Séance n°1 Débuts en SAS : Généralités - Présentation - Étape DATA.

L'objectif de ce cours est d'initier les étudiants à l'utilisation du logiciel SAS, tout en illustrant divers éléments de Statistique élémentaire. L'aspect initiation est important : la maîtrise de SAS est aujourd'hui indispensable à un étudiant issu de filière Statistique ; mais, un peu délicat à acquérir, elle nécessite une formation appropriée.

Vous trouverez sur ma page web (http://perso.lcpc.fr/guillaume.saint-pierre/Master2.html) les sujets et les corrections des TP, ainsi qu'une liste de liens utiles pour travailler avec SAS.

0. Généralités sur SAS

SAS -Statistical Analysis System- est un logiciel polyvalent qui traite pratiquement tous les domaines de la Statistique. De conception américaine, il est développé par la société SAS-Institute basée à Cary, en Caroline du Nord ; et a acquis, depuis sa mise sur le marché en 1976, une situation dominante dans de nombreuses branches d'activités économiques. SAS est donc un logiciel multi-facettes capable de gérer de gros volumes de données (tableaux de plusieurs gigaoctets) à des fins d'analyse ou de reporting automatisé. Autre avantage déterminant sur d'autres logiciels de Statistique présents dans l'industrie, tels que SPSS, Spad, Simca-P (en chimiométrie), etc., il est possible de programmer sous SAS ses propres procédures. Malheureusement, SAS compte aussi des points faibles : coût élevé, langage de programmation peu avancé, graphiques laids, etc. Des versions équivalentes de SAS existent pour chacun des environnements Unix et Windows (depuis 1990). La dernière version V9.2 est distribuée en France depuis 2010.

1. Mise en route et présentation

Pour accéder à SAS Unix depuis une machine de L'UTES :

1. Entrer votre login (normalement c'est le numéro de votre carte d'étudiant) puis votre mot de passe (celui attribué au moment de l'inscription administrative). Cette procédure est aussi valable pour ceux dont l'inscription n'est pas finalisée (inscription via internet) ;

2. Pour lancer SAS, cliquer sur le menu général, choisir l'onglet « Sciences », puis « Mathématiques », et cliquer sur SAS (serveur 1 ou 2) ;

3. Une fenêtre de connexion au serveur hébergeant SAS apparaît alors. Vous devez à nouveau taper votre mot de passe pour accéder à la ligne de commande.

Il n'y a pas de version de SAS pour Windows disponible dans les salles de l'UTES. Vous aurez néanmoins la possibilité d'installer une version de SAS Windows sur votre ordinateur personnel (sous réserve d'enregistrement de votre demande auprès de SAS-institute).
Pour disposer d'une version de SAS utilisable chez vous uniquement, vous devez vous rendre à cette adresse : http://www.sas.com/offices/europe/france/academic/licence-gratuite-SAS-domicile.html.
Remplissez le formulaire en ligne, imprimez-le, datez et signez, et ensuite donnez-le à votre enseignant qui remettra le document au CRI diffuseur du logiciel.

1.1 Brefs rappels d'Unix

Unix est, au même titre que **Windows**, un *système d'exploitation*, c'est-à-dire un ensemble de programmes permettant d'utiliser l'ordinateur. Entre autres, c'est lui qui gère l'accès aux différentes ressources (mémoire, affichage, etc.) de l'ordinateur. On dit qu'il est « multi-tâches » car il permet de faire fonctionner plusieurs programmes en même temps, et « multi-utilisateurs » car il permet à plusieurs utilisateurs de travailler en même temps sur la même machine. Votre travail est rangé sur le disque dans des *fichiers* et ces fichiers sont eux-mêmes rangés dans des *répertoires*. En fait, les répertoires sont des fichiers particuliers contenant une liste de fichiers, ils sont donc eux-mêmes rangés dans des répertoires. Le seul répertoire qui ne soit pas dans un autre répertoire est appelé *racine*. Cette organisation est souvent représentée par une structure arborescente.

