Université Lumière Lyon 2

M2 – SISE

Statistique et Informatique pour la Science des donnéEs

Support SAS

Statistique - Analyse des données – Modélisation

(Programmes d'application SAS)

Année Universitaire 2021-2022

R. Abdesselam
rafik.abdesselam@univ-lyon2.fr

Web : http://perso.univ-lyon2.fr/~rabdesse/fr/

Polycopiés : http://perso.univ-lyon2.fr/~rabdesse/Documents/

THE POWER TO KNOW®

INTRODUCTION au LOGICIEL SAS

Programmes d'application SAS

*Ce cours traite essentiellement de l'utilisation du logiciel SAS "**Statistical Analysis System**" ou "**Système d'Analyse Statistique**".*

L'objet de ce cours est d'initier les doctorants à la phase préparatoire des données pour l'analyse statistique des données avec le logiciel SAS.

L'accent est mis sur l'apprentissage et l'utilisation des instructions SAS permettant d'effectuer divers traitements ainsi que sur l'interprétation pratique des résultats fournis par SAS.

Chaque procédure est illustrée par un, voire plusieurs exemples de programmes SAS dont la syntaxe et les options sont commentées et les résultats obtenus interprétés.

Des exercices d'application sont proposés à la fin de chaque chapitre afin d'appliquer de nouveau les notions traitées.

Plan détaillé du cours

Chapitre 1 – Introduction au logiciel SAS

- *Présentation du système SAS, l'environnement SAS,*
- *Instructions et mots-clés*
- *Conception et exécution d'un programme SAS*
- *Procédures de gestion et de présentation des données*
- *Recodage et transformation des données*
- *Exercices d'application.*

Chapitre 2 – Statistique descriptive

- *Procédures de la statistique descriptive : means, univariate, corr, freq,*
- *Procédures graphiques :chart, plot,*
- *Exercices d'application.*

Chapitre 3 – Tests statistiques

- *Tests utilisant la procédure **means** : test sur une moyenne, test sur la différence de deux moyennes (échantillons indépendants),*
- *Tests utilisant la procédure **univariate** : test sur une moyenne, test sur la différence de deux moyennes (échantillons indépendants), test de normalité, test de Wilcoxon, test des signes,*
- *Tests utilisant la procédure **ttest** : Intervalles de confiance, test sur deux moyennes, test sur deux variances,*
- *Tests utilisant la procédure **anova** : analyse de la variance, complément à l'analyse de la variance,*
- *Tests utilisant la procédure **freq** : test d'indépendance, test d'homogénéité, test sur deux proportions,*
- *Tests utilisant la procédure **corr** : test sur le coefficient de corrélation linéaire, corrélation de rangs de Spearman,*
- *Tests utilisant la procédure **npar1way** : test U de Mann-Whitney, test de Kruskal-Wallis,*
- *Exercices d'application.*

Chapitre 4 – Modèles explicatifs de prédiction

- *Procédure **reg** : Régression linéaire simple et multiple, analyse de la variance, estimation des paramètres du modèle, détermination d'intervalles de confiance et prévision, Commande test en régression linéaire multiple,*
  - *Méthodes de sélection d'un ensemble de variables indépendantes : régression pas à pas, méthode d'introduction progressive, méthode d'élimination progressive, choix d'un modèle à l'aide du coefficient de détermination ajusté, choix d'un modèle à l'aide de la statistique cp de Mallows,*
  - *Vérification des hypothèses de base du modèle de régression linéaire : espérance nulle des résidus, homoscédasticité (variance constante), normalité des résidus, absence d'autocorrélation, absence de colinéarité,*
- *Procédure **logistic** : Régression logistique binaire, principe et estimation des paramètres, tests de Wald, score et likelihood ratio, courbe ROC,*
- *Exercices d'application.*

Chapitre 5 – Méthodes d'Analyse de données

Méthodes factorielles "Mapping"

- *Procédures **factor** et **princomp** : Analyse en composantes principales,*
- *Procédure **corresp** : analyse des correspondances simples et multiples*
- *Procédures **discrim**, **stepdisc** et **candisc** : Analyse discriminante (classement d'observations, efficacité, validité et conditions d'application de l'analyse),*

Classifications – Typologie "Clustering"

- *Procédure **fastclust** : classification sur individus, nuées dynamiques,*
- *Procédure **cluster** : classification hiérarchique ascendante,*
- *Procédure **varclust** : classification (hiérarchique et partition) des variables,*
- *Procédure **tree** : l'arbre de classification – dendogramme*
- *Exercices d'application.*

Quelques références bibliographiques

1- G. Baillargeon " Introduction au logiciel SAS'", Editions SMG (1996). Salle de travail.

2- G. Baillargeon " SAS régression'" , Editions SMG (1992). Salle de travail..

3- H.K. Kouomegni, O. Decourt " SAS'" , Applications et Métiers, 2ème Edition DUNOD, Paris (2007).

4- Consulter les bibliothèques d'exemples SAS.

5- Documentation SAS sur le serveur Web de SAS Institute.

6- Caillez F., Pages J.P. Introduction à l'Analyse des Données SMASH (1975).

7- Han J. , Kamber M. : Data Mining Concepts and Techniques, (2001).

8- Michael J.A. Berry, Gordon Linoff : Data Mining, Masson (1997).

9- http://dept.econ.yorku.ca/jasj/classes.html Cours C. Gourieroux, Econometric, Analysais of individual Risks

10- http://dorakmt.tripod.com/mtd/glosstat.html glossaire statistique

Référence(s) de base

Bouroche J.M., Saporta G. L'analyse des données, "Que sais-je?" N°1854 , PUF, 8ème édition (2002).

Ce fascicule de poche constitue une excellente introduction à l'analyse statistique multidimensionnelle. Il met l'accent sur l'interprétation intuitive des idées et concepts en n'ayant presque aucun recours à la notation mathématique. Il accorde aussi beaucoup d'importance à l'interprétation correcte des résultats.

Synthèse des principaux modules de programmation SAS'

Module SAS Programmation

*Etape **DATA***

- *import et saisie des données sous SAS*
- *création d'un fichier texte à partir d'une table SAS*
- *manipulation des données au cours d'une étape DATA*
- *chargement et fusion de tables SAS*

*Etape **PROC***

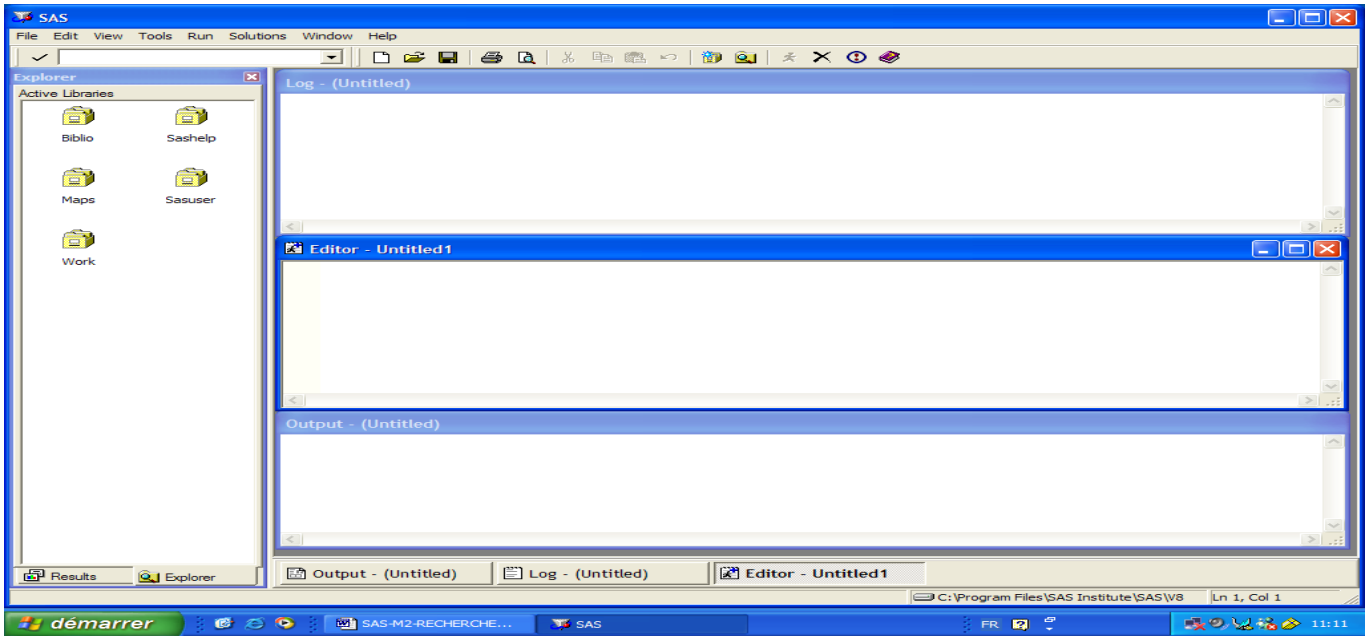
- *imprimer les observations d'une table SAS : PROC PRINT*
- *afficher le contenu d'une table : PROC CONTENTS*
- *effectuer des opérations globales sur les tables : PROC DATASETS*
- *créer des formats utilisateur : PROC FORMAT*
- *trier une table SAS selon des clés : PROC SORT*
- *afficher des comptages et des pourcentages : PROC FREQ*
- *établir des statistiques sommaires : PROC MEANS*
- *calculer des corrélations : PROC CORR*
- *analyser une variable en détail : PROC UNIVARIATE*
- *effectuer différents tests statistiques : PROC TTEST*
- *différents graphiques sommaires : PROC PLOT*

Autres modules SAS

- *langage matriciel : module SAS IML*
- *gestion de bases de données SGBD relationnels : SQL*
- *graphiques : SAS GRAPH*
- *macro langage SAS*

Environnement SAS

Les fenêtres de SAS : Log, Editor, Output et Explorer (Libraries/Results)



Chapitre 1 : Introduction au logiciel SAS

1 Introduction

*L'objectif principal de ce fascicule est l'initiation à l'utilisation du logiciel SAS (**Statistical Analysis System**). Le système SAS offre un large éventail de traitements séquentiel de données permettant d'assurer pratiquement toutes ou presque toutes les fonctions de gestion et d'analyse des données à l'aide des nombreuses procédures statistiques qu'il propose. La documentation importante de ce logiciel traduit sa richesse mais aussi sa complexité qui peut toutefois s'avérer un inconvénient pour certains utilisateurs. Comme il est visiblement impossible de couvrir SAS en entier, ce guide se veut donc qu'un résumé des aspects de base les plus intéressants pour être en mesure de comprendre comment élaborer un programme SAS et d'être suffisamment à l'aise avec certains mots-clés, certaines commandes nécessaires car beaucoup de programmes SAS se ressemblent et ne diffèrent que par les données utilisées et par les traitements effectués. Il existe donc des éléments communs à l'ensemble des programmes. Ce sont ces éléments qui seront d'abord présentés.*

1.1 Modes d'exécution d'un programme SAS

Il existe différents modes ou environnements d'exécution de programmes SAS version 9.4 sous l'interface du système d'exploitation Windows de l'Université Lumière Lyon 2.

