

Les tableaux de données en statistique

Eric Lecoutre

1^{er} août 2001

1 Les tableaux de données

En statistique appliquée, pas question de parler d'analyse sans disposer de données, alimentation quotidienne du statisticien. Ce dernier a d'ailleurs, peut-être à juste titre, la réputation de passer sa vie devant des pages couvertes de symboles barbares, sigma ou autres, et de chiffres. Ce document regroupe quelques notes ayant trait au traitement informatique des tableaux de données.

Pour satisfaire à la coutume, nous allons donc introduire ici un tableau (T) de chiffres :

23	21
45	56
36	30
22	25
16	23
5	14

Exercice 1 *Effectuer une analyse statistique du tableau T.*

L'exercice ne sera pas corrigé ici. Vous aurez même sûrement des difficultés à l'effectuer ! Et pourquoi ? Car un tableau statistique est **beaucoup plus** qu'un tableau de chiffres comme le tableau (T). Ainsi, vous devez en priorité connaître le contexte des données : comment ont-elles été produites, par qui et pour quoi ?

Le tableau (T) contient des informations relevées par un entomologiste lors d'une étude sur les fourmis. Avec cette information, pouvez-vous effectuer votre analyse ? En quoi cette information est-elle utile ?

Il manque encore des informations sur l'interprétation des chiffres eux-même. La plupart du temps, cette information consiste en une description des **variables** introduites lors de l'étude et de la **structure** du tableau (comment sont liées les variables). Qu'étudie-t-on ? Voici quatre scénarios différents :

Scénario 1 *Un entomologiste étudie le comportement des fourmis qui sortent d'une fourmilière. Il a observé 6 fourmis différentes qui se sont aventurées dehors (les 6 lignes du tableau). Pour ces 6 fourmis, il estime la distance parcourue en cm en dehors de la fourmilière (première colonne de chiffres), ainsi que la quantité de nourriture échangée lors de trophallaxies (deuxième colonne). Ce scénario correspond au tableau statistique «classique», où l'on trouve les individus étudiés en ligne et les variables d'intérêt en colonnes.*

Scénario 2 *Un entomologiste étudie le comportement des fourmis qui sortent d'une fourmilière. Il a attendu d'avoir 6 fourmis sorties. Durant 5 minutes, il mesure la distance parcourue (en cm). Au bout de 5 minutes, il augmente la température de 3°C en plaçant une lampe près de la fourmilière. Au cours des 5 minutes suivantes, il observe à nouveau la distance parcourue par les mêmes 6 fourmis. Ce scénario correspond à un tableau statistique très proche du précédent, mais avec la particularité de disposer de mesures répétées.*

Scénario 3 *Un entomologiste étudie le comportement des fourmis de 6 petites fourmilières très proches (variable qualitative : la fourmilière d'origine). Il différencie les fourmis qui sortent de la fourmilière de celles qui restent perpétuellement à l'intérieur (variable dichotomique, ou binaire). Il essaye de dénombrer le nombre total de fourmis et range chaque individu dans une catégorie formée par le croisement des deux variables. Ce scénario correspond au tableau statistique particulier qu'est la table de contingence. Dans ce tableau, les individus sont «à l'intérieur», alors que les variables sont «à l'extérieur».*

Scénario 4 *Un entomologiste étudie le comportement des fourmis de 2 fourmilières. Il a placé des petits morceaux de sucre à égales distances des deux entrées principales et essaye de comparer les performances des deux fourmilières. Pour ce faire, il attend que 3 fourmis de chaque fourmilière aient rappatrié un morceau de sucre jusqu'à leur entrée principale. Les trois premières fourmis sont de la première fourmilière. La première variable est la distance parcourue par la fourmi (en cm), la deuxième une mesure de la taille du morceau de sucre. Ce scénario correspond à nouveau à un tableau statistique «classique». Y manque la troisième variable «cachée», le facteur groupe 'fourmilière', pourtant nécessaire au statisticien puisque l'entomologiste souhaite comparer les performances des fourmilières.*

Exercice 2 *Imaginer un autre scénario. En quoi le scénario aide-t-il pour une analyse ?*

Usuellement, le jeu de données est fourni par une personne ou un organisme, dont vous obtenez aussi le «scénario». Ne jamais hésiter à poser des questions et demander un supplément d'information.

En résumé, la connaissance d'un «scénario» développé apporte :

- Le contexte de l'étude, qui conditionnera l'interprétation des résultats des analyses. Tous les milieux n'accordent pas la même importance aux p -valeurs : certains tests de comparaison de chaînes d'ADN en statistique génétique demandent des p -valeurs de l'ordre de 10^{-30} pour conclure...
- L'origine des données : comment ont-elles été obtenues ? Si l'on cherche à mesurer le niveau des étudiants de l'UCL au moyen d'un échantillonnage, il est important de savoir quels étudiants ont été interrogés. Si l'échantillon ne comporte que des ingénieurs, ou que des romanistes, il est clair que les résultats ne seront pas représentatifs de l'ensemble de l'UCL.
- La structure du tableau, indispensable avant toute analyse : type de tableau, variables en jeu, natures de variables (qualitatives ou quantitatives, facteurs de groupe ou non, endogène ou exogène). Pour un statisticien, la structure permet d'imaginer des questions possibles et d'orienter la recherche d'outils adaptés. Ainsi, si le tableau comprend une variable facteur à 2 modalités, on va déjà penser aux tests de comparaison de groupes (t de Student ou autre). Si le tableau est une table de contingence, on pensera immédiatement au test du χ^2 , à l'analyse des correspondances et au modèle loglinéaire.
- Enfin, et ce n'est pas le moindre, la question du «client» : si les données sont là, que souhaite-t-on en faire ? Pourquoi les a-t-on récoltées ? Dans certains des scénarios précédemment fournis, la question n'apparaît pas de manière précise. Un flou sur le but du client peut entraîner une mauvaise analyse. La compréhension des attentes du client ainsi que des possibilités (ou limitations...) du statisticien doit passer par une phase de communication.

Exercice 3 *Développer l'un des scénarios en ajoutant une ou plusieurs questions de la part d'un client. Dresser l'esquisse d'une analyse statistique permettant de répondre à la question (quel(s) outil(s) utiliser ?).*

2 Codage informatique des données

La structure des données, telle que décrite dans la section précédente, n'est finalement qu'une question de convention. Nul doute que cette structure pourrait être implicite avec un peu plus de précisions sur les côtés du tableau (en mettant les noms des variables en haut des colonnes par exemple).

Le codage informatique des données relève aussi de la convention. La structure que l'on souhaite donner aux chiffres doit pouvoir être compréhensible et exploitable par un programme.

Au niveau du contenu (information), sans structure, le tableau (T) n'est qu'une succession de 23 chiffres :

$$(T) \iff 23214556363022251623514$$

laquelle n'est pas très explicite et a été obtenue en adoptant une première convention : les chiffres sont lus prioritairement en horizontal.

La première étape, facile à gérer, est la différenciation des individus (lignes des tableaux statistiques «classiques»). Pour faciliter la lecture, la convention usuelle est d'utiliser le retour chariot (entrée) pour séparer les individus : tout comme sur une feuille, les individus sont sur des lignes distinctes.

Le tableau (T) devient alors :

```
2321 ¶
4556 ¶
3630 ¶
2225 ¶
1623 ¶
514 ¶
```

Noter que le caractère retour chariot (¶) n'est habituellement pas visible à l'écran (caractère non imprimable), mais est présent dans le fichier et occupe physiquement la même place que tout autre caractère, soit un octet (groupe de 8 bits : 8 chiffres binaires). Au niveau de la programmation, la lecture du fichier est maintenant plus facile : le fichier sera lu caractère par caractère, avec une petite fenêtre qui se déplace dans le fichier, jusqu'à ce que le contenu de la fenêtre soit le caractère retour chariot. On sait que l'on doit passer à la ligne et travailler avec un nouvel individu. Les individus peuvent être distingués au moyen d'un compteur qui sera alors incrémenté, pour obtenir l'algorithme :

Ouvrir le fichier

(le fichier est placé en mémoire vive et est prêt à être lu)

Initialiser une variable de comptage (individu) à 1

Répéter jusqu'à la fin du fichier

| Répéter jusqu'à la fin de la ligne (repérée par un retour chariot)

| | Lire les données de l'individu en cours

| Incrémenter le numéro d'individu

Fermer le fichier

Si l'information est déjà plus lisible, elle n'est pas encore exploitable : on sait séparer l'information du premier individu de celle du deuxième, mais on ne sait pas quelle est exactement les données de chaque individu. Comment décomposer les quatre chiffres 2321 du premier individu ? S'agit-il d'une mesure, de quatre variables qualitatives ou des deux valeurs 2 et 321 ?

