 variables.
NOTE: L'étape DATA a utilisé (Durée totale du processus) :
temps réel 0.13 secondes
temps processeur 0.01 secondes
3 ! /*Mode saisie des données*/
7 ;
8 RUN;
```

N'ayant que du bleu vous pouvez déjà vous dire que votre programme ne contient pas d'erreurs. Ce qui est bien le cas quand on regarde dans le détail : SAS vous dit que la table SAS a été enregistrée dans la « bibliothèque Work» avec le nom « Base ». Cette table contenant 3 observations et 3 variables, ce qui est effectivement le cas.

Pour enregistrer votre base dans une bibliothèque définie par l'utilisateur vous devez procéder comme suit :

```
LIBNAME test "D:\temp";
DATA test.base ; /*Stockage de la table base dans maLibrairie*/
INPUT nom$ var1 var2 ;
CARDS ;
Pierre 10 12
Paul 11 13
Jacques 8 14
;
RUN;
```

Dans ce tutorial nous n'entrerons pas dans d'autres détails de manipulation de bases de données sur SAS car cette partie à elle seule pourrait largement couvrir un nouveau tutorial.

Sachez que pour la gestion de bases de données sur SAS vous pouvez dans la version classique importer des données issues de fichier de formats très divers, manipuler vos données et créer de nouveaux fichiers.

IV-B - Procédure PROC

Les procédures PROC : Permettent l'analyse et le reporting.

IV-B-1 - Un premier exemple

La syntaxe classique d'une procédure est la suivante :

```
PROC nomProcédure / OPTIONS ;
□□
RUN;
```

Une procédure commence toujours par le mot clé PROC et finit toujours par le mot clé RUN. Entre ces deux mots clés vous pouvez paramétrer votre procédure de façon à obtenir le résultat voulu.

Je vais ci-dessous vous donner un exemple de procédure réalisant un calcul de moyenne : la PROC MEANS. Pour calculer notre moyenne nous devons spécifier à cette procédure le fichier contenant la variable dont on souhaite calculer la moyenne.

```
PROC MEANS DATA=base; /*Ceci est un commentaire pouvant être sur plusieurs lignes*/
RUN;
```

Une fois le code exécuté, allez dans la fenêtre « Journal » pour voir s'il s'est bien déroulé.

Vous devez avoir le texte ci-dessous dans votre journal :

```
9 PROC MEANS DATA=base; /*Ceci est un commentaire pouvant être sur plusieurs
lignes*/
10 RUN;
NOTE: 3 observations copiées de la table WORK.BASE.
NOTE: La procédure MEANS a utilisé (Durée totale du processus) :
temps réel 1.93 secondes
temps processeur 0.06 secondes
```

10 !

De même que pour l'étape DATA, il apparaît clairement que votre code ne contient pas d'erreur.

Nous allons maintenant générer volontairement une erreur « en omettant le premier point virgule » pour observer la réaction de SAS.

```
PROC MEANS DATA=base
RUN;
```

Nous visualisons dans la fenêtre « Journal » le texte suivant :

```
11 PROC MEANS DATA=base
12 RUN;
---
22
-
200
ERREUR 22-322: Erreur de syntaxe ; syntaxe requise : ;, (, ALPHA, CHARTYPE,
CLASSDATA, CLM, COMPLETETYPES, CSS, CV, DATA,
DESCEND, DESCENDING, DESCENDTYPES, EXCLNPWGT, EXCLNPWGTS,
EXCLUSIVE, FW, IDMIN, KURTOSIS, LCLM, MAX,
MAXDEC, MEAN, MEDIAN, MIN, MISSING, N, NDEC, NMISS, NONOBS, NOPRINT,
NOTHEADS, NOTRAP, NWAY, ORDER, P1,
P10, P25, P5, P50, P75, P90, P95, P99, PCTLDEF, PRINT, PRINTALL, PRINTALLTYPES,
PRINTIDS, PRINTIDVARS,
PROBT, Q1, Q3, QMARKERS, QMETHOD, QNTLDEF, QRANGE, RANGE, SKEWNESS,
STDDEV, STDERR, SUM, SUMSIZE, SUMWGT,
T, THREADS, UCLM, USS, VAR, VARDEF.
ERREUR 200-322: Le symbole n'est pas reconnu et sera ignoré.
```

On voit très clairement que SAS ne comprend pas notre code. Cet exemple illustre l'importance de la lecture du « journal » après l'exécution d'un programme, ainsi que l'utilité du point-virgule dans le langage SAS. Ceux-ci doivent toujours être placés à la fin d'une instruction.

Par exemple :

```
PROC MEANS DATA=base
MEAN; /*1ère instruction*/
VAR v1; /*2ème instruction*/
RUN; /*3ème instruction*/
```

Avec le « Journal » suivant :

