Université de Poitiers
Département de Mathématiques

1M09 – Statistique Descriptive
Master MMAS 1, année 2009–10

TRAVAUX PRATIQUES N° 1. — INTRODUCTION À SAS

1. Généralités sur le logiciel SAS

Le logiciel SAS (Statistical Analysis System) est développé et commercialisé par la société américaine SAS Institute Inc., située à Cary, en Caroline du Nord. Il a été conçu comme un logiciel de Statistique polyvalent, c'est-à-dire susceptible de traiter pratiquement tous les domaines de la Statistique. Il est assez ancien (initié dans années 1970) et est constamment enrichi de nouvelles méthodes. Aujourd'hui, SAS se veut un véritable système de gestion de l'Information plutôt qu'un simple logiciel de Statistique. La documentation papier de SAS est énorme et n'existe qu'en anglais. Cette documentation est aujourd'hui partiellement en ligne et accessible à partir de l'interface. On peut trouver sur Internet des cours synthétiques en français. Nous disposons en TP d'une version française l'interface. Un texte de référence en français est proposé dans :

[1] Azaïs J.M., Besse P., Cardot H., Couallier V. et Croquette A., *SAS sous Unix : Logiciel hermétique pour système ouvert*, `http://www.lsp.ups-tlse.fr/Besse/pub/saspdf.pdf`.

Même si nous serons amenés à utiliser la version Windows de SAS, il est vivement conseillé de vous y reporter avant de vous lancer dans la consultation de l'aide en ligne. Une référence pour une introduction à SAS se trouvant à la BU :

[2] Kontchou Kouomegni, H. et Decourt, O., *SAS : maîtriser SAS Base et SAS Macro SAS 9 et versions antérieures*, Dunod (2004). (réf. BU 004.43(SAS) KON)

Vous trouverez d'autres ouvrages dans la même classification.

2. Connexion dans la salle BE11

Se connecter à ENS/UFR. Il est ensuite important de créer un répertoire spécial sous lequel, tous les programmes, tables et sorties SAS seront sauvegardées (on pourra, par exemple, l'appeler `TPSAS`).

1. Si ce n'est pas déjà fait, créer un raccourci vers l'*Explorateur* sur le bureau : sélectionner `Menu démarrer/Tous les programmes/Accessoires/Explorateur Windows` puis « clic droit » pour choisir `Envoyer vers/Bureau`. Puis il suffira de cliquer deux fois sur le raccourci pour ouvrir l'*Explorateur*.
2. Ouvrir l'*Explorateur*. Créer le répertoire de nom `TPSAS` à partir de `Fichier/Nouveau-/Dossier` de la fenêtre de l'*Explorateur*.
3. Enfin, créer un raccourci vers le répertoire `TPSAS` en sélectionnant le répertoire `TPSAS` sous l'*Explorateur* et en procédant comme en 1.

3. Les fenêtres principales de SAS

Entrer dans SAS en faisant sélectionnant SAS à partir du menu `Démarrer/Tous les programmes` (la version actuelle est la version 9.1.3). On pourra là aussi créer un raccourci sur le bureau avec la même manipulation que dans la précédente section. Une fois lancé, plusieurs fenêtres s'ouvrent à l'écran comme en Figure 1 et constituent ce qui est appelé le *Display Manager Service* ou (DMS).

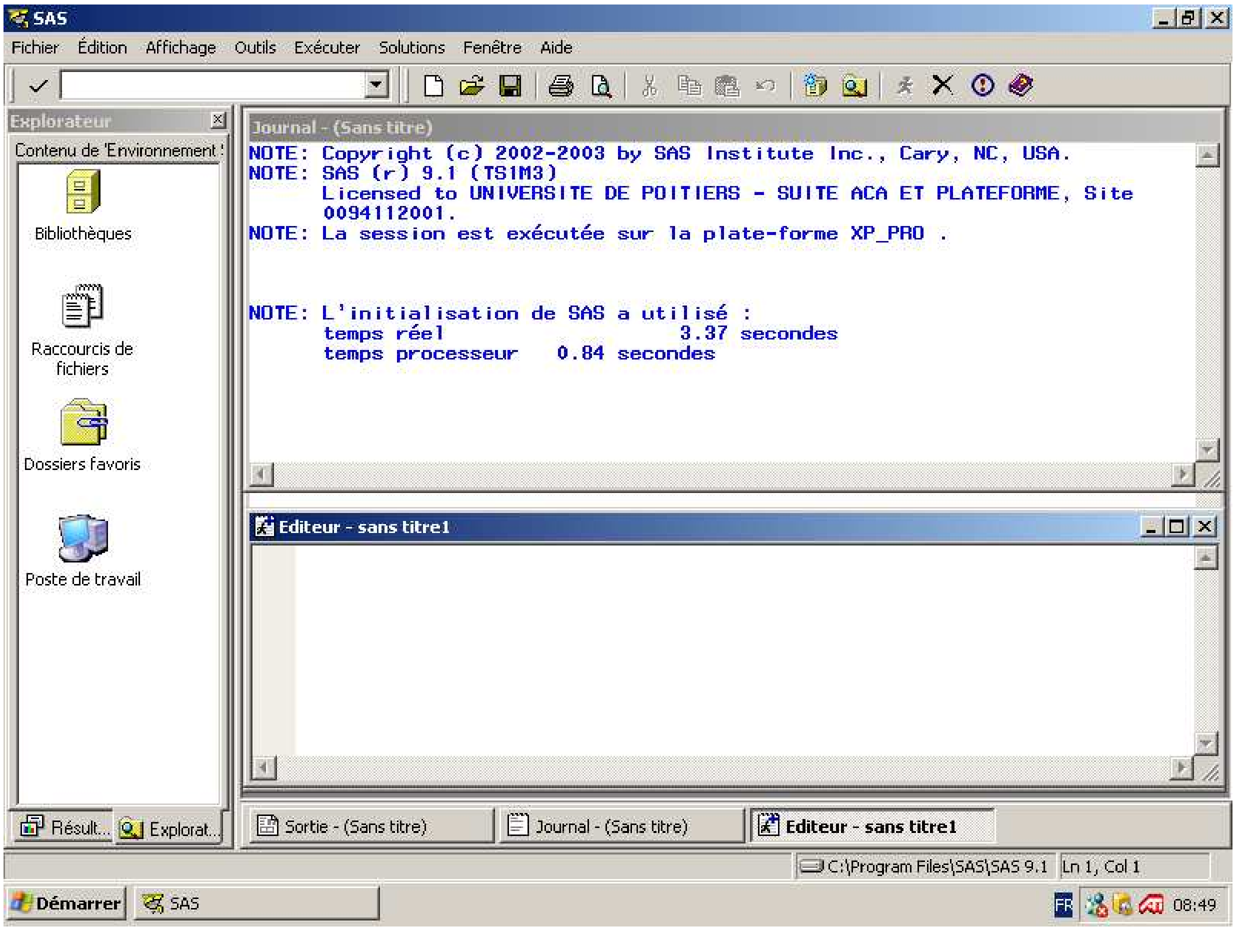

Figure 1. — Aspect de l'interface après lancement de SAS

Les 3 fenêtres les plus utiles apparaissent dans la partie droite de la fenêtre principale (Figure 1) :

- *Éditeur* (Editor) : c'est l'éditeur de texte de SAS, dans lequel on entre tout programme à exécuter.
- *Journal* (Log) : fenêtre dans laquelle s'affichent, au cours de l'exécution d'un programme, le programme lui-même, séquence par séquence (en noir) et les commentaires du système SAS sur ce programme (en bleu) ; le cas échéant, s'affichent également ici un message d'avertissement, lorsqu'un problème non fatal est détecté (précédé de *warning*, en vert) ou un message d'erreur, lorsqu'une erreur fatale est détectée (précédé de *error*, en rouge) ;
- *Sortie* (Output) : fenêtre dans laquelle s'affichent tous les résultats obtenus à partir d'un programme s'exécutant correctement.

Seules les deux fenêtres *Journal* et *Éditeur* sont ouvertes lors du lancement. La fenêtre *Sortie* recouvrira les deux précédentes lors de l'affichage des résultats textuels (ou graphiques, et dans ce cas, c'est la fenêtre *Graphique* qui prend la place). Pour retrouver le *Journal* et l'*Éditeur*, il suffit de cliquer sur les onglets correspondants dans la barre en bas de la fenêtre (cf Figure 1). De manière générale, cette manipulation permet de rendre la fenêtre sélectionnée active. Ces trois fenêtres possèdent sensiblement les mêmes « menus déroulants » sur la partie haute de la fenêtre principale (le menu contextuel en haut de la fenêtre principale dépend de la fenêtre active).

Notons que les résultats (textuels et graphiques) des différentes exécutions d'une session SAS apparaissent dans le menu placé dans la fenêtre dans la partie gauche de la fenêtre principale (*voir* onglet *Résultat*).

On quitte SAS à partir du menu déroulant `Fichier/Sortie` de la fenêtre principale. Il existe divers modes de traitement de données avec SAS. Nous ne nous intéresserons ici qu'au mode interactif.

1. PROGRAMMATION SAS. Cela consiste à écrire un programme SAS dans la fenêtre *Éditeur* (ou sous un éditeur de texte quelconque). On sélectionne à la souris le texte correspondant et on l'exécute via le menu déroulant `Exécuter/Soumettre` (ou en cliquant sur le bouton dans la barre d'outils principale. Nous reviendrons sur la structure globale d'un programme SAS plus loin.

2. SAS/INSIGHT

   (menu déroulant `Solutions/Analyse/Analyse Interactive des Données`).

   Il permet un traitement interactif immédiat et puissant des données uni- ou bidimensionnelles. On peut réaliser des graphiques déjà très élaborés. Les possibilités de SAS/INSIGHT seront partiellement exploitées dans les TP. Quelques méthodes d'analyses multidimensionnelles sont disponibles (en particulier l'ACP). Notons qu'il existe un module, sur le principe « tout à la souris » de SAS/INSIGHT, nommée *Enterprise Miner* et largement utilisé dans le milieu des entreprises.

Sur la Figure 1, une seconde fenêtre apparaît sur la partie gauche de l'interface. Deux onglets figurent au bas de cette fenêtre, *Explorateur* et *Résultats*. Il suffit de cliquer sur un onglet pour rendre la fonction d'exploration de répertoires, ou de résultats des analyses, active. Une barre d'outils (de boutons) contextuelle apparaît alors et permet un accès direct à quelques fonctionnalités. Par exemple, si la fonction d'exploration est active, on voit apparaître au moins les quatre icônes suivantes :

1. `Bibliothèques` qui donne accès à l'ensemble des bibliothèques actives durant la session (*voir* la prochaine sous-section) ;
2. `Dossiers Favoris` qui donne accès au contenu des dossiers `My Documents` (`Mes Documents`) et `My Desktop` (`Bureau`) de Windows ;
3. `Poste de Travail` qui donne accès au contenu de l'ensemble du poste de travail ;
4. `Raccourci de fichiers` qui donne accès, en particulier, aux raccourcis sur des fichiers externes à SAS ou sur des répertoires.

4. **Données**

Un fichier de données n'est reconnu, lu et traité par SAS que s'il est dans un format spécifique. De même, les fichiers produits en sortie d'une procédure SAS sont conformes à ce format spécifique. Nous parlerons de table SAS (*SAS data set*) pour un fichier mis dans un tel format. Une table SAS sauvegardée sur disque apparaît sous la forme ⟨*nomfichier*⟩`.sas7bdat`. Le TP proposera diverses manipulations sur ces tables.

Notons dès maintenant que chaque commande SAS est terminée par un point-virgule.

4.1. LES BIBLIOTHÈQUES

Une bibliothèque (*library*) est un nom logique, un raccourci, attribué par SAS à un répertoire pouvant contenir des données SAS. Une fois déclarée, on n'a plus à utiliser le nom absolu du répertoire (adresse physique sur le disque). Ainsi, pour utiliser le répertoire `TPSAS` créé en section 2 sans avoir à redonner son chemin absolu dans l'arborescence du disque dur, il suffit d'allouer un bibliothèque `TPSAS` une fois pour toute en début de session via :

```
LIBNAME TPSAS "Z:\...\TPSAS";
```

Une fois allouée, la lecture ou l'écriture d'une table ⟨*nomtable*⟩ dans la bibliothèque (i.e. dans le répertoire d'alias) `TPSAS` se fait au moyen :

```
TPSAS.nomtable
```

l'ensemble des bibliothèques actives durant la session SAS apparaît dans la fenêtre de gauche de l'interface une fois avoir sélectionné l'onglet *Explorateur*. Il est également possible de réaliser une création de bibliothèque via le bouton  de la barre d'outils active lorsque la fonction *Explorateur* a été choisie.

Au démarrage d'une session SAS, au moins trois bibliothèques sont allouées automatiquement. Elles apparaissent en cliquant dans la fenêtre *Explorateur/Bibliothèques* :

1. `WORK` : bibliothèque temporaire qui est purgée en fin de session. Il est donc déconseillé de l'utiliser pour stocker votre travail.
2. `SASHELP` : contient essentiellement de l'aide sur le fonctionnement du logiciel et des bases de données proposées par SAS.
3. `SASUSER` : c'est une bibliothèque personnelle permanente. Elle contient des informations de personnalisation d'une session SAS et peut éventuellement être utilisée pour y écrire vos propres données
4. `MAPS` : apparaît selon la configuration du logiciel. Elle contient un fond de cartes fournis par SAS (la majorité du globe et une carte départementale de la France) utilisable via certains modules SAS (SAS/GRAPH).

5. La structure d'un programme SAS

Un programme SAS comprend :

1. une étape `DATA` : elle est constituée des entrées-sorties (lectures et écritures de données) ;
2. une étape `PROC` : application d'une procédure ou d'un enchaînements de procédures aux données définies par l'étape `DATA` ;

Un programme SAS exécutable est stocké sur le disque sous la forme ⟨*nomfichier*⟩`.sas`.

5.1. Gestion des données avec SAS : étape DATA

Il existe 2 possibilités pour lire un fichier de données dans un programme SAS.

Tout d'abord, il est possible d'acquérir directement les données, en les incluant dans le programme, au moyen de la commande `CARDS` (*voir* l'Exercice 1). Cette façon de procéder est adaptée aux petits jeux de données et est suffisante pour une première prise en main du logiciel.

Ensuite, on peut lire des données préalablement enregistrées dans un fichier ASCII extérieur à SAS, au moyen de la commande `INFILE` (*voir* l'Exercice 2). Dans tous les cas, une déclaration des variables doit être effectuée au moyen de la commande `INPUT`. Enfin, noter que les données ne seront réellement utilisables qu'une fois transformées en table SAS, ce qui se fait par la commande `DATA`, suivie du nom que l'on souhaite donner à cette table. La commande `DATA` doit être placée en début de séquence, avant même la lecture des données. Ainsi, une séquence de lecture de données se présente en général de la façon suivante :

```
DATA <nom de la table SAS>;
     INFILE "<nom absolu du fichier des données ASCII>";
     INPUT <liste des noms des variables>;
RUN;
```

Noter que, dans la liste des variables, le séparateur est un blanc (et pas une virgule ou un point virgule).