La deuxième étape est donc de spécifier l'emplacement des observations précisément pour chaque individu. Pour ce faire, il existe plusieurs possibilités. Deux méthodes sont principalement utilisées : le codage avec séparateur et le codage à taille fixe.

2.1 Codage avec séparateur

Pour séparer les individus, un caractère spécial (le retour chariot ¶) est utilisé. Selon le même principe, on adopte la convention de séparer les observations (variables) par un caractère particulier, lequel peut être l'espace, la tabulation (caractère spécial) ou tout autre caractère. La virgule (,), le point-virgule (;) ou le dollar (\$) sont souvent rencontrés, mais rien n'empêche d'utiliser une lettre si les données sont toutes numériques.

Le tableau (T) devient alors :

```
23 21¶
45 56¶
36 30¶
22 25¶
16 23¶
5 14¶
```

ou encore

```
23$21¶
45$56¶
36$30¶
22$25¶
16$23¶
5$14 ¶
```

L'algorithme de lecture est ajusté en ajoutant un contrôle de flux lors de la lecture d'une ligne de type «Répéter... jusqu'à ce que...» : on utilise un compteur de variable et l'on «remplit» les variables une à une en lisant les données jusqu'à rencontrer le séparateur (à préciser en entrée de l'algorithme).

Le codage avec séparateur a comme avantage indéniable la simplicité de l'algorithme de lecture. De plus, il est très peu contraignant et ne limite pas la taille des observations : pour une variable texte, une observation peut être composée d'autant de caractères que l'on veut, pourvu que l'on place un séparateur pour en marquer la fin. En revanche, un fichier avec séparateur n'est pas toujours très lisible, comme l'illustre l'extrait de fichier suivant, portant sur quelques villes belges, et utilisant le caractère § comme séparateur :

```
Bruxelles§5786290§65.20§Grande§2
Louvain-la-Neuve§42351§95.30§Petite§1
Grez-Doiceau§10548§41.95§Petite§3
```

```
Court-St-Etienne§5952§84.20§Petite§4
Mons§241030§65.25§Grande§1
```

Exercice 4 *En se basant sur l'algorithme fourni précédemment, écrire un algorithme développé pour lire les fichiers à séparateur.*

2.2 Codage fixe

Dans le codage à taille fixe, chaque variable occupe un nombre de caractères fixe, à préciser. Chaque variable est donc limitée à un certain nombre de caractères, qui seront tous utilisés (éventuellement remplis avec des espaces, un caractère «nul» au niveau de l'information).

Pour le tableau (T), chaque variable nécessite deux caractères (nombres à 2 chiffres). Le tableau (T), en codage fixe, donne donc :

```
23·21 ¶
45·56 ¶
36·30 ¶
22·25 ¶
16·23 ¶
·5·14 ¶
```

les espaces étant repérés par un · (l'espace est un caractère non imprimable).

Si ici l'on voit peu la différence avec le même tableau codé avec l'espace comme séparateur, la différence est plus flagrante avec l'extrait sur les villes :

```
Bruxelles      5786290 65.20 Grande  2
Louvain-la-Neuve 42351   95.30 Moyenne  1
Grez-Doiceau   10548   41.95 Petite   3
Court-St-Etienne 5952    84.20 Petite   4
Mons           241030 65.25 Grande   1
*****
|      V1 16      | V2  7 | V3 5| V4 7 | V5 1
```

Les deux dernières lignes ne sont pas présentes dans le fichier mais sont rajoutées ici pour bien comprendre le processus d'encodage : la première variable (nom de la ville) est codée sur 16 caractères. Toutes les observations utilisent ces 16 caractères, en complétant les noms courts par autant d'espaces que nécessaire.

Un grand avantage pour le fichier à taille fixe est sa grande lisibilité. Un simple éditeur de texte permet une première exploration du contenu. Cependant, pour de gros volumes de données, son emploi est déconseillé car il occupe plus de place physique (ne pas oublier que chaque espace compte pour un caractère). De plus, sa lecture est plus

pénible, puisqu'un algorithme de lecture nécessite en entrée un paramètre par variable, à comparer avec l'unique paramètre de l'algorithme de lecture des fichiers à séparateur.

2.2.1 Fichiers texte et éditeur de texte

Les fichiers de données au format texte encodés en taille fixe ou avec des séparateurs portent la plupart du temps l'**extension** TXT ou DAT (pour data). L'extension d'un fichier - le groupe de trois lettres à la fin du nom du fichier - sert à déterminer son **format**, soit le procédé d'écriture employé lors de l'écriture physique. Une extension TXT désigne en principe toujours un fichier au format texte, généré par un éditeur de texte et directement lisible avec de tous petits logiciels tel le notepad de Windows. Les tableaux créés avec Excel portent l'extension XLS et ne sont lisibles que sous Excel (vous pouvez essayer d'ouvrir un fichier XLS avec un éditeur de texte : rien n'est lisible et l'écran sera rempli de caractères bizarres non compris par votre éditeur). Le logiciel SPSS, lui, utilise l'extension SAV. De même, il s'agit là d'un format propriétaire, directement lisible uniquement sous SPSS. L'extension servira donc à savoir quel logiciel exécuter pour ouvrir vos données. Un bon réflexe, lorsque vous êtes en présence d'un fichier portant l'extension TXT ou DAT est de l'ouvrir tout de suite avec un éditeur de texte et de déterminer s'il s'agit d'un encodage avec séparateur (lequel ?) ou à taille fixe.

2.2.2 Les données manquantes

Le traitement des données manquantes nécessite encore d'adopter une convention. Hélas, ici, rien d'universel n'existe encore et tout est permis. Le procédé le plus simple consiste à utiliser l'espace. Dans un fichier avec séparateur, on pourra gagner de l'espace en n'utilisant aucun caractère ! Le séparateur utilisé est alors répété, ce qui ne rend pas la lecture très claire lorsque le séparateur en question est l'espace.

Lorsqu'un caractère est utilisé, on apprécie le point (.), conforme avec la notation interne dans le logiciel SAS et proposé en option dans la plupart des logiciels de statistique.

Il est déconseillé d'utiliser un chiffre et notamment le 0. Le chiffre pourrait être considéré comme une valeur observée et fausserait toutes les analyses (moyennes, écart-types etc). Pour les variables qualitatives ayant peu de modalités (souvent le cas lors d'enquêtes), beaucoup de logiciels utilisent toutefois le chiffre 9 ou une valeur négative comme -1. Penser à y faire attention si vous voulez analyser le fichier sous un logiciel générique tel que SPlus.

2.2.3 Rôle de la première ligne

Dans les fichiers texte, la première ligne peut jouer un rôle particulier et comporter le nom des variables. Lors de la lecture des données, l'algorithme de lecture doit alors en

tenir compte et n'initialiser le compteur des individus qu'à partir de la seconde ligne.

Si l'on souhaite incorporer une telle première ligne dans un fichier de données, on limitera le nom des variables à 8 caractères, afin d'assurer une compatibilité maximale avec les logiciels de statistique. Dans les fichiers à codage fixe, on aura de toutes façons intérêt à utiliser des noms les plus courts possibles, puisque le nombre de caractères utilisés conditionnera souvent le nombre de caractères à utiliser pour coder la variable.

Exercice 5 *Expliciter le principe de l'algorithme de lecture d'un fichier TXT quelconque. Peut-on se passer de l'utilisateur et déterminer automatiquement le type de codage utiliser ? Expliquer.*

2.3 Autres codages

Les codages avec séparateur ou à taille fixe sont les codages les plus fréquemment rencontrés, mais il peut arriver que vous rencontriez un codage un peu plus exotique.

Le logiciel DidaStat3, par exemple, utilise un découpage de l'information basé sur la convention suivante :

Première ligne : nom du tableau

Deuxième ligne : nombre d'individus, espace, nombre de variables.

Troisième ligne : type de tableau (1 : tableau «classique», 2 : mesures répétées, 3 : table de contingence). Pour une table de contingence, nombre de modalités des variables, séparées par des espaces. Pour les mesures répétées, nombre de répétitions.

Noms des variables, retour à la ligne entre chaque nom. Puis les observations, idem.