Ce cours sera axé sur le mode ''programmation''; d'exécution de type non-interactif. Ce mode d'exécution, Batch SAS, se caractérise par les étapes suivantes :

- *la création au préalable d'un fichier externe nommé **«exemple.sas»**, contenant les instructions SAS, au moyen de l'éditeur intégré **Program Editor**,*
- *l'appel de la demande d'exécution du programme contenu dans exemple.sas par la commande **Submit** du menu RUN,*
- *le programme SAS s'exécute immédiatement et occupe la session. A la fin de l'exécution du programme, deux fichiers externes de résultats sont créés :*
  - *le premier **«exemple.log»** contient l'énoncé du programme source et les commentaires en couleurs (remarques, notes, avertissements et messages d'erreurs éventuels), des commandes du programme exécutées par le système SAS,*
  - *le second **«exemple.lst»** contient les résultats produits par les procédures SAS.*

1.2 Éditeur de SAS

Pour débuter une session SAS, cliquer simplement sur l'icône SAS. L'éditeur de SAS est composé de trois fenêtres principales qui composent l'environnement SAS de programmation. Diverses fenêtres cohabitent pour permettre simultanément de consulter les données, de rédiger un programme, de consulter les erreurs commises, ou les résultats lors de l'exécution d'un programme SAS.

- *La fenêtre **Program Editor** ou l'**Enhanced Editor**[1] (éditeur amélioré) sert à la rédaction des énoncés des commandes qui seront ensuite exécutées,*
- *La fenêtre **Log** (ou journal) affiche le déroulement et les messages provenant du système SAS lors de l'exécution d'un programme. Les messages renvoyés sont de trois types : **NOTE** (en bleu) signifie qu'aucune anomalie dans le traitement demandé ; **WARNING** (en vert) est un message d'avertissement, il est possible que des erreurs (non fatales) aient eu lieu pendant le traitement ; **ERROR** (en rouge) pour signaler des erreurs importantes de traitement lors de l'exécution du programme.*
- *La fenêtre **Output** (ou de sortie) affiche les sorties résultats du programme sous forme de listing. Elle est complétée par deux autres fenêtres : la fenêtre **Graph** pour les graphiques, et la fenêtre **Results Viewer** qui permet d'afficher les résultats mis en forme à l'intérieur du logiciel SAS.*

L'environnement SAS est constitué de 3 fenêtres principales :

SAS Program Editor (F5) éditeur de texte de SAS dans lequel on écrit les commandes à exécuter ; SAS Log (F6) fenêtre de gestion de la session, compilation des commandes ligne par ligne ; **SAS Output (F7)** listing des sorties d'un programme SAS.

*La **fenêtre Log** est très importante pour s'assurer de la bonne exécution d'un programme avec les notes en bleu, les avertissements (warnings) ou erreurs non fatales en vert …et les erreurs fatales en rouge ! Les **fenêtres Log** et **Output** ne sont pas purgées automatiquement par SAS entre les exécutions ; il est donc conseillé d'en effacer régulièrement le contenu en utilisant la commande **Edit > Clear all** ou **l'icône « page blanche** ». Dans la*

[1] *Cet éditeur propose de choisir la police dans laquelle tout ou une partie du programme est rédigé, signale les mot-clés par des nuances de gras ou d'italique, et permet de replier/déplier le code ainsi q'un coloriage automatique de la syntaxe.*

version 9.3, les sorties 'texte', dont certaines agrémentées de graphiques, s'affichent par défaut dans une fenêtre HTML Results Viewer au lieu de la fenêtre listing Output.

Pour éditer une fenêtre ou se déplacer d'une fenêtre à l'autre, il suffit de l'ouvrir dans le menu VIEW soit de positionner le curseur sur la fenêtre en question si elle est déjà ouverte.

Deux autres fenêtres peuvent être consultées : ***Explorer*** *: pour visualiser les tables de données SAS et la gestion des bibliothèques et* ***Results*** *: pour naviguer rapidement parmi les résultats SAS.*

1.3 Exécution d'une session SAS

Pour exécuter les énoncés entrés, cliquer sur l'icône ***(le petit bonhomme)*** *de la commande Submit sur la barre d'outils SAS ou dans le menu RUN. Il permet l'exécution du programme écrit dans la fenêtre active.*

Après l'exécution, examiner le contenu de la fenêtre Log pour voir les messages du système à propos de la session SAS. Si la fenêtre Log indique certaines erreurs, retourner à la fenêtre Program Editor et faire afficher à l'écran les énoncés précédemment exécutés à l'aide de la commande ***Recall Last Submit*** *du menu RUN pour effectuer les corrections nécessaires et exécuter de nouveau.*

La commande ***Save*** *du menu FILE permet de sauvegarder le contenu d'une fenêtre : un programme écrit dans Program Editor sous le nom* ***«exemple.sas»*** *ou encore les résultats d'exécution de Output sous le nom* ***«exemple.lst»****. Elle permet aussi de sauvegarder des modifications effectuées sur un fichier déjà nommé.*

La commande ***Clear All*** *du menu EDIT permet d'effacer le contenu d'une fenêtre.*

La commande ***Exit*** *du menu FILE termine une session SAS.*

Enfin, de nombreux raccourcis sont accessibles à partir des icônes de la barre d'outils SAS ou encore en actionnant le bouton droit de la souris.

1.4 Programme SAS

Un programme SAS se décompose en deux étapes (types) d'instructions :

- *un (ou des) bloc(s)* ***DATA*** *pour les données : c'est un regroupement d'instructions SAS permettant de créer une, voire plusieurs tables SAS à partir de données se trouvant dans des fichiers ASCII ou des tables SAS créées auparavant. C'est à l'intérieur d'un bloc DATA que les transformations à apporter aux données sont effectuées.*
- *Un (ou des) bloc(s) PROC pour les procédures : permet l'analyse des données à l'aide d'une procédure spécifique. Les différentes procédures d'un programme SAS seront détaillées par la suite.*

Ces deux parties sont généralement indépendantes, il est donc inutile de ré-exécuter l'instruction ***data****, dont les informations des données sont déjà stockées dans une table SAS (permanente ou temporaire), lors d'un enchaînement ou de la mise au point de procédures.*

1.5 Instructions de base d'un programme SAS

Il est important de se familiariser rapidement avec certains ***mots-clés*** *et certaines* ***commandes****, nécessaires à l'élaboration d'un programme SAS, que l'on retrouve dans de nombreux programmes qui ne diffèrent que par les données utilisées et/ou par les traitements effectués. Il existe donc des instructions communes à l'ensemble des programmes SAS.*

SAS étant principalement un logiciel de programmation. L'étape DATA et les procédures sont les deux familles principales d'instructions d'un programme SAS.

- *L'étape* ***DATA*** *regroupe un ensemble d'instructions de programmation commençant par une ligne* ***DATA*** *et se terminant par l'instruction* ***RUN ;*** *qui clôt cette étape. Entre ces deux mot-clés, on utilisera des instructions pour la lecture d'une ou de plusieurs tables, d'un fichier externe, le calcul de nouvelles variables ou la modification de variables existantes.*
- *Les procédures sont des programmes déjà écrits de SAS, à la syntaxe beaucoup plus rigoureuse que celle de l'étape DATA, où il faut surtout renseigner un certain nombre de paramètres avec/sans d'options, qui accomplissent chacune une tache spécialisée. Chaque procédure commence par une instruction* ***PROC*** *suivie de son nom, et s'achève sur une instruction* ***RUN ;*** *qui demande l'exécution de cette procédure.*

La plupart des exemples utilisés pour illustrer l'application de ces instructions ainsi que de diverses procédures SAS, utilisent les données d'un échantillon de 27 petites voitures concurrentes de moins de 3,80 mètres ''Le marché belge des petites voitures''.

On dispose de neuf variables ou caractéristiques (sept quantitatives et deux qualitatives) : le prix en milliers de francs belges, la consommation urbaine, la cylindrée, la vitesse maximum, le volume maximum du coffre, le rapport poids-puissance, la longueur du véhicule, la puissance fiscale et de la marque du constructeur.

