Introduction à SAS
Statistical Analysis System (SAS)

Manipulation des données: partie 1

Romain BROUTIN, Mackenson GREFIN, Antoine KRIEGER, Yasmine MEDKOUR

Plan de la séance:

- Histoire de SAS
- Comment installer SAS ?
- L’environnement de travail
- Bibliothèques
- Programmation
- Création / Importation de données
- Formats
- Conditions
- Boucles

Histoire de SAS :

Anthony J.Barr

- Création en 1960
- Programme d'analyse de variance
- Enrichissement pour faire des régressions multiples
- Ajout au fil du temps d'autres méthodes d'analyses
- Introduction à la gestion de bases de données et au reporting
- Actuellement très répandu sur le marché et l'un des leaders mondiaux du marché de la Business Intelligence

Installation de SAS

- Afin d'installer SAS à l'aide sur votre poste de travail, voir le fichier pdf Installation SAS.pdf (pages 1 à 4)

- Assurez vous au préalable de disposer d'une machine virtuelle sur votre poste de travail. Si ce n'est pas le cas: https://www.virtualbox.org/wiki/Downloads

- Pour installer la version universitaire, rendez-vous sur le site de SAS https://www.sas.com/en_us/software/university- edition/download-software.html

- Ensuite, suivez pas-à-pas les instructions

- Il vous sera demandé de vous créer un compte. Pour cela, vous devrez entrer votre adresse mail universitaire

- Si vous rencontrez une erreur concernant le kernel lors du démarrage de la machine virtuelle, il vous faudra rechercher dans vos dossier les fichiers VBoxUSBMon.inf VBoxDrv.inf et puis procéder à leur installation.

SAS on-line

- Il existe une version online gratuite voir fichier : Installation SAS.pdf (pages 5 à 7)
- Rendez-vous sur le lien suivant : SAS OnDemand for Academics Control Center
- Créer votre compte (l'adresse mail utilisée pour l'installation de SAS avec une machine virtuelle peut être réutilisée)
- Suite à la création de votre compte vous devez l'activer grâce à un lien qui vous sera envoyé par mail
- Vous pouvez désormais vous connecter à SAS on-line

Installation de SAS

- Avantages/ inconvénients :

	SAS sur PC	SAS OnDemand
Avantages	• Pas d'accès à internet obligatoire	• Possibilité de changer d'ordinateur
Inconvénients	• Obligation d'avoir sa machine à disposition	• Obligation d'voir un accès à internet

Pourquoi SAS ?

- SAS est un logiciel statistique qui permet la gestion de données et d'effectuer des analyses statistiques plus ou moins poussées (statistique descriptive, test de conformité, régressions linéaires…).

- Le langage de SAS peut se décomposer en trois parties: une partie DATA qui permet de créer et d'importer des bases de données, une autre partie permet la modification des data frames, et une partie procédure qui permet d'effectuer des analyses statistique.

- SAS fonctionne sur différents systèmes d'exploitations : Windows, Macintosh, Unix, Linux, Mainframe.

- SAS dispose de son propre langage de programmation : le macro langage.

- Fonctionne en modules.

- Ce langage permet de répondre à trois besoins qui sont la création et la gestion de base de données, l'analyse de ces base de données et la création et la diffusion de rapport de synthèse et de listing.

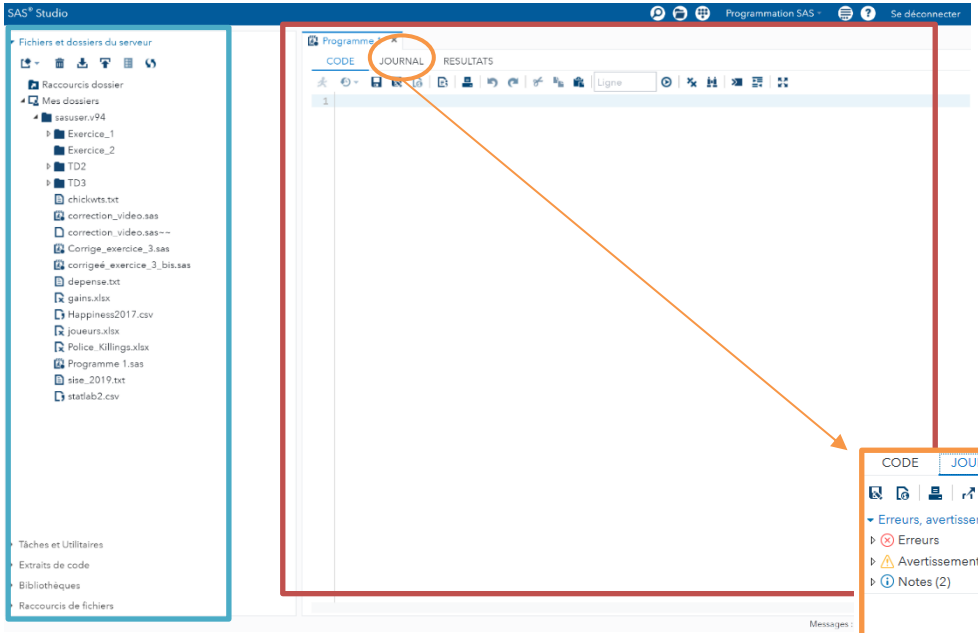
SAS : l'environnement de travail : les bonnes pratiques