Le fichier au format DidaStat3 correspondant au scénario 3 serait le suivant :

```
FOURMIS
316 2
3 6 2
ORIGINE
TYPE
23
45
36
22
16
5
21
56
30
```

25

23

14

Exercice 6 *Imaginer de nouvelles règles d'encodage, donner l'allure du tableau (T) dans votre format, ainsi que l'algorithme de lecture correspondant.*

L'algorithmique est très importante, car si vous avez affaire à un format exotique comme le format **DS3**, aucun logiciel de statistique standard ne saura lire vos données. Vous devrez alors utiliser un logiciel générique qui propose un environnement de programmation avancée tel SAS ou SPlus et écrire votre fonction d'importation des données.

2.4 Pourquoi utiliser des formats propriétaires ?

La plupart des logiciels de statistique utilisent leur format de données (datasets SAS avec l'extension **SD2**, fichiers SPSS avec l'extension **SAV** etc) et on peut se demander pourquoi. L'explication est simple : chaque programmeur a son idée de ce que peut être un bon codage optimal des données.

Considérons un fichier texte encodé à taille fixe. Pour des grands tableaux (beaucoup d'individus et/ou de variables), il est évident qu'il y a un gaspillage de place physique due aux espaces fréquemment rencontrés. La place occupée devient même un problème important lorsqu'il s'agit d'effectuer des analyses sur le tableau, puisque pour profiter de temps de calcul plus rapides, l'entièreté du tableau doit être placée en mémoire vive (la RAM de l'ordinateur).

3 Lecture des données, importation

3.1 Lecture dans un éditeur de texte

La figure 1 est une capture d'écran lors de l'ouverture d'un fichier dans l'éditeur de texte Textpad. Comme dans tout logiciel respectant les standards Microsoft, l'ouverture d'un fichier s'effectue par le menu File...Open (Fichier...Ouvrir). La boîte de dialogue qui apparaît est alors standard et s'utilise de la manière suivante : dans la partie supérieure se trouve les éléments de navigation qui permettent de changer le répertoire courant (voir la section suivante pour plus de détails sur l'organisation des données grâce aux répertoires). Au milieu sont recensés les fichiers du répertoire courant conformes au filtre spécifié dans la partie inférieure. Le filtre le plus utilisé est le type de fichier. Ici, seuls les fichiers portant l'extension **TXT** ou **INI** sont affichés. Si votre fichier porte une extension qui n'est pas dans la liste proposée (**DAT** pour data par exemple), vous pouvez l'ouvrir en demandant l'affichage de tous les fichiers (*.*).

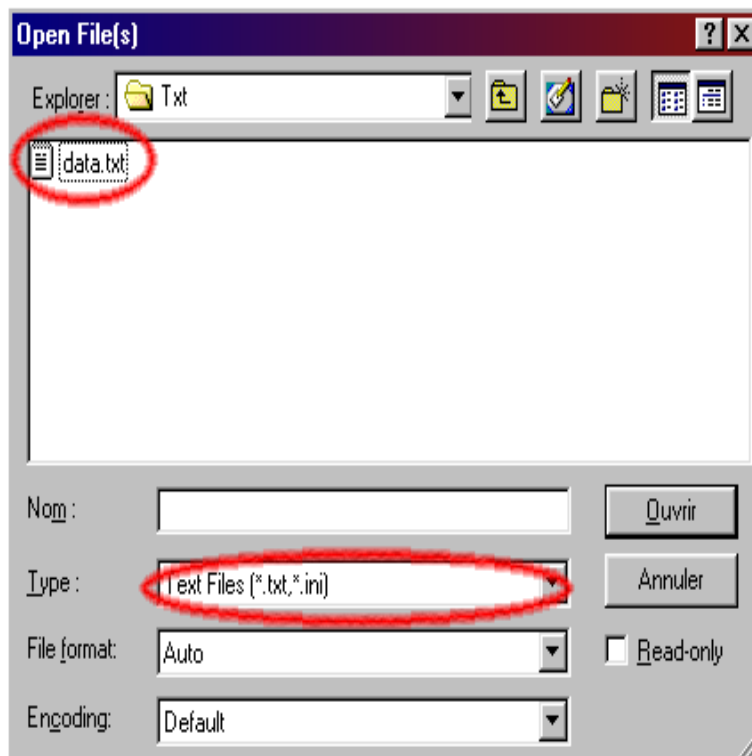


FIG. 1 – Ouverture d'un fichier dans un éditeur

3.2 Importation dans Excel

Le tableur Excel dispose de son format de fichier (extension XLS). Les fichiers dans ce dernier format ne sont pas lisibles dans un éditeur de texte. Ils comprennent des informations supplémentaires aux données : tailles des cellules, couleurs, polices utilisées, bordures, formules, noms des plages etc. Aussi pour mettre en forme avec Excel vos données en profitant de tous ces agréments, vous devrez d'abord *importer* votre fichier de données.

Grâce à la section précédente, l'importation ne devrait pas être trop difficile. Excel dispose en effet d'un assistant d'importation, qui vous guidera au travers plusieurs étapes.

Ainsi, la première étape est le choix du type d'encodage, parmi les deux grands types déjà rencontrés : taille fixe ou à séparateur.

La figure 2 montre la deuxième étape de l'importation d'un fichier avec séparateur : le choix du ou des caractères délimiteurs.

Dès l'importation finie, ne pas oublier d'enregistrer les données au format Excel.

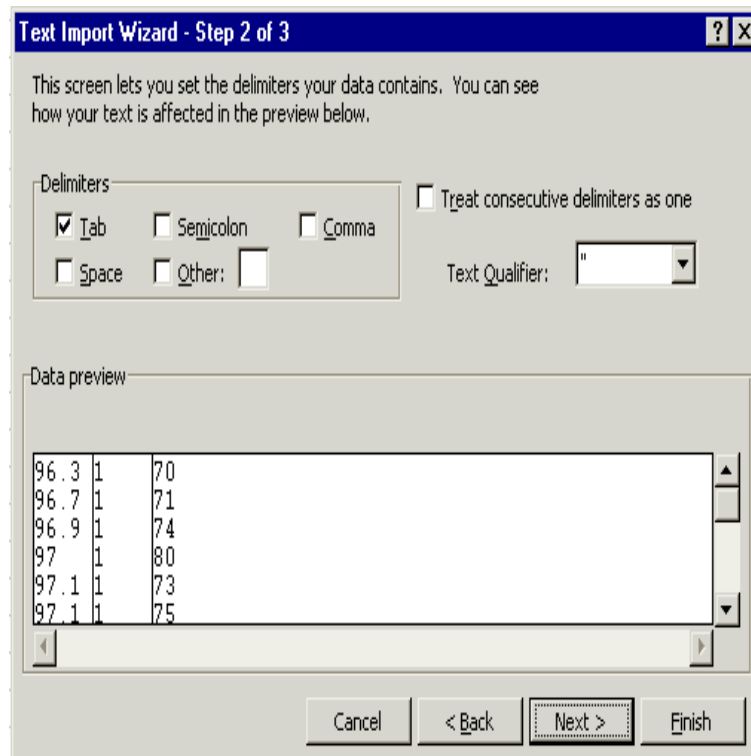


FIG. 2 – Excel : choix du séparateur (seconde étape de l'importation)

3.3 Importation dans SAS

Le logiciel SAS dispose lui aussi d'un assistant pour importer les données (File... Import). Cependant, le logiciel est tout à son avantage lors d'une importation grâce au code. En effet, si même l'étape d'importation est implémentée en code SAS, vous pouvez lancer l'analyse sur un autre fichier avec un minimum d'effort. Grâce au macro-langage SAS, des compagnies peuvent aussi traiter les données de tous les jours de manière automatique.

La manipulation des données en SAS passe par un bloc DATA. Les options lors d'une importation sont très nombreuses et permettent de lire des fichiers exotiques (un individu réparti sur plusieurs lignes ou plusieurs individus par ligne par exemple).

Le principe général est le suivant : SAS se sert d'un buffer (petite mémoire) intermédiaire lors de la lecture du fichier. Une partie du fichier est lue (souvent une ligne), placée dans le buffer, traité dans ce buffer selon les instructions du bloc DATA et une fois analysée est écrite dans le dataset.

L'objectif de ce document n'étant pas de donner un cours de programmation SAS, nous nous contenterons de disséquer un code SAS déjà écrit pour importer un fichier très particulier, le fichier LOG d'un serveur Windows NT, dont voici les premières lignes :

```
02/03/99;14:16:17;Print;Information;None;10;sdekeyser;DIDAC01;4356984;1
02/03/99;14:17:26;Print;Information;None;10;sdekeyser;DIDAC01;4355904;1
02/03/99;14:17:28;Print;Information;None;10;nbakanibona;DIDAC01;38897;1
03/03/99;15:08:43;Wins;Error;None;4202;N/A;DIDAC01;An attempt to connect
03/03/99;15:14:09;Wins;Error;None;4202;N/A;DIDAC01;An attempt to connect
```

Ce fichier LOG recense les événements d'impression reçus par le serveur : requêtes par utilisateur, nom du fichier, nombre de page, taille en octets etc. Dans certains cas, il y a une erreur (si l'impression est annulée par exemple). Nous ne nous intéressons qu'aux événements de type 10 (6ème information) qui correspondent à des impressions effectivement réalisées, ceci permettant de mieux contrôler l'utilisation de l'imprimante.