Par ailleurs, chaque procédure SAS réalisant un traitement statistique produit un certain nombre de résultats. Ces résultats sont, soit affichés dans la fenêtre *Sortie*, soit enregistrés dans une table SAS particulière. Dans ce dernier cas, la procédure `PRINT` permet d'afficher le contenu de cette table SAS dans la fenêtre *Sortie*. On peut ensuite archiver le contenu de la fenêtre *Sortie* dans un fichier ASCII (ou RTF) en utilisant le menu déroulant `Fichier/Enregistrer sous`.

5.2. Les procédures SAS

Le cœur d'un programme SAS est en fait un enchaînement de procédures, chacune réalisant un traitement homogène. Les procédures de base sont répertoriées dans le volume *SAS procedures guide*. En dehors de la procédure `PRINT`, déjà citée, les principales procédures sont les suivantes :

- `SORT` : range le fichier selon les valeurs croissantes ou décroissantes d'une variable quantitative spécifiée par `BY`.
- `RANK` : calcule une variable $\langle rang \rangle$ pour chaque variable quantitative déclarée (déclaration obligatoire).
- `STANDARD` : permet le calcul des valeurs centrées et réduites associées à une variable quantitative donnée (nécessite les options `MEAN=0` et `STD=1`).
- `MEANS`, `UNIVARIATE` : servent à la description élémentaire de variables quantitatives (nombre d'observations, minimum, maximum, moyenne, écart-type, ...) ; `UNIVARIATE` est plus élaborée.
- `PLOT` : réalise le nuage de points relatif à 2 variables quantitatives sans mise en forme particulière ; la procédure `GPLOT` génère des graphiques plus esthétiques.
- `CHART` : réalise différents graphiques pour une variable qualitative ; là encore, `GCHART` offre des graphiques de plus grande qualité.
- `CORR` : calcul de la matrice des corrélations (ainsi que la matrice des variances-covariances) d'un ensemble de variables quantitatives.
- `FREQ` : sert à la description élémentaire de variables qualitatives (effectifs, fréquences, ...). Elle permet aussi de croiser 2 ou plusieurs variables, de déterminer des profils, ...).

On invoque une telle procédure par un bloc d'instructions démarrant par `PROC`, suivi du nom de la procédure ainsi que d'éventuelles options, et qui s'achève par l'instruction `RUN` suivi d'un « `;` ».

Exemple :

```
PROC UNIVARIATE;
RUN;
```

5.3. Instructions globales

Diverses commandes générales peuvent être rajoutées au début d'un programme SAS.

- `LIBNAME` : alloue une bibliothèque.

- `OPTION(S)` : il est ici question des options système et à spécifier en début de programme. On les déclare avec la commande `OPTION(s)` (le « s » est facultatif). En voici trois :
  1. `PAGESIZE`, qui spécifie le nombre de lignes dans une page de *Sortie* ;
  2. `LINESIZE`, qui spécifie le nombre de caractères par ligne dans une page de *Sortie* ;
  3. `NODATE`, qui supprime l'impression de la date dans les sorties.

- `TITLE` : permet de placer un titre en haut de chaque page des sorties textuelles et graphiques.

  ```
  TITLE "Ceci est un titre";
  ```

  On peut obtenir un titre sur plusieurs niveaux (jusqu'à 10) en spécifiant `TITLE1`, `TITLE2`, ..., `TITLE10`. Un titre reste actif durant toute une session tant que l'on ne réinitialise pas la commande. Pour cela, il suffit de vider le contenu en exécutant l'instruction

  ```
  TITLE;
  ```

- `FOOTNOTE` (note de pied-de-page) : permet de placer un titre en bas de chaque page des sorties. Elle permet également une meilleure mise-en-page des sorties SAS sur une imprimante. Cette commande dispose des mêmes propriétés et extensions que `TITLE`.

  ```
  FOOTNOTE "Ceci apparaîtra en bas de page";
  ```

- `GOPTION(S)` : spécifie des options valables pour les graphiques. Elles restent actives durant toute la session SAS tant que l'on ne réinitialise pas. Pour des graphiques en noir et blanc :

  ```
  GOPTIONS COLORS=(BLACK);
  ```

  Pour réinitialiser l'option précédente, il suffit de faire

  ```
  GOPTIONS RESET=COLORS;
  ```

  On réinitialise toutes les options avec

  ```
  GOPTIONS RESET=ALL;
  ```

- Commentaires : des commentaires peuvent être insérés n'importe où dans un programme SAS de la façon suivante :

  ```
  /* ceci est un commentaire */
  ```

  (surtout commode au sein d'une ligne de commande)

  ```
  * ceci en est un autre;
  ```

5.4. L'ODS

L'*Output Delivery System* ou ODS spécifie la destination de tout ce qui n'est pas un calcul. Par défaut, c'est la destination *Listing* qui est active, c'est-à-dire la fenêtre *Sortie*. Il est possible de rediriger la destination des sorties SAS vers un format PDF, HTML, RTF. Cela concerne par exemple la sortie d'une instruction `PRINT` ou d'un graphique par une procédure graphique par exemple. Elle permet, en particulier, d'obtenir des sorties mises en forme récupérables dans les applications associés aux types de fichiers mentionnés ci-dessus. Notons qu'il existe une destination au format LaTeX mais dont l'exploitation au sein d'un fichier LaTeX classique n'est pas toujours aisée. Le fichier LaTeX (par défaut avec un suffixe `.ltx`) est à exécuter de sorte d'obtenir un résultat au format PDF.

6. **Exercices**

Allouer la bibliothèque `TPSAS` via `LIBNAME`.

Exercice 1 (*Saisie de tables de données*). — Reproduire et faire marcher le programme SAS suivant (noter que l'instruction `INPUT` est placée avant `CARDS`). À cette occasion, utiliser les fonctionnalités de la fenêtre *Éditeur*.

```
DATA TPSAS.TableElection04;
     ATTRIB Nom LABEL="Nom du parti";
     ATTRIB Pourcentage LABEL="Pourcentage des voix obtenues";
     INPUT Nom$ Pourcentage;
     CARDS;
     Ext.G 3.3
     PC 5.2
     PS 28.9
     Verts 7.4
     Div.G 5.9
     Div.D 10.7
     UDF 11.9
     UMP 16.6
     FN 9.8
     Ext.D 0.3
     ;
RUN;

PROC PRINT;
RUN;

PROC PRINT NOOBS;
RUN;
```

Q 1. Une fois avoir exécuté, avec succès, le code précédent, sauvegarder le texte édité dans un programme SAS (`Fichier/Enregistrer sous`).

Q 2. Voir également SAS/INSIGHT pour consulter et apporter des modifications à la table.

Exercice 2 (*Création d'une table SAS à partir d'un fichier ASCII*). — Récupérer les fichiers `yeux-cheveux.txt` et `notes.txt` à l'adresse

`http://www-math.univ-poitiers.fr/~???/1m09/yeux-cheveux.txt`

`http://www-math.univ-poitiers.fr/~???/1m09/notes.txt`

Q 1. Reproduire et faire marcher le programme SAS suivant (noter que l'instruction `INPUT` est placée après `INFILE`) :

```
DATA TPSAS.YeuxChev1;
     INFILE "Z:\...\TPSAS\TP1\yeux-cheveux.txt" DLM=",";
     INPUT _IDCOURT_ V1$ V2$;
     LABEL V1="Couleur Yeux" V2="Couleur Cheveux";
RUN;

PROC PRINT NOOBS LABEL;
RUN;
```

```
PROC PRINT NOOBS LABEL;
     VAR V1 V2;
RUN;
```

Q 2. Un autre exemple avec des options système pour l'impression. Les données sont relatives à 88 étudiants d'une université anglaise ayant passé 5 épreuves notées par des notes entières de 0 à 100. Les 5 épreuves sont respectivement : Mécanique, Algèbre Linéaire, Algèbre des Structures, Analyse et Statistique.

```
OPTIONS PAGESIZE=64 LINESIZE=78 NODATE;

FOOTNOTE "Fichier des notes";
TITLE "TP No 1 Exercice 2";

DATA TPSAS.notes;
     INFILE "Z:\...\notes.txt";
     INPUT ind MECA ALIN ALGB ANLS STAT;
RUN;

PROC PRINT NOOBS;
RUN;
```

Exercice 3 (*Manipulation de tables*). — Q 1. On reprend le fichier `YeuxChev1` de l'Exercice 2 dans lequel on élimine la variable `_IDCOURT_`.

```
DATA TPSAS.YeuxChev2;
     SET TPSAS.YeuxChev1 (KEEP=V1 V2);
RUN;

PROC PRINT NOOBS;
RUN;
```

Création de la table de contingence associée au croisement des variables `V1` et `V2`.

```
PROC FREQ DATA=TPSAS.YeuxChev2;
     TABLE V1*V2 / OUT=TPSAS.Tab.cont.yeuxchev2;
RUN;
```

On pourra sauvegarder la table en format RTF à partir de `Fichier/Enregistrer sous` avec la fenêtre *Sortie* active.

Q 2. On va créer deux tables en sélectionnant des individus de la table `notes.sas7bdat` puis reconstruire la table originale en empilant les deux tables. Faire d'autres essais en fixant des conditions sur les notes.

```
DATA TPSAS.base1.note;
     SET TPSAS.notes (WHERE =(ind <= 30));
RUN;

PROC PRINT;
RUN;

DATA TPSAS.base2.note;
     SET TPSAS.notes (WHERE=(ind > 30));
RUN;

DATA TPSAS.base.note.complete;
     SET TPSAS.base1.note TPSAS.base2.note;
RUN;
```

```
PROC PRINT;
RUN;
```

Q 3. Création de nouvelles variables par « calculs » à partir d'une table, ici `TPSAS.notes`. On calcule quelques indicateurs supplémentaires pour chaque individu.

```
DATA TPSAS.notes.gene;
     SET TPSAS.notes;
     ATTRIB Resultat LABEL="Résultat final" FORMAT=$7.;
     ATTRIB Moyenne.generale LABEL="Moyenne générale du candidat" FORMAT=6.2;
     Score.total=SUM(meca,alin,algb,anls,stat);
     Moyenne.generale=MEAN(meca,alin,algb,anls,stat);
     Note.max=MAX(meca,alin,algb,anls,stat);
     Note.min=MIN(meca,alin,algb,anls,stat);
     IF Moyenne.generale >= 50 THEN DO;
          Resultat="Admis";
     END;
     ELSE DO;
          Resultat="Ajourné";
     END;
RUN;
```

Q 4. Création de deux tables découpant le fichier `TPSAS.notes` en deux par sélection de variables et reconstitution par `MERGE`.

```
DATA TPSAS.base.c1b.note;
     SET TPSAS.notes (KEEP=ind meca alin stat);
RUN;

PROC PRINT NOOBS;
RUN;

DATA TPSAS.base.c2b.note;
     SET TPSAS.notes (KEEP=algb anls stat );
RUN;

PROC PRINT NOOBS;
RUN;

DATA TPSAS.base.cbis.note.complete;
     MERGE TPSAS.base.c1b.note TPSAS.base.c2b.note;
RUN;

PROC PRINT NOOBS;
RUN;
```

Comme précédemment mais avec une variable commune dans les deux tables et sans utiliser l'instruction `BY` pour spécifier la variable commune, ici `IND` (mais il y a écrasement des valeurs de la dernière table citée).

```
DATA TPSAS.base.c1.note;
     SET TPSAS.notes (KEEP=ind meca algb stat);
RUN;

PROC PRINT NOOBS;
RUN;
```

```
DATA TPSAS.base.c2.note;
     SET TPSAS.notes (DROP=ind meca algb stat);
RUN;

PROC PRINT NOOBS;
RUN;

DATA TPSAS.base.c.note.complete;
     MERGE TPSAS.base.c1.note TPSAS.base.c2.note;
RUN;

PROC PRINT;
RUN;
```

Comme précédemment mais avec deux variables communes `ind` et `stat`, et spécification de `ind` comme variable de jointure dont les valeurs permettront d'identifier les observations ou lignes. Attention les variables de jointure ou communes doivent être totalement ordonnées pour l'identification des individus et réaliser une fusion.

```
DATA TPSAS.base.c1t.note;
     SET TPSAS.notes (KEEP=ind meca alin stat);
RUN;

DATA TPSAS.base.c2t.note;
     SET TPSAS.notes (KEEP=ind stat algb anls);
RUN;

DATA TPSAS.base.cter.note.complete;
     MERGE TPSAS.base.c1t.note TPSAS.base.c2t.note;
     BY ind;
RUN;

PROC PRINT NOOBS;
RUN;
```

Exercice 4 (*ODS*). — Quelques exemples de redirection des sorties des procédures vers des formats de type PDF, RTF ou HTML (récupérable sous EXCEL).

Q 1. Une table en PDF, HTML ou RTF.

```
ODS PDF;
PROC PRINT DATA=TPSAS.TableElection04 NOOBS;
RUN;
ODS PDF CLOSE;

ODS HTML;
PROC PRINT DATA=TPSAS.TableElection04 NOOBS;
RUN;
ODS HTML CLOSE;

ODS RTF;
PROC PRINT DATA=TPSAS.TableElection04 NOOBS;
RUN;
ODS RTF CLOSE;
```

Q 2. Une redirection de la table de contingence croisant les variables $\langle yeux \rangle$ et $\langle cheveux \rangle$ vers un fichier PDF, puis LaTeX.

```
ODS PDF;
PROC FREQ DATA=TPSAS.YeuxChev2;
     TABLE V1*V2;
RUN;
PROC PRINT;
RUN;
ODS PDF CLOSE;

ODS latex STYLE=journal FILE="Z:\...\TPSAS\toto.tex";
PROC FREQ data=TPSAS.YeuxChev2;
     TABLES V1*V2;
RUN;
ODS latex CLOSE;
```

À l'issue de ce premier TP, il est vivement conseillé de se reporter au cours polycopié en ligne pour bien s'approprier les premières manipulations de SAS. Pour cela, nous conseillons de lire attentivement les chapitres 1 et 2. Il est également conseillé, en dehors des séances de TP, de s'entraîner à la manipulation de SAS.

$***$ Fin $***$

Université de Poitiers
Département de Mathématiques

1M09 – Statistique Descriptive
Master MMAS 1, année 2009–10

TRAVAUX PRATIQUES N° 2. — FONCTIONS GRAPHIQUES SOUS SAS

Résumé. — L'objet de ce TP est de se familiariser avec des procédures de statistique descriptive de données uni/bidimensionnelle. Une part importante sera consacrée aux procédures graphiques.