On distingue 4 types de fenêtres:

- L'éditeur : permet d'écrire du code SAS et programmer
- Le journal (ou LOG) : montre comment SAS réagit au code exécuté et permet ainsi de mettre en évidence les erreurs
- La fenêtre de résultats (ou OUTPUT) : affiche les résultats des analyses statistiques lancées par l'intermédiaire des procédures ou fonctions SAS
- L'explorateur : permet la navigation parmi les objets SAS (données et catalogue) et parmi l'espace de travail

SAS : l'environnement de travail : les bonnes pratiques

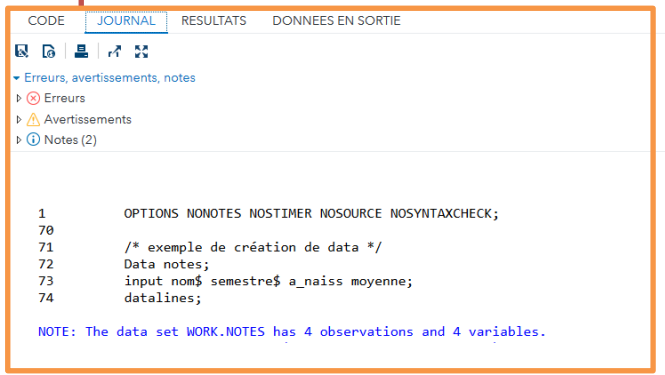
L'explorateur



L'éditeur

La fenêtre RESULTATS affiche les sorties des procédures.

La fenêtre DONNEES EN SORTIE affiche les data frame.



Journal

Bibliothèques (ou librairies)

Une bibliothèque est un répertoire dans lequel sont stockées les tables. Il existe des bibliothèques permanentes ou temporaires (WORK). On accède aux bibliothèque à l'aide de l'explorateur.

Nous avons 2 types de bibliothèques :

- Une bibliothèque permanente, on peut y stocker des tables pour les sessions suivantes.

  - Pour la créer : `LIBNAME TD1 '/mes documents/TD';`

  - Pour accéder à une table dans cette librairie : `DATA TD1.Nom_de_la_table;`
    `...`

- Une bibliothèque temporaire, appelée "Work". Les tables stockées dans cette bibliothèques sont temporaires.

  - La bibliothèque temporaire est déjà créée

  - Pour accéder à une table : `DATA Nom_de_la_table;`
    `...`

SAS : les bonnes pratiques

La programmation SAS peut être découpée en trois parties :

- Les étapes DATA permettent, au même titre que le langage SQL, une manipulation des données
- Les procédures PROC : elles permettent de faire des analyses statistiques de bases de données, des graphiques et de légères manipulation de données
- Le langage MACRO : il sert à automatiser les programmes et réaliser des manipulations plus sophistiquées sur les données

Les commentaires sont mis dans une « balise » : `/* Mon commentaire */`

Création d'une table via DATA

Les lignes de code suivantes permettent de créer une table de données qui n'existait pas (« from scratch »)

```
Data base; /* Nom de la nouvelle table */
length ville $10; /* spécifie la longueur de la variable ville*/
INPUT ville$ code_departement; /* Déclaration des variables et formats*/
CARDS; /* Ecriture des données*/
Lyon 69
Marseille 13
Lille 59
;
RUN;/* Fin de l'instruction DATA*/
```

À noter qu'il est aussi possible d'utiliser DATALINES au lieu de CARDS.

Données en sortie :

	ville	code_departement
1	Lyon	69
2	Marseille	13
3	Lille	59

Remarque :
Dans cet exemple, le DELIMITER est un espace (‘ ‘).

Importation de données

Les lignes de codes suivantes permettent d'importer des données à partir d'un fichier existant sur notre machine de travail. **DMBS** permet de spécifier le format du fichier.

- Via PROC IMPORT :

Fichier Excel

```
/* Import de données .xlsx */
PROC IMPORT
    DATAFILE='/mon_chemin/fichier.xlsx' /* Chemin du fichier source */
    OUT=ma_table /* Nom de la table SAS en sortie */
    DBMS=xlsx /* Format du fichier source */
    REPLACE; /* Permet de remplacer la table SAS si
                elle existe déjà */
    GETNAMES=yes; /* Considère que la première ligne
                     contient les noms des variables */
    SHEET='adult'; /* Pour préciser la feuille
                      contenant les données */
RUN;
```

Fichier texte ou csv

```
/* Import de données en .txt ou .csv */
PROC IMPORT
    ...
    DBMS=tab / csv
    ...;
    DELIMITER=';';
    ...
RUN;
```

Remarque :
On spécifie le DELIMITER en fonction du fichier:

- **DELIMITER**=';'
- **DELIMITER**=' '
- **DELIMITER**=','

Modification d'une table SAS dans une étape DATA :