L'importation d'un tel fichier dans SAS peut être réalisée par le code suivant :

```
LIBNAME PROJET 'Z:\SAS';
FILENAME ficdata 'Z:\DATA\log.dat';

DATA PROJET.LOG(DROP=HEURE TEMP1 TEMP2 SERVEUR TYPE ID);
  INFILE ficdata DELIMITER=';' MISSOVER;
  INPUT JOUR DDMYY8. HEURE $ TYPE $ TEMP1 $ TEMP2 $ ID @;
  IF (ID=10 AND TYPE='Print') THEN
    DO;
      INPUT NOM $ SERVEUR $ TAILLE PAGES;
      OUTPUT;
    END;
RUN;
```

Le principe de lecture est le suivant : on crée d'abord la librairie de travail PROJET qui assigne le répertoire Z:\SAS, ainsi qu'un lien vers le fichier de données. A l'intérieur du bloc DATA, on spécifie que le traitement est à effectuer depuis un fichier extérieur au moyen de l'instruction INFILE. Noter la spécification de l'option DELIMITER. Une première partie de la ligne est placée dans le buffer avec l'instruction INPUT. Le symbole @ placé en fin d'instruction précise que l'on doit laisser le curseur de déplacement (lecture) à l'endroit où il est : la ligne n'est pas fini, la prochaine lecture de la ligne continuera ici. Seules les 6 premières variables nous intéressent en effet pour l'instant, la 6ème déterminant si le reste est pertinent ou non. Un test est effectué sur la valeur de cette 6ème variable est effectué, si la valeur 10 est rencontrée (impression correctement réalisée), le reste de la ligne est une observation. On place le reste de la ligne dans le buffer (INPUT), remplaçant par cela le buffer précédent, et on écrit le contenu du buffer dans le dataset (instruction OUTPUT).

Le nombre d'instructions à utiliser est relativement limité. La lecture d'un tel fichier dans un autre logiciel de statistique est difficile. Enfin, si l'on souhaite traiter un deuxième fichier provenant d'un autre serveur, seule la deuxième instruction est à changer !

3.4 Importation dans SPlus

La procédure doit maintenant devenir assez habituelle : l'importation de données passe par la gestion d'un fichier et se trouve donc dans le menu File (File... Import data... From file). La boîte de dialogue qui apparaît est assez complexe et se répartit en trois onglets. Le premier onglet (data specs) permet de choisir le type de fichier (noter le grand nombre de formats reconnus par SPlus : Excel, Matlab, Gauss, SPSS, Stata, SAS,...), ainsi que le chemin et le nom du fichier.

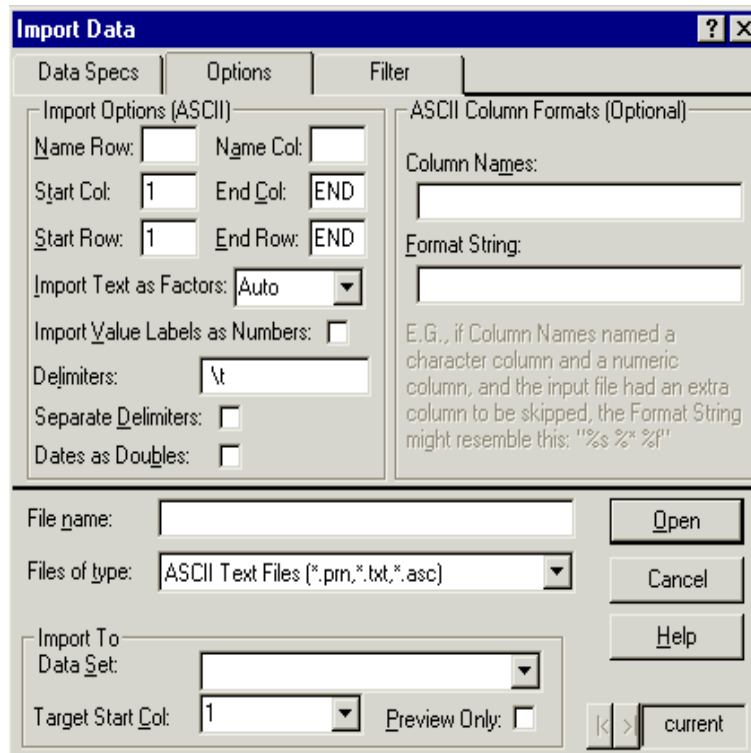


FIG. 3 – SPlus : l'onglet option de la boîte de dialogue importation)

Le deuxième onglet (Options, figure 3) recense toutes les options d'importation : le delimitateur pour les fichiers encodés avec séparateur (la valeur par défaut est «\t», correspondant à une tabulation). Si votre fichier comprend le nom des variables, vous devez

en indiquant la localisation (quelle ligne ?) dans la zone «Name Col». Enfin, vous pouvez spécifier un nom précis au jeu de données pour son utilisation en tant que dataframe dans SPlus (zone : «Import to dataset»).

4 Organisation des données

4.1 Les répertoires DOS

Sur un ordinateur, il ne se trouve pas que les fichiers de données de vos analyses... Le système d'exploitation, indispensable pour communiquer avec l'ordinateur, est lui aussi composé de fichiers. Windows NT, par exemple, est constitué d'environ 4000 fichiers, dont la plupart ne sont pas lisibles par l'utilisateur (extensions SYS pour des fichiers système, DRV pour des drivers, passerelles entre le matériel et le système d'exploitation, EXE pour des programmes, les exécutables, etc.).

On comprend donc l'intérêt de pouvoir **ranger** ces fichiers. C'est là le rôle des **répertoires**. Pour un disque dur, le répertoire est l'équivalent des intercalaires d'un classeur.

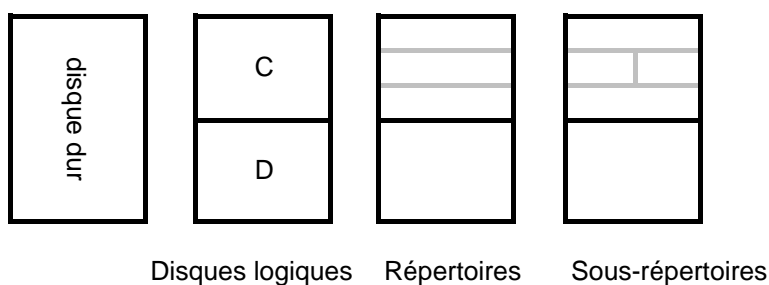


FIG. 4 – Découpage du disque dur

Revenons un peu en arrière pour étudier la syntaxe employée dans les premiers systèmes d'exploitation. Lorsqu'on utilisait un support externe pour ranger ou stocker l'information (carte perforée, bande magnétique ou disquette), il fallait un moyen de distinguer ce support du disque dur interne de l'ordinateur.

Chaque support s'est vu attribué une lettre. Un ordinateur est prévu pour contenir jusqu'à deux lecteurs de disquette, dont les lettres réservées sont A et B. Le disque dur principal, où doit se trouver en principe le système d'exploitation, a donc la lettre C.

Le système d'exploitation DOS, *Disk Operating System*, comme son nom l'indique, est une interface utilisée pour gérer l'ordinateur. C'est un programme particulier qui interprète des **commandes** et sait comment les exécuter. La commande `dir` affiche la liste de tous les fichiers présents dans un répertoire (`ls` sous Unix). Pour différencier la lettre de support d'une commande, on utilise le caractère réservé : (deux points).

Ainsi, **A:** désigne le lecteur **A**, donc le premier lecteur de disquette de l'ordinateur. Et la commande **dir A:** liste les fichiers présents sur la disquette (voir les commandes plus bas).

Compliquons un peu l'histoire : un même disque dur peut se voir attribuer plusieurs lettres ! Un seul disque dur physique peut en effet être découpé en lecteurs logiques, ce qui permet déjà de ranger l'information. Si votre disque est de grande capacité, vous pouvez le morceler et faire comme si vous disposiez de plusieurs disques durs, l'un servant pour les programmes (système d'exploitation, traitement de texte etc.), un autre pour stocker vos fichiers de travail, un troisième pour ranger les médias (images, vidéos, sons), encore un autre pour les jeux etc. Chacun de ces lecteurs aura une lettre attribuée par ordre alphabétique : **C**, **D**, **E**, **F**, etc.