Quelques commandes de base sur les fichiers et les répertoires :

- **pwd :** Indique le nom du répertoire de travail (print working directory) ;
- **cd :** Renvoie au home directory (change directory) ;
- **cd.. :** Renvoie au répertoire « père » de votre répertoire courant (c'est-à-dire celui situé juste au-dessus dans la hiérarchie) ;
- **cd < *répertoire* > :** Va dans le répertoire de nom *répertoire* ;
- **ls :** Donne la liste des fichiers du répertoire courant ;
- **ll** : C'est une abréviation pour ls –la où 'l' signifie 'en ligne' et 'a' signifie 'tout' (all) ;
- **mkdir < *nom* > :** Crée le répertoire *nom* (make directory) ;
- **rm < *fichier* > :** Efface le ou les fichiers spécifiés (remove) ;
- **rmdir < *répertoire* > :** Efface le ou les répertoires spécifiés. Il faut que ces répertoires soient vides !
- **cp < *nom1* > < *nom2* > :** Copie le fichier de nom *nom1* sous le nom *nom2* (copy) ;
- **mv < *nom1*> < *nom2* > :** Change le nom du fichier *nom1* en *nom2* (move).
- Attention, sous Linux, un chemin de fichier s'écrit de la façon suivante : `"/home/11111/tpsas/exemple.txt"` où `/home/11111/` est votre « home », auquel on peut accéder en utilisant le raccourci `~` (ex : « `~/tpsas/exemple.txt` »). Le code 11111 représente votre login.

Pour pouvoir travailler, vous aurez besoin des fichiers fournis par l'enseignant. Utilisez les commandes précédentes pour copier dans votre « home » le dossier datasas présent sur son « home ».

Créer tout d'abord un dossier TPSAS dans votre « home » en utilisant la commande mkdir.
Copier ensuite le dossier datasas indiqué par l'enseignant dans votre espace personnel. Il vous a spécialement ouvert des droits d'accès pour ce dossier.

```
cp -R /home/saintpierreg/TPSAS/datasas  /home/XXXXXX/TPSAS/datasas
```

Où XXXXXX est votre login (N° de carte étudiant)
Taper la commande `man cp` pour voir ce que fait l'option –R.

1.2 Présentation de SAS

L'environnement SAS est constitué de 3 fenêtres principales :

- **SAS Program Editor :** éditeur de texte de SAS dans lequel on tape les commandes à exécuter ;
- **SAS Log :** fenêtre de gestion de la session, les commandes y sont compilées ligne par ligne ;
- **SAS Output :** pour visualiser les sorties d'un programme SAS.

Remarques La fenêtre *Log* est très importante pour s'assurer de la bonne exécution d'un programme avec les notes en bleu, les avertissements (warnings) en vert …et les erreurs en rouge ! Notons aussi que les fenêtres *Log* et *Output* ne sont pas purgées automatiquement par SAS entre les exécutions. Il est conseillé d'en effacer régulièrement le contenu en utilisant la commande `Edit > Clear all`.

D'autres fenêtres sont aussi utiles :

- **SAS Explorer :** pour visualiser et manipuler les fichiers et les librairies SAS ;
- **SAS Toolbox :** pour donner des ordres à l'interface SAS ;
- **SAS Results :** pour visualiser et manipuler les résultats de SAS textuels et graphiques.

Dans la fenêtre *Program Editor* un programme SAS sera toujours composé :

1. D'une ou plusieurs **étapes DATA :** définition / lecture / modification d'un ou plusieurs tableaux de données ;

2. Puis d'une ou plusieurs **étapes PROC :** toutes les procédures portant sur les tableaux de données.

Modules SAS : Environ 40 modules (payants) tendent à compléter les fonctionnalités du logiciel parmi lesquels : **SAS/BASE** (instructions pour la manipulation de données, les statistiques descriptives élémentaires, l'édition de rapports et la programmation dans les langages SAS de base, SQL et macro), **SAS/STAT** (procédures de modélisation, de classification et de statistiques descriptives), **SAS/GRAPH** (édition de graphiques haute résolution), **SAS/INSIGHT** (analyse statistique interactive), **SAS/IML** (Interactive Matrix Language), **SAS/SQL** (Structured Querry Language), **SAS/ETS** (Econometrics and Time Series : étude des séries chronologiques), **SAS/AF** (Application Facility : création d'interfaces), **SAS/OR** (recherche opérationnelle), **SAS/FSP** (Full Screen Products), **SAS/ASSIST** (interface clique-bouton pour créer des programmes SAS)…