Préliminaires. — Se connecter à ENS/UFR et comme lors de la première séance, allouer une bibliothèque de stockage de vos fichiers via l'instruction `LIBNAME`. Reprendre le même répertoire que lors de la première séance pour pouvoir accéder aux fichiers de données déjà créés (ici `TPSAS`).

1. **Des rappels sur quelques instructions globales**

Diverses commandes générales peuvent être rajoutées au début d'un programme SAS.

- `OPTION(S)` : il est ici question des options système et à spécifier en début de programme. On les déclare avec la commande `OPTION(S)` (le « s » est facultatif). En voici trois :
  1. `PAGESIZE`, qui spécifie le nombre de lignes dans une page de *Sortie*.
  2. `LINESIZE`, qui spécifie le nombre de caractères par ligne dans une page de *Sortie*.
  3. `NODATE`, qui supprime l'impression de la date dans les sorties.
- `TITLE` : permet de placer un titre en haut de chaque page des sorties textuelles et graphiques.

  ```
  TITLE "Ceci est un titre";
  ```

  On peut obtenir un titre sur plusieurs niveaux (jusqu'à 10) en spécifiant `TITLE1`, `TITLE2`, …, `TITLE10`. Un titre reste actif durant toute une session tant que l'on ne réinitialise pas la commande. Pour cela, il suffit de vider le contenu en exécutant l'instruction

  ```
  TITLE;
  ```

- `FOOTNOTE` (note de pied-de-page) : permet de placer un titre en bas de chaque page des sorties. Elle permet également une meilleure mise-en-page des sorties SAS sur une imprimante. Cette commande dispose des mêmes propriétés et extensions que `TITLE`.

  ```
  FOOTNOTE "Ceci apparaîtra en bas de page";
  ```

- Commentaires : des commentaires peuvent être insérés n'importe où dans un programme SAS de la façon suivante :

  ```
  /* ceci est un commentaire */
  ```

  (surtout commode au sein d'une ligne de commande)

  ```
  * ceci en est un autre;
  ```

- `GOPTION(S)` : spécifie des options valables pour les graphiques. Elles restent actives durant toute la session SAS tant que l'on ne réinitialise pas. Pour des graphiques en noir et blanc :

  ```
  GOPTIONS COLORS=(BLACK);
  ```

Pour réinitialiser l'option précédente, il suffit de faire

```
GOPTIONS RESET=COLORS;
```

On réinitialise toutes les options avec

```
GOPTIONS RESET=ALL;
```

- PATTERN⟨n⟩ : permet de spécifier la n-ième couleur employée pour l'ensemble des graphiques d'une session.

```
PATTERN1 COLOR=blue;
PATTERN2 COLOR=yellow;
PATTERN3 COLOR=black;
PATTERN5 COLOR=red;
PATTERN4 COLOR=green;
PATTERN6 COLOR=purple;
```

Il est recommandé, après chaque appel à une procédure graphique, de faire suivre l'instruction `RUN;` par `QUIT;`

2. **Données unidimensionnelles**

2.1. Variable quantitative

2.1.1. *Variable quantitative discrète. — Diagramme en bâtons*

Exercice 1. — Q 1. Créer une table SAS de nom ⟨*NbreEnfants*⟩ associée à Tab. 1. Cette création peut être réalisée via la saisie en ligne des données (cf TP n° 1) ou via `Menu Outils/Éditeur` de tables SAS

Nb enfants	0	1	2	3	4	5	6
Effectif	235	183	285	139	88	67	3
Fréquence	0.235	0.183	0.285	0.139	0.088	0.067	0.03

Table 1. — Nombre d'enfants à partir de 1000 couples

Q 2. Voici un premier code qui trace un diagramme en bâtons (de base) à partir de la procédure `PLOT`. Cette procédure est essentiellement dédiée à la construction de graphes bidimensionnels.

```
/* AXISn définit le style employé pour le n-ième type d'axes */

AXIS1 LABEL=("Nb Enfants" JUSTIFY=RIGHT);
AXIS2 LABEL=("Effectifs") ORDER=(0 TO 290 BY 10) offset=(0,);

/* ORDER fixe le domaine sur le second axe et le placement
d'une marque ici tous les 10 unités OFFSET pour figer
l'écart entre l'origine du dessin et la 1ère marque (.,)
et entre la dernière marque (ou ticks) (,.) et la fin du
tracé de l'axe */

PROC GPLOT DATA=TPSAS.NbreEnfants;
     SYMBOL1 INTERPOL=NEEDLE VALUE=DOT;
     PLOT Effectif*Nb_enfants / HAXIS=AXIS1 VAXIS2=AXIS2;
RUN;
QUIT;
```

Q 3. Un second code d'obtention d'un diagramme en bâtons de forme classique à partir de la procédure GCHART.

```
AXIS1 LABEL=("Nb Enfants" JUSTIFY=RIGHT);
AXIS2 LABEL=("Effectif") ORDER=(0 TO 290 BY 10) offset=(0,);

PROC GCHART DATA=TPSAS.NbreEnfants;
     VBAR Nb_enfants / DISCRETE SUMVAR=Effectif MAXIS=AXIS1 RAXIS=AXIS2;
     /* l'option PATTERNID=MIDPOINT permet d'avoir une couleur par bâton */
RUN;
QUIT;
```

Fonction de répartition

Exercice 2. — Q 1. À partir du fichier `NbreEnfants`, on construit une nouvelle table SAS de nom `NbreEnfantsPlus` qui reprend les trois variables de `NbreEnfants`, auxquelles on adjoint deux nouvelles variables :

- ⟨*Eff_cumule*⟩ : effectifs cumulé ;
- ⟨*Freq_cumule*⟩ : fréquences cumulées qui seront calculées à partir des variables initiales.

L'objectif est de disposer des données nécessaires à la construction de la fonction de répartition de la variable ⟨*Nb_enfants*⟩.

```
DATA TPSAS.NbreEnfantsPlus;
     SET TPSAS.NbreEnfants;
     BY Nb_enfants;
     RETAIN Eff_cumule Freq_cumule (0,0);
     Eff_cumule=Eff_cumule+Effectif;
     Freq_cumule=Freq_cumule+Freq;
     ATTRIB Eff_cumule LABEL="Effectifs cumulés";
     ATTRIB Freq_cumule LABEL="Fréquences cumulées";
RUN;
```

Q 2. Construction du diagramme en bâtons des effectifs et fréquences cumulés à partir de GCHART : faire selon le principe de l'Exercice 1.3.

Q 3. Un autre code construisant la fonction de répartition empirique avec GPLOT :

```
PROC GPLOT DATA=TPSAS.NbreEnfantsPlus;
     AXIS1 LABEL=("Nbre enfants" JUSTIFY=RIGHT) ORDER=(0 TO 6 BY 1);
     AXIS2 LABEL=("Fréquence" JUSTIFY=RIGHT "Cumulée")
          ORDER=(0 TO 1 BY 0.1) OFFSET=(1,);
     SYMBOL1 INTERPOL=STEP COLOR=blue;
     PLOT Freq_cumule*Nb_enfants / HAXIS=AXIS1 VAXIS=AXIS2 GRID;
     /* GRID permet d'avoir une grille en pointillés sous-jacente
     au graphique */
RUN;
QUIT;
```

2.1.2. *Variable quantitative continue. — Fonction de répartition*

Exercice 3. — Q 1. Créer une table SAS de nom `Ampoule` associée à Tab. 2 avec un nom de variable ⟨*duree*⟩.

91.6	35.7	251.3	24.3	5.4	67.3	170.9	9.5	118.4	57.1

Table 2. — Durées de vie de 10 ampoules

Q 2. On obtient un histogramme basé sur cinq classes et à pas fixe $h = 52$:

```
PROC GCHART DATA=TPSAS.ampoule;
     VBAR DUREE / LEVELS=5 MIDPOINTS=26 TO 234 BY 52 TYPE=FREQ;
     /* Remplacer FREQ par CFREQ pour avoir un histogramme des
     effectifs cumulés */
RUN;
```

Q 3. À partir de l'histogramme précédent, créer une nouvelle table SAS `AmpouleHisto` comportant deux variables, ⟨*freq_cumule*⟩ et ⟨*midpoint*⟩, donnant les fréquences cumulées et les abscisses des points-milieux des classes. On inclura une ligne initiale supplémentaire comportant les données $(0, 0)$. Puis tracer le polygone des fréquences cumulées avec le code :

```
AXIS1 LABEL=("Midpoint" JUSTIFY=RIGHT);
AXIS2 LABEL=("Fréquences" JUSTIFY=RIGHT "Cumulées");
AXIS3 LABEL=("Fréquences" JUSTIFY=left "Cumulées");

PROC GBARLINE DATA=TPSAS.Ampoulehistobis;
     BAR MIDPOINT / DISCRETE SUMVAR=Freq_cumule AXIS=AXIS2;
     PLOT / SUMVAR=Freq_cumule RAXIS=AXIS3;
RUN;
QUIT;
```

Q 4. Explorer la fonctionnalité de SAS/INSIGHT,

```
Outils/Analyse Interactive des Données/Analyze/Distribution (Y)
```

sur la table initiale.

2.2. Variable qualitative

2.2.1. *Diagrammes en bâtons et en barres.* — Exercice 4 (*Diagrammes en bâtons*). — Il s'agit ici de représenter la variable ⟨*couleur des yeux*⟩ dans une population donnée via un diagramme en bâtons

```
AXIS1 LABEL=("Effectifs") ORDER=(0 TO 240 BY 20);

PROC GCHART DATA=TPSAS.Tab_cont_yeuxchev2;
     VBAR V1 / SUMVAR=COUNT RAXIS=AXIS1 INSIDE=SUM;
RUN;
```

Exercice 5 (*Diagrammes en barres*). — Avec des barres verticales :

```
AXIS1 LABEL=NONE ORDER=(0 TO 215);

PROC GCHART DATA=TPSAS.Tab_cont_yeuxchev2 (FIRSTOBS=1 OBS=4);
     VBAR V1 / NOAXIS SUMVAR=COUNT SUBGROUP=V2 RAXIS=AXIS1
          INSIDE=SUM OUTSIDE=SUM;
RUN;
```

Puis la seconde ⟨*couleur des yeux*⟩ :

```
AXIS1 LABEL=NONE ORDER=(0 TO 215);

PROC GCHART DATA=TPSAS.Tab_cont_yeuxchev2 (FIRSTOBS=5 OBS=8);
     VBAR V1 / NOAXIS SUMVAR=COUNT SUBGROUP=V2 RAXIS=AXIS1
          INSIDE=SUM OUTSIDE=SUM;
RUN;
```

Avec des barres horizontales :

```
AXIS1 ORDER=(0 TO 220 BY 10) LABEL=NONE;
AXIS2 LENGTH=1cm;

PROC GCHART DATA=TPSAS.Tab_cont_yeuxchev2 (FIRSTOBS=5 OBS=8);
     HBAR V1 / SUMVAR=COUNT SUBGROUP=V2 RAXIS=AXIS1 MAXIS=AXIS2
          OUTSIDE=SUM SUMLABEL="Effectif total"
          CTEXTSIDE=PURPLE REF=7 75 194;
RUN;
QUIT;
```

Là encore vous pouvez remplacer les instructions `HBAR` et `VBAR` par `HBAR3D` et `VBAR3D` pour voir l'effet obtenu.

2.3. Diagramme en secteurs

Exercice 6. — Reprendre la table SAS `TableElection04` créée lors du TP n° 1.

Q 1. Tracer un diagramme en camembert avec le code suivant :

```
PROC GCHART DATA=TPSAS.tableelection04;
     PIE Nom / SUMVAR=pourcentage NOHEADING SLICE=OUTSIDE
     PERCENT=OUTSIDE ASCENDING OTHER=0 VALUE=NONE;
     /* COUTLINE=SAME donne la même couleur au bord
     et à l'intérieur d'une tranche */
RUN;
QUIT;
```

Une autre présentation avec l'utilisation de l'option globale `LEGEND` qui permet de remplacer la légende par défaut proposée par la procédure :

```
LEGEND1 LABEL=NONE
     POSITION =(LEFT MIDDLE) /* position de la légende par rapport
               au graphique */
     OFFSET=(4,) ACROSS=1 /* nombre de colonnes pour la légende */
     VALUE=(COLOR=BLACK)
     SHAPE=BAR(4,1.5); /* spécification de la taille de la bo\^\i te
               pour chaque catégorie */

PROC GCHART DATA=TPSAS.tableelection04;
     PIE nom / SUMVAR=pourcentage NOHEADING ASCENDING
          COUTLINE=SAME OTHER=0 PERCENT=OUTSIDE LEGEND=LEGEND1;
RUN;
QUIT;
```

Enfin, voir le résultat en remplaçant l'instruction `PIE` par `PIE3D`.

3. Données bidimensionnelles

3.1. Boîtes à moustaches en parallèle

Exercice 7. — Récupérer le fichier `pento.txt` à l'adresse

`http://www-math.univ-poitiers.fr/~???/1m09/pento.txt`

Ensuite, créer la table SAS correspondante via une étape `DATA` (comme dans le TP n° 1). Les trois variables sont ⟨*code*⟩, ⟨*rythme*⟩, et ⟨*facteur*⟩. La première et la troisième sont de type chaîne de caractères, la seconde est numérique.

Q 1. Une version de base :

```
PROC GPLOT DATA=TPSAS.pento;
     SYMBOL1 INTERPOL=BOX;
     AXIS1 LENGTH=5cm OFFSET=(1cm,1cm);
     AXIS2 LENGTH=5cm;
     PLOT rythme*facteur=1 / VAXIS=AXIS2 VMINOR=1 HAXIS=AXIS1;
RUN;
QUIT;

GOPTIONS RESET=ALL;
```

Q 2. Une version avec `BOXPLOT` :

```
PROC BOXPLOT DATA=TPSAS.pento;
     PLOT rythme*facteur / CBOXES=black CBOXFILL=yellow
          ALLLABEL=VALUE CTEXT=black;
     INSET NOBS / HEIGHT=3 HEADER="Nombre d'observations" POSITION=NW;
RUN;
```

3.2. Diagrammes en barres en parallèle

Pour illustrer les deux prochains diagrammes, on reprend la base SAS `YeuxChev1` du TP n° 1. Une étape préliminaire consiste à créer une table SAS sauvegardant la table complète générée par la procédure `FREQ`, à savoir la table de contingence mais également les « tableaux » de profils-lignes et colonnes. Pour cela exécuter le code :

```
PROC FREQ DATA=TPSAS.YeuxChev1;
     TABLE V1*V2 / OUT=TPSAS.Tab_freq_yeuxchev3 OUTPCT;
     /* l'option OUTPCT permet de compléter la sauvegarde */
RUN;

PROC PRINT DATA=TPSAS.Tab_freq_yeuxchev3;
RUN;
```

Les deux variables contenant les profils-lignes et profils-colonnes sont nommée par SAS `PCT_ROW` et `PCT_COL`.