Avec le développement des réseaux, on sait utiliser un périphérique de stockage à distance. Là encore, la désignation s'effectue grâce à une lettre. Il est d'usage (mais pas nécessaire) de commencer l'alphabet à rebours. A l'Institut, l'accès au disque dur du serveur se fait par la lettre **Z**.

Puis un lecteur logique est encore découpé au moyen des répertoires, lesquels permettent un rangement aussi précis que souhaité, grâce à leur imbrication : chaque répertoire peut lui même être séparé au moyen de répertoires. Par conséquent, l'utilisation de répertoires fournit à un lecteur une structure en arbre, tous les répertoires dépendant d'un répertoire parent. Le répertoire parent ultime, dont dépend tous les répertoires, s'appelle la racine et utilise le caractère backslash ****.

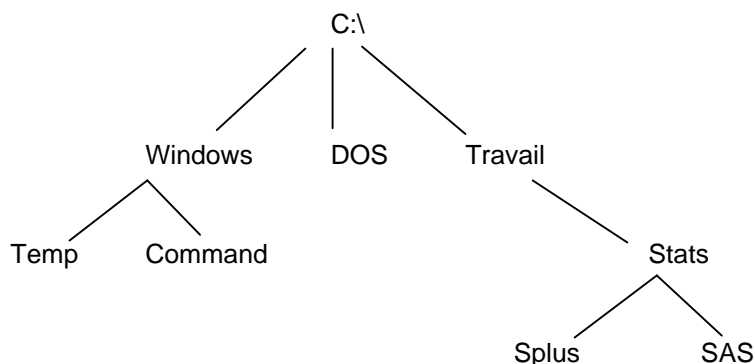


FIG. 5 – Structure en arbre des répertoires

La figure 5 illustre un découpage en répertoires. Le lecteur **C** (disque dur) est d'abord séparé par trois répertoires (**Windows**, **DOS** et **Travail**). Dans le répertoire **Travail**, on a créé un sous-répertoire qui s'appelle **Stats**. Notons que des fichiers peuvent être trouvés dans n'importe quel répertoire : aussi bien dans le répertoire (faute) **Travail**, que dans le répertoire **Stats**.

Pour pouvoir employer un fichier particulier, il faut que l'on «soit» dans le répertoire (on n'accède pas à une fiche de révision sans d'abord trouver et ouvrir la farde dans laquelle elle est), ou que l'on précise le **chemin** d'accès complet du fichier.

Si le fichier `FOURMIS.DAT` est présent dans le répertoire Stats, son chemin complet (absolu) est `C:\TRAVAIL\STATS\FOURMIS.DAT`, ce qui se décompose de la manière suivante : `C:` pour le lecteur C (disque dur principal), `\` pour le répertoire racine, dont on commence l'exploration en «entrant» dans le répertoire `TRAVAIL`, puis dans son sous-répertoire (`\`) `STATS`, où se trouve (`\`) le fichier `FOURMIS.DAT`.

Si le répertoire courant est le répertoire Windows, on pourra désigner le même fichier en utilisant un chemin relatif (comment accéder au fichier en partant de là où l'on est). Le point (`.`) désigne le répertoire courant, et deux points (`..`) le répertoire parent (un niveau au dessus, le répertoire qui contient le répertoire courant). Le chemin relatif est alors : `..\..\TRAVAIL\STATS\FOURMIS.DAT`

Les deux modes d'accès à un fichier (absolu ou relatif) sont toujours possibles. Le mode relatif offre plus de possibilités : si vous programmez une analyse qui écrit un rapport de manière automatique, vous pouvez copier votre programme sur un autre disque dur, il fonctionnera encore.

4.2 Quelques commandes DOS de gestion

Normalement, les systèmes d'exploitation entièrement graphiques, tel Windows, permettent de passer outre l'apprentissage des commandes DOS. Cependant, pour un statisticien, il est intéressant de connaître quelques une de ces commandes. En effet, le seul moyen de manipuler des fichiers à l'intérieur d'un environnement de programmation comme SPlus ou SAS est de faire appel à ces commandes.

<code>:</code>	Utilisation d'un lecteur logique dans un chemin (A :, C :)
<code>\</code>	Séparation de répertoire. Seul, désigne la racine d'un lecteur
<code>.</code>	Répertoire courant
<code>..</code>	Répertoire parent (qui contient le répertoire en cours)
<code>?</code>	Caractère joker spécial. Remplace tout autre caractère
<code>*</code>	Caractère joker spécial. Remplace toute suite de caractères
<code>dir</code>	Pour Directory. liste le contenu d'un répertoire
<code>cd</code>	Pour Call Directory. Change le répertoire courant
<code>md</code>	Pour Make Directory. Crée un sous-répertoire dans le répertoire courant
<code>rd</code>	Pour Remove Directory. Supprime un répertoire (il doit être vide)
<code>copy</code>	Pour copier des fichiers d'un répertoire à un autre
<code>del</code>	Pour effacer des fichiers

Exemples

En travaillant avec l'arbre de la figure 5 :

`dir c:\` affiche la liste du répertoire racine. Il s'agit donc de quelques fichiers systèmes (AUTOEXEC.BAT, CONFIG.SYS et quelques autres. Ne jamais y toucher) et des répertoires WINDOWS, DOS et TRAVAIL.

`md \TRAVAIL\STATS\EXCEL` crée un sous-répertoire EXCEL dans le répertoire STATS. Noter l'emploi du caractère \ pour partir de la racine du disque (on peut donc «se trouver» dans n'importe quel répertoire courant, la commande fonctionnera).

`copy C:\TRAVAIL\STATS\SAS*.* .` copie tous les fichiers (désignés par *.* : n'importe quel nom suivi de n'importe quelle extension) du répertoire SAS dans le répertoire courant (.).

Exercice 7 *Le répertoire courant est Z:\PERSONAL. Donner une suite de commandes permettant de créer un répertoire COURS à la racine du lecteur Z, contenant le sous-répertoire STAT2430. Y copier le fichier WINCHAT.INI, présent dans le répertoire WINNT du disque dur local. Recommencer l'exercice en trouvant d'autres commandes possibles (changer par exemple les chemins absolus en chemins relatifs).*

Exercice 8 (SPlus) *Ecrire une fonction pour effectuer une analyse sommaire du tableau (T) selon l'un des scénarios. Les sorties doivent être redirigés vers un fichier texte. L'emplacement du fichier texte doit être passé en argument de la fonction (chemin absolu).*

4.3 SPlus : bases et répertoires `_Data`

Le logiciel SPlus utilise deux répertoires très particuliers à connaître. L'un se nomme `_Prefs` et regroupe les paramètres de lancement (c'est ici que seront inscrites les préférences d'environnement propres à l'utilisateur). L'autre, très important, est le répertoire `_Data`, aussi appelé **base**, où sont stockés les objets manipulés par l'utilisateur lors de son travail. Ces objets sont tout aussi bien les données importées que les fonctions écrites ou les vecteurs manipulés.

Dans le répertoire `_Data`, SPlus utilise un format propriétaire, aussi bien pour la structure du fichier que pour l'attribution du nom des fichiers. En principe, tout doit se passer de manière transparente pour l'utilisateur, ce qui fait qu'il n'y a pas besoin d'aller modifier des éléments dans ce répertoire (n'effacez d'ailleurs aucun des fichiers présents ici...).

La fonction SPlus `ls()` liste les objets de votre répertoire `_Data`. Une visualisation plus poussée des objets présents dans votre base de travail peut s'effectuer grâce à l'explorateur d'objets inclus dans SPlus. L'explorateur permet de séparer les différents types d'objet (matrices, vecteurs, fonctions, jeux de données, etc.)