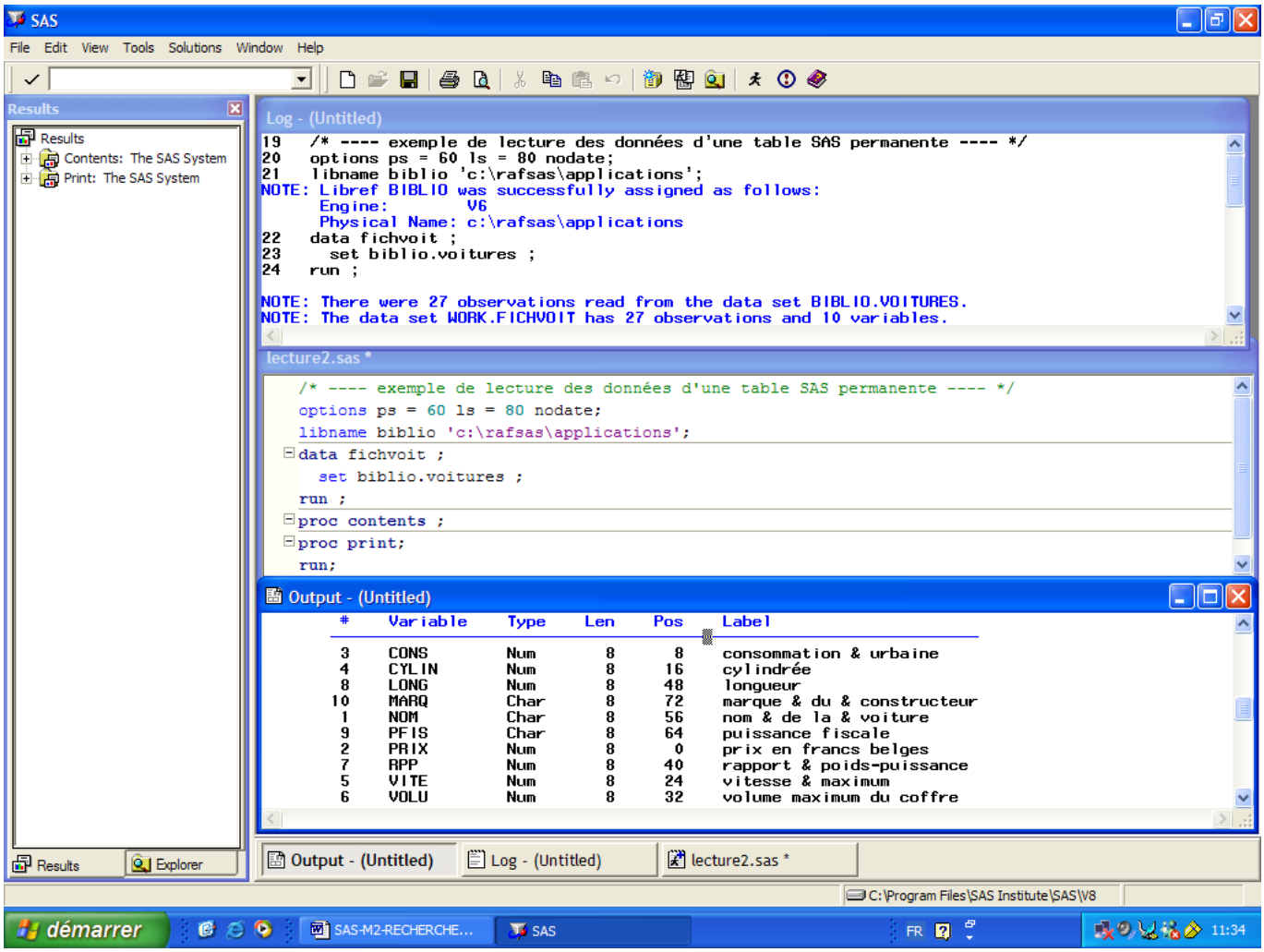
2 Lecture des données

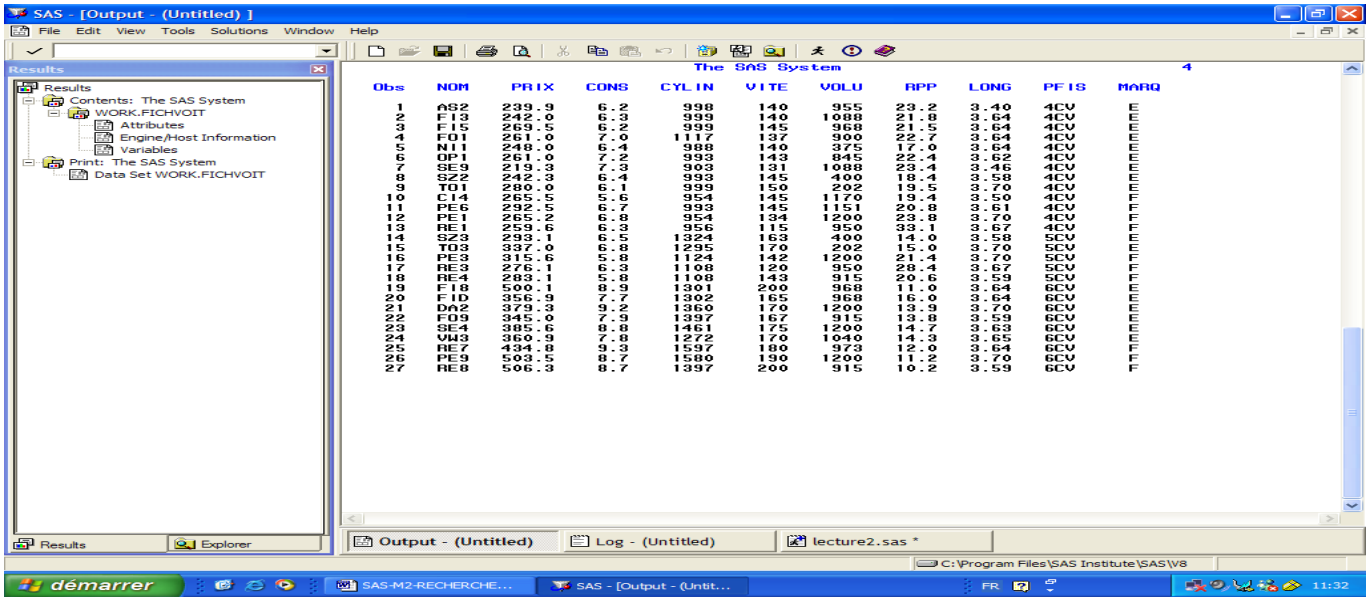
Le logiciel SAS effectue divers traitements (statistiques, économétriques, graphiques, etc.) sur des données provenant de diverses sources. Il doit donc être en mesure de lire facilement et efficacement les données à traiter. Cette étape est primordiale car si les données ne sont pas lues correctement, les résultats seront évidemment erronés.

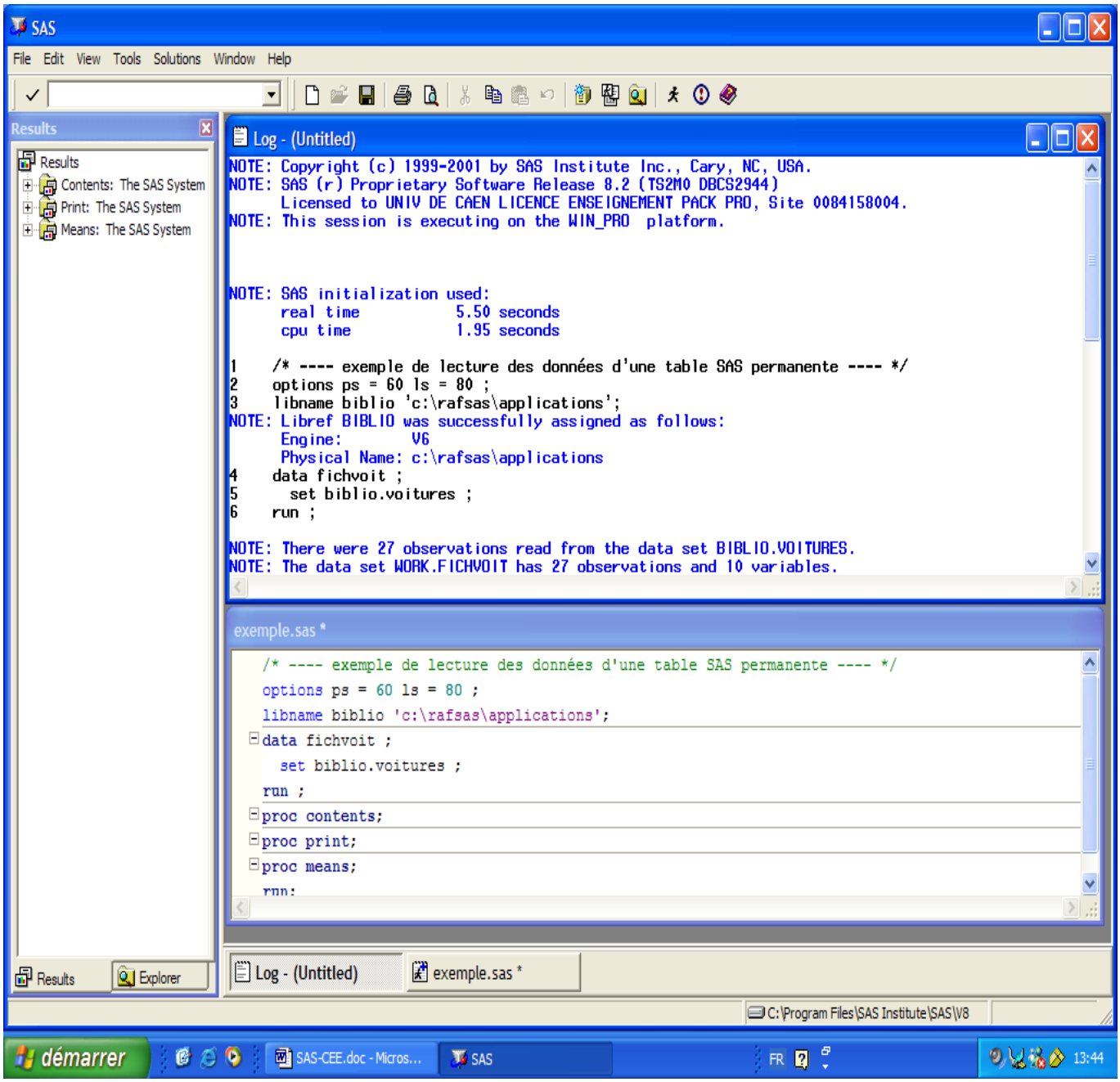
Il existe deux formats de données: ASCII (format externe au système SAS) ou SAS (format qui peut être lu par le système SAS directement). Les principales caractéristiques des deux types de fichiers sont les suivantes : un fichier ASCII peut être visualisé et modifié directement à l'aide de n'importe quel éditeur de texte; il est toutefois nécessaire de passer par le système SAS pour effectuer les mêmes opérations sur un fichier SAS. Par contre, lorsqu'un fichier SAS est défini, on peut y référer à l'intérieur du système SAS sans devoir redéfinir à chaque fois les variables associées aux différentes valeurs des observations, ce qui n'est pas le cas d'un fichier ASCII.

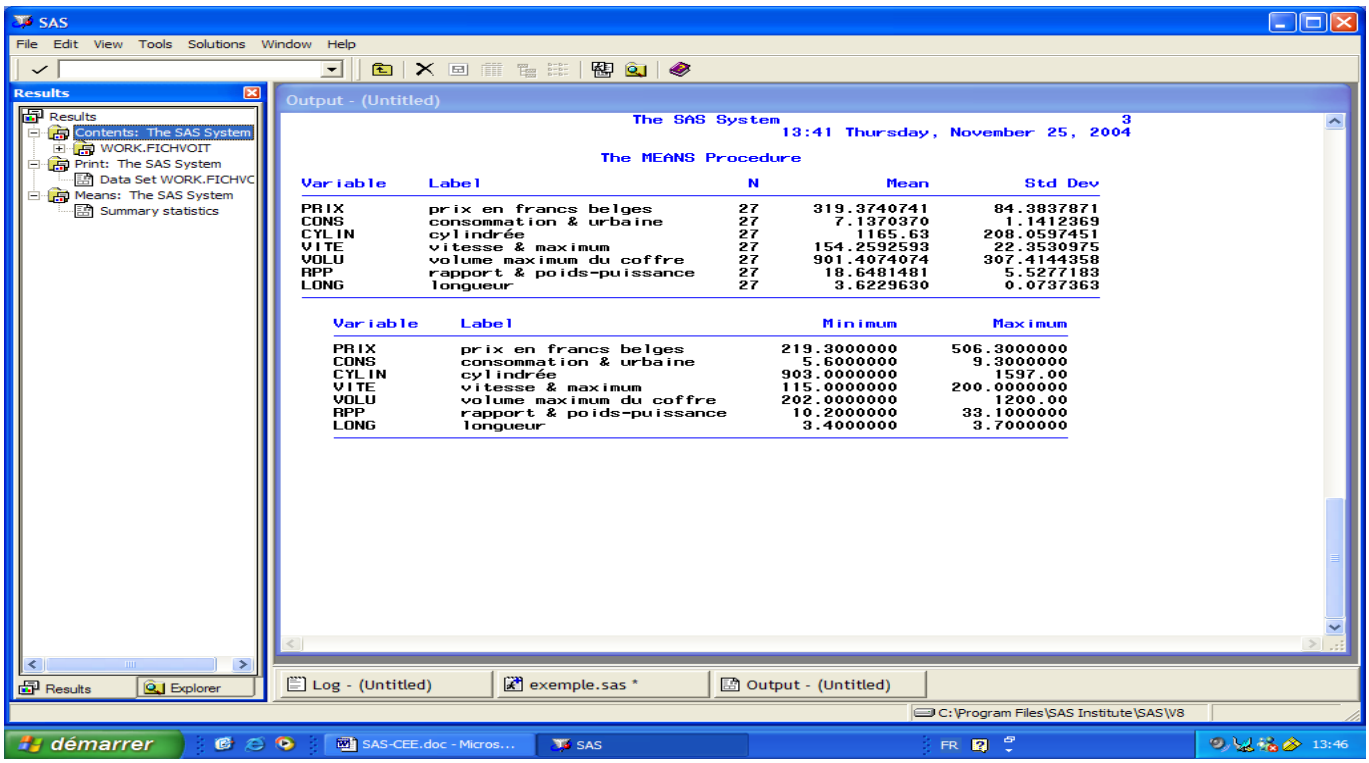
2.1 Importation des données à partir de SAS

On peut importer différents types de fichiers de données (base (***.dbf**), Excel (***.xls**), Tab Delimited File (***.txt**), lotus (***.wq1**), …,etc.) à partir de la commande ***Import*** du menu *FILE* de SAS. Il suffit de spécifier le chemin d'accès au fichier de données, choisir la bibliothèque de son choix (WORK par défaut) ainsi que le nom de la table SAS à créer. A noter que toute table SAS créée dans la bibliothèque WORK ne sera que temporaire c'est-à-dire qu'elle sera détruite lorsque la session SAS sera terminée.




