Préparation des données et statistiques élémentaires

Résumé

Tuteuriel de préparation des données. La syntaxe de l'étape Data et celle des principales procédures (`univariate, freq, tabulate, corr, rank, sort`*) de statistique élémentaire sont illustrées en suivant le déroulement classique d'une préparation et première exploration de données.*

Plan des tuteuriels :

- *Prise en main*
- *Préparation des données*
- *Graphiques*
- *Macros-commandes*
- *Bases de données*

La préparation des données (*data munging*) est une étape essentielle qui nécessite des outils efficaces donc parfois complexes. L'étape `data` offre un large éventail de possibilités avec une spécificité qui peut s'avérer importante pour des données massives : les données *ne sont pas* toutes stockées en mémoire mais lues et écrites séquentiellement par bloc (*buffer*).

1 Lecture des données

1.1 syntaxe

Création d'une table SAS par lecture d'un fichier.

Étape Data

data <sasuser.>table-sas ;
infile fileref ou 'nom-de-fichier' <dlm= 'carac' lrecl=nn> ;
input liste de variables et spécifications ;

L'exécution intègre implicitement une boucle : lire jusqu'à la fin le fichier désigné par `infile` ligne à ligne selon le format de `input` et écrire chaque ligne dans la table SAS temporaire ou permanente de la ligne `data`.

Options

La liste des variables définit chaque identificateur ; il est suivi du caractère "$" pour préciser le type éventuellement alphanumérique d'une variable de type chaîne de caractères ou encore qualitative.

- *dlm=';'* si le délimiteur entre les valeurs est un ';' ou '`09x`' pour un caractère de tabulation, par défaut un ou plusieurs espaces ;
- *dsd* deux délimiteurs successifs sont interprétés comme une valeur manquante, sinon insérer un caractère '.' ;
- *lrecl* si la ligne est très longue, majorant du nombre de caractères d'un enregistrement,
- *firstobs* numéro de la ligne à laquelle commencer la lecture,
- *obs* numéro de la dernière ligne à lire.

1.2 Exemple

Ce sont les mêmes données (variables) du tutoriel précédent mais elles sont lues à partir d'un fichier brut `statlab.dat` à télécharger. Le travail va donc consister à détailler les étapes qui ont permis de mettre en forme le fichier précédemment utilisé.

Visualiser le fichier dans un éditeur de texte. Repérer que les valeurs sont séparées par des espaces et que chaque individu ou observation est décrite sur 3 lignes. Exécuter les commandes ci-dessous d'importation après en avoir étudié en détail et commenté la syntaxe.

```
data sasuser.statlab;
infile "statlab.dat";
/* le "/" fait passer à la ligne la lecture */
input sexenf $ gsenf $ tenf_n penf_n tenf_10 penf_10 /
gsmere $ agem_n pmere_n consm_n $ tmere_10 pmere_10 /
consm_10 $ agep_n consp_n $ tpere_10 ppere_10 revf_n /
  revf_10;
run;
```

Visualiser la table obtenue dans l'explorateur. L'essentiel du travail consiste à

transformer les variables pour les rendre plus explicites : coder les modalités des variables qualitatives et préciser les unités de celles quantitatives.

1.3 Autres syntaxes

Lecture formatée

Le format libre ci-dessus n'est plus utilisable lorsque, pour des raisons d'économies d'espace disque, les données sont collées ; il faut alors indiquer explicitement, à la suite du nom de chaque variable, les positions (ou champs) concernés. C'est aussi nécessaire pour forcer la lecture de caractères "espace" dans des chaînes de caractères.

```
data sasuser.fich1;
    infile '~/data/fich.dat';
    input var1 1-12 var2 $ 13-18 var3 19-25;
run;
```

Importation

Le tutoriel précédent montrait un exemple d'importation de fichier au format de type excel `.csv` : première ligne avec les noms des variables, valeurs séparées par des "," et marque décimale le point. De manière réciproque, une table SAS peut être exportée dans un format donnée, texte (ascii) ou `.csv`.

```
proc import datafile="nom-de-fichier.csv"
      out=sasuser.table-sas dbms=dlm replace;
getname=yes;
datarow=2;
run;
/* traitements*/
proc export data= sasuser.table-sas
     outfile="nom-de-fichier2.xls"
     dbms=excel;
```

Le module SAS/Access autorise la lecture et l'écriture directes de fichiers au format propriétaire Excel mais ceci nécessite de posséder la licence Microsoft afférente sur l'ordinateur utilisé.