Malheureusement, si vous travaillez souvent sous SPlus, votre répertoire `_Data` va se remplir et gérer tous vos objets deviendra vite fastidieux («J'ai une fonction qui

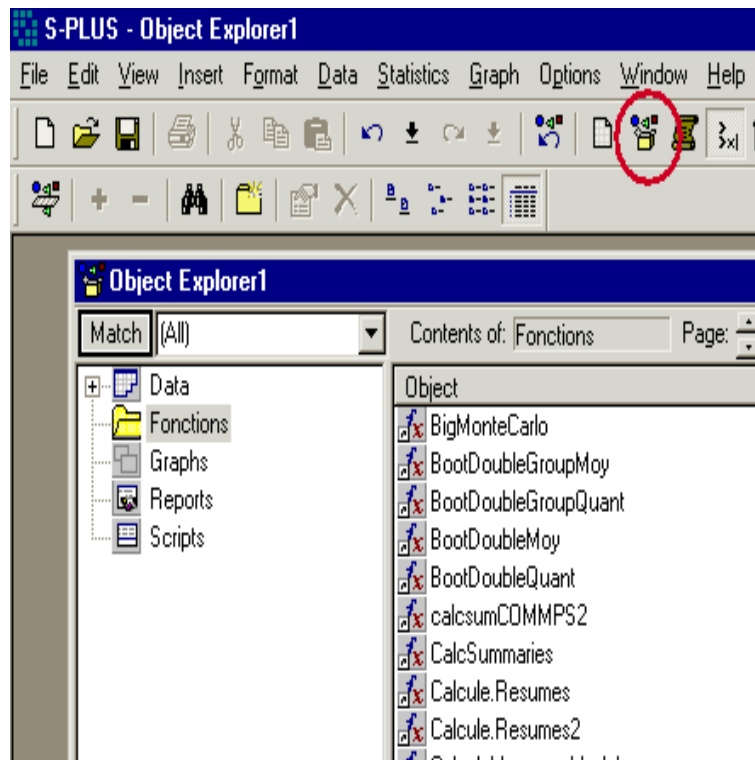


FIG. 6 – Capture d'écran : l'explorateur d'objets de SPlus

fait cela, comment l'avais-je appelée, déjà ?» est une question que l'on entend souvent à l'Institut...). Aussi, il faut aussi prévoir un système de rangement dans SPlus plus poussé que la simple séparation objets prévue par l'explorateur d'objets.

La solution passe par la gestion de plusieurs répertoires `_Data`. Vous n'êtes en fait pas limités dans le nombre de bases ouvertes en simultanément. A preuve un appel à la fonction `search()` qui liste les bases actuellement ouvertes. Au lancement de SPlus, 14 bases sont déjà ouvertes (toutes ne correspondent pas à des répertoires `_Data` mais recensent les fonctions SPlus internes).

Pour créer une nouvelle base, il suffit de créer un nouveau répertoire `_Data` et de l'attacher en tant que base à la commande `attach()`, qui s'emploie de la manière suivante :

```
attach("chemin du répertoire _Data", pos=numéro de base)
```

Lorsque l'on fait référence à un objet SPlus, le logiciel va chercher l'objet correspondant dans les bases en les parcourant de la première (la base de travail) à la dernière (au lancement la base 14). C'est ainsi qu'un appel à la fonction `mean` sans argument permet

de voir le code de la fonction : cette dernière est présente dans l'une des 14 bases disponibles (plus précisément celle qui est en deuxième position lors du lancement, comme le montre `ls(pos=2)`).

Pour détacher une base (on ne travaille plus avec les objets présents dans le répertoire correspondant), on utilise la commande `detach(what=nombre)`. On pourra copier des objets d'une base à une autre en utilisant la commande `objcopy` (voir la syntaxe dans l'aide de SPlus).

Afin de ne pas mélanger tous vos objets (et notamment vos fonctions), il est conseillé de créer un répertoire `_Data` pour chaque analyse que vous avez à faire (ou pour chaque projet, chaque cours). Un moyen ensuite de travailler directement dans ce répertoire est de l'attacher en tant que base en première position. Attention : pour disposer dans SPlus du symbole `\` de séparation de répertoire, il est nécessaire d'en utiliser deux ! Aussi la manipulation sera :

```
md Z:\COURS\STAT2430\SPLUS\_DATA (sous DOS)
> attach("z:\\cours\\stat2430\\splus\\_data", pos=1) (sous SPlus juste après le lancement)
```

Exercice 9 (SPlus) *Ecrire une fonction pour sauvegarder vos données SPlus. Elle doit prendre en paramètre un chemin de sauvegarde, y créer un répertoire `_Data` (utiliser la fonction `dos()` de SPlus), l'attacher comme base en deuxième position, y copier tous vos objets SPlus et le détacher.*

4.4 SAS : les librairies

Sous le logiciel SAS, on retrouve aussi cette approche de la gestion des données par une gestion transparente d'objets placés dans des répertoires physique. L'équivalent de la base SPlus est la **librairie** SAS. Comme pour SPlus, on a grand intérêt à utiliser le système parallèle de gestion offert par le logiciel, afin d'éviter la manipulation directe des objets, codés dans un format interne. De plus, le logiciel SAS existe aussi sous des gros systèmes dont la gestion des répertoires est très différente de celle du DOS sur PC. Aussi, l'utilisation d'un système parallèle assure la compatibilité du code entre les différentes versions de SAS.

La déclaration d'une librairie se fait par la commande `LIBNAME` :

```
LIBNAME NOMDELIBRAIRIE 'CHEMIN ABSOLU DU REPERTOIRE';
```

Le nom de la librairie, dont on se servira par la suite pour faire référence au répertoire, ne doit pas faire plus de 8 caractères. La déclaration suivante est valide :

```
LIBNAME MALIB 'c:\travail\stats\sas';
```

Une librairie SAS peut comprendre des jeux de données (appelés `datasets` sous SAS), des formats et des graphiques.

La figure 7 montre les deux accès au module de gestion de librairie (Menu `Globals...Access... Display Libraries` ou l'icône du gestionnaire de fichiers).

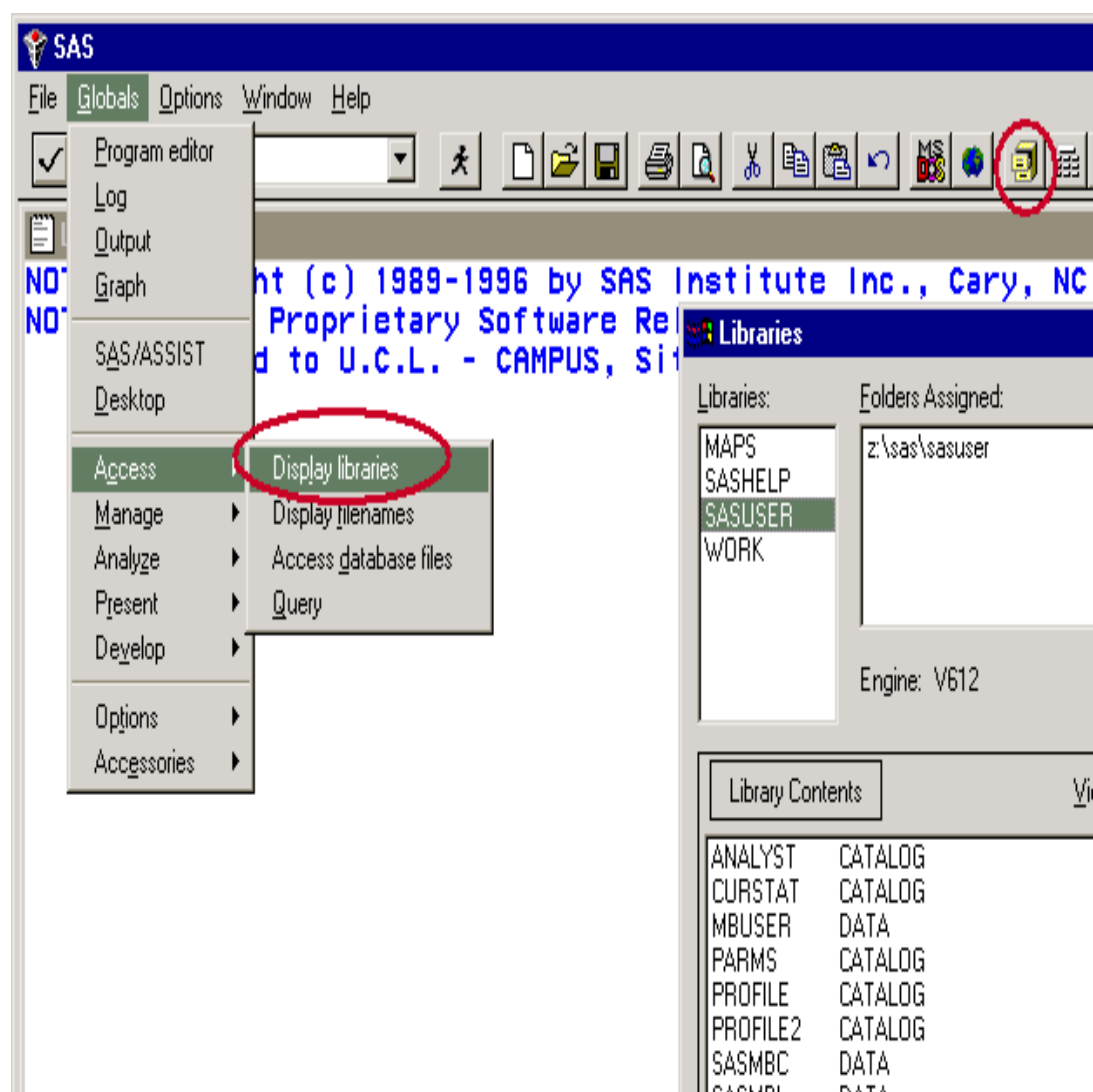


FIG. 7 – Capture d'écran : Gestion des librairies dans SAS

On emploiera un bloc data pour copier un jeu de données d'une librairie à une autre :

```
DATA malib1.data1;
    SET malib2.data1;
RUN;
```