1.3 Rédaction de programmes SAS

L'écriture de programmes se fait normalement en utilisant la fenêtre *Program Editor*, mais pour des raisons pratiques il est recommandé d'utiliser un éditeur de texte différent. En effet, la version Unix de l'éditeur de programme SAS est peu conviviale, et certaines fonctionnalités classiques ne sont pas disponibles. Il est recommandé d'utiliser un des programmes suivants : *emacs, xemacs, gedit, ou kwrite*.

Pour lancer l'un de ces éditeurs de texte aller dans le menu principal `/Outils/Konsole Terminal` puis taper le nom de l'éditeur choisi, suivi de & (ceci afin de continuer à pouvoir taper des commandes Unix dans la Konsole).

- **N'oubliez pas d'enregistrer régulièrement vos programmes avec l'extension .sas**
- **Pour les exécuter, il suffit de faire un copier/coller dans l'éditeur de programme de SAS et de lancer la compilation :** `Menu > Run > Submit` ou icône « bonhomme » dans les Tools ou touche F3.

2. Étape DATA

Un fichier de données ne peut être reconnu, lu ou traité par SAS que s'il est dans un format spécifique. Nous appellerons *Table SAS* un tel fichier écrit dans ce format. Il existe des tables SAS temporaires et d'autres permanentes. Une séquence de lecture des données se présente donc de la façon suivante :

```
DATA < nom de la Table SAS >;
 INPUT < liste des variables >;
 CARDS;
 < les données sont entrées " à la main " >
 ;
RUN;
```

L'instruction **CARDS** signale le début de la saisie des observations (une seule observation par ligne). Les données sont alors séparées par des espaces, une valeur manquante étant représentée par un point. Notons que, dans la liste des variables, le séparateur est encore un blanc. Enfin, l'instruction **RUN** (facultative) termine l'étape DATA.

Exemple de programme

```
00001   /* exemple de programme; */          → ligne de commentaire
00002   data exemple;
00003   input nom$ naiss CSP$ auto$ sin98 sin99;
00004   cards;
00005   Pierre 65 lib golf 0 10
00006   Paul 48 arti 306 0 0
00007   Jacques 61 cadre 205 51 0
00008   Carole 70 lib AX 27 0
00009   Caroline 65 cadre golf 0 0
00010   Nathalie 62 cadsup 306 0 5
00011   ;
00012   run;
```

Remarque La numérotation des lignes de code est spécifique à la version Unix. Cette numérotation n'apparaît pas lorsque l'on utilise emacs ou xemacs.

Variables SAS : Le nom des variables nominales est toujours suivi du symbole **« $ »**. Les variables numériques peuvent être qualitatives (par ex. dichotomiques / binaires ou catégorielles…) ou quantitatives (discrètes ou continues). Le nom d'une variable comporte au plus 8 caractères, commence par une lettre « A-Z » ou « _ » (underscore) et ne doit pas contenir d'espace(s) ou de symboles spéciaux tels que « & », « % », « $ », « # », etc.

Compilation : `Menu > Run > Submit` ou icône « bonhomme » dans les *Tools* ou touche **F3**.

SAS est un langage interprété. Lorsque l'on lance un programme, les lignes de code défilent lues une à une dans la fenêtre Log. Ici, un message indique la création de la table *exemple* qui compte 6 lignes et 6 colonnes. Sur la version Unix, les lignes de code disparaissent de la fenêtre Program Editor lors de l'exécution. Celles-ci peuvent être « rappelées » ensuite avec la commande `Menu > Run > Recall Last Submit`.

Librairies SAS : Toute nouvelle table est par défaut conservée dans la **librairie temporaire Work** et ne reste en mémoire que le temps de la session SAS en cours. Pour pérenniser l'existence d'une table, il faut l'enregistrer au sein même du programme dans une **librairie définitive**. L'instruction **LIBNAME** indique alors le chemin d'un répertoire où doivent être stockées les tables permanentes (en « .sas7bdat »).