SAS - [Output - (Untitled)]
File Edit View Tools Solutions Window Help
Results
Results
Contents: The SAS System
WORK.FICHVOIT
Attributes
Engine/Host Information
Variables
Print: The SAS System
Data Set WORK.FICHVOIT
The SAS System 4
Obs NOM PRIX CONS CYLIN VITE VOLU APP LONG PFIS MARQ
1 AS2 239.9 6.2 998 140 955 23.2 3.40 4CV E
2 FI3 242.0 6.3 999 140 1088 21.8 3.64 4CV E
3 FI5 269.5 6.2 999 145 968 21.5 3.64 4CV E
4 FO1 261.0 7.0 1117 137 900 22.7 3.64 4CV E
5 NI1 248.0 6.4 988 140 375 17.0 3.64 4CV E
6 OP1 261.0 7.2 993 143 845 22.4 3.62 4CV E
7 SE9 219.3 7.3 903 131 1088 23.4 3.46 4CV E
8 SZ2 242.3 6.4 993 145 400 18.4 3.58 4CV E
9 TO1 280.0 6.1 999 150 202 19.5 3.70 4CV E
10 CI4 265.5 5.6 954 145 1170 19.4 3.50 4CV F
11 PE5 292.5 6.7 993 145 1151 20.8 3.61 4CV F
12 PE1 265.2 6.8 954 134 1200 23.8 3.70 4CV F
13 RE1 259.6 6.3 956 115 950 33.1 3.67 4CV F
14 SZ3 293.1 6.5 1324 163 400 14.0 3.58 5CV E
15 TO3 337.0 6.8 1295 170 202 15.0 3.70 5CV E
16 PE3 315.6 5.8 1124 142 1200 21.4 3.70 5CV F
17 RE3 276.1 6.3 1108 120 950 28.4 3.67 5CV F
18 RE4 283.1 5.8 1108 143 915 20.6 3.59 5CV F
19 FI8 500.1 8.9 1301 200 968 11.0 3.64 6CV E
20 FI0 356.9 7.7 1302 165 968 16.0 3.64 6CV E
21 DA2 379.3 9.2 1360 170 1200 13.9 3.70 6CV E
22 FO3 345.0 7.9 1397 167 915 13.8 3.59 6CV E
23 SE4 385.6 8.8 1461 175 1200 14.7 3.63 6CV E
24 VW3 360.9 7.8 1272 170 1040 14.3 3.65 6CV E
25 AE7 434.8 9.3 1597 180 973 12.0 3.64 6CV F
26 PE9 503.5 8.7 1580 190 1200 11.2 3.70 6CV F
27 RE8 506.3 8.7 1397 200 915 10.2 3.59 6CV F
Results Explorer
Output - (Untitled) Log - (Untitled) lecture2.sas *
C:\Program Files\SAS Institute\SAS\V8
démarrer
SAS-M2-RECHERCHE...
SAS - [Output - (Untit...
FR
11:32


SAS
File Edit View Tools Solutions Window Help
Results
Results
Contents: The SAS System
Print: The SAS System
Means: The SAS System
Log - (Untitled)
NOTE: Copyright (c) 1999-2001 by SAS Institute Inc., Cary, NC, USA.
NOTE: SAS (r) Proprietary Software Release 8.2 (TS2M0 DBCS2944)
Licensed to UNIV DE CAEN LICENCE ENSEIGNEMENT PACK PRO, Site 0084158004.
NOTE: This session is executing on the WIN_PRO platform.
NOTE: SAS initialization used:
real time 5.50 seconds
cpu time 1.95 seconds
1 /* ---- exemple de lecture des données d'une table SAS permanente ---- */
2 options ps = 60 ls = 80 ;
3 libname biblio 'c:\rafsas\applications';
NOTE: Libref BIBLIO was successfully assigned as follows:
Engine: V6
Physical Name: c:\rafsas\applications
4 data fichvoit ;
5 set biblio.voitures ;
6 run ;
NOTE: There were 27 observations read from the data set BIBLIO.VOITURES.
NOTE: The data set WORK.FICHVOIT has 27 observations and 10 variables.
exemple.sas *
/* ---- exemple de lecture des données d'une table SAS permanente ---- */
options ps = 60 ls = 80 ;
libname biblio 'c:\rafsas\applications';
data fichvoit ;
set biblio.voitures ;
run ;
proc contents;
proc print;
proc means;
run;
Results Explorer
Log - (Untitled) exemple.sas *
C:\Program Files\SAS Institute\SAS\V8
démarrer
SAS-CEE.doc - Micros...
SAS
FR
13:44



2.2 Saisie directe des données

*Si les données ne sont pas déjà dans un fichier, celles-ci peuvent être entrées à la main à l'intérieur d'un programme SAS par le biais du mot-clé **CARDS** sous l'instruction **INPUT**.*

Programme 2.2.1 :

```
/* ---- exemple de saisie des données  ---- */
data voitures ;
input  nom$  prix  cons  cylin  vite volu  rpp  long  pfis$  marq$ ;
cards ;
      AS2  239.9  6.2  998  140  955  23.2  3.40  4CV  E
      CI4  265.5 5.6  954  145  1170  19.4  3.50 4CV  F
     ...
     VW3  360.9  7.8  1272  170  1040  14.30  3.65  6CV  E
run;
proc print data = voitures ;
run ;
```

***data** est le mot-clé, suivi d'un nom choisi, qui spécifie au système SAS de lire des données et de les organiser en table SAS sous le nom spécifié, ici voitures.*

***cards** est le mot-clé qui indique au système SAS que les données qui doivent être lues, sur les lignes suivantes jusqu'à ce qu'il atteigne l'instruction **run** qui indique la fin de la lecture des données. Chaque ligne de données doit contenir autant de données qu'il y a de variables définies dans l'instruction **input**. A noter que les données doivent être séparées par au moins un espace-blanc et aucune ligne de données ne doit se terminer par un point-virgule.*

*La procédure **proc print** est ajoutée et, si nécessaire, l'option **data =** voitures, pour imprimer la table SAS voitures afin de vérifier que les données ont été correctement saisies. Et enfin, le mot-clé **run** exécute la procédure.*

Remarque :

- *A noter, que sans l'option **data =** nom_table, SAS lit automatiquement la dernière table SAS créée. Cette option, ajoutée à une procédure, permet ainsi de changer cette lecture par défaut en spécifiant la table de données SAS de son choix. Elle permet également d'éviter les confusions lorsqu'on travaille avec plus d'une table de données SAS.*

Programme 2.2.2 :

```
/* - Saisie de données - Plusieurs observations sur la même ligne */
data voitures ;
input  nom$  prix  cons  cylin  vite volu  rpp  long  pfis$  marq$  @@ ;
cards ;
  AS2  239.9  6.2  998  140  955  23.2  3.40  4CV  E CI4  265.5 5.6  954  145  1170  19.4  3.50
4CV  F        VW3  360.9  7.8  1272  170  1040  14.30  3.65  6CV  E
run;
```

Lorsque le volume des données est très réduit, celles-ci peuvent être intégrées au programme, avec la commande ***cards****, en mettant plusieurs observations par ligne. En effet, le double caractère "@@" qui termine la commande* ***input****, a pour effet de maintenir un article dans le tampon mémoire ''buffer'' de lecture jusqu'à ce qu'il soit complètement lu.*

2.3 Lecture d'un fichier de données en format ASCII

Pour lire des données (ASCII) provenant d'un fichier externe et les transformer en un ensemble de données sous format SAS, dans la fenêtre Program Editor, on doit utiliser dans l'ordre les instructions SAS suivantes : ***DATA****,* ***INFILE****,* ***INPUT*** *et* ***RUN.***

Cet enchaînement constitue un bloc DATA. Le programme ci-dessous, illustre la lecture et l'impression des données de l'exemple d'application.

Programme 2.3 :

```
/* ---- exemple de lecture des données à partir d'un fichier ---- */
data voitures ;
options ps = 60 ls = 80 ;
options nodate ;
title ;
footnote 'Résultats '
infile  'c:|rafsas|applications|donnees|voit.prn' ;
input  nom$  prix  cons  cylin  vite volu  rpp  long pfis$ marq$ ;
proc print ;
run ;
```

Examinons chacune des lignes de ce programme.

data *est le mot-clé utilisé pour indiquer au système SAS qu'on souhaite créer un fichier de données (data set) ou table SAS, il est suivi du nom du fichier de données qui sera créé, ici voitures, utilisé pour représenter l'ensemble des données. Ce nom attribué à la table SAS ne doit pas excéder huit caractères.*

options *pagesize* ***ps = 60*** *et linesize* ***ls = 80*** *cette instruction ''optionnelle'' est utilisée pour une meilleure présentation des résultats. Ainsi, dans les fichiers résultats, la commande* ***ps = 60*** *indique que chaque page contient 60 lignes tandis que la commande* ***ls = 80*** *détermine le nombre maximum de caractères par ligne. Par défaut, sans les options format,* ***ps = 60*** *et* ***ls = 132****.*

options nodate *''optionnelle '' permet d'éliminer l'heure et la date du jour dans le rapport imprimé.*

title *''optionnelle'' est utilisée pour éliminer le titre par défaut ''****The SAS System****''. On peut également titrer les résultats en ajoutant la commande* ***title i 'Marché des petites voitures'*** *qui permet de rajouter le titre contenu entre apostrophes sur la ligne i (i doit être un nombre entier entre 1 et 10).*

footnote *''optionnelle'' est utilisée pour ajouter une note de bas de page à la procédure de sortie.*

infile *précise l'endroit où se trouve les données. Il identifie un fichier de données externe, ici voit.prn, dans lequel on souhaite lire à l'aide d'une instruction* ***input****. A noter que le nom du fichier y compris le chemin d'accès à ce fichier, doivent être entre apostrophes.*

input *est l'instruction qui décrit l'arrangement des valeurs d'une observation du fichier d'entrée et assigne au logiciel SAS les variables à lire ainsi que l'ordre de lecture. Elle suppose que les valeurs des variables sont séparées par au moins un blanc ou espace, et que les valeurs manquantes sont spécifiées par un point. Notons que certaines variables de la liste sont suivies du symbole ''****$****'', cela indique qu'elles sont de type alphanumérique (contiennent des valeurs alphabétiques), les autres variables sont numériques ou quantitatives. Il est fortement conseillé de déclarer les variables qualitatives de type alphanumérique.*

proc print *est la procédure ''facultative'' qui permet d'imprimer les données, dans la fenêtre* ***Output****, pour vérifier si l'on a effectué correctement la lecture des données.*

run *est le mot-clé qui permet d'exécuter une procédure. Ici, toutes les instructions SAS précédentes seront exécutées.*

Remarques : ———————————————————————————————————————

- *Chaque ligne du programme (sauf exceptions) se termine par un point-virgule ‘’* ***;***‘’.
- *Tous les programmes SAS (sauf exception) doivent débuter par le mot-clé* ***data****.*
- *Des règles précises sont établies dans chaque procédure SAS pour le traitement des valeurs manquantes ou non disponibles. S’il y a des données manquantes, elles doivent être indiquées par un* ***point*** *(****.****) et non un* ***espace blanc****. Lors de l’affichage des données à l'aide d'une procédure SAS, le* ***point*** *est utilisé pour représenter une* ***valeur numérique manquante*** *et un* ***blanc*** *pour représenter une* ***valeur alphanumérique manquante****.*
- *Une variable alphanumérique peut contenir jusqu’à* ***200 caractères*** *et une variable numérique jusqu’à* ***32 caractères****.*
- *Pour faciliter la lecture d’un programme, on peut y insérer des commentaires qui seront ignorés par SAS lors de l’exécution. Pour cela, les deux syntaxes suivantes sont possibles :* ***/* commentaire */*** *ou* ****commentaire;*** *. La première forme permet également d'éliminer temporairement l'exécution d’un bloc d'instructions en insérant au début du bloc les symboles ‘’****/****‘’ et à la fin, ‘’*****/****’’. Il est également autorisé d’insérer des lignes blanches, de façon à rendre le programme plus lisible.*
- *A noter qu’un programme SAS peut être introduit indifféremment en* ***minuscules*** *ou en* ***majuscules****.*

———————————————————————————————————————

3 Création d'une table SAS permanente

Dans les exemples précédents, les tables SAS créées sont temporaires. C'est-à-dire qu'elles sont accessibles au cours de la session SAS seulement. Lorsque la session est terminée, les tables SAS temporaires sont éliminées. Si les données sont sauvegardées de manière permanente, il sera possible d'y référer lors d'une session ultérieure. On pourra alors procéder directement à une analyse des données sans devoir recréer la table SAS.