4.5 SAS Enterprise Guide : cas particulier

Enterprise Guide ne permet pas la gestion des données, en ce qu'il gère un projet complet d'analyses regroupant de multiples jeux de données *déjà* rangés. Lorsque vous créez un projet avec Enterprise Guide, les données ne sont ajoutés à l'analyse qu'en tant que **lien**. C'est à dire que vous indiquez à EG le chemin absolu de vos fichiers et qu'il en crée une copie locale dans un répertoire temporaire lorsqu'il en a besoin. Ce procédé comporte un avantage incontestable : les données originales ne sont jamais modifiées et restent sécurisées.

Par conséquent, si vous employez EG, vous devrez être particulièrement vigilant au rangement de vos fichiers dans des répertoires.

Attention : la version d'Enterprise Guide présente à l'Institut n'accepte pas toujours les chemins réseau. Si le logiciel ne veut pas incorporer votre fichier dans un projet, copiez vos données en local, dans un répertoire du disque **C** où vous avez l'accès en écriture (comme le répertoire **C:\WINNT\TEMP**, par exemple).

5 Manipulations

Dans cette dernière section, nous reprenons en revue tout ce qui a été vu, en le remplaçant dans le contexte d'une analyse statistique, étape par étape. Bien entendu, ce ne sont là que des conseils et tout un chacun est entièrement libre de planifier son analyse comme il l'entend. Ces conseils sont surtout valables à long terme, lorsque votre disque dur doit contenir les multiples fichiers propres à beaucoup d'analyses.

5.1 Utilisation des répertoires

Au cours d'une analyse, on est souvent amené à utiliser plusieurs logiciels, certains engendrant beaucoup de fichiers (le logiciel SPAD crée une dizaine de fichiers au cours d'une analyse factorielle). Aussi est-il conseillé de se servir au maximum des possibilités de classement des répertoires.

Voici un exemple de ma classification personnelle, portant sur le premier travail du cours STAT2430 (figure 8).

Un premier répertoire est créé, dédié à l'analyse. Des sous-répertoires pour chaque logiciel (Txt pour les données mises au format texte, Excel pour les données mises en forme, Word pour le rapport, Statistica pour les graphiques, SPlus pour les analyses etc.). Ainsi, les fonctions créées sous SPlus ne sont pas mélangées avec d'autres fonctions SPlus et sont facilement retrouvables.

Pour une analyse qui va durer en longueur, il est prudent de ranger dans des sous-répertoires de dates (12-03-2001) et d'effectuer des **copies de sauvegarde** régulièrement.

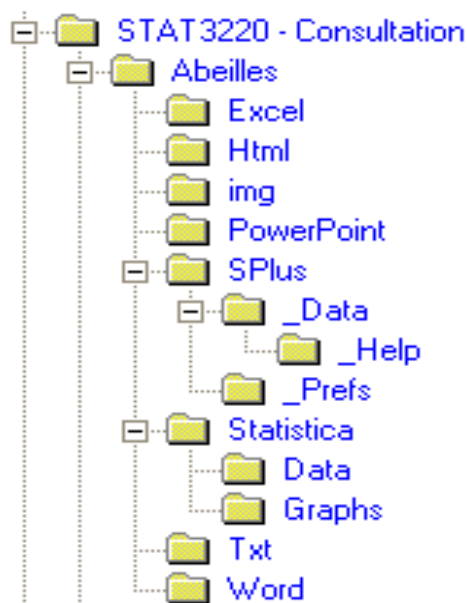


FIG. 8 – Des répertoires pour ranger

5.2 Format texte : aperçu et plate-forme d'échange

La toute première étape est de regarder le contenu du fichier et s'il est dans un format particulier, d'utiliser le logiciel qui l'a généré pour l'exporter au format texte. Le format texte est le seul format réellement universel, en ce sens que tous les logiciels de statistique savent importer des fichiers texte (alors que tous ne lisent pas des fichiers Excel par exemple). Profiter d'un traitement de texte comme Textpad ou WinEdt (salle didactique) pour regarder le nombre d'individus, de variables, les données manquantes ou aberrantes, voir si vous comprenez bien le fichier etc. A quelle(s) question(s) souhaitez-on répondre ?

5.3 Format Excel : Mise en forme des données, petits calculs, petites retouches

Si le fichier n'est pas trop grand (limitations : 65536 lignes et 256 colonnes), il est utile à ce stade de l'ouvrir avec Excel pour disposer d'un tableau mis en forme. La mise en forme n'est pas seulement qu'esthétique... Si une variable précise est d'intérêt (variable à expliquer au moyen d'une régression par exemple), on peut la mettre en premier. On peut aussi séparer les variables continues des variables discrètes, réordonner les individus selon une ou plusieurs variables, repérer les minimum et maximum par des couleurs, voir

si les données manquantes n'obéissent pas à certaines règles (du genre : si l'information d'une variable manque alors systématiquement d'autres variables ne sont pas observées).

À ce stade, on doit arriver à mieux sentir l'analyse et à se faire une idée a priori sur des résultats, que l'on devra montrer après. Pour avancer prudemment, on peut utiliser Excel pour réaliser quelques calculs simples et des graphiques intermédiaires, que l'on souhaite mettre en forme avant de montrer au client : «Voyez ici le phénomène : il semble que ces deux groupes ont des moyennes très différentes (montrer un tableau mis en forme et un graphique). Est-ce bien la preuve de cette différence qui vous intéresse ?»

Excel est aussi le logiciel idéal pour effectuer des petites retouches sur les données : remplacements de valeurs, suppressions d'observations, recodages, ajout d'une variable etc. Penser à utiliser les fonctionnalités du classeur Excel : il est conseillé de toujours garder une copie des données originales. Je conseille de garder ces données sur une feuille, d'utiliser une seconde feuille pour les mettre en forme, une troisième pour préparer des tableaux résumés et graphiques et 'encore une autre pour manipuler ces données avant exportation. Note : l'exportation en format texte s'effectue via le menu File... Save as (format TXT) et ne permet d'exporter qu'une feuille à la fois. Toujours enregistrer l'ensemble du classeur au format XLS avant l'exportation.

5.4 Calculs intensifs/itératifs, fonctions personnalisées

Il arrive souvent qu'au cours d'une analyse une même tâche doive être répétée plusieurs fois : effectuer des sous-analyses par groupes, traiter plusieurs fichiers etc. A moins d'être prêt à se transformer en machine humaine, point de salut sans automatisation du processus, laquelle passe par... de la programmation. A l'Institut, seuls les logiciels SPlus et SAS vous permettront d'arriver au résultat souhaité.

Le logiciel SAS, pour la majorité de ces procédures, propose l'option BY, laquelle permet d'appliquer la même procédure à plusieurs sous-groupes définis par un facteur. L'équivalent dans SPlus est à chercher dans les fonctions de la famille apply (moins immédiat).

	taille	poids	sexe
1	175	69	2
2	179	83	1
3	193	96	1
4	182	93	1
5	168	55	2
6	172	86	2

Exemple SAS

```
PROC MEANS data=work.data1 n min max range mean var;
  VAR taille poids;
```

```
BY sexe;
RUN;
```

Exemple SPlus

```
> tapply(data[,1],data[,3],summary)
$"1":
  Min. 1st Qu. Median  Mean 3rd Qu. Max.
  179   180.5   182 184.7   187.5  193

$"2":
  Min. 1st Qu. Median  Mean 3rd Qu. Max.
  168    170    172 171.7   173.5  175
```

Enfin, l'un et l'autre de ces logiciels permettent de définir vos fonctions personnalisées et de construire automatiquement des rapports.