2 Statistiques élémentaires

Une première exploration permet de guider les transformations à venir, notamment les recodages des variables qualitatives.

2.1 Procédure `Univariate`

Regroupe toutes les statistiques unidimensionnelles des variables quantitatives et, en option, par classe d'une variable qualitative (`by`). La procédure `Means` ne diffère que par la présentation plus synthétique des résultats.

Syntaxe

proc univariate <options> ;
var liste de variables ;
by <descending> variable ;
weight variable ;
output <out=table sas> <liste de statistiques> ;

Options

La liste des options permet de préciser les résultats attendus.

- *data=table sas* indique le nom de la table par défaut, la dernière créée,
- *normal* pour obtenir des tests de normalité,
- *plot* pour obtenir les graphiques, si la commande *by* est employée, les boîtes sont affichées en parallèle,
- *vardef=* précise le diviseur dans le calcul de la variance (df, n, wdf, wgt).

Commandes

by suivi du nom d'une variable qualitative indique que les statistiques sont calculées par groupe d'observations ; la table doit être triée.

output indique le nom du fichier et la liste des statistiques qui y seront enregistrées.

var liste des variables concernées par la procédure, par défaut, toutes les variables quantitatives.

weight nom de la variable contenant les pondérations des observations.

Exemple

Tester les outils ci-dessous pour évaluer le plus efficace à la prise en compte d'un nombre important de variables.

```
proc univariate data=sasuser.statlab;
/* par défaut toutes les variables quantitatives */
run;
proc means data=sasuser.statlab;
/* par défaut toutes les variables quantitatives */
run;
```

Vérifier la cohérence des variables quantitatives

2.2 Procédure `freq`

Cette procédure concerne les variables qualitatives : effectif des modalités, tables de contingence simples et multiples, en option le test du (χ^2).

Syntaxe

proc freq <options> ;
by <descending> variable ;
tables liste des croisements requis </ options> ;
weight variable ;

Options

- *data=table sas* indique le nom de la table, par défaut, la dernière créée,
- *order=freq* édition ordonnée par effectifs décroissants,

Commandes

by suivi du nom d'une variable qualitative indique que les statistiques sont calculées par groupe d'observations ; la table doit être triée.

tables liste des croisements exprimés sous une des formes : a*b, a*(b c), (a b)*(c d), (a- -d)*c. Les options précisent les résultats et statistiques demandées ; la plus utile est `chisq` qui exécute un test du χ^2, d'autres permettent d'éviter certaines éditions (profils).

weight nom de la variable contenant les pondérations des observations.

Exemple

```
proc freq data=sasuser.statlab;
/* variables qualitatives */
table sexenf gsenf gsmere consm_n consm_10 consp_n;
run;
```

Analyser les effectifs des variables qualitatives. Noter que des groupes sanguins sont particulièrement rares de même que certaines classes de consommation de cigarettes.

3 Étape *data* de transformations

L'étape Data intègre un langage de programmation à la syntaxe spécifique [1] (`if, then, else, do`). Une variable de ce langage est une variable statistique de la table.

3.1 Syntaxe

data <sasuser.>table_out ;

set <sasuser.>table_in ;

... *instructions* ;

run ;

Chaque observation ou ligne de `table_in` est lue, transformée par exécution des `instructions` puis enregistrée sur `table_out`. Par défaut, toutes les variables de `table_in` sont considérées et recopiées sur `table_out` ainsi que celles qui ont été créées par les `instructions` mais il est possible d'en laisser tomber (`drop`) ou de n'en conserver (`keep`) que certaines.

Keep *et* Drop

Ces commandes peuvent apparaître comme des options des commandes `data`, `set` ou d'autres procédures :

data `table_out (drop=var1 var2)` ;

set `table_in (keep=var1 var2 var3)` ;

1. Calquée sur un langage (PL/1) des années 60 lancé sans succès par IBM

ou encore comme commandes d'une étape data :

`keep|drop var1 var2`

L'utilisation de `drop` ou `keep` dépend du nombre relatif de variables à éliminer par rapport au nombre à conserver.