Programme 3.1 : ———————————————————————————————————————

```
/* ---- exemple de création d'une table SAS permanente ---- */
libname biblio 'c:|rafsas|applications';
data biblio.voitures ;
infile 'c:|rafsas|applications|donnees|voit.prn' ;
input  nom$  prix  cons  cylin  vite volu  rpp  long pfis$ marq$ ;
label nom = 'nom & de la & voiture'
      prix = 'prix en francs belges'
      cons = 'consommation & urbaine'
      cylin = 'cylindrée'
      vite = 'vitesse & maximum'
      volu = 'volume maximum du coffre'
      rpp = 'rapport & poids-puissance'
      long = 'longueur'
      pfis = 'puissance fiscale'
      marq = 'marque & du & constructeur';
run;
```

———————————————————————————————————————

libname *est l’instruction qui permet de créer une table SAS permanente, il suffit de donner à la table SAS un nom en deux parties, plutôt qu'un nom simple. La première partie du nom, ici* ***biblio****, est le nom de référence de la bibliothèque personnelle de travail qui sera créée (ou qui est déjà créée) et associée au répertoire de son choix, ici* ***c:\rafsas\applications****.*

Ensuite, l’instruction ***Data*** *biblio.voitures entraîne la création de la table SAS voitures dans la bibliothèque biblio du répertoire c:\rafsas\applications. Cette table SAS sera permanente, c’est-à-dire qu’elle restera après la fin de l’exécution du programme SAS. En fait, c’est un fichier de données* ***voitures.sas7bdat*** *qui sera créé dans ce répertoire et sera toujours associé à la table SAS voitures. L’extension sas7bdat est spécifique au système SAS pour reconnaître la table permanente.*

voitures.sas7bdat

Par défaut, si on omet le nom de sa propre bibliothèque, la table SAS sera enregistrée dans la bibliothèque ***work*** *associée au répertoire* ***c:\sas\saswork*** *sous le nom* ***work.voitures*** *ou tout simplement voitures. Pour associer la première partie du nom à un répertoire, on utilise toujours une instruction* ***Libname****.*

*La commande **label**, sous l'instruction **input** permet de libeller clairement les variables. Cette description, qui ne doit pas excéder **40 caractères**, sera imprimée en entête des colonnes de chaque variable. Seule la dernière variable décrite à l'aide de cette option se termine par un **point-virgule ' ;',***

Remarques : ———————————————————————————————————

- *Si les données sont déjà dans une table SAS permanente, il n'est pas nécessaire de débuter le programme par un bloc DATA.*
- *La table SAS permanente biblio.voitures est conservée dans la bibliothèque biblio, rattachée au répertoire 'c:\rafsas\applications". Les données sont enregistrées dans la base ou fichier de données voiture.sd2.*
- *Avant de travailler sur les données de sa propre bibliothèque, il faut d'abord y faire référence c'est-à-dire la positionner comme bibliothèque de travail à partir de l'icône "**libraries-petits tiroirs"**. Par défaut, WORK est la bibliothèque de travail.*

———

*Le programme suivant, illustre la commande **set** qui permet de lire et de créer une nouvelle table SAS à partir des données d'une table SAS permanente déjà créée.*

Programme 3.2 : ———————————————————————————————————

```
/* ---- exemple de lecture des données d'une table SAS permanente ---- */
options ps = 60 ls = 80 ;
libname biblio 'c:|rafsas|applications';
data fichvoit ;
  set biblio.voitures ;
run ;
```

———

*L'instruction **libname** est identique à celle du programme précédent.*

*L'instruction **data** spécifie le nom, ici fichvoit, de la table SAS à créer et qui contiendra les données de la table permanente biblio.voitures.*

*La commande **set** nom_table est utilisée pour lire une table SAS permanente, ici biblio.voitures. A noter que l'instruction **input** qui permet la description des variables de la table n'est plus utilisée vu que toute l'information est enregistrée dans la table permanente y compris les libellés des variables.*

*Le mot-clé **run** exécute la lecture des données et la création de la table SAS fichvoit.*

*Le programme suivant, illustre, à partir des commandes **drop** et **keep** ou options de l'instruction **data (drop=** ou **keep=)** , comment on peut éliminer ou sélectionner certaines variables d'une table SAS.*

*L'option entre parenthèses **(keep=** liste_de_variables**)** de l'instruction **data** permet de sélectionner certaines variables de la table SAS permanente spécifiée par la commande **set**, ici biblio.voitures. Ces variables sont conservées dans la table SAS nommée, ici voit1, de l'instruction **data**.*

*De même, l'option entre parenthèses **(drop=** liste_de_variables**)** de l'instruction **data** permet d'éliminer certaines variables de la table SAS permanente spécifiée par la commande **set**, ici biblio.voitures. Les variables restantes sont conservées dans la table SAS nommée, ici voit2, de l'instruction **data**.*

Les deux tables SAS créées voit1 et voit2 sont identiques; elles contiennent les mêmes variables à savoir la liste nom, prix, pfis et marq.

*On peut aussi utiliser, illustration 2, les commandes **keep** et **drop** plutôt que les options du même nom pour effectuer aboutir aux même résultats. La table SAS.*

Programme 3.3 :

```
/* ---- exemple de création d'une table SAS à partir d'un sous-ensemble de variables-- */
/* illustration 1 :  Options keep et drop de l'instruction data  */
options ps = 60 ls = 80 ;
data voit1 (keep= nom prix pfis marq)
data voit2 (drop= cons  cylin  vite volu  rpp  long) ;
set biblio.voitures ;
proc contents data=voit1 ;
proc contents data=voit2 ;
run;
/* illustration 2 : les commandes keep et drop  */
options ps = 60 ls = 80 ;
data voit11;
set biblio.voitures ;
keep nom prix pfis marq;
run;
data voit22;
set biblio.voitures ;
drop cons  cylin  vite volu  rpp  long ;
run;
```

6 LES PROCEDURES : PROC

Jusqu'à présent, on a vu comment créer une table SAS et comment modifier l'information contenue dans celle-ci. On peut à présent utiliser les procédures SAS pour analyser les données. Un programme simple est généralement constitué d'un bloc DATA et est suivi d'un ou plusieurs blocs PROC. On peut toutefois utiliser plusieurs blocs DATA à l'intérieur d'un même programme, et les blocs DATA et PROC n'ont pas à apparaître dans un ordre précis. En effet, si les données qu'on souhaite analyser sont déjà dans un fichier SAS permanent, il n'est pas nécessaire de débuter le programme par un bloc DATA.

L'instruction ***PROC*** *indique au système SAS qu'on souhaite utiliser une procédure particulière déjà définie pour l'analyse de données. Elle commence par le mot PROC et est suivie du nom de la procédure à exécuter. Il est possible de préciser dans le bloc PROC le nom de la table de données SAS avec laquelle on souhaite travailler (* ***option data =*** *), sur quelles variables l'analyse doit porter (* ***commande var****) ou encore que l'on souhaite étudier les données par groupe (****commande by*** *). Si aucune précision n'est faite, la procédure utilise la dernière table SAS créée, toutes les variables du fichier (ou toutes les variables numériques s'il s'agit d'une procédure de calcul) et l'ensemble des observations (plutôt que des groupes d'observations).*

4.1 Procédures de gestion des données - Utilitaires

Certaines procédures sont utiles pour vérifier le bon déroulement de la phase de lecture des données et de la création de la table SAS. Elles permettent notamment de décrire les caractéristiques de la table, d'imprimer les données ou encore de trier les observations selon les valeurs ou catégories d'une ou plusieurs variables.

4.1.1 PROC CONTENTS

Le programme suivant, illustre les caractéristiques de la table SAS biblio.voitures.

Programme 4.1.1:

```
/* ---- Caractéristique de la table SAS ---- */
proc contents data=biblio.voitures ;
run;
```

La procédure proc contents permet de décrire le contenu d'une table SAS. Elle présente entre autres les libellés des variables, leurs types, le nombre de colonnes qu'elles occupent et leurs positions.

Le mot-clé run exécute la procédure dont voici les résultats que l'on peut consulter dans la fenêtre OUTPUT.

```
                          The CONTENTS Procedure
Data Set Name: BIBLIO.VOITURES                        Observations:         27
Member Type:   DATA                                   Variables:            10
Engine:        V6                                     Indexes:              0
Created:       15:05 Monday, February 26, 2001        Observation Length:   80
Last Modified: 15:05 Monday, February 26, 2001        Deleted Observations: 0
Protection:                                           Compressed:           NO
Data Set Type:                                        Sorted:               YES
Label:

                -----Engine/Host Dependent Information-----
Data Set Page Size:         8192
Number of Data Set Pages:   1
First Data Page:            1
Max Obs per Page:           102
Obs in First Data Page:     27
Number of Data Set Repairs: 0
File Name:                  C:\RAFSAS\applications\voitures.sd2
Release Created:            6.12.00
Host Created:               WIN_95

       -----Alphabetic List of Variables and Attributes-----
#    Variable    Type    Len    Pos    Label
-------------------------------------------------------------------
3    CONS        Num       8     16    consommation urbaine
4    CYLIN       Num       8     24    cylindrée
8    LONG        Num       8     56    longueur
10   MARQ        Char      8     72    marque du constructeur
1    NOM         Char      8      0    nom de la voiture
9    PFIS        Char      8     64    puissance fiscale
2    PRIX        Num       8      8    prix en francs belges
7    RPP         Num       8     48    rapport poids-puissance
5    VITE        Num       8     32    vitesse maximum
6    VOLU        Num       8     40    volume maximum du coffre
```

4.1.2 PROC SORT

*La procédure **proc sort**, qui doit obligatoirement se situer avant toutes les procédures utilisant la commande **by**, permet d'ordonner (trier) les observations d'une table SAS d'après les valeurs d'une ou plusieurs variables. Par défaut, le tri est croissant ou alphabétique. Cette procédure est toujours accompagnée de la commande **by**, nécessaire pour grouper les données avant d'effectuer une analyse par groupe. La syntaxe générale de cette procédure se présente sous la forme suivante :*

Syntaxe - options - commandes

proc sort *<options > ;*
by *<descending> variable;*

Programme 4.1.2.1 :

```
/* ---- La table SAS biblio.voitures est permanente ---- */
proc sort data=biblio.voitures ; by prix ; run ;
```

Dans la bibliothèque biblio, les observations de la table SAS permanente voitures sont triées par valeur croissante du prix.