Exemple SPlus

```
> MonAnalyse<-function(vecteur,valeur=180)
  {
    cat("\n *** Statistiques descriptives ***\n")
    print(summary(vecteur))
    cat("\n *** Test: Taille=180? ***      \n ")
    print(t.test(vecteur,mu=valeur))
    invisible(return(1))
  }

> tapply(data[,1],data[,3],MonAnalyse)

*** Statistiques descriptives ***
  Min. 1st Qu. Median  Mean 3rd Qu. Max.
  179   180.5   182 184.7   187.5  193

*** Test: Taille=180? ***

One-sample t-Test

data:  vecteur
t = 1.0966, df = 2, p-value = 0.3872
```

```

alternative hypothesis: true mean is not equal to 180
95 percent confidence interval:
 166.3558 202.9775
sample estimates:
mean of x
 184.6667

```

```

*** Statistiques descriptives ***
Min. 1st Qu. Median Mean 3rd Qu. Max.
 168    170    172 171.7  173.5  175

```

```

*** Test: Taille=180? ***

```

One-sample t-Test

```

data: vecteur
t = -4.11, df = 2, p-value = 0.0544
alternative hypothesis: true mean is not equal to 180
95 percent confidence interval:
 162.9427 180.3907
sample estimates:
mean of x
 171.6667

```

5.5 Gestion de gros fichiers, bases de données

Gérer des gros fichiers de données pose des problèmes au statisticien. Comme déjà vu, Excel est limité à 65536 observations, ce qui est peu pour beaucoup de besoins actuels. Et même avec 50000 observations, l'utilisation d'Excel n'est pas toujours évidente : ce dernier place en effet le fichier ouvert entièrement en mémoire vive... De toutes façons, la gestion de gros fichiers nécessite de grosses machines (processeur rapide et surtout grande quantité de mémoire vive, RAM).

Avec de gros fichiers, un simple calcul de moyenne peut être fortement ralenti. Si le logiciel SPlus n'est censé être limité que par la mémoire disponible sur votre ordinateur, son emploi est à déconseiller pour traiter les très gros fichiers.

En fait, seul le logiciel SAS tire son épingle du jeu lorsqu'il s'agit de gérer de très grandes quantités de données. D'ailleurs, ce n'est pas un hasard si c'est un des seuls logiciels statistiques à pouvoir gérer des bases de données.

5.5.1 Les bases de données

On entend de plus en plus parler des bases de données et d'analyse statistique des bases de données. Le terme base de données (en anglais, database) provient du monde informatique et a priori ne concerne pas du tout le statisticien... Les bases de données reposent sur une théorie du stockage de l'information. Tout comme le tableau statistique est plus qu'un tableau de nombres, la base de données est un ensemble regroupant beaucoup d'informations :

- **Plusieurs tableaux de données.** On ne travaille plus avec une table, mais avec plusieurs. Ceci est certainement ce qui gêne le plus le statisticien, toujours habitué à ne travailler qu'avec un unique tableau «à plat». Noter que la division en plusieurs tables n'existe que pour permettre une optimisation de la place physique occupée. En effet, toute base de données peut être ramenée à un unique tableau à plat, lequel peut dans certains cas s'avérer énorme. La reconstruction de cet unique tableau s'effectue via :
- **Des liens entre ces tableaux.** Les différentes tables doivent toujours partager de l'information en commun. Un lien décrit la dépendance structurelle entre les informations. La figure 9 décrit une base de données portant sur les stocks d'une chaîne commerciale. La table «principale» est celle des stocks (articles, magasins). Cette table est liée à une table qui apporte de l'information sur les articles. Cette information placée dans la table principale serait redondante : un même article étant présent dans plusieurs magasins, la description de l'article serait répétée pleins de fois. La table principale est aussi reliée à la table des villes au moyen d'une table intermédiaire. De même, on optimise l'espace physique en évitant de répéter plusieurs fois l'information (plusieurs magasins sont présents par ville).
- **Des droits d'accès :** la base de données est prévue pour partager de l'information. Certaines personnes doivent pouvoir ajouter de l'information, d'autres la consulter, et encore d'autres n'y avoir aucun accès. L'accès à l'information se fait via :
- **Des requêtes.** Une requête est une question que l'on peut poser à la base de données. Le gestionnaire d'un magasins peut vouloir disposer tous les soirs de la liste des articles de son magasin dont les stocks ont baissé de 50% dans la journée. La requête est l'équivalent d'une fonction qui interrogera la base et lui retournera une réponse sous la forme d'un état.
- **Des états.** Aussi appelés vues. Décritent la mise en forme visuelle des interrogations de la base. Si le gestionnaire peut poser sa question, sous quelle forme sortira son rapport ?

La base de données est donc le moyen le plus abouti à ce jour de gérer l'information, aussi bien au niveau structurel que pratique (accès, optimisation de la place physique occupée, droits d'accès, automatisation du processus d'acquisition de données, etc.). Les bases de données sont utilisées dans le monde entier et dans des endroits parfois

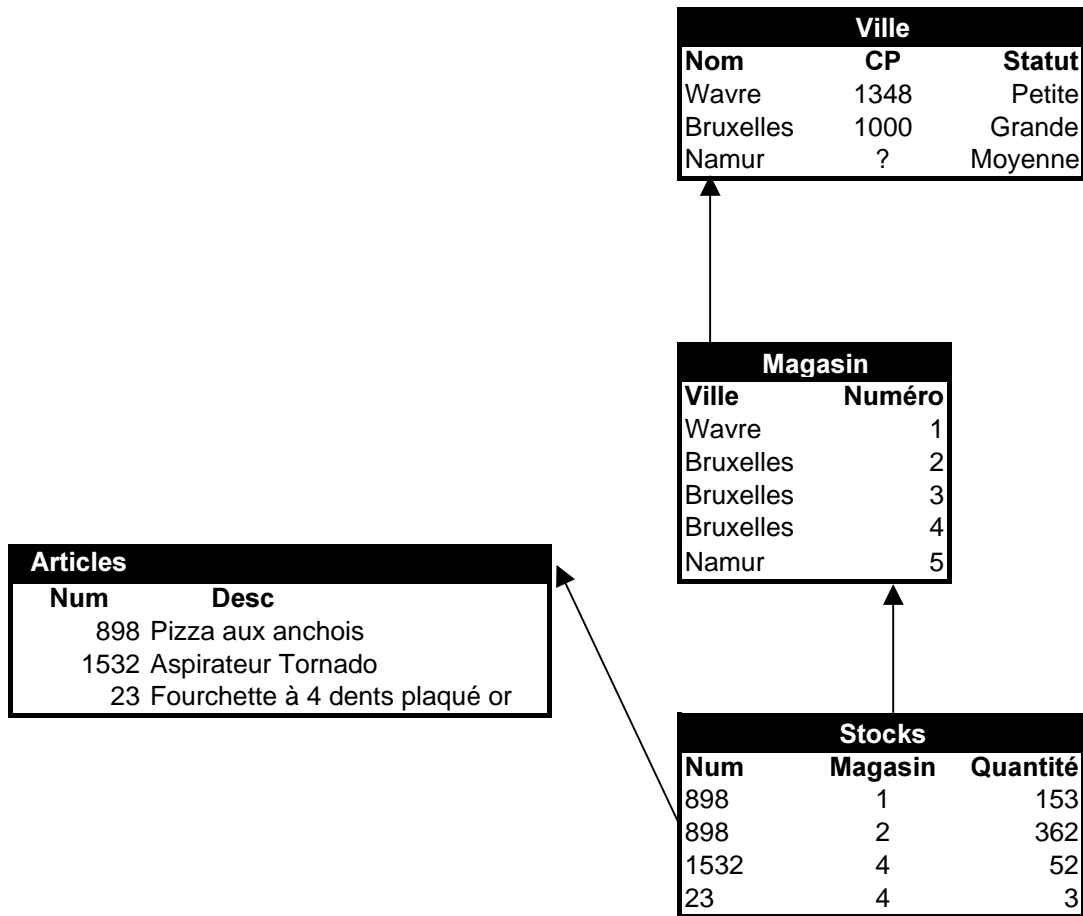


FIG. 9 – Illustration d'une base de données

insoupçonnés. A la SCNB, vous consultez une borne pour disposer des horaires des trains. La borne consulte une base de données (elle y a donc un droit d'accès), construit à la volée la requête correspondant à votre question (Liste des trains partant de LLN direction Bruxelles aujourd'hui), et vous retourne un rapport visible à l'écran.

La culture statistique se concentre surtout sur les tableaux statistiques mais va lentement être gagnée par le monde des bases de données. Aussi est-il bon pour un statisticien de connaître un minimum de la théorie des bases de données. Sous le logiciel SAS, la manipulation des bases de données est possible via le langage universel SQL.