Au sein d'une étape DATA il est possible d'effectuer des modifications:

```
RENAME datenais=date_de_naissance;  /* Renommer une variable */
KEEP nom moyenne age;  /* Garder exclusivement des variables */
DROP metier auto;  /* Supprimer des variables */
MERGE TAB1 … TABk; /* Fusion horizontale */
```

Mais aussi des créations de variable :

```
Distance_km = Distance_miles / 0.62137;
Vitesse = Distance_km / Temps;
IF Vitesse > 200 THEN Categorie = 'Sport';
ELSE IF ...
```

Exemples :

RENAME

(Suite de l'exemple diapo 12)

```
data base;
set base;
rename ville=Ville_A;
run;
```

DROP

```
data base (drop=code_departement);
   set base;
run;
```

KEEP

```
data base;
set base;
keep Ville_A ;
run;
```

Modification d'une table SAS :

- Suppression d'un individus à l'aide de l'index :

```
/* suppression de la ligne 2 en fonction de l'index*/
data donnees
   set donnees;
   if _N_=2 then delete;
run;
```

Dans cet exemple, **_N_** spécifie l'index de la ligne qu'on veut supprimer.

- Ajout d'une observation à la fin d'un DATASET en spécifiant les colonnes:

```
data base;
set base end=eof;
output;
if eof then
do;
Ville_A='Paris';
code_departement=75;
output;
end;
run;
```

Données en sortie :

	Ville_A	code_departement
1	Lyon	69
2	Marseille	13
3	Lille	59
4	Paris	75

Les formats :

- Il existe des formats prédéfinis sous SAS :

Données « caractère »		Dates SAS (nombres de jours depuis le 01/01/1960)	
$2.	Ab	DDMMYY10.	24/11/2006
$UPCASE2.	AB	MMYYS7.	11/2006
Données numériques		YEAR4.	2006
5.	12345	FRADFWDX.	24 novembre 2006
7.2	1234.56	FRADFWKX.	vendredi 24 novembre 2006
NUMX7.2	1234,56	FRADFMN.	novembre
NLNUM10.2	1 234,56	FRADFDWN.	Vendredi

Exemple de création de données avec le format date:

```
Data table1;
input nom$ semestre$ a_naiss note1 note2;
format a_naiss DDMMYY10.; /*Précise que la variable a_naiss est de ce format*
informat a_naiss DDMMYY10.; /* pour afficher sous ce format*/
DATALINES;
Ricco S1 23.10.1953 7
Julien S2 12031967 2
Omar S1 29/06/1940 3
Fadila S1 24/09/1980 14
;
run;
```

- Il est aussi possible d'en créer via la PROC FORMAT :

```
PROC FORMAT;
    VALUE matières
    "Inf." ="Informatique"
    "Stat."="Statistique"
    ;
RUN;
```

- Utilisation :

```
DATA Notes;
    SET Notes;
    FORMAT
        UE matières.
        Moyenne 7.2
        ;
RUN;
```

Les conditions :

- IF … ELSE IF … ELSE :

A l'aide du code IF et THEN, on créé la variable reg à partir de dep. Cette nouvelle variable sera du type string.

```
DATA france;
    SET france;
    IF dep = 67 OR dep = 68 THEN reg = "Alsace";
    ELSE IF dep = 69 OR ... THEN reg = "Rhône-Alpes";
    ...
    ELSE reg = "Inconnu";
RUN;
```

- SELECT :

A l'aide du code SELECT et WHEN, on créé une variable Description en fonction de la valeur de la variable CSP.

```
DATA france;
    SET france;

    SELECT (CSP) ;
        WHEN(0) Description = "Agriculteurs exploitants";
        WHEN(1) Description = "Salariés de l'agriculture";
        WHEN(2) Description = "Patrons de l'industrie et du commerce";
        ...
        WHEN(7) Description = "Personnels de services";
        OTHERWISE Description = "Autres catégories";
    END;
RUN;
```

Les boucles :

L'étape DATA permet de créer une table nommée, puis la boucle permet de calculer à chaque itération la valeur des variables. La boucle DO WHILE est équivalente à une boucle while tandis que la boucle DO TO est équivalente à une boucle for.

- DO WHILE :

```
DATA A;
     a=0;
     i=1;
     DO WHILE(a<5);
          a=a+i;
          OUTPUT;
     END;
RUN;
```

L'option OUTPUT (ici insérée dans la boucle) permet de sortir la valeur des variables à chaque itération dans la table A.

Données en sortie :

	a	i
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1

- DO TO :

```
DATA B;
     DO i=0 TO 10 BY 1;
          b=i*10;
     END;
     OUTPUT;
RUN;
```

L'option OUTPUT (ici insérée à l'extérieur de la boucle) permet de sortir la valeur des variables après la dernière itération dans la table B.

Données en sortie :

	i	b
1	11	100

Sources et références :

- Sources :
  - https://support.sas.com/en/documentation.html
  - https://thesasreference.wordpress.com
  - https://od-datamining.com/knwbase/

- Références :
  - Fonctions : http://acp.coursinfostat.free.fr/DOC_SAS/SAS1/SAS1_Fonctions.htm