Exercice 8. — Ce code fait la juxtaposition des diagrammes en barres pour chaque classe de profils :

```
AXIS1 LABEL=NONE;
PROC GCHART DATA=TPSAS.Tab_cont_yeuxchev3;
     VBAR V1 / SUMVAR=PCT_ROW SUBGROUP=V2 RAXIS=AXIS1 INSIDE=SUM;
RUN;
```

```
AXIS1 LABEL=NONE;
PROC GCHART DATA=TPSAS.Tab_cont_yeuxchev3;
     VBAR V2 / SUMVAR=PCT_COL SUBGROUP=V1 RAXIS=AXIS1 INSIDE=SUM;
RUN;
```

3.3. Nuage de points

Récupérer la table SAS `denrees.sas7bdat` à l'adresse

`http://www-math.univ-poitiers.fr/~???/1m09/denrees.sas7bat`

Ouvrir la base pour voir sa structure.

Exercice 9. — Q 1. On obtient différents nuages de points en croisant deux des variables de la base via le code :

```
PROC GPLOT DATA=TPSAS.denrees;
     SYMBOL1 INTERPOL=NONE VALUE=DOT;
     PLOT V1*V3 / GRID;
RUN;
QUIT;
```

Q 2. Un nuage de points avec les étiquettes des individus associés aux points du nuage :

```
PROC GPLOT DATA=TPSAS.denrees;
     SYMBOL1 INTERPOL=NONE VALUE=DOT;
     PLOT V1*V3=_idlong / GRID;
RUN;
QUIT;
```

Q 3. Génération de combinaisons deux à deux d'un ensemble de variables :

```
SYMBOL1 INTERPOL=NONE VALUE=DOT COLOR=red;
SYMBOL2 INTERPOL=NONE VALUE=DOT COLOR=blue;

PROC GPLOT DATA=TPSAS.denrees;
     PLOT V1*V3=1 V1*V4=2/ GRID;
RUN;
QUIT;

GOPTIONS RESET=ALL;
```

ou (et constater la différence)

```
PROC GPLOT DATA=TPSAS.denrees;
     PLOT V1*(V3 V4);
RUN;
QUIT;

GOPTIONS RESET=ALL;
```

Q 4. Vous pourrez explorer la fonctionnalité de SAS/INSIGHT,

`Outils/Analyse Interactive des Données/Analyze/Scatter Plot (Y X)`,

pour obtenir divers nuage de points associée à la base `denrees.sas7bat`

Remarques. — *a*) Utiliser l'option `OVERLAY` pour superposer plusieurs graphiques.

b) Pour le scatter-plot 3D, utiliser la procédure `SCATTER`.

c) Il est possible de faire une mise au point et sauvegarder des graphiques lors d'une session SAS. L'utilisation de l'instruction `ODS` permet d'obtenir une sortie en postscript. Noter que cette sortie est disponible sous le nom générique `sasprt.ps` placée en général dans le répertoire `"C:\Document and Settings\VotreNom"`.

4. **Quelques procédures statistiques élémentaires**

4.1. MEANS et SUMMARY

Exercice 10 (*Sorties comparées des deux procédures*). — On applique les deux procédures aux 6 variables numériques de la base `denrees.sas7bat`. Le résultat est alors stocké dans une table SAS.

Q 1. Procédure `MEANS` :

```
PROC MEANS DATA=TPSAS.denrees MAXDEC=2;
     VAR V1-V6;
     OUTPUT OUT=TPSAS.Means_denrees;
RUN;
```

Il est possible de demander le calcul d'autres indicateurs. Par exemple :

```
PROC MEANS DATA=TPSAS.denrees MEAN STD CV MAX MIN MEDIAN MAXDEC=2;
     VAR V1-V6;
RUN;
```

Il est possible d'obtenir encore d'autres informations, comme un intervalle de confiance pour la moyenne d'une variable ainsi que le résultat d'un test statistique de nullité de cette moyenne.

Cette procédure peut également être utilisée pour calculer des caractéristiques numériques de variables pour des sous-populations, en spécifiant une variable qualitative définissant les différents sous-groupes. Par exemple, reprendre la base `pento.sas7bat` et calculer des résumés de la variable $\langle rythme \rangle$ par classes définies par la variable $\langle facteur \rangle$:

```
PROC MEANS DATA=TPSAS.pento MAXDEC=2;
     VAR rythme;
     CLASS facteur;
RUN;
```

Q 2. Procédure `SUMMARY` : essentiellement les mêmes caractéristiques excepté que, par défaut, aucun affichage n'est fourni.

4.2. UNIVARIATE

La procédure `UNIVARIATE` fournit elle aussi des résumés statistiques pour des variables numériques. Cependant, elle propose de nombreux résumés supplémentaires et en particulier des quantités classiques liés à la distribution d'une variable comme les quantiles et des résultats de tests de normalité. L'option `PLOT` permet d'obtenir des premières représentations graphiques (en particulier le `QQPLOT` introduit en TD) mais dont la qualité est très faible.

Par exemple appliquons cette procédure aux données de la base `notes_gene.sas7bat` créée lors du premier TP. On regarde les variables $\langle MECA \rangle$, $\langle STAT \rangle$ et enfin $\langle Moyenne\_generale \rangle$.

```
PROC UNIVARIATE DATA=TPSAS.notes_gene NORMAL PLOT;
     VAR STAT Moyenne_generale;
RUN;
```

4.3. FREQ

La procédure `FREQ` a déjà été utilisée dans le TP n° 1, pour obtenir la table de contingence croisant deux variables qualitatives puis dans la sous-section 3.2 pour stocker les profils-lignes et colonnes. Des options sont proposées pour tester, par exemple, l'indépendance des deux variables à partir de diverses statistiques dont celle du χ^2.

```
PROC FREQ DATA=TPSAS.Yeuxchev1;
     TABLE V1*V2 / CHISQ;
RUN;
```

4.4. CORR

La procédure `CORR` permet, en particulier, le calcul de la matrice de (covariance avec l'option `COV`) corrélation d'un ensemble de variables numériques ainsi que de tests d'hypothèses de nullité de ces coefficients de corrélation. On applique cette procédure aux 6 variables de la base `denrees.sas7bat` avec sauvegarde dans la table `corr_denrees.sas7bat` :

```
PROC CORR DATA=TPSAS.denrees OUT=TPSAS.corr_denrees;
     VAR V1-V6;
RUN;
```

*** Fin ***

TRAVAUX PRATIQUES N° 3. — ACP SOUS SAS

Résumé. — L'objet de ce TP est l'exploitation d'une procédure classique d'analyse multidimensionnelle, l'analyse en composantes principales. De plus, une brève illustration du concept de macro sera proposée.

Préliminaires

1. Se connecter à ENS/UFR et comme lors de la première séance, allouer une bibliothèque de stockage de vos fichiers via l'instruction `LIBNAME`. Reprendre le même répertoire que lors de la première séance pour pouvoir accéder aux fichiers de données déjà créés (ici `TPSAS`).
2. Récupérer le fichier compressé comportant les données du jour à l'adresse

   `http://www-math.univ-poitiers.fr/~???/1m09/datatp3.zip`

   Extraire la totalité des fichiers dans `TPSAS`.

1. ACP via SAS/INSIGHT

On reprend les données du fichier `denrees.sas7bdat` utilisée dans le second TP. Ces données ont été utilisées pour illustrer le cours sur l'ACP. Nous allons retrouver les principaux résultats avec SAS/INSIGHT.

Accéder au module de traitement interactif des données via

`Solutions/Analyses/Analyse de données interactives`

et ouvrir alors la table `denrees` dans le répertoire `TPSAS`.

1. Tracer la matrice des nuages de points croisant les variables 2 à 2 afin de dépister d'éventuelles relations non linéaires qui poseraient des problèmes : une relation non linéaire entre deux variables n'est pas prise en compte en ACP (linéaire).
   - Choisir `Analyze/Scatterplot` dans le menu principal : sélectionner les variables $\langle V_1 – V_8\rangle$ en `X`, les variables $\langle V_1 – V_8\rangle$ en `Y` et la variable $\langle$*_IDLONG_*$\rangle$ en `Label`.
   - Exécuter l'analyse en composantes principales.
   - Choisir `Analyze/Multivariate (YX)` puis
   - Sélectionner les variables $\langle V_1\rangle$ à $\langle V_8\rangle$ dans `Y` et la variable $\langle$*_IDLONG_*$\rangle$ en `Label`.
   - Bouton `Method` : cocher `Correlation Matrix` pour une ACP réduite, `N` comme diviseur de la variance (*variance divisor*).
   - Bouton `Output` : décocher `UNIVARIATE` et `CORR` ; cocher `Principal Component Analysis` (il est possible ici également d'obtenir une matrice de nuages de points).
   - Bouton `Principal Component Options` : décocher `Automatic` et cocher `All` ; cocher dans `Output components : All` ; dans `Component Plots : First 2 components` (représentera alors le nuage des individus sur le premier plan factoriel ; si `3 components` alors le nuage des individus dans le repère des trois premiers axes factoriels).

2. Choisir la dimension à partir des valeurs propres. Est-ce facile sans graphique ?
3. Pour aider, tracer les boîtes à moustaches (ou *box-plot*) des composantes principales via `Analyse/Box Plot` : Sélectionner `PCV1` à `PCV8` dans `Y`.
4. *a*) Que vaut la variance de chacune des composantes (Faire `Analyse/Distribution (Y)` et sélectionner les variables $\langle V_1 - V_8 \rangle$) ?

   b) De façon générale, pour s'assurer de la robustesse d'un axe que l'on voudrait conserver, il est bon de vérifier que l'analyse calculée sans individus influents est identique, c'est-à-dire qu'elle conduit aux mêmes axes factoriels. Dans le cas contraire, une discussion avec les commanditaires ou spécialistes du domaine concerné s'impose. S'agit-il d'une erreur de mesure, d'échantillonnage ? Faut-il ou non conserver une observation atypique dans les données ou faut-il conserver une composante avec des valeurs atypiques ? Cliquer sur ces points, puis `Edit/Observations/Exclude in calculation` réexécute les calculs sans ces points. L'axe 2 est-il modifié (corrélations variables facteurs) ?
5. « L'habillage » des points de nuage des individus sur le premier plan factoriel peut être réalisé via une sélection par encadrement de la zone d'intérêt au pointeur de la souris, puis en allant dans le menu `Edit/Observations`. Le graphique et les tableaux peuvent être sauvegardés (individuellement ou par sous-groupe via une sélection à la souris) comme un graphique ou une image via le menu `File/Graphics Catalog`. Il y aura sauvegarde dans le répertoire choisi d'un catalogue de graphiques de nom « insight » par défaut.
6. Enfin, notons que le cercle des corrélations n'est pas proposé par SAS/INSIGHT.

2. **ACP via la procédure** `PRINCOMP`

Exercice 1. — SAS dispose de la procédure `PRINCOMP` pour réaliser l'ACP normée (ou non).

Q 1. Mettre cette procédure en œuvre, sur les données dans la table `denrees.sas7bdat`, à l'aide des commandes suivantes :

```
PROC PRINCOMP DATA=TPSAS.denrees;
RUN;
```

Qu'obtient-on comme résultats ? Commenter.

Q 2. Faire ensuite :

```
PROC PRINCOMP DATA=TPSAS.denrees
     VARDEF=N /* COV si ACP non normée */
     OUT=TPSAS.denrees_acp1 /* Données initiales
          + Composantes principales */
     OUTSTAT=TPSAS.denrees_acp2; /* Table stockant les résumés
     de base pour les variables initiales + Corrélation/covariances
     + Valeurs propres et Variables principales */
RUN;

PROC PRINT DATA=TPSAS.denrees_acp1 NOOBS;
RUN;

PROC PRINT DATA=TPSAS.denrees_acp2 NOOBS;
RUN;
```

Explorer les sorties standard de la procédure `PRINCOMP` avec le contenu des tables `denrees_acp1.sas7bdat` et `denrees_acp2.sas7bdat`.

- Noter que dans `denrees_acp1.sas7bdat`, la variable $\langle prin \rangle_\ell$ donne le vecteur des coordonnées des individus sur l'axe factoriel u_ℓ, i.e. le vecteur $C_\ell = \big([Xu_\ell](i)\big)_i$, i.e. la ℓ-ième composante principale avec les notations du cours. Par contre dans le fichier `denrees_acp2.sas7bdat`, la ligne $\langle prin \rangle_\ell$ désigne la composante principale ℓ comme combinaison linéaire des variables originales :

$$C_\ell = \sum_j u_\ell(j) \left[\frac{X_j - \bar{X}_j}{s_{X_j}} \right]$$

  avec les notations du cours.
- Noter que les coefficients de corrélation variables-composantes principales ne sont pas disponibles.

On rappelle que cet élément donne les éléments de construction de la représentation graphique du nuage des variables.

3. Macro SAS et ACP

Dès que l'on souhaite écrire des programmes SAS suffisamment généraux afin, par exemple, de les appliquer à différents jeux de données, il est nécessaire, par souci d'efficacité, de faire appel aux ressources de SAS permettant de définir des macro-variables et des macro-commandes. Le principe général consiste à associer une chaîne de caractères, une suite de commandes ou, un texte à un identificateur. Par la suite, toute occurrence de cet identificateur est remplacée par le texte désigné au cours d'un traitement préalable à l'exécution proprement dite.

Le pré-processeur implicitement invoqué reconnaît différents objets : variables, commentaires, commandes, fonctions, arguments, qui lui sont propres (précédés des caractères `&` ou `%`) ; ils lui confèrent les possibilités d'un langage de programmation rudimentaire mais structuré. Les macro-variables et macro-commandes sont connues, sauf déclaration explicite contraire (`%GLOBAL`, `%LOCAL`), dans l'environnement dans lequel elles sont déclarées : globalement pour toute une session SAS ou localement à l'intérieur d'une macro.

3.1. Macro-variables

La déclaration d'une macro-variable consiste à associer par la commande `%LET` une chaîne de caractères à un identificateur :

```
%LET nomvar1=v1;
%LET nomvar2=v2;
%LET varlist=csp pao paa vio via pdt;
```

Certaines déclarations sont implicites : compteur d'une boucle `%DO`, paramètres d'une macro-commande, ... Dans la suite du programme, les références à une macro-variable sont précédées du caractère `&` :

```
TITLE "\'Etude des variables &varlist";

PROC PLOT DATA=TPSAS.denrees;
     PLOT &nomvar1*&nomvar2;
RUN;
```

Une macro-variable peut contenir des commandes ou instructions SAS mais, dans ce cas, il est préférable de définir une macro-commande.