Fonctions

Le langage reconnaît les expressions arithmétiques et la plupart des fonctions mathématiques usuelles (`round, sin, log, sqrt, ...`), les fonctions de gestion de chaînes de caractères (`lenght, scan, substr, ...`), celles spécifiques aux différentes lois de probabilités (quantiles) et d'autres à usage plus statistique (`sum, mean, min, max, var, std, ...`). Ces dernières s'appliquent à une liste de valeurs avec la syntaxe suivante :

sum `(var1,` **of** `var10-var20, var 25)`

Transformation des variables quantitatives

Il s'agit simplement de changer les échelles des variables de pouce en cm, d'onces en kg et de $ en ?.

```
data statlab1;
set sasuser.statlab;
ET0 = tenf_n*2.54;
EP0 = penf_n*0.4536;
ET10= tenf_10*2.54;
EP10= penf_10*0.4536;
MP0 = pmere_n*0.4536;
MT  = tmere_10*2.54;
MP10= pmere_10*0.4536;
PT  = tpere_10*2.54;
PP10= ppere_10*0.4536;
RF0 = revf_n*0.74;
RF10= revf_10*0.74;
MA0 = agem_n;
PA0 = agep_n;
drop sexenf--revf_10;
run;
```

3.2 Structures conditionnelles et variables qualitatives

Différentes syntaxes permettent de découper une variable quantitative en classes, renommer des modalités de façon explicite, en regrouper.

`if, then, else`

```
data statlab2;
set sasuser.statlab;
if sexenf="1.0" then ESx="M";
                else ESx="F";
keep ESx;
run;
```

Vérifier la table obtenue.

`select, when`

Certains groupes sanguins sont trop rares. Seule l'information concernant le rhésus de l'enfant et de la mère est prise en compte dans les regroupements. Commenter les deux types de syntaxe utilisés.

```
data statlab3;
set sasuser.statlab;
if gsenf = "1.0" or
   gsenf = "2.0" or
         gsenf = "3.0" or
         gsenf = "4.0" then ERh="Rh+";
                            else ERh="Rh-";
select ;
when (gsmere in ("1.0","2.0","3.0","4.0"))
                                  MRh="Rh+";
when (gsmere in ("5.0","6.0","7.0","8.0" "9.0"))
                                  MRh="Rh-";
otherwise MRh="inc";
end;
keep ERh MRh ;
run;
```

Vérifier la table ; dans quel cas la syntaxe avec un `IF` ne donnerait pas les mêmes résultats que celle utilisant *select* ?

Même opération avec les consommations de cigarettes.

```
data statlab4;
set sasuser.statlab;
select (consm_n);
when ("1.0") MCig0 = "0cig";
when ("2.0") MCig0 = "1-10cig";
when ("3.0") MCig0 = "1-10cig";
when ("4.0") MCig0 = ">10cig";
when ("5.0") MCig0 = ">10cig";
otherwise MCig0 = "inc";
end;
select (consm_10);
when ("1.0") MCig10 = "0cig";
when ("2.0") MCig10 = "1-10cig";
when ("3.0") MCig10 = "1-10cig";
when ("4.0") MCig10 = ">10cig";
when ("5.0") MCig10 = ">10cig";
otherwise MCig10 = "inc";
end;
select (consp_n);
when ("1.0") PCig0 = "0cig";
when ("2.0") PCig0 = "1-10cig";
when ("3.0") PCig0 = "1-10cig";
when ("4.0") PCig0 = ">10cig";
when ("5.0") PCig0 = ">10cig";
otherwise PCig0 = "inc";
end;
keep MCig0 MCig10 PCig0 ;
run;
```

Vérifier la table, quel problème dans le nombre de caractères ?

3.3 Sélection d'observations

Il peut être nécessaire de filtrer des observations, par exemple en éliminant celles présentant une donnée manquante codées en SAS par "." pour une variable fixée.

```
data sasuser.table1;
    set sasuser.table2 ;
    if var1 = '.' then delete;
run;
```

SAS passe à la suivante sans écrire dans la table en création. Ou encore par sélection implicite. Ou enocre

```
data sasuser.table1;
    set sasuser.table2 ;
    if var1 = 'bon' ;
/* sinon SAS passe a la suivante*/
    ...
run;
```

Des commandes spécifiques à l'étape Data ne sont pas décrites : `retain`, `return`, `put`, `output`, `missing`, `list`, `link`, `label`, `goto`, `do`, `array`.

(`array`) définit le vecteur de la ligne en cours et peut être indicé. Cette options est largement employée et illustrées dans les macros-commandes disponibles.

3.4 Autres transformations

Les exécutions suivantes ne sont pas indispensables sur les données étudiées, elles illustrent les possibilités d'autres procédures.

Procédure `rank`

Il est courant de devoir découper une variables quantitatives en classes pour la rendre qualitative. Le plus raisonnable et de construire des classes d'effectifs égaux et donc de choisir des bornes basées sur les quantiles de la distribution.