*Il est possible de trier selon 2 ou plusieurs variables, dans ce cas le tri est effectué sur la première variable et si certaines valeurs de la première variable sont identiques, celles-ci seront ordonnées en fonction des valeurs de la deuxième variable et ainsi de suite. On peut toutefois, en utilisant l'option **descending**, trier par ordre décroissant.*

Programme 4.1.2.2 :

```
/* ---- La table SAS biblio.voitures est permanente ⇒ sauvegarde des résultats ---- */
proc sort data=biblio.voitures out=biblio.resultri;  /*création de la table contenant les résultats */
by marq descending prix ; run ;
```

Les observations de la table SAS d'origine biblio.voitures sont regroupées par la marque et ordonnées, par ordre décroissant du prix, à l'intérieur de chaque groupe. Le résultat de ce tri est enregistré dans la table SAS sous le nom biblio.resultri.

*L'option **out=** est ajoutée pour indiquer qu'il faut conserver les données ordonnées dans une nouvelle table SAS, ici biblio.resultri, sinon par défaut, il les enregistre sous le même nom que la table d'origine.*

*L'option **nodup** peut-être ajoutée si on veut éliminer de la table SAS toutes les observations identiques.*

4.1.3 PROC PRINT

*La procédure **proc print** permet d'imprimer dans OUTPUT et d'afficher dans LOG le contenu d'une table SAS. Elle est donc très utile pour vérifier si on a effectué correctement le transfert des données, la concaténation de tables SAS et les transformations apportées aux données au cours d'un bloc DATA. La syntaxe générale de cette procédure peut se résumer ainsi :*

Syntaxe - options - commandes

```
proc print  <options > ;
by <descending> variable;
var  liste de variables ;
```

Programme 4.1.3 :

```
/* ---- La table SAS voitures est permanente ---- */
proc print label ; run;
/* ---- tri de la table SAS    */
proc sort data=biblio.voitures; by marq ; run;
/* ---- impression des variables prix et pfis selon la variable marq     */
proc print label noobs round split = '&';
by marq ; var prix pfis ; run ;
```

Plusieurs options sont disponibles avec cette procédure. Les plus courantes sont :

***label** : permet d'afficher le libellé des variables,*

***noobs** : permet d'éliminer le numéro des observations qui est indiqué par le logiciel SAS (OBS),*

***split = '&'** : permet de présenter la description de la variable sur plusieurs lignes. Ainsi, par exemple la description de la variable marq apparaîtra sur 3 lignes. Le caractère '&' utilisé par cette option est arbitraire en autant que l'on utilise le même dans la description de la variable avec l'option label.*

***round** : spécifie qu'il faut arrondir les résultats à deux décimales.*

*La commande **var**, située après la commande **by**, indique que ne seront affichées que les valeurs des variables prix et pfis par groupe d'observations de la variable marq.*

4.2 Procédures de transformations des données

4.2.1 PROC RANK

*La procédure **proc rank** permet de créer de nouvelles variables de rangs déterminées à partir des variables quantitatives de la table SAS. La syntaxe générale de cette procédure peut se résumer ainsi :*

Syntaxe - options - commandes

```
proc rank  <options > ;
by <descending> variable;
ranks liste de nouvelles variables ;
var  liste de variables ;
```

*Dans la première partie du programme, la procédure **proc rank,** détermine les rangs des observations de la variable prix spécifiée dans la commande **var**, et les recopie dans une nouvelle table SAS biblio.resulta1 spécifiée par l'option **out=**. Ainsi, dans cette table de sortie SAS, on retrouve les variables initiales et la nouvelle variable créée dont le nom ou le libellé, ici rgprix, est spécifié par la commande **ranks**. Par défaut et sans option, cette procédure détermine les rangs et les valeurs égales (ex-æquo) sont affectées du rang moyen.*

Programme 4.2.1 :

```
/*  rangs de variables continues */
```

```
proc rank  data=biblio.voitures out=biblio.resulta1;
ranks rgprix ; var prix ; run ;
proc print data=biblio.resulta1 ; run ;
/*  rangs de variables continues par groupe  */
proc sort data=biblio.voitures ;  /*  tri de la table par groupe  */
by marq ; run ;
proc rank  groups=2 data=biblio.voitures out=biblio.resulta2;
ranks rgprix ; by marq ; var prix ; run ;
proc print data=biblio.resulta2 ; run;
```

Cette procédure dispose d'autres options de calcul que les rangs ; l'option,

***fraction** : détermine les valeurs de la fonction de répartition,*

***groups =** n : permet de spécifier le nombre de valeurs de rangs utilisées et ainsi de découper en classes une variable quantitative avec des effectifs sensiblement égaux,*

***normal=blom** : les valeurs d'une distribution normale, plutôt que, par défaut, les valeurs des rangs,*

***ties = .mean** ou **high** ou encore **low**, permet de spécifier la façon de gérer les ex-æquo.*

*Dans la deuxième partie du programme, la commande **by** a été rajoutée, suivie du nom d'une variable qualitative marq qui indique que les calculs doivent se faire par groupe d'observations de cette variable ; la table SAS doit être triée selon les modalités de cette variable.*

*La commande **ranks** doit être spécifiée si l'on veut que les variables initiales soient recopiées en sortie. Sinon, les variables gardent le même nom. Il y a une correspondance terme à terme entre les noms des deux listes de variables.*

*Enfin, par défaut c'est-à-dire sans la commande **var**, toutes les variables continues de la table SAS seront traitées.*

4.2.2 PROC STANDARD

*La procédure **proc standard** permet le centrage et la réduction ou la ``standardisation'' de variables quantitatives. La syntaxe générale de la procédure est de la forme suivante :*

Syntaxe - options - commandes

***proc standard** <options > ;*
***by** <descending> variable;*
***var** liste de variables ;*
***weight** variable;*

Programme 4.2.2 :

```
/*  standardisation de variables  */
proc standard print mean=0 std=1 data=biblio.voitures out=biblio.resultat;
var prix cons ; run ;
proc print data=biblio.resultat ; run ;
```

*la commande **var** suivie des variables prix et cons, indique à la procédure **proc standard** la standardisation de ces deux variables (par défaut, c'est-à-dire sans la commande **var**, toutes les variables continues de la table SAS seront standardisées) selon les options suivantes :*

***print** : pour l'impression des moyennes et des écarts-types des variables traitées,*

***mean=**0 et **std=**1 : indique la nouvelle valeur de la moyenne et de l'écart-type,*

***data=**biblio.voitures : spécifie le nom de la table SAS, par défaut, la dernière créée,*

***out=biblio**.resultat : spécifie le nom de la table SAS, dans la bibliothèque personnelle biblio, qui contiendra les variables initiales et celles standardisés.*

On peut également ajouter d'autres options :

***replace** : pour indiquer le remplacement de toute donnée continue manquante (spécifiée dans la table par un point '.') soit remplacée par la nouvelle valeur de la moyenne,*

***vardef** : précise le diviseur dans le calcul de la variance (df, n, wdf, wgt).*

Ainsi que d'autres commandes :

by *: suivie du nom d'une variable qualitative, spécifie au système que la standardisation doit se faire selon les groupes ou les modalités de cette variable ; la table d'entrée SAS doit être triée.*

weight *: suivie du nom de la variable contenant les pondérations des observations pour la standardisation.*

Résultats du programme 4.2.2 :

```
                                The STANDARD Procedure
                                Standard
Name            Mean            Deviation               N    Label

PRIX        319.374074          84.383787              27    prix en francs belges
CONS          7.137037           1.141237              27    consommation & urbaine
```

Obs	NOM	PRIX	CONS	CYLIN	VITE	VOLU	RPP	LONG	PFIS	MARQ
1	AS2	-0.94182	-0.82107	998	140	955	23.2	3.40	4CV	E
2	FI3	-0.91693	-0.73345	999	140	1088	21.8	3.64	4CV	E
3	FI5	-0.59104	-0.82107	999	145	968	21.5	3.64	4CV	E
4	FO1	-0.69177	-0.12008	1117	137	900	22.7	3.64	4CV	E
5	NI1	-0.84583	-0.64582	988	140	375	17.0	3.64	4CV	E
6	OP1	-0.69177	0.05517	993	143	845	22.4	3.62	4CV	E
7	SE9	-1.18594	0.14280	903	131	1088	23.4	3.46	4CV	E
21	RE1	-0.70836	-0.73345	956	115	950	33.1	3.67	4CV	F
22	PE3	-0.04473	-1.17157	1124	142	1200	21.4	3.70	5CV	F
23	RE3	-0.51282	-0.73345	1108	120	950	28.4	3.67	5CV	F
24	RE4	-0.42987	-1.17157	1108	143	915	20.6	3.59	5CV	F
25	RE7	1.36787	1.89528	1597	180	973	12.0	3.64	6CV	F
26	PE9	2.18201	1.36953	1580	190	1200	11.2	3.70	6CV	F
27	RE8	2.21519	1.36953	1397	200	915	10.2	3.59	6CV	F

4.2.3 PROC FORMAT

*La procédure **proc format**, permet de rendre plus ''présentables'' le rapport de résultats d'une analyse. Elle s'utilise en début de programme avant l'instruction **data** tout comme la commande **value**. Cette procédure permet d'effectuer certains changements sans avoir à réécrire toutes les données.*

Programme 4.2.3 :

```
/* ---- Donner un nom au modalités d'une variable nominale ---- */
proc format ;
value $a  '4CV' = '4 chevaux fiscaux'  '5CV' = '5 chevaux fiscaux'  '6CV' = '6 chevaux fiscaux' ;
value $b  'F' = 'marque française'    'E' = 'marque étrangère' ;
data biblio.voitures ;
infile 'c:|rafsas|applications|données|voit.dat' ;
input  nom$  prix  cons  cylin  vite  volu  rpp  long pfis$ marq$ ;
format pfis a. ;
format marq b. ;
run ;
proc print label noobs ; run ;
```

*La commande **value** permet de modifier les valeurs d'une variable par d'autres valeurs. Ainsi, nous remplaçons par exemple, les modalités des variables pfis et marq.*

Le nom du format a ''arbitraire'' doit être précédé du symbole ''$'' si les valeurs de la variable sont alphanumériques.

*Ensuite, après l'instruction **input**, par exemple, la commande **format marq b.;** indique que les valeurs de la variable marq seront modifiées selon le format b (ne pas oublier, dans la commande, le point ''.'' après le nom du format).*

4.3 Autres transformations des données

Opérateurs arithmétiques

**	*puissance (exposant)*
*	*multiplication*
/	*division*
+	*addition*
-	*soustration*

Opérateurs logiques

< ou LT	*inférieur strictement à*
<= ou LE	*inférieur ou égal à*
> ou GT	*supérieur strictement à*
>= ou GE	*supérieur ou égal à*
= ou EQ	*égal à*
^= ou NE	*différent de*
OR	*ou*
& ou AND	*et*
^ ou NOT	*négation*

*Une opération de comparaison produit la valeur '**1**', si la comparaison est vraie et la valeur '**0**', si elle est fausse. La priorité d'exécution va d'abord aux éléments entre parenthèses, puis à (1) **, NOT; (2) *, /; (3) -, +; (4) EQ, NE, LT, GT, LE, GE; (5) AND et (6) OR.*

SAS compte également un bon nombre de fonctions qui peuvent être utilisées pour transformer les données. Voici un résumé des plus intéressantes