3.2. Macro-commandes

La déclaration d'une macro-commande (ou macro tout court) suit les mêmes principes. Elle peut inclure des paramètres qui deviennent des macro-variables locales :

```
%MACRO impdat(in);
%* ceci est un commentaire;
     PROC PRINT DATA=&in NOOBS;
     RUN;
%MEND impdat;
```

l'exécution d'une macro est invoquée en faisant précéder son nom du caractère `%` :

```
/* Appel à la macro */
%impdat(TPSAS.denrees);
```

Une autre macro :

```
%MACRO plot(in,yvar,xvar);
     PROC PLOT DATA=&in;;
     PLOT &yvar*&xvar;
     RUN;
%MEND plot;
```

et son appel :

```
%plot(tp3.denrees,&nomvar1,&nomvar2);
```

3.3. Bibliothèque de macros

Une macro-commande doit être déclarée avant toute utilisation au cours d'un même session SAS. Lorsque certaines sont régulièrement sollicitées, il est économique de les stocker dans une bibliothèque de macros spécifiques à un utilisateur. Les règles suivantes sont à respecter :

1. Chaque déclaration de macro et une seule est enregistrée dans un fichier le nom de cette macro suivi de l'extension `.sas` : par exemple `impdat.sas`.
2. Tous ces fichiers sont regroupés dans un ou des répertoires qui constituent la ou les bibliothèques. Créer un répertoire `MesMacros` dans le dossier `TPSAS` et sauvegarder la macro sous `impdat.sas` dans ce répertoire.
3. Le ou les chemins d'accès sont spécifiés une fois pour toute dans le fichier `config.sas` du répertoire d'accueil de l'utilisateur ou lors d'une session SAS par la commande (*voir* la librairie `SASUSER`) via :

   ```
   OPTIONS SASAUTOS=(SASAUTOS "C:\...\TPSAS\MesMacros") MAUTOSOURCE MRECALL;
   ```

3.4. Des macros associées à l'ACP

Nous allons utiliser les fichiers `temp.txt` et `temp_tp.sas7bdat` extraits de l'archive `datatp3.zip` en début de TP.

Exercice 2 (*Une première macro de lecture de données*). — Q 1. Soumettre le programme de macro-commande ci-dessous à SAS. Il est adapté à la lecture de tout fichier texte de nom « in » contenant une première colonne d'identificateurs (⟨*ident*⟩) des observations suivie de plusieurs colonnes de variables (⟨*listevar*⟩) séparées, par défaut, par des espaces. La table crée est de nom ⟨*out*⟩ :

```
%MACRO lecacp(in,out,ident,listevar,dlm=" ");
     DATA &out;
     INFILE &in DLM=&dlm;
     INPUT &ident$ &listevar;
     RUN;
%MEND lecacp;
```

Sauvegarder cette macro dans votre répertoire `MesMacros`.

Q 2. Exécuter cette macro dans SAS afin de créer la table SAS à partir du fichier texte `C:\...\TPSAS\temp.txt` la table `temp.sas7bdat` :

```
%lecacp("C:\...\TPSAS\temp.txt", tp3.temp,ville,
     janv fevr mars avri mai juin juil aout sept oct nov dec);
%impdat(TPSAS.temp);
```

Récupérer le fichier compressé comportant une liste de macros à l'adresse

`http://www-math.univ-poitiers.fr/~???/1m09/macrostp3.zip`

Extraire tous les fichiers de l'archive `macrostp3.zip` dans votre répertoire `MesMacros`. Il s'agit d'adaptations de macros dues à P. Besse.

Exercice 3 (*Une macro donnant les tableaux d'indicateurs*). — Nous allons utiliser la macro `%acp` dont l'entête est le suivant :

```
%MACRO acp(dataset, ident, listev, red=, q=3, poids=);
%* ACP de dataset;
%*         ident : variable contenant les identificateurs des individus;
%*        listev : liste des variables (numériques);
%*    par defaut : réduites sinon red=cov;
%*             q : nombre de composantes retenues;
%*         poids : variable de pondération;
%*          pvar : nombre de variables;
%* options édition;
%global pvar;
```

Exécuter successivement les appels à la macro :

```
%acp(TPSAS.denrees,_IDLONG_, v1--v8);
%acp(TPSAS.tempf_tp,ville, janv--dec, red=cov);
%acp(TPSAS.tempf_tp,ville, janv--dec);
```

Sauvegarder pour chaque appel, le fichier de résultats sous le format liste (`.lst`).

Exercice 4 (*Des macros donnant les graphiques de base*). — Après avoir utilisé la macro `%acp`, exécuter successivement les quatre macros suivantes qui vont donneront les graphiques classiques d'une ACP :

```
/* \'ebouli des valeurs propres */
%gacpsx;
/* Boîtes à moustaches en parallèle des différentes
valeurs propres */
%gacpbx;
/* Nuage des individus sur le premier plan factoriel */
%gacpix;
/* Cercle des corrélations croisant les deux premières
composantes principales par défaut */
```

```
%gacpvx;
/* Cercle des corrélations croisant les composantes principales Nos 2 et 3 */
%gacpvx(x=2,y=3);
```

L'entête de la dernière macro :

```
%macro gacpvx(x=1, y=2, nc=4, coeff=1);
%* Graphique des variables avec cercle des corrélations;
%*  x : numéro axe horizontal;
%*  y : numéro axe vertical;
%* nc : nombre max de caractères;
```

Exercice 5 (*Variables supplémentaires*). — Dans le cas où des variables supplémentaires ou illustratives sont présentes dans le fichier de données, il est classique d'analyser leur corrélation avec les composantes principales obtenues. Pour cela, on utilisera la séquence suivante. Elle est à exécuter après l'application de la macro `%acp` car elle utilise la table `coorindq` créée par la macro. Noter que tous ces fichiers se trouvent stockés dans la librairie `WORK`.

Pour la table `tempf_tp`, quatre variables sont disponibles : $\langle Lat \rangle$ (latitude), $\langle Long \rangle$ (longitude), $\langle Tmoy \rangle$ (température moyenne), $\langle Amp \rangle$ (amplitude).

```
DATA tp3.corrvarsup;
     SET coorindq (KEEP=Prin1-Prin3);
     SET tp3.tempf_tp (KEEP=Lat Long Tmoy Amp);
RUN;

PROC PRINT NOOBS;
RUN;

PROC CORR DATA=tp3.corrvarsup NOSIMPLE OUT=tp3.rescorrvarsup;
     VAR PRIN1-PRIN3;
     WITH Lat Long Tmoy Amp;
RUN;
```

Vous pourrez transformer ce code sous forme de macro et l'inclure dans votre « librairie » de macros.

4. Analyse en composantes principales à rendre

Exercice 6 (*Évaluation sensorielle de 8 eaux minérales gazeuses*). — On compare 8 eaux gazeuses du point de vue de leur description sensorielle. Les données sont fournies dans la table `eaux_gaz.sas7bdat`. En lignes figurent les 8 eaux et en colonnes les 13 variables : 12 descripteurs sensoriels et une note d'appréciation globale du produit. Ces variables sont en fait des moyennes de notes attribuées par les membres d'un jury. Les notes individuelles variant entre 0 et 10, les moyennes ont, bien entendu, une amplitude plus faible.

La variable d'identification, les 12 descripteurs sensoriels et la note globale possèdent les noms suivants :

Produit	IntBulles	NbreBulles	TailleBulles	
HeteroBulles	Efferv	IntenGus	Amer	Sucree
Acide	Salee	Alcaline	IntCrepit	NoteGlobale

Vous devez réaliser une ACP sur ces données avec les 12 descripteurs sensoriels comme variables actives et la note d'appréciation générale comme variable illustrative.

Q 1. Analyser la qualité globale de la projection du nuage des individus sur le premier plan factoriel.

Q 2. Faire une synthèse de l'ACP si on retient que les deux premières composantes principales.

Q 3. Pourquoi la variable « Appréciation globale » a-t-elle été introduite en tant que variable illustrative ? À quoi sert-elle comme variable illustrative ? Est-elle bien représentée sur le premier cercle des corrélations ? Interpréter sa projection.

*** Fin ***

TP N° 4. — AFC ET RÉGRESSION LINÉAIRE

Résumé. — L'objet de ce TP est l'exploitation de autres procédures classiques d'analyse multidimensionnelle, l'Analyse Factorielle des Correspondances et la régression linéaire.

Préliminaires

1. Se connecter à ENS/UFR et comme lors de la première séance, allouer une bibliothèque de stockage de vos fichiers via l'instruction `LIBNAME`. Reprendre le même répertoire que lors de la première séance pour pouvoir accéder aux fichiers de données déjà créés (ici `TPSAS`).
2. Récupérer le fichier compressé comportant les données du jour à l'adresse

   `http://www-math.univ-poitiers.fr/~???/1m09/datatp4.zip`

   Extraire la totalité des fichiers sous `TPSAS`.

1. AFC via la procédure `CORRESP` et SAS/INSIGHT

Contrairement aux autres techniques factorielles qui ont été développées dans SAS il y a de nombreuses années, la procédure `CORRESP` est plus récente. En conséquence, son usage pour mettre en œuvre une analyse des correspondances simple ou multiple répond bien aux besoins des présentations récentes de ces techniques. Il n'est donc pas indispensable de faire appel à des macros pour en faciliter l'usage, seules quelques manipulations de base sont nécessaires pour obtenir des graphiques.

Exercice 1 (*Acquisition des données*). — Il s'agit des résultats du premier tour des élections présidentielles de 1995. On connaît, pour chacun des 95 départements métropolitains et la Corse (les deux départements ont été agrégés en un seul identifié par le n° 20) les informations suivantes :

- le nombre d'inscrits ;
- le nombre de votants ;
- le nombre de suffrages exprimés ;
- et, dans l'ordre, le nombre de voix des candidats : Villiers, Le Pen, Chirac, Laguiller, Cheminade, Jospin, Voynet, Balladur et Hue.

Lire les données et calculer le nombre des abstentions, des votes blancs ou nuls.

```
DATA TPSAS.elec95;
     INFILE "C:\...\TPSAS\elec95.txt" dlm="09"x;
     INPUT num$ Inscrits Votants Exprimes Villiers Le_Pen Chirac
          Laguill Cheminad Jospin Voynet Balladur Hue poids;
     Abstent=Inscrits-Votants;
     Blancs=Votants-Exprimes;
RUN;
```

1.1. AFC simple

Exercice 2 (*AFC*). — Q 1. Exécuter le code suivant réalisant l'AFC du tableau `elec95.sas7bdat` :

```
PROC CORRESP DATA=TPSAS.elec95 OBSERVED OUT=TPSAS.AFC_elec95;
     VAR Abstent Blancs Villiers Le_Pen Chirac Laguill Cheminad
          Jospin Voynet Balladur Hue;
     ID num;
RUN;
```

Q 2. Exploration avec SAS/INSIGHT :

2.a) Ouvrir la table de la bibliothèque `TPSAS.AFC_elec95` dans SAS/INSIGHT. Déclarer la variable $\langle num \rangle$ en label et construire les nuages des modalités pour les deux premiers axes via le menu `Analyze/Scatter Plot (YX)`.

2.b) Commenter le vote Villiers et la Vendée, ainsi que le vote Chirac et la Corrèze.

Ces deux départements sont mis en « supplémentaire » dans l'analyse suivante.

Exercice 3 (*AFC sans la Vendée et la Corrèze*). — Exécuter le code suivant qui permet via la spécification de l'option `WEIGHT` d'éliminer les départements de la Vendée et de la Corrèze de l'analyse.

```
PROC CORRESP DATA=TPSAS.elec95 OBSERVED
     OUT=tp4.AFC_elec95_ssCorrezeEtVendee DIM=3;
     VAR Abstent Blancs Villiers Le_Pen Chirac Laguill Cheminad
          Jospin Voynet Balladur Hue;
     ID num;
     WEIGHT poids;
RUN;
```

Q 1. Choisir le nombre d'axes, refaire les représentations graphiques. Comparer à l'analyse précédente.

Q 2. On pourra faire une AFC croisant les départements avec les candidats : Le Pen, Hue, Chirac, Balladur.

Remarque. — Lorsqu'on réalise une AFC avec la procédure `CORRESP`, il convient de noter les points suivants.

a) Lorsque le fichier des données est constituée de la table de contingence à analyser (c'est le cas ici), l'instruction `VAR` est nécessaire pour déclarer les variables intervenant en colonnes (elles doivent être quantitatives puisqu'elles contiennent des effectifs).

b) Toujours dans ce cas, la commande `ID` permet de déclarer la variable intervenant en lignes (elle est nécessaire pour pouvoir représenter ces lignes dans le graphique ; elle est qualitative puisqu'elle contient des étiquettes).

c) Si le fichier des données comporte en lignes les individus et en colonnes les 2 variables qualitatives codées, la commande `VAR` doit être remplacée par la commande `TABLES`.

1.2. Comparaison avec une ACP

Commenter le rôle de la métrique du χ^2 dans le cadre en comparant les résultats avec ceux de l'ACP des taux de suffrage exprimés pour chacun des candidats.

```
DATA tp4.telec95;
     SET tp4.elec95;
     villiers=villiers/inscrits;
     le_pen=le_pen/inscrits;
     chirac=chirac/inscrits;
     laguill=laguill/inscrits;
     cheminad=cheminad/inscrits;
     jospin=jospin/inscrits;
     voynet=voynet/inscrits;
     balladur=balladur/inscrits;
     hue=hue/inscrits;
run;
%acp(telec95,num,villiers--hue);
%gacpsx;
%gacpvx;
%gacpix;
```

2. Régression linéaire

2.1. Régression linéaire simple via SAS/INSIGHT

Le fichier `suitincom.txt` contient, pour 47 immeubles locatifs d'une grande ville américaine, le revenu net obtenu et le nombre d'appartements de l'immeuble. Les deux variables sont

- ⟨*Revenu*⟩ : *net operating income*;
- ⟨*Nbappart*⟩ : *number of suites.*

Exercice 4. — Q 1. Création du fichier de données : exécuter le programme de lecture des données

```
DATA suitinco;
     INFILE "C:\...\TPSAS\suitincom.txt" DLM="09"x;
     INPUT Revenu Nbappart;
RUN;
```

Q 2. Visualiser le nuage de points via le module `Analyze/Scatterplot` de SAS/INSIGHT. Puis effectuer une régression linéaire simple avec le module `Analyze/Fit` avec le revenu net comme variable à expliquer, et le nombre d'appartements comme variable explicative. On explorera différentes options des sorties via les menus `Graphs` et `Curves`. Cela permet en particulier d'effectuer le contrôle des sorties proposées.