C'est obtenu par la procédure `rank` dont l'objectif initial est de remplacer les valeurs numériques par leur rang pour l'obtention de statistiques *robustes*.

```
proc rank data=statlab1 out=statlab5 groups=3;
var ET0 EP0 ET10 EP10;
run;
```

Vérifier le rôle de cette procédure sur le contenu de la table, même chose en retirant l'option `group=3`.

Procédure `standard`

Centrer et réduire les variables quantitatives est un grand classique. Il est pris en charge implicitement par les procédures statistiques (i.e. ACP) mais pas toujours.

```
proc standard data=statlab1 out=statlab5
   mean=0 std=1 print;
var ET0 EP0 ET10 EP10;
run;
```

Vérifier le rôle de cette procédure sur le contenu de la table.

Procédure `sort`

Ce n'est pas nécessaire non plus mais trier des tables selon une variable clef est indispensable avant une fusion de tables.

```
proc sort data=statlab1 out=statlab5;
   by EP0;
run;
```

Même chose sur le contenu de la table.

Bien d'autres procédures ne sont pas décrites : `transpose, score...`.

4 Concaténations de tables

4.1 Concaténation verticale et fusion

Cette opération consiste à compléter une table SAS par une ou plusieurs autres contenant les mêmes variables mesurées sur d'autres observations. Elle est très simple à réaliser, il suffit de mentionner toutes ces tables dans la commande `set` en renommant, éliminant, conservant éventuellement certaines variables. S'il n'y a pas une bonne correspondance entre les variables, des données manquantes sont générées.

```
data sasuser.concvtable;
    set sasuser.table1 (rename=(var1=var15))
        sasuser.table2 (rename=(var5=var15));
run;
```

Si chacune des tables est triée sur la même clé, ce tri peut être conservé dans la table concaténée, il s'agit alors d'une **fusion**, en spécifiant la variable clé dans une commande `by`.

```
data sasuser.fusiontable;
    set sasuser.table1 (rename=(var1=var15))
        sasuser.table2 (rename=(var5=var15));
    by vartri;
run;
```

4.2 Concaténation horizontale

Les mêmes unités statistiques ont été observées sur des paquets de variables contenues dans des tables SAS distinctes. La table regroupant toutes les variables est obtenue en utilisant plusieurs fois la commande `set`.

```
data sasuser.conchtable;
    set sasuser.table1;
    set sasuser.table2;
run;
```

Le même résultat peut être obtenu avec la commande `merge` qui réalise également une *jointure*. Elle permet, en plus, de contrôler la bonne correspondance des lignes de chaque table (triée) suivant les valeurs d'une clé et introduit, le cas échéant, des données manquantes.

```
data sasuser.mergetable;
```

```
    merge sasuser.table1 sasuser.table2;
    by varcom;
run;
```

4.3 Construction de la table résultante

Il suffit de concaténer horizontalement les tables ainsi crées afin de regrouper les variables transformées. Implicitement, l'ordre des observations est inchangé d'une table à l'autre.

```
/*  concaténation horizontale*/
data sasuser.statlabT;
set statlab1;
set statlab2;
set statlab3;
set statlab4;
run;
```

Vérifier la bonne construction de la table. Bien entendu, l'ordre des variables a été modifié par rapport à l'original mais cela n'est pas significatif.

4.4 Exportation

Il s'agit maintenant de construire le fichier dans un format de tableur qui pourra être ainsi transféré vers d'autres logiciels comme Excel ou R.

```
/*  exportation au format csv */
proc export data = sasuser.statlabT
   outfile="statlab3.csv"
         dbms=csv replace;
        putnames=yes;
run;
```

Vérifier le contenu du fichier dans un éditeur. Attention, il ne sera pas lu tel quel dans une version française d'Excel avec "," comme séparateur !

5 D'autres statistiques élémentaires

Quelques résultats et statistiques complémentaires.

5.1 procédure `tabulate`

La procédure tabulate est complexe à utiliser avec une syntaxe spécifique. Elle permet de créer des *tableaux de bord* (reporting) synthétiques qui compilent, croisent récapitulent les données.

Syntaxe

proc tabulate <options> ;
class <liste$_1$ > ;
var <liste$_2$ > ;
table ... ;
run ;

Options

Chaque ligne de commande a un rôle très spécifique en fonction de la syntaxe et de la ponctuation.