Les fonctions

***ABS(** argument **) :** retourne la valeur absolue d'une valeur numérique.*
***CEIL(** argument **) :** retourne le plus petit entier _ argument.*
***COS(** argument **) :** retourne le cosinus d'un angle en radians.*
***EXP(** argument **) :** fonction exponentielle, élève e (_ 2.71828) à une puissance spécifique.*
***FLOOR(** argument **) :** retourne le plus grand entier _ argument.*
***INPUT(** argument, informat **) :** définit un format de lecture pour une valeur. Le format spécifié détermine si le résultat est numérique ou caractère.*
***LAGn(** argument **) :** décale la valeur de l'argument de n observations.*
***LENGTH(** argument **) :** retourne la longueur d'une chaîne de caractère.*
***LOG(** argument **) :** donne le logarithme népérien (en base e) de l'argument.*
***LOGn(** argument **) :** donne le logarithme en base n de l'argument.*
***MAX(** argument, argument,...**) :** donne la valeur la plus élevée parmi les valeurs non manquantes (au moins deux) des arguments.*
***MEAN(** argument, argument,...**) :** donne la valeur moyenne des arguments non manquants.*
***MIN(** argument, argument,...**) :** donne la plus petite valeur parmi les valeurs non manquantes des arguments.*
***MOD(** argument1, argument2 **) :** calcule le reste de argument1/argument2 (modulo).*
***NORMAL(** seed **) :** génère un nombre pseudo-aléatoire distribué selon une N(0,1). La valeur du "seed" peut être soit 0, soit un nombre entier impair de 5-7 chiffres. Par exemple.: x=m+s*normal(seed) est utilisé pour générer un nombre distribué normalement avec moyenne m et écart-type s.*
***RANGE(** argument, argument,...**) :** donne l'écart entre la plus grande et la plus petite valeur non manquante.*
***ROUND(** argument, unitéarrond **) :** arrondit l'argument à l'unité d'arrondissement près.*
***SIGN(** X **) :** retourne la valeur -1 si X<0, 0 si X=0 et 1 si X>0.*
***SIN(** argument **) :** donne le sinus d'un angle en radians.*
***SQRT(** argument **) :** donne la racine carrée de l'argument.*
***STD(** argument, argument,...**) :** calcule l'écart-type des valeurs non manquantes des arguments.*
***SUM**(argument, argument,...) : calcule la somme des valeurs des arguments.*
***TAN(** argument **) :** calcule la tangente d'un angle en radians.*
***VAR(** argument, argument,...**) :** calcule la variance des valeurs non manquantes.*

*L'**argument** d'une fonction peut être un nombre, une variable ou encore, une liste de nombres ou de variables selon le cas. Une fonction ou une formule peut également être utilisée comme argument.*

4.4 Autres options et commandes importantes

L'étape data est capable d'interpréter un langage de programmation évolué. On y retrouve les structures : ***if***, ***then***, ***else***, ***do****. La différence fondamentale est qu'une étape data peut être assimilée, en général, à une lecture de la table à traiter. Elle inclut implicitement une boucle considérant chacune des observations ; une ``variable'' du langage est une colonne ou variable statistique. La syntaxe habituelle est la suivante :*

Syntaxe - commandes

```
data  <work. >table_out ;
set   <work. >table_in ;
....   instructions ;
run;
```

Programme 4.4.1 :

```
/*  Création multiple de tables SAS  */
data France Etranger;
set biblio.voitures;
if marq='F' then output France ;
else output Etranger;
run ;
```

Dans l'exemple ci-dessus, deux tables SAS (France et Etranger) sont créées à partir de la table SAS biblio.voitures. Selon la valeur de la variable marq, l'observation courante est écrite soit dans la table France, soit dans la table Etranger.

Programme 4.4.2 :

```
/* illustration 1 Concaténation en lignes de tables SAS  */
data total1 ;
set France Etranger;
run;
/* illustration 2 Concaténation en colonnes de tables SAS selon une variable */
proc sort data=France;
by pfis ;
run ;
proc sort data=Etranger;
by pfis ;
run;
data total2 ;
set France Etranger;
by pfis ;
run;
```

Dans l'illustration 1 de l'exemple 4.4.2, la table SAS total1 contient les observations de la table France suivies de celles de la table Etranger. Les données sont empilées de manière à faire correspondre les variables des deux tables. A noter que si des variables sont présentes dans une table, mais pas dans l'autre, les valeurs de ces variables seront enregistrées comme valeurs manquantes pour cette dernière.

Le programme de l'illustration 2 effectue la même opération de concaténation en regroupant les observations par catégorie de la variable pfis spécifiée par la commande ***by****. Les deux tables ont au préalable été ordonnées par catégorie de la variable pfis à l'aide de la procédure* ***sort****.*

Chapitre 2 : Statistique descriptive

Les résultats des procédures SAS de la statistique descriptive permettent de résumer les données d'une table SAS sous forme d'indices ou de mesures statistiques accompagnés de représentations graphiques.

1 PROC MEANS

*La procédure **proc means** permet d'obtenir les moyennes ainsi que d'autres indices statistiques de variables numériques d'une table SAS.*

*Par défaut, c'est-à-dire sans options, cette procédure fournit les statistiques sommaires suivantes (mot-clés de l'option de la statistique) : la taille de l'échantillon (**n**), la moyenne arithmétique (**mean**), l'écart-type (**std**), les valeurs minimum (**min**) et maximum (**max**).*

*Avec options, il faut spécifier le mot-clé de l'option de la statistique désirée y compris ceux des statistiques sommaires précédentes. En plus, on peut obtenir : le nombre d'observations manquantes ou non-réponses (**nmiss**), l'étendue (**range**), la somme (**sum**), la somme des carrés (**uss**), la somme des carrés corrigés (**css**), la variance (**var**), l'erreur-type de la moyenne (**sdterr**), les coefficients de variation (**cv**), d'asymétrie (**skewness**) et d'aplatissement (**kurtosis**).*

Programme 1.1 :

```
/* ---- La table SAS voitures est permanente ---- */
proc means  n  mean cv   data=biblio.voitures ;
run ;
```

Cet exemple calcule les statistiques demandées en options (la taille de l'échantillon, la moyenne et le coefficient de variation) pour toutes les variables numériques de la table SAS biblio.voitures.

*La procédure **proc means** est très souvent utilisée avec les commandes :*

***var** : pour obtenir les statistiques de certaines variables de la table seulement.*

***by** : pour obtenir les statistiques par groupe d'observations de l'échantillon pour pouvoir, par exemple, les comparer ; la table doit être triée à l'aide de la procédure **proc sort**.*

***weight** : indique au système SAS le nom de la variable_poids contenant les pondérations des observations. Dans ce cas, la procédure calcule une moyenne et variance pondérées.*

La syntaxe générale de la procédure est de la forme suivante :

Syntaxe - options - commandes

***proc means** <options > ;*
***var** liste de variables ;*
***by** <descending> variable;*
***weight** variable;*
***output** <out=table SAS> < liste de statistiques>;*

*Dans la première illustration de l'exemple ci-dessous, la procédure **proc means** calcule les statistiques spécifiées pour les variables prix et consommation. Dans la deuxième illustration, la procédure calcul les mêmes statistiques sur les mêmes variables pour chaque groupe de la variable puissance fiscale. Enfin, dans la troisième illustration, en utilisant la commande **class**, on obtient exactement les mêmes résultats que ceux de la deuxième illustration, mais présentés différemment. La commande **class** a un effet similaire à celui de la commande **by** mais a l'avantage de ne pas nécessiter un tri de la table SAS par la procédure **proc sort.***

*La commande **output** permet de conserver les résultats de la procédure means dans une table SAS dont le nom est spécifié à l'aide de l'option **out=**. Les options permettent également de préciser les variables à inclure dans la table SAS de sortie. Le moyen le plus simple d'indiquer les statistiques à inclure et les noms des variables contenant ces statistiques est le suivant :*

***output out=** table SAS stat1=variable1 variable2 … stat2=variable1 variable2 … ;*

*où stat est le mot-clé associé à la statistique qu'on souhaite calculer et la liste de noms de variables après le signe d'égalité "=" nomme les statistiques calculées pour les variables correspondantes de la commande **var**. Cette liste peut être plus courte que la liste de la commande **var**.*

Programme 1.2 :

```
/*  illustration 1  */
proc means  mean std   data=biblio.voitures ;
var prix cons ; run ;
/* illustration 2 */
proc sort   data=biblio.voitures ;
by pfis ; run ;
proc means   mean std   data=biblio.voitures ;
var prix cons ; by pfis ; run ;
/* illustration 3 */
proc means  mean std   data=biblio.voitures ;
var prix cons ; class pfis ; run ;
```

Programme 1.3 :

```
/*  illustration de la commande output out= */
proc means clm alpha=0.01 data=biblio.voitures ;
var prix cons ; by marq ;
output out=biblio.resultat  mean=prixmoy consmoy  min=prixmin  max=consmax
run ;
proc print data=biblio.resultat; run;
```

*Dans cet exemple, comme la variable marq précisée dans la commande **by** prend deux valeurs différentes, la table SAS biblio.resultat contient deux observations, une pour chaque valeur de la variable marq (sans la commande **by**, la table SAS de sortie ne compterait qu'une seule observation). Chacune des deux observations de la table SAS comprend les variables marq (variable de la commande **by**), prixmoy, consmoy, prixstd, consstd, et prixmax. Les quatre dernières variables (celles de la commande **output**) représentent respectivement le prix moyen, la consommation moyenne, le prix minimal et la consommation maximale. Notons que, en plus d'écrire dans la table biblio.resultat, La procédure **means** affiche à l'écran les statistiques calculées par défaut. Si on veut éliminer l'affichage à l'écran, il faut ajouter l'option **noprint** au bout de la procédure **proc means**. L'option **CLM** permet d'établir des intervalles de confiance avec un risque d'erreur alpha choisi (alpha = 0.01) . Par défaut, alpha = 5%.*

Résultats du programme 1.3 :

Procédure MEANS

Variable	Libellé	Borne inférieure de l'IC à99% pour la moy.	Borne supérieure de l'IC à99% pour la moy.
PRIX	prix en francs belges	274.2486742	364.4994739
CONS	consommation & urbaine	6.5267448	7.7473293
CYLIN	cylindrée	1054.37	1276.89
VITE	vitesse & maximum	142.3056305	166.2128880
VOLU	volume maximum du coffre	737.0132767	1065.80
RPP	rapport & poids-puissance	15.6921242	21.6041721
LONG	longueur	3.5835315	3.6623945

The SAS System

marque du constructeur=E

The MEANS Procedure

Variable	Label	N	Mean	Std Dev	Minimum	Maximum
PRIX	prix en francs belges	17	307.1117647	73.7725040	219.3000000	500.1000000
CONS	consommation urbaine	17	7.2176471	1.0168796	6.1000000	9.2000000

marque du constructeur=F

Variable	Label	N	Mean	Std Dev	Minimum	Maximum
PRIX	prix en francs belges	10	340.2200000	100.6403807	259.6000000	506.3000000
CONS	consommation urbaine	10	7.0000000	1.3751768	5.6000000	9.3000000

Obs	MARQ	_TYPE_	_FREQ_	PRIXMOY	CONSMOY	PRIXMIN	CONSMAX	RUN
1	E	0	17	307.112	7.21765	219.3	500.1	9.2
2	F	0	10	340.220	7.00000	259.6	506.3	9.3

Remarques :

- *Il est à noter que les résultats de la procédure **means** sont inclus dans ceux fournis par la procédure **univariate**. La procédure **means** diffère par la présentation des résultats résumés sous la forme d'un tableau de synthèse plus facile à analyser.*
- *Il est à noter que la procédure **summary** est pratiquement similaire à la procédure **means**, elle ne diffère que dans le choix des options par défaut.*

2 PROC FREQ

*Vu que seules les variables continues sont utilisées par la procédure **proc means**, la procédure **proc freq** résume l'information de variables continues discrètes, avec un nombre fini de valeurs distinctes, et de variables nominales ou alphanumériques. D'un point de vue descriptif, elle permet d'obtenir des tableaux de répartition à une dimension (tris à plat) et des tables de contingences (tris croisés). Elle permet également d'effectuer des tests statistiques non-paramétriques que nous verrons un peu plus loin.*