Discutez la validité du modèle obtenu.

Q 3. Proposer d'autres modèles. Pour tester ces nouveaux modèles, on ouvre une nouvelle session SAS/INSIGHT pour laquelle on charge le fichier `suitinco.sas7bdat`. Puis dans le menu `Edit/Variable`, choisir successivement les transformations de variables associés à vos nouveaux modèles. Un fois appliqué, les nouvelles variables transformées apparaissent dans la base. La sauvegarder via le menu `File/Save/data`.

Effectuer les régressions linéaires. On notera que chaque application adjoint à la base les deux variables résiduelle et prévision associées au modèle testé. En particulier, cela permet de construire le graphique des résidus tel que présenté dans le cours.

Discutez la validité de chacun des modèles obtenus. Puis les comparer.

2.2. La régression multiple

Dans le cadre d'un objectif prédictif, ce TP compare les implémentations SAS des techniques visant à la recherche de modèles de régression linéaire multiple parcimonieux ainsi qu'une des techniques de régression biaisée (régression sur composantes principales) pour des données présentant un problème de colinéarité ou multilinéarité.

2.2.1. *Les données.* — Les données décrivent les résultats comptables de 80 entreprises du Royaume Uni. ⟨*RETCAP*⟩ est la variable à prédire. Elles sont décomposées en deux fichiers de 40 entreprises chacun : `ukcomp1.txt` et `ukcomp2.txt`. La constitution de ces deux fichiers a été réalisée par tirage aléatoire.

Voici un descriptif des 13 variables :

`RETCAP`	*Return on capital employed*
`WCFTDT`	*Ratio of working capital flow to total debt*
`LOGSALE`	*Log to base 10 of total sales*
`LOGASST`	*Log to base 10 of total assets*
`CURRAT`	*Current ratio*
`QUIKRAT`	*Quick ratio*
`NFATAST`	*Ratio of net fixed assets to total assets*
`FATTOT`	*Gross sixed assets to total assets*
`PAYOUT`	*Payout ratio*
`WCFTCL`	*Ratio of working capital flow to total current liabilities*
`GEARRAT`	*Gearing ratio (debt-equity ratio)*
`CAPINT`	*Capital intensity (ratio of total sales to total assets)*
`INVTAST`	*Ratio of total inventories to total assets*

Table 1. — La variable à expliquer avec les 12 variables explicatives

1. Création du fichier `ukcomp1.sas7bdat` à partir de la lecture du fichier `ukcomp1.txt` avec pour liste de variables

RETCAP	GEARRAT	CAPINT	WCFTDT	LOGSALE	LOGASST	CURRAT
QUIKRAT	NFATAST	INVTAST	FATTOT	PAYOUT	WCFTCL	

```
DATA TPSAS.ukcomp1;
     INFILE "C:\...\TPSAS\ukcomp1.txt" dlm="09"x;
     INPUT RETCAP GEARRAT CAPINT WCFTDT LOGSALE LOGASST CURRAT
          QUIKRAT NFATAST INVTAST FATTOT PAYOUT WCFTCL;
RUN;
```

2. Faire de même pour le fichier `ukcomp2.sas7bdat` à partir de `ukcomp2.txt` mais en nommant cette fois-ci ⟨*RETCAP2*⟩ la variable de retour sur capital.

2.2.2. *Modèle complet.* — Exercice 5. — Q 1. Effectuer une régression multiple via `SAS/INSIGHT` avec un modèle complet sur le fichier `ukcomp1.sas7bdat`. Commenter la qualité du modèle obtenu.

Q 2. On va retrouver les résultats avec la procédure classique `REG`.

2.a) Tout d'abord effectuer l'appel de base suivant :

```
OPTIONS NODATE NONUMBER;

PROC REG DATA=TPSAS.ukcomp1;
     MODEL RETCAP = WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
          NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT;
RUN;
```

2.b) Ensuite, on souhaite extraire certaines informations fournies par la procédure et les sauvegarder dans une table :

```
PROC REG DATA=TPSAS.ukcomp1 ADJRSQ CP
     OUTEST=TPSAS.ukcomp1_estim Tableout;
     /* outest permet de stocker les infos sur les estimations
     fournies par la table de sortie standard de SAS pour la procédure */
     MODEL RETCAP = WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
          NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT
          / CLB CLM CLI R P;
     OUTPUT OUT=TPSAS.ukcomp1_regcomplet
          P=predite R=residu LCLM=Borneinf_dr
          UCLM=Bornesup_dr /* borne de l'IC pour la droite */
          LCL=Borne_inf_prev
          UCL=Bornesup_prev;/* borne de l'IC de pr\'evision */
          /* Indique les indicateurs à stocker
          dans ukcomp1_regcomplet en dehors des
          variables employées dans le modèle */
RUN;

PROC PRINT DATA=TPSAS.ukcomp1_regcomplet (DROP =GEARRAT--WCFTCL);
RUN;

PROC PRINT DATA=TPSAS.ukcomp1_estim;
RUN;
```

Q 3. Enfin, il est possible de réaliser le test de nullité d'un sous-ensemble de coefficients du modèle (contre le modèle complet). Il suffit pour cela d'utiliser l'instruction `TEST` comme dans l'exemple suivant :

```
PROC REG DATA=TPSAS.ukcomp1;
     MODEL RETCAP = WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
          NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT;
     TEST WCFTCL=0, WCFTDT=0, GEARRAT=0;
RUN;
```

2.2.3. *Sélection manuelle de modèles par élimination.* — SAS propose des algorithmes de sélection automatique des variables. Néanmoins il est nécessaire de savoir se « débrouiller » avec des outils plus limités proposés par d'autres logiciels.

Exercice 6. — Mettre en œuvre la procédure de sélection dite « backward » à l'aide de SAS/INSIGHT :

Q 1. Choisir, parmi les variables explicatives, celle X_j pour lequel le test de Student (H_0 : $a_j = 0$) est le moins significatif, c'est-à-dire de plus grande p-valeur.

Q 2. La retirer du modèle et recalculer l'estimation. Il suffit pour cela de sélectionner le nom de la variable dans le tableau (`TYPE III`) et d'exécuter la commande `DELETE` du menu `edit` de la même fenêtre.

Q 3. Arrêter le processus lorsque tous les coefficients sont considérés comme significativement différents de 0 (à 5 %). Attention, la variable ⟨*intercept*⟩ ne peut pas être considérée au même titre que les autres variables et est à conserver dans tout modèle. Noter la séquence des modèles ainsi obtenus.

Q 4. Comparer avec la procédure automatique identique descendante ou « backward » :

```
PROC REG DATA=TPSAS.ukcomp1;
     MODEL RETCAP = WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
          NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT
          / SELECTION=BACKWARD;
          /* choix de la procédure de sélection */
RUN;
```

Dans cette procédure, il y a arrêt lorsque les variables restant dans le modèle ont une p-valeur plus petite que le seuil `SLSTAY` (= 0.1 par défaut ; il est possible de le spécifier dans l'appel).

Q 5. Comparer avec la procédure ascendante ou « forward » par ajout de variables :

```
PROC REG DATA=TPSAS.ukcomp1;
     MODEL RETCAP = WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
          NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT
          / SELECTION=FORWARD;
RUN;
```

Dans cette procédure, il y a lorsque les variables en dehors du modèle ont une p-valeur plus grande que le seuil `SLENTRY` (= 0.5 par défaut ; il est possible de le spécifier dans l'appel), c'est-à-dire une p-valeur plus grande que le seuil pour « entrer » dans le modèle courant.

Q 6. Comparer enfin avec la modification dite « stepwise » de la stratégie « forward » :

```
PROC REG DATA=TPSAS.ukcomp1;
     MODEL RETCAP = WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
          NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT
          / SELECTION=STEPWISE;
RUN;
```

Dans cette procédure, les seuil `SLENTRY` et `SLSTAY` pour entrer et sortir sont fixés par défaut à 0.15.

2.2.4. *Sélection automatique du modèle.* — Exercice 7. — Parmi les trois types d'algorithmes disponibles dans SAS et les différents critères de choix, une des façons les plus efficaces consistent à choisir les options appropriées de la procédure `REG`. dans le code ci-dessous, tous les modèles (parmi les plus intéressants selon l'algorithme de Furnival et Wilson) sont considérés. Seul le meilleur pour chaque niveau (via l'otion `BEST`) c'est-à-dire pour chaque valeur p du nombre de variables explicatives (hors constante), sont donnés. Il est alors facile de choisir celui minimisant l'un des critères globaux R^2, R^2_{ajust}, C_p, ...

```
PROC REG data=TPSAS.ukcomp1;
     MODEL RETCAP = WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
          NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT
          / SELECTION=RSQUARE ADJRSQ CP BEST=1;
RUN;
```

Sélectionner le modèle de C_p minimum et celui de R^2 ajusté maximum.

2.2.5. *Prévision.* — On se propose de comparer les prévisions calculées sur le deuxième ensemble d'entreprises du fichier `ukcomp2.sas7bdat` à l'aide de modèles estimés sur le premier ensemble, avec les valeurs effectivement observées. Enchaîner les étapes suivantes :

1. Concaténer verticalement les fichiers par

   ```
   DATA tp4.ukcomp;
        SET tp4.ukcomp1 tp4.ukcomp2;
   RUN;
   ```

2. Estimer le modèle sur les 40 premières observations et prévoir les 40 suivantes pour le modèle complet, le modèle maximisant le R^2 ajusté et celui minimisant le C_p. Il suffit pour cela déstimer dans `INSIGHT` (ou avec la procédure `REG`) le modèle expliquant $\langle RETCAP \rangle$. Les calculs sont faits en excluant les données manquantes et donc sur les 40 premières observations. En revanche, les prédictions sont calculées pour toutes. Pour effectuer les calculs et ajouter séquentiellement les résultats dans la base, utiliser le bouton `Apply` plutôt que `Ok`.
3. Comparer les valeurs prédites et les valeurs observées de $\langle RETCAP2 \rangle$. Pour cela, calculer la somme des carrés des erreurs pour les trois modèles. La somme des carrés des erreurs est simplement calculée dans `INSIGHT` en étudiant la distribution de la nouvelle variable $\langle P\_RETCAP \rangle - \langle RETCAP2 \rangle$ des écarts entre observations sur les 40 entreprises de test et valeurs prédites. Le paramètre `USS` (unmodified sum of squares) fournit la bonne valeur.

Cette procédure qui consiste à tirer au hasard un sous-échantillon, estimer un modèle sur ce sous-échantillon dit d'apprentissage, calculer l'erreur de prédiction sur le reste de l'échantillon dit de validation ou test peut être itérée pour estimer la distribution, propre à chaque modèle, de l'erreur moyenne de prédiction ou celle de tout autre estimateur. On rejoint l'objet des méthodes dites de ré-échantillonnage ou bootstrap qui substituent des simulations aux hypothèses classiques de normalité.

2.3. Régression sur composantes principales

L'approche suivante qui peut, dans certaines situations, donner de bons résultats se déroule en deux étapes.

1. Calcul des « variables principales » deux à deux non corrélées et engendrant le même espace que les variables explicatives par une ACP.

```
PROC PRINCOMP data=tp4.ukcomp1 OUT=comp;
     VAR WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
          NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT;
RUN;
```

2. Régression sur ces variables principales avec une sélection automatique des variables.

```
PROC REG DATA=comp;
     MODEL retcap = prin1--prin12 / SELECTION=RSQUARE CP BEST=1;
RUN;
QUIT;
```

Comparer les valeurs des C_p. Calculer les erreurs de prévision sur le deuxième échantillon d'entreprises du fichier `ukcomp2.sas7bdat` en ré-estimant le modèle sélectionné dans SAS/INSIGHT à partir des données de la table des composantes principales `work.comp`.