- Les variables qualitatives contenues dans la liste 1 servent à définir des groupes d'observations sur lesquels des statistiques seront calculées. Ce sont ces variables qui définiront les lignes et les colonnes du tableau à calculer.
- Les variables contenues dans la liste 2 doivent nécessairement être numériques. C'est sur ces dernières que l'on pourra effectuer des opérations.
- l'instruction table permet de préciser et définir l'architecture du tableau. En particulier, il est possible de concaténer, croiser ou encore regrouper des catégories.

la virgule marque la limite entre les lignes et les colonnes, un espace indique une juxtaposition de deux éléments dans une même dimension (ligne ou colonnes) et l'étoile signifie que l'on imbrique deux éléments dans une même dimension. Typiquement, on utilisera donc la syntaxe suivante :

```
table (lignes) , (colonnes * cellules);
```

Entre les parenthèses de la dimension `ligne`, on trouve des noms de variables citées dans class. Il en va de même entre les parenthèses de la dimension colonne.

Pour les cellules, on retrouve soit une statistique comme une fréquence ou un pourcentage, soit une variable de calcul (déjà citée dans var) suivie d'une

étoile et de sa statistique.

Exemples

```
proc  tabulate data = sasuser.statlabT;
var EP0 EP10 ET0 ET10;
class ESx ERh MRh MCig0 MCig10 PCig0;
table ESx, EP0*mean ET0*mean;
table MCig0, ESx*(EP0*mean EP10*mean);
run;
```

5.2 Procédure `univariate`

D'autres résultats de la procédure `univariate`.

```
proc  univariate data = sasuser.statlabT normal;
  histogram / normal (color=red)
  kernel(c = 0.25 0.50 0.75 1.00
            l = 1 20 2 34 noprint);
run;
```

Attention au volume des résultats produits, il ne s'agit pas de faire exploser celui d'un rapport. Seuls les résultats pertinents en fonction de l'objectif doivent être extraits.

5.3 Procédure `freq`

Même chose pour les variables qualitatives et les croisements de variables qualitatives.

```
proc freq data = sasuser.statlabT ;
table ESx ERh MRh MCig0 MCig10 PCig0;
table (ESx ERh MRh MCig0 MCig10 PCig0) *
   (ESx ERh MRh MCig0 MCig10 PCig0)/
         chisq plots=freqplot;
run;
```

Noter les croisements aboutissant à un test significatif. Dommage mais cette procédure ne semble pas pourvoir produire un `mosaic plot` des vecteurs profiles.

5.4 Procédure `corr`

Étude élémentaire de la structure de corrélation des variables. Les limitations imposées par l'exécution de la procédure nécessite d'aller consulter la documentation afin de pouvoir obtenir les résultats.

```
proc corr data = sasuser.statlabT
 plots (MAXPOINTS=20000) =matrix (hist nvar=all );
run;
```

Le nombre de variables est limité mais il est possible de considérer une matrice de nuages rectangulaires pour tous les représenter.

```
proc corr data = sasuser.statlabT
 plots=matrix (hist nvar=all nwith=all);
 var ET0 EP0 ET10 EP10 MP0 MT ;
 with MP10 PT PP10 RF0 RF10 MA0 PA0;
run;
```

6 Documentation

L'apprentissage de SAS passe également par celui de la structure extrêmement complexe de la documentation en ligne. Celle papier occupait plusieurs mètres sur une étagère. Devenue en ligne en format numérique, le volume physique est restreint mais la complexité ne fait qu'augmenter.

Chercher l'aide sur la procédure `corr`.

`Aide > Aide SAS et Documentation`

Bien sûr, un index et un onglet de recherche par mot clef sont accessibles ; y entrer `corr` n'aide finalement pas beaucoup.

A l'usage, le plus simple est d'ouvrir la page

`SAS Products > SAS Procédures`

pour rechercher la bonne procédure parmi celle ordonnées alphabétiquement. Apprécier le nombre considérables de procédures proposées. Attention, toutes ne sont pas accessibles, cela dépend de la licence souscrite et donc de l'accès ou non à certains modules. De cette façon, la documentation des procédures est facilement accessible et très complète, notamment à propos de la

bibliographie sur les méthodes utilisées. La documentation concernant la syntaxe de l'étape `Data` est elle plus obscure.

Trouver la procédure `corr` et repérer la syntaxe des options et commandes qui ont finalement été sélectionnées.

Bien sûr, pour procéder de la sorte il faut connaître *a priori* la procédure que l'on veut utiliser. Une initiation élémentaire est donc nécessaire ; c'est l'objectif de ce cours mais, attention, une infime partie des capacités du logiciel et de ses modules est couverte.