Exemple : Création d'une librairie définitive « malib »

Attention, il faut au préalable créer le dossier auquel libname fait référence, en utilisant des commandes Unix (commande mkdir) entrées dans la Konsole.

```
libname malib '/home/XXXXXX/TPSAS/malib';

data malib.exemplebis;      /* exemplebis est permanente */
 set exemple;               /* exemple est temporaire */
run;
```

Quelques informations sur les autres librairies de SAS dans Explorer :

SASHELP contient des données fournies par SAS à des fins d'exercices (comme *class*, *air*, *shoes*, etc…) ;

SASUSER est une bibliothèque personnelle (chaque utilisateur a sa propre *SASUSER*) permanente (contrairement à *WORK*). Cette bibliothèque contient entre autres les informations de personnalisation de la session SAS (barres d'outils, par exemple). Il est possible d'y écrire ses données, même si, en général, il est préférable d'allouer une bibliothèque séparée pour le stockage des données.

Quelques commandes pour modifier une Table SAS dans une étape DATA :

Attention, ces lignes de commande doivent être incluse dans une étape data de la façon suivante, afin de permettre la modification des données (vous ne pouvez pas modifier un jeu de données si il est ouvert par vous dans une fenêtre viewtable) :

```
data exemple;
set exemple;
IF nom='Jacques' THEN naiss=50;/*modifications des données exemple*/
run;
```

- `IF nom = 'Jacques' THEN naiss = 50;` *modification d'une valeur* ;
- `RENAME naiss = naissance;` *changer le nom d'une variable* ;
- `KEEP nom sin98 sin99;` ⇔ `DROP naiss CSP auto;` *enlever des variables* ;
- `INPUT varsup; cards; …;` *ajouter une variable supplémentaire* ;
- `sintotal = sin98 + sin99;` *ajouter une variable fonction des précédentes* ;
- `if sin98 = 0 then DELETE;` *enlever des observations*.
- SAS reconnaît les fonctions mathématiques et statistiques suivantes : **ABS**, **ARCOS**, **ARSIN**, **ATAN**, **COS**, **COSH**, **SIN**, **SINH**, **EXP**, **LOG** (népérien), **MIN**, **MAX**, **RANGE**, **SQRT**, **N** (nb de valeurs disponibles), **NMISS** (nb de valeurs manquantes), **SUM**, **TAN**, **TANH**, **MEAN**, **VAR, STD…**
- Une instruction très utile et souvent utilisée est la suivante :
- `SET TAB1; SET TAB2;` ⇔ `MERGE TAB1 TAB2;`
  fusionner 2 tables contenant des variables différentes et des individus identiques ;

Attention ! Avant de concaténer 2 tables, les lignes doivent être triées dans le même ordre (avec la **PROC SORT** et l'instruction **BY**).

Importer un tableau de données depuis un fichier externe :

Il est possible d'importer sous SAS des fichiers Excel « **.**xls » ou « **.**csv » (Comma Separated Value, délimiteur point-virgule). L'instruction **INFILE** indique le chemin du fichier à importer.

- On peut importer des données sous SAS directement en utilisant l'étape data. La syntaxe pour un fichier « **.**csv » est la suivante :

```
DATA < nom de la Table SAS >;
 INFILE '~/ … / fichier.csv'    delimiter = ';';   /* ou dlm = ';' */
 INPUT < liste des variables >;
RUN;
```

L'option `delimiter = ';'` permet de préciser quel est le caractère utilisé dans les données originales pour sépare les colonnes. Il est possible de spécifier d'autres caractères que ' ;' (Cela dépendra de vos données originales).

- Syntaxe pour un fichier Excel (attention, cela ne marche que sous Windows, ne marchera pas dans les salles de l'UTES) :

```
filename file DDE "Excel|m:\courssas\[import.xls]feuil1!L2C1:L19C4";
/* DDE pour Dynamic Data Exchange, Excel désignant l'" Engine ". */

DATA < nom de la Table SAS >;
 INFILE file;
 INPUT < liste des variables >;
RUN;
```

Les données à importer sont dans l'onglet 'feuil1' du fichier Excel *import.xls*, de la ligne 2 à la ligne 19 et sur les 4 premières colonnes. Le fichier Excel doit impérativement être ouvert pour mener cette opération.