```
PROC MEANS DATA=base
MEAN; /*1ère instruction*/
VAR v1; /*2ème instruction*/
RUN; /*3ème instruction*/
NOTE: 3 observations copiées de la table WORK.BASE.
NOTE: La procédure MEANS a utilisé (Durée totale du processus) :
temps réel 0.01 secondes
temps processeur 0.01 secondes
```

Le dernier exemple illustre une petite partie des nombreuses options des procédures. En effet si vous ne voulez que le calcul de moyennes dans votre sortie, vous devez le signifier dans votre procédure à travers l'ajout d'options (dans cet exemple l'ajout de l'option MEAN permet l'obtention de la moyenne seule).

Pour visualiser les résultats statistiques de votre code vous devez cliquer sur la fenêtre « Sortie ».

IV-B-2 - Plus de détails

Le nombre de procédures sur SAS est considérable, chaque méthode statistique possède sa où ses propres procédures. Chaque procédure possède elle aussi un grand nombre d'options permettant de paramétrer tout aussi bien les entrées (bases de données) que les sorties (résultats). Ce qui fait qu'au final SAS est un logiciel dont on ne peut connaître toutes les options : l'utilisation de l'aide en ligne est à ce titre très précieuse de même que le forum et la **FAQ SAS** de developpez.com

L'aide en ligne SAS donne une description des diverses procédures tant au niveau des sorties générées qu'au niveau des syntaxes. Cette aide en ligne est le premier outil à prendre en considération pour tout développeur SAS.

Pour illustrer la diversité des options SAS, je vais prendre l'exemple d'un calcul du chi2. Cette mesure sert à quantifier le degré de dépendance (identique à la notion corrélation entre variable numérique) d'au moins deux variables qualitatives (non numérique). Une variable qualitative étant une variable du style : 1 = homme, 2 = femme.

La base de données initiale peut être de deux formes :

SEXE	MALADIE
1	0
1	1
2	0
1	0
2	0
1	0
1	1
2	1
2	1
2	1
2	0
2	0

```

DATA base;
INPUT sexe maladie;
CARDS;
1 0
1 1
2 0
1 0
2 0
1 0
1 1
2 1
2 1
2 1
2 0
2 0
;
RUN;
    
```

Ou de cette forme (tableau de contingence):

	Maladie	
	0	1
Homme	3	2
Femme	4	3

```
DATA base2;
INPUT sexe maladie effectif;
CARDS;
1 0 3
1 1 2
2 0 4
2 1 3
;
RUN;
```

Une des procédures permettant entre autre le calcul du chi2 est la procédure FREQ, celle-ci intègre bien évidemment la possibilité de lecture des deux types de tables ci-dessus dans ces options.

Pour la première table :

```
PROC FREQ DATA=base;
TABLE sexe*maladie/CHISQ;
RUN;
```

Pour la deuxième table :

```
PROC FREQ DATA=base2;
TABLE sexe*maladie/CHISQ;
WEIGHT effectif;
RUN;
```

L'ajout de l'option WEIGHT spécifie l'existence d'une variable de poids, SAS comprend alors que la base de données est sous la forme d'un tableau de contingence.

Pour calculer la valeur du test du chi2 vous devez rajouter l'option « CHISQ », l'ajout d'options se faisant avec la / option1 option2 □ .

Dans cet exemple sans cet ajout :

```
PROC FREQ DATA=base2;
TABLE sexe*maladie;
WEIGHT effectif;
RUN;
```

Nous n'aurions pas obtenu le calcul du chi2 car cette option n'est pas définie par défaut.

Le but des deux premiers exemples étaient de pouvoir montrer les principales utilisations des options des procédures en vue de l'obtention du résultat souhaité. Je vais ci-après vous indiquer comment récupérer vos résultats dans une table et dans un fichier Word.

IV-B-3 - La gestion des sorties

Pour imprimer les informations de cette table dans la fenêtre résultat vous devez utiliser la procédure PRINT. Cette dernière étant assez différente des autres car elle ne fait qu'imprimer les données d'une table, contrairement aux autres procédures qui font des calculs sur les bases.

```
PROC MEANS DATA=base;
OUTPUT OUT=sortie;
PROC PRINT DATA=sortie ;
RUN;
```

L'autre notion de gestion des sorties sous SAS couvre la présentation des résultats dans un éditeur à partir de SAS. Nous allons ci-dessous détailler le code « imprimant les résultats » dans l'éditeur de texte Word.

```
ODS RTF FILE="D:\resultats.doc";
PROC MEANS DATA=base;
RUN;
ODS RTF CLOSE;
```

IV-C - Langage MACRO

Une macro commence par « %MACRO nomFonction ; » et finit par « %MEND ; ». Les paramètres de la macro sont initialisés de la façon suivante, par exemple pour deux paramètres par1 et par2 : « %MACRO maFonction(par1,par2) ».

```
%MACRO maFonction(par1,par2) ;
PROC MEANS DATA=&par1 ; /*Pour appeler les paramètres on utilise le &*/
VAR &par2 ;
RUN;
%MEND ;
%maFonction(base,v1) ;/*On exécute la macro*/
```

Le but de ce présent article étant une présentation générale de SAS il ne nous est pas apparu utile d'approfondir les subtilités du langage MACRO dans ce tutorial.

Conclusion

Au terme de ce tutoriel vous posséder maintenant une vue d'ensemble du fonctionnement de SAS tant au niveau des principales fonctionnalités qu'au niveau du langage informatique classiquement utilisé.

Ressources utilisées pour la rédaction de l'article

La communauté SAS de developpez.com

La FAQ
Le Forum

Internet

Le site de SAS
SAS Wikipedia
Anthony J . BARRE
L'histoire de SAS Institue Corporation

Livre

Maitriser SAS BASE et SAS MACRO (SAS 9 et versions antérieures) : Hélène Kontchou Kouomegni et Olivier decourt

Remerciements

Je remercie **Fleur-Anne.Blain**, **Adrien Artero** et **Steelspirit** pour avoir collaboré à la réalisation de ce tutoriel et pour leurs précieuses relectures.