*** Fin ***

```
input statsmac;

% "Couleur des cheveux" "Couleur des yeux" "effectifs"
data(yeuxcheveux)
"blond", "bleu", 15,
"brun", "bleu", 9,
"noir", "bleu", 3,
"roux", "bleu", 7,
"blond", "gris ou vert", 13,
```

```
"brun", "gris ou vert", 17,
"noir", "gris ou vert", 10,
"roux" ,"gris ou vert", 7,
"blond", "marron", 7,
"brun", "marron", 13,
"noir", "marron", 8,
"roux", "marron", 5;

usedata(yeuxcheveux, cheveux$, yeux$, effectif);
n := 0 for i = 1 upto effectif.n: +effectif[i] endfor;
declare(x$, n); declare(y$, n); declare(alea, n);
k := 0;
for i = 1 upto yeux.n:
     for j = 1 upto effectif[i]:
          k := k+1;
          x[k] = yeux[i]; y[k] = cheveux[i];
          alea[k] = uniformdeviate 1;
     endfor
endfor
sort alea, x, y;
storedata(tmp, x, y);DLM := ", ";
writedata(tmp, "yeux-cheveux.txt");

% Un autre exemple avec des options syst\`eme pour l'impression. Les
% donn\'ees sont relatives \`a 88 \'etudiants d'une universit\'e
% anglaise ayant pass\'e 5 \'epreuves not\'ees par des notes enti\`eres
% de 0 \`a 100. Les 5 \'epreuves sont respectivement~: M\'ecanique, Alg\`ebre
% Lin\'eaire, Alg\`ebre des Structures, Analyse et Statistique.

n := 88;

for i = 1 upto n:
     mu := round(min(max(60+15*normaldeviate,0),100));
     sigma := uniformdeviate(15);
     write ""
     for j = 1 upto 5:
          &
          decimal(round(min(max(mu+sigma*normaldeviate,0),100)))
          &" "
     endfor
     to "notes.txt";
endfor
write EOF to "notes.txt";
```

```
TITLE "TP1. --- Introduction à SAS";

* LIBNAME TPSAS "C:\Mes documents\sas";

/* Différents éléments pour commencer

TPSAS.nomtable;

DATA <nom de la table SAS>;
     INFILE "<nom absolu du fichier des données ASCII>";
     INPUT <liste des noms des variables>;
RUN;
```

```
PROC UNIVARIATE;
RUN;

TITLE "Ceci est un titre";

TITLE;

FOOTNOTE "Ceci apparaîtra en bas de page";

GOPTIONS COLORS=(BLACK);

GOPTIONS RESET=COLORS;

GOPTIONS RESET=ALL;

*/

/* ceci est un commentaire */

* ceci en est un autre;

TITLE "TP1, Exercice 1. --- Saisie de tables de données";
FOOTNOTE "Elections 2004";

DATA elections04;
     ATTRIB Nom LABEL="Nom du parti";
     ATTRIB Pourcentage LABEL="Pourcentage des voix obtenues";
     INPUT Nom$ Pourcentage;
     CARDS;
     Ext.G 3.3
     PC 5.2
     PS 28.9
     Verts 7.4
     Div.G 5.9
     Div.D 10.7
     UDF 11.9
     UMP 16.6
     FN 9.8
     Ext.D 0.3
     ;
RUN;

PROC PRINT;
RUN;

PROC PRINT NOOBS;
RUN;

TITLE "TP1, Exercice 2. --- Création d'une table à partir d'un fichier ASCII";
FOOTNOTE "Q1 : Couleurs des yeux et des cheveux";

DATA yeux_cheveux1;
     INFILE "C:\Mes documents\sas\yeux-cheveux.txt" DLM=",";
     INPUT _IDCOURT_ V1$ V2$;
     LABEL V1="Couleur Yeux" V2="Couleur Cheveux";
RUN;

PROC PRINT NOOBS LABEL;
RUN;
```

```
PROC PRINT NOOBS LABEL;
     VAR V1 V2;
RUN;

OPTIONS PAGESIZE=64 LINESIZE=78 NODATE;

FOOTNOTE "Q2 : Fichier des notes";

DATA notes;
     INFILE "C:\Mes documents\sas\notes.txt";
     INPUT ind MECA ALIN ALGB ANLS STAT;
RUN;

PROC PRINT NOOBS;
RUN;

TITLE "TP1, Exercice 3. --- Manipulations de tables";
FOOTNOTE "Q1 : Couleurs des yeux et des cheveux";

DATA yeux_cheveux2;
     SET yeux_cheveux1 (KEEP=V1 V2);
RUN;

PROC PRINT NOOBS;
RUN;

PROC FREQ DATA=yeux_cheveux2;
     TABLE V1*V2 / OUT=Tab.cont.yeuxchev2;
RUN;

FOOTNOTE "Q2 : Fichier de notes";

DATA base1.note;
     SET notes (WHERE =(ind <= 30));
RUN;

PROC PRINT;
RUN;

DATA base2.note;
     SET notes (WHERE=(ind > 30));
RUN;

DATA base.note.complete;
     SET base1.note base2.note;
RUN;

PROC PRINT;
RUN;

FOOTNOTE "Q3 : Fichiers de notes, calculs";

DATA notes.gene;
     SET notes;
     ATTRIB Resultat LABEL="Résultat final" FORMAT=$7.;
     ATTRIB Moyenne.generale LABEL="Moyenne générale du candidat" FORMAT=6.2;
     Score.total=SUM(meca,alin,algb,anls,stat);
     Moyenne.generale=MEAN(meca,alin,algb,anls,stat);
     Note.max=MAX(meca,alin,algb,anls,stat);
```

```
     Note.min=MIN(meca,alin,algb,anls,stat);
     IF Moyenne.generale >= 50 THEN DO;
       Resultat="Admis";
     END;
     ELSE DO;
       Resultat="Ajourné";
     END;
RUN;

FOOTNOTE "Q4 : Fichier de notes, création par sélection";

DATA base.c1b.note;
     SET notes (KEEP=ind meca alin stat);
RUN;

PROC PRINT NOOBS;
RUN;

DATA base.c2b.note;
     SET notes (KEEP=algb anls stat );
RUN;

PROC PRINT NOOBS;
RUN;

DATA base.cbis.note.complete;
     MERGE base.c1b.note base.c2b.note;
RUN;

PROC PRINT NOOBS;
RUN;

DATA base.c1.note;
     SET notes (KEEP=ind meca algb stat);
RUN;

PROC PRINT NOOBS;
RUN;

DATA base.c2.note;
     SET notes (DROP=ind meca algb stat);
RUN;

PROC PRINT NOOBS;
RUN;

DATA base.c.note.complete;
     MERGE base.c1.note base.c2.note;
RUN;

PROC PRINT;
RUN;

DATA base.c1t.note;
     SET notes (KEEP=ind meca alin stat);
RUN;
```

```
DATA base.c2t.note;
     SET notes (KEEP=ind stat algb anls);
RUN;

DATA base.cter.note.complete;
     MERGE base.c1t.note base.c2t.note;
     BY ind;
RUN;

PROC PRINT NOOBS;
RUN;

TITLE "TP1, Exercice 4. --- ODS";
FOOTNOTE "Q1 : Elections 2004";

ODS PDF;
PROC PRINT DATA=elections04 NOOBS;
RUN;
ODS PDF CLOSE;

ODS HTML;
PROC PRINT DATA=elections04 NOOBS;
RUN;
ODS HTML CLOSE;

ODS RTF;
PROC PRINT DATA=elections04 NOOBS;
RUN;
ODS RTF CLOSE;

FOOTNOTE "Q2 : Couleurs des yeux et des cheveux";

ODS PDF;
PROC FREQ DATA=yeux_cheveux2;
     TABLE V1*V2;
RUN;
PROC PRINT;
RUN;
ODS PDF CLOSE;

ODS latex STYLE=journal FILE="C:\Mes documents\sas\toto.tex";
PROC FREQ data=yeux_cheveux2;
     TABLES V1*V2;
RUN;
ODS latex CLOSE;

TITLE; FOOTNOTE; /* réinitialisation */

TITLE "TP2. --- Fonctions graphiques sous SAS";

/* Quelques éléments pour commencer

GOPTIONS COLORS=(BLACK);

GOPTIONS RESET=COLORS;

GOPTIONS RESET=ALL;
```

```
PATTERN1 COLOR=blue;
PATTERN2 COLOR=yellow;
PATTERN3 COLOR=black;
PATTERN5 COLOR=red;
PATTERN4 COLOR=green;
PATTERN6 COLOR=purple;

*/

/* AXISn définit le style employé pour le n-ième type d'axes */

TITLE "TP2, Exercice 1. --- Diagramme en bâtons";
FOOTNOTE "Q1 : nombre d'enfants";

DATA NbreEnfants;
     INPUT Nb_enfants Effectif;
     CARDS;
     0 235
     1 183
     2 285
     3 139
     4 88
     5 67
     6 3
     ;
RUN;

PROC PRINT DATA=NbreEnfants;
RUN;

FOOTNOTE "Q2 : diagramme en bâtons avec GPLOT";

AXIS1 LABEL=("Nb Enfants" JUSTIFY=RIGHT);
AXIS2 LABEL=("Effectifs") ORDER=(0 TO 290 BY 10) offset=(0,);

/* ORDER fixe le domaine sur le second axe et le placement
d'une marque ici tous les 10 unités OFFSET pour figer
l'écart entre l'origine du dessin et la 1ère marque (.,)
et entre la dernière marque (ou ticks) (,.) et la fin du
tracé de l'axe */

PROC GPLOT DATA=NbreEnfants;
     SYMBOL1 INTERPOL=NEEDLE VALUE=DOT;
     PLOT Effectif*Nb_enfants / HAXIS=AXIS1 VAXIS2=AXIS2;
RUN;
QUIT;

FOOTNOTE "Q3 : diagramme en bâtons avec GCHART";

AXIS1 LABEL=("Nb Enfants" JUSTIFY=RIGHT);
AXIS2 LABEL=("Effectif") ORDER=(0 TO 290 BY 10) offset=(0,);

PROC GCHART DATA=NbreEnfants;
     VBAR Nb_enfants / DISCRETE SUMVAR=Effectif MAXIS=AXIS1 RAXIS=AXIS2;
     /* l'option PATTERNID=MIDPOINT permet d'avoir une couleur par bâton */
RUN;
QUIT;
```

```
TITLE "TP2, Exercice 2. --- Fonction de répartition";
FOOTNOTE "Q1 : nombre d'enfants cumulé";

DATA NbreEnfantsPlus;
     SET NbreEnfants;
     BY Nb_enfants;
     RETAIN Eff_cumule Freq_cumule (0,0);
     Eff_cumule=Eff_cumule+Effectif;
     Freq_cumule=Freq_cumule+Freq;
     ATTRIB Eff_cumule LABEL="Effectifs cumulés";
     ATTRIB Freq_cumule LABEL="Fréquences cumulées";
RUN;

PROC PRINT DATA=NbreEnfantsPlus;
RUN;

FOOTNOTE "Q2 : diagramme en bâtons avec GCHART";

/* à voir */

FOOTNOTE "Q3 : tracé de la courbe avec GPLOT";

PROC GPLOT DATA=NbreEnfantsPlus;
     AXIS1 LABEL=("Nbre enfants" JUSTIFY=RIGHT) ORDER=(0 TO 6 BY 1);
     AXIS2 LABEL=("Fréquence" JUSTIFY=RIGHT "Cumulée")
       ORDER=(0 TO 1 BY 0.1) OFFSET=(1,);
     SYMBOL1 INTERPOL=STEP COLOR=blue;
     PLOT Freq_cumule*Nb_enfants / HAXIS=AXIS1 VAXIS=AXIS2 GRID;
     /* GRID permet d'avoir une grille en pointillés sous-jacente
     au graphique */
RUN;
QUIT;

TITLE "TP2, Exercice 3. --- Fonction de répartition";
FOOTNOTE "Q1 : ampoules";

DATA ampoule;
     INPUT duree @@;
     CARDS;
     91.6 35.7 251.3 5.4 67.3 170.9 9.5 118.4 57.1
     ;
RUN;

PROC PRINT DATA=ampoule;
RUN;

FOOTNOTE "Q2 : histogramme avec GCHART";

PROC GCHART DATA=ampoule;
     VBAR DUREE / LEVELS=5 MIDPOINTS=26 TO 234 BY 52 TYPE=FREQ;
     /* Remplacer FREQ par CFREQ pour avoir un histogramme des
     effectifs cumulés */
RUN;

FOOTNOTE "Q3 : polygone des fréquences cumulées";
```

```
AXIS1 LABEL=("Midpoint" JUSTIFY=RIGHT);
AXIS2 LABEL=("Fréquences" JUSTIFY=RIGHT "Cumulées");
AXIS3 LABEL=("Fréquences" JUSTIFY=left "Cumulées");

PROC GBARLINE DATA=Ampoulehistobis;
     BAR MIDPOINT / DISCRETE SUMVAR=Freq_cumule AXIS=AXIS2;
     PLOT / SUMVAR=Freq_cumule RAXIS=AXIS3;
RUN;
QUIT;

FOOTNOTE "Q4 : avec SAS/INSIGHT...";

TITLE "TP2, Exercice 4. --- Diagramme en bâtons";
FOOTNOTE "Couleurs des yeux et des cheveux";

AXIS1 LABEL=("Effectifs") ORDER=(0 TO 240 BY 20);

PROC GCHART DATA=Tab_cont_yeuxchev2;
     VBAR V1 / SUMVAR=COUNT RAXIS=AXIS1 INSIDE=SUM;
RUN;

TITLE "TP2, Exercice 5. --- Diagramme en barres";
FOOTNOTE "Couleurs des yeux et des cheveux";

TITLE2 "verticales";

AXIS1 LABEL=NONE ORDER=(0 TO 215);

PROC GCHART DATA=Tab_cont_yeuxchev2 (FIRSTOBS=1 OBS=4);
     VBAR V1 / NOAXIS SUMVAR=COUNT SUBGROUP=V2 RAXIS=AXIS1
       INSIDE=SUM OUTSIDE=SUM;
RUN;

AXIS1 LABEL=NONE ORDER=(0 TO 215);

PROC GCHART DATA=Tab_cont_yeuxchev2 (FIRSTOBS=5 OBS=8);
     VBAR V1 / NOAXIS SUMVAR=COUNT SUBGROUP=V2 RAXIS=AXIS1
       INSIDE=SUM OUTSIDE=SUM;
RUN;

TITLE2 "horizontales";

AXIS1 ORDER=(0 TO 220 BY 10) LABEL=NONE;
AXIS2 LENGTH=1cm;

PROC GCHART DATA=Tab_cont_yeuxchev2 (FIRSTOBS=5 OBS=8);
     HBAR V1 / SUMVAR=COUNT SUBGROUP=V2 RAXIS=AXIS1 MAXIS=AXIS2
       OUTSIDE=SUM SUMLABEL="Effectif total"
       CTEXTSIDE=PURPLE REF=7 75 194;
RUN;
QUIT;

TITLE2; /* réinitialisation */

TITLE "TP2, Exercice 6. --- Diagramme en secteurs";
FOOTNOTE "Q1 : Elections 2004, camembert";
```

```
PROC GCHART DATA=elections04;
     PIE Nom / SUMVAR=pourcentage NOHEADING SLICE=OUTSIDE
     PERCENT=OUTSIDE ASCENDING OTHER=0 VALUE=NONE;
     /* COUTLINE=SAME donne la m{\accent 94 e}me couleur au bord
     et à l'intérieur d'une tranche */
RUN;
QUIT;

FOOTNOTE "Q1 : Elections 2004, camembert avec légende";

LEGEND1 LABEL=NONE
     POSITION =(LEFT MIDDLE) /* position de la légende par rapport
         au graphique */
     OFFSET=(4,) ACROSS=1 /* nombre de colonnes pour la légende */
     VALUE=(COLOR=BLACK)
     SHAPE=BAR(4,1.5); /* spécification de la taille de la boîte
         pour chaque catégorie */

PROC GCHART DATA=elections04;
     PIE nom / SUMVAR=pourcentage NOHEADING ASCENDING
       COUTLINE=SAME OTHER=0 PERCENT=OUTSIDE LEGEND=LEGEND1;
RUN;
QUIT;

FOOTNOTE "Q1 : Elections 2004, camembert 3D avec légende";

PROC GCHART DATA=elections04;
     PIE3D nom / SUMVAR=pourcentage NOHEADING ASCENDING
       COUTLINE=SAME OTHER=0 PERCENT=OUTSIDE LEGEND=LEGEND1;
RUN;
QUIT;

TITLE "TP2, Exercice 7. --- Boîtes à moustaches";
FOOTNOTE "Pento";

DATA pento;
     INFILE "pento.txt";
     INPUT nom$ rythme facteur;/* ??? */
RUN;

PROC PRINT DATA=pento;
RUN;

FOOTNOTE "Q1 : tracé de pento avec GPLOT";

PROC GPLOT DATA=pento;
     SYMBOL1 INTERPOL=BOX;
     AXIS1 LENGTH=5cm OFFSET=(1cm,1cm);