- Plus simplement, nous pourrons encore importer des données depuis un fichier externe avec la commande `File > Import Data` en précisant ensuite la nature et le chemin du fichier source.

- Encore une autre façon d'importer des données, consiste à utiliser une procédure SAS spécialement conçue pour cela

```
proc import datafile='~/TPSAS/datasas/Vins_de_Bordeaux.xls'
   out=< nom de la Table SAS de sortie > replace;
   sheet=< nom de la feuille du fichier excel >;
   getnames=yes;
run;
```

Il est fortement recommandé de consulter l'aide de SAS sur internet (http://support.sas.com/onlinedoc/913/docMainpage.jsp) pour connaître les différentes options disponibles (et elles sont en général nombreuses). Essayez d'importer le fichier `Vins_de_Bordeaux.csv` à l'aide de la procédure IMPORT en regardant l'aide en ligne pour vous aider.

Il existe une procédure équivalente pour l'exportation qui sera vue plus en détail au TP n°2. Les lignes suivantes permettent d'exporter les données Fil2 dans un fichier .txt dont les colonnes sont délimitées par des espaces.

```
PROC EXPORT DATA= Fil2
            OUTFILE= "~/TPSAS/datasas/Vins_de_Bordeaux.txt"
            DBMS=DLM REPLACE;
            DELIMITER='20'x; /* delimiter = space */
RUN;
```

Remarque Il est également possible de récupérer des données sur internet en « **.**html » ou « **.**htm ». Pour ce faire, on peut copier (commandes *copier / coller*) les données dans un fichier via un éditeur de texte (*nedit* ou *emacs* sous Unix, le *bloc-note* ou *Word* sous Windows, par exemples), puis les formater pour ensuite les importer dans SAS.

Exercice n°1

- Vous avez déjà copié le répertoire « *datasas* » qui contient les tables SAS utilisées dans ce cours. Vous disposez donc des fichiers *champ08.txt* et *champ08.csv* dans *datasas*.

Ces tables contiennent les résultats du championnat 2007/08 de football de deuxième division. Les variables indiquent respectivement pour chaque club : le nom *club*, le nombre total de victoires *vic*, de matchs nuls *nul* et de défaites *def*, le nombre total de buts marqués *bp* et encaissés *bc*.

- Dans un premier temps, avec la commande INFILE, importer ces données sous SAS dans une table *tab1* à partir des deux types de fichiers sources. Donner un titre à cette table (avec l'instruction TITLE).

- Créer ensuite une table *clt* contenant sur une colonne et par ordre croissant tous les entiers compris de 1 à 20. Afficher les tables *tab1* et *clt*.

- À partir de la table *tab1*, créer une table *tab2* qui comptera 2 variables supplémentaires : *pts*, le nombre total de points obtenus par une équipe sachant qu'une victoire rapporte 3 points et un match nul 1 point ; *diff*, la différence entre les nombres de buts marqués et encaissés. Afficher la table *tab2*.

- À partir de la table *tab2*, créer une table *tab3* qui reprend les données de *tab2* triées par ordre décroissant de points *pts* puis par ordre décroissant de différence de buts *diff*. Afficher la table *tab3*.

- Enfin, dans une table *champ*, fusionner les tables *clt* et *tab3* et ajouter une variable *result* qui vaut « + » si l'équipe est classée dans les trois premières places, « - » si elle est classée dans les trois dernières places ou « = » sinon. Afficher la table *champ* puis effacer les tables *tab1*, *tab2*, *tab3* et *clt*.

Exercice n°2

Utiliser un moteur de recherche (par exemple, *Google*) pour trouver un tableau de données sur internet puis le rapatrier sous SAS.

Indication : Récupérer les données dans un fichier « **.**txt » via un éditeur de texte (l'un de ceux présentés plus haut). Importer ensuite ces données en utilisant l'instruction INFILE.