     AXIS2 LENGTH=5cm;
     PLOT rythme*facteur=1 / VAXIS=AXIS2 VMINOR=1 HAXIS=AXIS1;
RUN;
QUIT;

GOPTIONS RESET=ALL;

FOOTNOTE "Q2 : tracé de pento avec BOXPLOT";
```

```
PROC BOXPLOT DATA=pento;
     PLOT rythme*facteur / CBOXES=black CBOXFILL=yellow
       ALLLABEL=VALUE CTEXT=black;
     INSET NOBS / HEIGHT=3 HEADER="Nombre d'observations" POSITION=NW;
RUN;

TITLE "TP2, Exercice 8. --- Diagrammes en Barres en parallèle";
FOOTNOTE "Couleurs des yeux et des cheveux";

PROC FREQ DATA=yeux_cheveux1;
     TABLE V1*V2 / OUT=Tab_freq_yeuxchev3 OUTPCT;
     /* l'option OUTPCT permet de compléter la sauvegarde */
RUN;

PROC PRINT DATA=Tab_freq_yeuxchev3;
RUN;

FOOTNOTE "Diagrammes en barres pour chaque classe de profils";

AXIS1 LABEL=NONE;
PROC GCHART DATA=Tab_cont_yeuxchev3;
     VBAR V1 / SUMVAR=PCT_ROW SUBGROUP=V2 RAXIS=AXIS1 INSIDE=SUM;
RUN;

AXIS1 LABEL=NONE;
PROC GCHART DATA=Tab_cont_yeuxchev3;
     VBAR V2 / SUMVAR=PCT_COL SUBGROUP=V1 RAXIS=AXIS1 INSIDE=SUM;
RUN;

TITLE "TP2, Exercice 9. --- Nuage de points";
FOOTNOTE "Denrees";

DATA TPSAS.denrees;
     INFILE "denrees.txt";
     INPUT _idlong V1 V2 V3 V4;/* ??? */
RUN;

FOOTNOTE "Q1 : nuage de points par croisement de deux variables";

PROC GPLOT DATA=denrees;
     SYMBOL1 INTERPOL=NONE VALUE=DOT;
     PLOT V1*V3 / GRID;
RUN;
QUIT;

FOOTNOTE "Q2 : nuage de points avec étiquettes";

PROC GPLOT DATA=denrees;
     SYMBOL1 INTERPOL=NONE VALUE=DOT;
     PLOT V1*V3=_idlong / GRID;
RUN;
QUIT;

FOOTNOTE "Combinaisons deux à deux";

SYMBOL1 INTERPOL=NONE VALUE=DOT COLOR=red;
SYMBOL2 INTERPOL=NONE VALUE=DOT COLOR=blue;
```

```
PROC GPLOT DATA=denrees;
     PLOT V1*V3=1 V1*V4=2/ GRID;
RUN;
QUIT;

GOPTIONS RESET=ALL;

PROC GPLOT DATA=denrees;
     PLOT V1*(V3 V4);
RUN;
QUIT;

FOOTNOTE "Q4 : avec SAS/INSIGHT...";

TITLE "TP2, Exercice 10. --- MEANS et SUMMARY";
FOOTNOTE "Q1 : MEANS appliquée aux données denrees";

GOPTIONS RESET=ALL;

PROC MEANS DATA=denrees MAXDEC=2;
     VAR V1-V6;
     OUTPUT OUT=Means_denrees;
RUN;

PROC MEANS DATA=denrees MEAN STD CV MAX MIN MEDIAN MAXDEC=2;
     VAR V1-V6;
RUN;

FOOTNOTE "Q2 : MEANS appliquée aux données pento";

PROC MEANS DATA=pento MAXDEC=2;
     VAR rythme;
     CLASS facteur;
RUN;

TITLE "TP2. --- La procédure UNIVARIATE";
FOOTNOTE "Notes générales";

PROC UNIVARIATE DATA=notes_gene NORMAL PLOT;
     VAR STAT Moyenne_generale;
RUN;

TITLE "TP2. --- La procédure FREQ";
FOOTNOTE "Couleurs des yeux et des cheveux";

PROC FREQ DATA=Yeuxchev1;
     TABLE V1*V2 / CHISQ;
RUN;

TITLE "TP2. --- La procédure CORR";
FOOTNOTE "Matrice des corrélations de denrées";

PROC CORR DATA=denrees OUT=corr_denrees;
     VAR V1-V6;
RUN;

TITLE; FOOTNOTE; /* réinitialisation */

TITLE "TP3. --- ACP sous SAS";
```

```
TITLE "TP3. --- ACP via SAS/INSIGHT...";

TITLE "TP3, Exercice 1. --- ACP via PRINCOMP";
FOOTNOTE "Q1 : ACP sur les denrées";

PROC PRINCOMP DATA=denrees;
RUN;

FOOTNOTE "Q2 : ACP sur les denrées";

PROC PRINCOMP DATA=denrees
     VARDEF=N /* COV si ACP non normée */
     OUT=denrees_acp1 /* Données initiales
       + Composantes principales */
     OUTSTAT=denrees_acp2; /* Table stockant les résumés
     de base pour les variables initiales + Corrélation/covariances
     + Valeurs propres et Variables principales */
RUN;

PROC PRINT DATA=denrees_acp1 NOOBS;
RUN;

PROC PRINT DATA=denrees_acp2 NOOBS;
RUN;

TITLE "TP3. --- Macro SAS et ACP";
FOOTNOTE "Macro-variables";

%LET nomvar1=v1;
%LET nomvar2=v2;
%LET varlist=csp pao paa vio via pdt;

FOOTNOTE "Étude des variables &varlist";

PROC PLOT DATA=denrees;
     PLOT &nomvar1*&nomvar2;
RUN;

FOOTNOTE "Macro-commandes";

%MACRO impdat(in);
%* ceci est un commentaire;
     PROC PRINT DATA=&in NOOBS;
     RUN;
%MEND impdat;

/* Appel à la macro */
%impdat(denrees);

%MACRO plot(in,yvar,xvar);
     PROC PLOT DATA=&in;;
     PLOT &yvar*&xvar;
     RUN;
%MEND plot;

%plot(tp3.denrees,&nomvar1,&nomvar2);

/* chemin d'accès à une bibliothèque de macros
OPTIONS SASAUTOS=(SASAUTOS "C:\...\TPSAS\MesMacros") MAUTOSOURCE MRECALL;
*/
```

```
TITLE "TP3, Exercice 2. --- Macro de lecture de données";
FOOTNOTE;

%MACRO lecacp(in,out,ident,listevar,dlm=" ");
     DATA &out;
     INFILE &in DLM=&dlm;
     INPUT &ident$ &listevar;
     RUN;
%MEND lecacp;

%lecacp("C:\...\TPSAS\temp.txt", tp3.temp,ville,
     janv fevr mars avri mai juin juil aout sept oct nov dec);
%impdat(temp);

TITLE "TP3, Exercice 3. --- Macros de P. Besse";
FOOTNOTE "ACP's";

/* Retouver la macro sur le web !
%MACRO acp(dataset, ident, listev, red=, q=3, poids=);
%* ACP de dataset;
%*        ident : variable contenant les identificateurs des individus;
%*       listev : liste des variables (numériques);
%*   par defaut : réduites sinon red=cov;
%*            q : nombre de composantes retenues;
%*        poids : variable de pondération;
%*         pvar : nombre de variables;
%* options édition;
%global pvar;
*/

%acp(denrees,_IDLONG_, v1--v8);
%acp(tempf_tp,ville, janv--dec, red=cov);
%acp(tempf_tp,ville, janv--dec);

TITLE "TP3, Exercice 4. --- Macros pour graphiques de base";
FOOTNOTE;

/* \'ebouli des valeurs propres */
%gacpsx;
/* Boîtes à moustaches en parallèle des différentes
valeurs propres */
%gacpbx;
/* Nuage des individus sur le premier plan factoriel */
%gacpix;
/* Cercle des corrélations croisant les deux premières
composantes principales par défaut */
%gacpvx;
/* Cercle des corrélations croisant les composantes principales No 2 et No 3*/
%gacpvx(x=2,y=3);

/ * entete de macro
%macro gacpvx(x=1, y=2, nc=4, coeff=1);
%* Graphique des variables avec cercle des corrélations;
%*  x : numéro axe horizontal;
%*  y : numéro axe vertical;
```

```
%* nc : nombre max de caractères;
*/

TITLE "TP3, Exercice 5. --- Variables supplémentaires";
FOOTNOTE "Données tmpf_tp";

DATA tp3.corrvarsup;
     SET coorindq (KEEP=Prin1-Prin3);
     SET tp3.tempf_tp (KEEP=Lat Long Tmoy Amp);
RUN;

PROC PRINT NOOBS;
RUN;

PROC CORR DATA=tp3.corrvarsup NOSIMPLE OUT=tp3.rescorrvarsup;
     VAR PRIN1-PRIN3;
     WITH Lat Long Tmoy Amp;
RUN;

TITLE "TP3, Exercice 6. --- Eaux minérales gazeuzes";
FOOTNOTE "Non disponible";

TITLE; FOOTNOTE; /* réinitialisation */

TITLE "TP4. --- AFC et régression linéaire";

TITLE "TP4, Exercice 1. --- Acquisition des données";
FOOTNOTE "Elections 1995";

DATA elec95;
     INFILE "C:\...\TPSAS\elec95.txt" dlm="09"x;/* curieux */
     INPUT num$ Inscrits Votants Exprimes
     Villiers Le Pen Chirac Laguill Cheminad
     Jospin Voynet Balladur Hue poids;
     Abstent=Inscrits-Votants;
     Blancs=Votants-Exprimes;
RUN;

TITLE "TP4, Exercice 2. --- AFC simple";
FOOTNOTE "Q1 : Elections 1995";

PROC CORRESP DATA=elec95 OBSERVED OUT=AFC_elec95;
     VAR Abstent Blancs Villiers Le Pen Chirac Laguill Cheminad
       Jospin Voynet Balladur Hue;
     ID num;
RUN;

FOOTNOTE "Q2 : avec SAS/INSIGHT...";

TITLE "TP4, Exercice 3. --- AFC simple";
FOOTNOTE "Elections 1995 sans la Vendée ni la Corrèze";

PROC CORRESP DATA=elec95 OBSERVED
     OUT=tp4.AFC_elec95_ssCorrezeEtVendee DIM=3;
     VAR Abstent Blancs Villiers Le_Pen Chirac Laguill Cheminad
       Jospin Voynet Balladur Hue;
     ID num;
     WEIGHT poids;
RUN;
```

```
FOOTNOTE "Q1 : graphiques à voir...";

FOOTNOTE "Q2 : autres éliminations...";

TITLE "TP4. --- Comparaison avec une ACP";
FOOTNOTE "Elections 1995";

DATA tp4.telec95;
     SET tp4.elec95;
     villiers=villiers/inscrits;
     le_pen=le_pen/inscrits;
     chirac=chirac/inscrits;
     laguill=laguill/inscrits;
     cheminad=cheminad/inscrits;
     jospin=jospin/inscrits;
     voynet=voynet/inscrits;
     balladur=balladur/inscrits;
     hue=hue/inscrits;
run;
%acp(telec95,num,villiers--hue);
%gacpsx;
%gacpvx;
%gacpix;

TITLE "TP4, Exercice 4. --- Régression linéaire simple";
FOOTNOTE "Q1 : Revenus immobiliers";

DATA suitinco;
     INFILE "C:\...\TPSAS\suitincom.txt" DLM="09"x;
     INPUT Revenu Nbappart;
RUN;

FOOTNOTE "Q2 : avec SAS/INSIGHT...";
FOOTNOTE "Q3 : avec SAS/INSIGHT...";

TITLE "TP4. --- Régression linéaire multiple";
FOOTNOTE "Q1 : Revenus immobiliers 1";

DATA ukcomp1;
     INFILE "C:\...\TPSAS\ukcomp1.txt" dlm="09"x;
     INPUT RETCAP GEARRAT CAPINT WCFTDT LOGSALE LOGASST CURRAT
       QUIKRAT NFATAST INVTAST FATTOT PAYOUT WCFTCL;
RUN;

FOOTNOTE "Q2 : Revenus immobiliers 2";

DATA ukcomp2;
     INFILE "C:\...\TPSAS\ukcomp2.txt" dlm="09"x;
     INPUT RETCAP2 GEARRAT CAPINT WCFTDT LOGSALE LOGASST CURRAT
       QUIKRAT NFATAST INVTAST FATTOT PAYOUT WCFTCL;
RUN;

TITLE "TP4, Exercice 5. --- Régression linéaire multiple";
FOOTNOTE "Q1 : Revenus immobiliers 1 avec SAS/INSIGHT...";

FOOTNOTE "Q2 : Revenus immobiliers 1 avec REG";
```

```
OPTIONS NODATE NONUMBER;

PROC REG DATA=ukcomp1;
     MODEL RETCAP = WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
       NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT;
RUN;

PROC REG DATA=ukcomp1 ADJRSQ CP
     OUTEST=ukcomp1_estim Tableout;
     /* outest permet de stocker les infos sur les estimations
     fournies par la table de sortie standard de SAS pour la procédure */
     MODEL RETCAP = WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
       NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT
       / CLB CLM CLI R P;
     OUTPUT OUT=ukcomp1_regcomplet
       P=predite R=residu LCLM=Borneinf_dr
       UCLM=Bornesup_dr /* borne de l'IC pour la droite */
       LCL=Borne_inf_prev
       UCL=Bornesup_prev;/* borne de l'IC de pr\'evision */
       /* Indique les indicateurs à stocker
       dans ukcomp1_regcomplet en dehors des
       variables employées dans le modèle */
RUN;

PROC PRINT DATA=ukcomp1_regcomplet (DROP =GEARRAT--WCFTCL);
RUN;

PROC PRINT DATA=ukcomp1_estim;
RUN;

FOOTNOTE "Q2 : Revenus immobiliers 1 avec REG et TEST";

PROC REG DATA=ukcomp1;
     MODEL RETCAP = WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
       NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT;
     TEST WCFTCL=0, WCFTDT=0, GEARRAT=0;
RUN;

TITLE "TP4, Exercice 6. --- Sélection backward/forward";
FOOTNOTE;

PROC REG DATA=ukcomp1;
     MODEL RETCAP = WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
       NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT
       / SELECTION=BACKWARD;
       /* choix de la procédure de sélection */
RUN;

PROC REG DATA=ukcomp1;
     MODEL RETCAP = WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
       NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT
       / SELECTION=FORWARD;
RUN;
```

```
PROC REG DATA=ukcomp1;
     MODEL RETCAP = WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
       NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT
       / SELECTION=STEPWISE;
RUN;

TITLE "TP4, Exercice 7. --- Sélection backward/forward";
FOOTNOTE;

PROC REG data=ukcomp1;
     MODEL RETCAP = WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
       NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT
       / SELECTION=RSQUARE ADJRSQ CP BEST=1;
RUN;

TITLE "TP4. --- Prévision";
FOOTNOTE;

DATA tp4.ukcomp;
     SET tp4.ukcomp1 tp4.ukcomp2;
RUN;

TITLE "TP4. --- Régression sur les composantes principales";
FOOTNOTE;

PROC PRINCOMP data=tp4.ukcomp1 OUT=comp;
     VAR WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
       NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT;
RUN;

PROC REG DATA=comp;
     MODEL retcap = prin1--prin12 / SELECTION=RSQUARE CP BEST=1;
RUN;
QUIT;
```