*Des options peuvent être ajoutées à cette procédure. La plus courante consiste à spécifier l'ordre des valeurs de la variable qui apparaîtront dans la distribution de fréquences. Par défaut, c'est-à-dire sans options, les valeurs de la variable seront ordonnées de manière croissante. L'option **order = data** présente les valeurs selon leur ordre d'apparition dans l'ensemble des données alors que l'option **order = freq** présente les valeurs selon leurs fréquences de manière décroissante.*

*La commande **tables** est utilisée pour spécifier la ou les variables qualitatives à traiter. Si on ne spécifie pas cette commande, la procédure **proc freq** produit les tableaux de fréquences à une dimension de toutes les variables de la table SAS.*

*On peut également inclure une commande **by** suivie d'une variable qualitative si on veut établir une distribution de fréquences par catégorie de la variable qualitative spécifiée c'est-à-dire, procéder à l'analyse des données par groupe d'observations. La table SAS doit être préalablement triée par la procédure **proc sort**.*

La syntaxe générale de la procédure est de la forme suivante :

Syntaxe - options - commandes

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

Programme 2.1 :

```
/* Tris à plat  */
proc freq  data=biblio.voitures ;
tables marq pfis ;
run ;
/* tri croisé */
proc freq  data=biblio.voitures ;
tables marq*pfis ;
run ;
/* commande by */
proc sort  data=biblio.voitures ;
by marq ;       /* trie de la table - observations ordonnées selon la marque */
run ;
proc freq order=data   data=biblio.voitures ;
tables pfis ;
by marq ;
run ;
```

*Dans la première illustration, pour chaque variable, la procédure **proc freq** fournit, le tableau de fréquences : absolues (effectifs), relatives, absolues cumulées et relatives cumulées.*

*L'illustration 2 produit un tableau de fréquences à deux dimensions ''tableau de contingence'' représentant la distribution conjointe des modalités des deux variables spécifiées dans la commande **tables**. Ainsi, la procédure fournit : la répartition en effectifs, en pourcentages, en pourcentages lignes et colonnes. Le symbole '*****' est utilisé pour lier les deux variables; les modalités de la première variable seront positionnées en ligne et celles de la deuxième variable en colonne.*

*Pour un tableau à trois dimensions par exemple, avec la commande **tables** v1*v2*v3, on obtient pour chaque modalité de la variable v1 un tableau à deux dimensions des modalités des deux autres variables. D'autres croisements peuvent être spécifier sous une des formes suivantes : v1*(v2 v3), (v1 v2)*(v3 v4), (v1- -v6)*v7. Enfin, plusieurs tableaux peuvent être obtenus à partir d'une même commande **tables**:*

*Dans la troisième illustration, les observations de la table SAS biblio.voitures sont ordonnées, selon les modalités de la variable marque par la procédure **proc sort**, afin d'obtenir ensuite la distribution de fréquences de la variable puissance fiscale dans chaque groupe ou modalité de la variable marque.*

Il est fréquent que l'on veuille transformer les données d'une variable continue en variable qualitative avec un certain nombre de modalités ou catégories. Le programme suivant effectue ce type de ''recodage''.

Programme 2.2 :

```
/* Création d'une variable qualitative : classes de prix */
data biblio.voit2 ;
set biblio.voitures ;
if prix <= 270  then clprix = 3;
else if  270 < prix <= 370 then clprix=2 ;
else clprix = 1; run ;
proc freq ;
tables  (pfis marq)*clprix  clprix  / out=biblio.freqclas;
title 'Répartitions : classes de prix - puissance fiscale x classes de prix ⫫ marque x classes de prix' ; run ;
```

Résultats du programme 2.2 :

Répartitions : classes de prix - puissance fiscale x classes de prix ⫫ marque x classes de prix 1

The FREQ Procedure

Table of PFIS by CLPRIX

PFIS(puissance fiscale) CLPRIX

Frequency / Percent / Row Pct / Col Pct	1	2	3	Total
4CV	0	2	11	13
	0.00	7.41	40.74	48.15
	0.00	15.38	84.62	
	0.00	20.00	100.00	
5CV	0	5	0	5
	0.00	18.52	0.00	18.52
	0.00	100.00	0.00	
	0.00	50.00	0.00	
6CV	6	3	0	9
	22.22	11.11	0.00	33.33
	66.67	33.33	0.00	
	100.00	30.00	0.00	
Total	6	10	11	27
	22.22	37.04	40.74	100.00

Table of MARQ by CLPRIX

MARQ(marque & du & constructeur) CLPRIX

Frequency / Percent / Row Pct / Col Pct	1	2	3	Total
E	3	6	8	17
	11.11	22.22	29.63	62.96
	17.65	35.29	47.06	
	50.00	60.00	72.73	
F	3	4	3	10
	11.11	14.81	11.11	37.04
	30.00	40.00	30.00	
	50.00	40.00	27.27	
Total	6	10	11	27
	22.22	37.04	40.74	100.00

The FREQ Procedure

CLPRIX	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	6	22.22	6	22.22
2	10	37.04	16	59.26
3	11	40.74	27	100.00

Remarques :

- *Une commande **weight** peut être utilisée si on dispose d'une variable de pondération qui permet de modifier la contribution des observations au calcul des fréquences.*
- *Deux options de la commande **tables** peuvent être ajoutées, l'option **missing** concerne les valeurs manquantes, spécifie à la procédure **proc freq** de considérer ces valeurs comme non manquantes et de les inclure dans le calcul des fréquences. L'option **out=** nom_table SAS, permet de créer une table de sortie des résultats. Lorsque la commande **tables** est suivie d'une liste de tableaux à réaliser, seul le dernier tableau est conservé dans la table de sortie. Dans l'exemple 2, la table SAS de sortie biblio.freqclas ne contient que les valeurs de la répartition de la variable clprix créée.*

3 PROC CORR

*La procédure **proc corr** permet d'étudier les liaisons entre variables continues. Elle fournit un certain nombre de résultats de tests associés à des indices de liaisons comme le coefficient de corrélation linéaire de Bravais-Pearson, le coefficient de corrélation de rangs de Spearman, le tau de concordance de Kendall ainsi que d'autres moins usuels. Elle peut également calculer des coefficients de corrélation pondérés et des corrélations partielles. De plus, certaines statistiques sommaires des variables sont présentées.*

La syntaxe générale de la procédure est de la forme suivante :

Syntaxe - options - commandes

proc corr *<options > ;*
by *<descending> variable;*
var *liste de variables ;*
weight *variable;*
with *liste de variables ;*

Programme 3.1:

```
/* 1ère partie : Corrélation */
proc corr  data=biblio.voitures ;
var prix cons volu ; run ;
/* 2ème partie : corrélation entre certains couples de variables avec options*/
proc corr cov nosimple pearson spearman   data=biblio.voitures ;
var prix cons ; with rpp  vite ; run ;
```

Résultats du programme 3.1:

The CORR Procedure

3 Variables: PRIX CONS VOLU

Simple Statistics

Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
PRIX	27	319.37407	84.38379	8623	219.30000	506.30000
CONS	27	7.13704	1.14124	192.70000	5.60000	9.30000
VOLU	27	901.40741	307.41444	24338	202.00000	1200

Pearson Correlation Coefficients, N = 27

Prob > |r| under H0: Rho=0

	PRIX	CONS	VOLU
PRIX	1.00000	0.81387	0.21854
prix en francs belges		<.0001	0.2735
CONS	0.81387	1.00000	0.29462
consommation urbaine	<.0001		0.1358
VOLU	0.21854	0.29462	1.00000
volume maximum du coffre	0.2735	0.1358	

Avec la première partie du programme, on obtient d'abord les statistiques de base (n, mean, std, sum, min et max) pour chaque variable spécifiée dans la commande ***var****. Puis les coefficients de corrélation (linéaire de Pearson par défaut - sans option) sont obtenus.*

Le nombre en dessous de la corrélation est la probabilité ***Prob>|R|*** *qui correspond au test de nullité du coefficient. Si cette probabilité est inférieure à un risque d'erreur* α *(ou à* 2α *pour un test unilatéral), nous favorisons l'hypothèse alternative d'existence d'une dépendance linéaire autrement dit le coefficient de corrélation linéaire est significativement différent de zéro.*

Par exemple, la valeur du coefficient de corrélation linéaire entre le prix et le volume est égale à 0,21854 et nous obtenons ***Prob>|R|****= 0,2735 ce qui est supérieur à* α *= 5%. Nous favorisons l'hypothèse nulle à savoir que le coefficient de corrélation n'est pas significativement différent de zéro; on peut conclure, avec un risque d'erreur* α *= 5%, qu'il n'existe pas de dépendance linéaire entre le prix et le volume.*

Résultats du programme 3.1 (suite):

The CORR Procedure
2 With Variables: RPP VITE
2 Variables: PRIX CONS

Covariance Matrix, DF = 26

		PRIX	CONS
RPP	rapport poids-puissance	-358.180627	-4.305313
VITE	vitesse maximum	1724.437749	19.909259

Pearson Correlation Coefficients, N = 27
Prob > |r| under H0: Rho=0

	PRIX	CONS
RPP	-0.76789	-0.68247
rapport poids-puissance	<.0001	<.0001
VITE	0.91422	0.78044
vitesse maximum	<.0001	<.0001

Spearman Correlation Coefficients, N = 27
Prob > |r| under H0: Rho=0

	PRIX	CONS
RPP	-0.81484	-0.61623
rapport poids-puissance	<.0001	0.0006
VITE	0.86863	0.62855
vitesse & maximum	<.0001	0.0004

Dans la deuxième partie du programme, la commande ***with*** *suivie d'une liste de variables, permet d'obtenir les mêmes statistiques que précédemment mais uniquement pour les variables de cette liste avec celles de la liste de la commande* ***var****. De plus, on peut préciser certaines options dans la procédure* ***proc corr****. Parmi les options disponibles, on a (liste non-exhaustive) :*

- ***Spearman, Kendall, Hoeffding, Pearson*** *permet de calculer toutes les mesures de corrélation spécifiées,*
- ***cov*** *permet de calculer les coefficients de covariance,*
- ***Best=*** *k permet d'imprimer seulement les k meilleurs coefficients de corrélation (les k plus élevés en valeur absolue),*
- ***nosimple*** *qui permet d'éliminer l'impression des statistiques sommaires des variables.*

Remarques :

- *Sans la commande* ***var****, la procédure* ***proc corr*** *fournit les statistiques sommaires et les coefficients de corrélation de toutes les variables continues de la table SAS).*
- *La commande* ***partial*** *permet de calculer des corrélations partielles et Une commande* ***weight*** *des coefficients de corrélation pondérés*

---

4 PROC UNIVARIATE

*La procédure **proc univariate** permet d'obtenir un nombre important d'indices statistiques utiles dans le cadre d'une étude statistique ''univariée'' descriptive complète de variables continues. Elle fournit les principaux paramètres de tendance centrale (moyenne, médiane, mode, quantiles), de dispersion (variance, intervalle interquartiles, ect.) de forme (aplatissement, asymétrie) ainsi que des représentations graphiques (diagramme en feuilles, diagramme en boîte). Elle est plus complète que les procédures **proc means** ou **proc summary**. De plus, elle effectue de nombreux tests statistiques paramétriques (moyenne, différence de deux moyennes, normalité) et non-paramétriques (Wilcoxon, signes) qui seront traités au prochain chapitre.*

La syntaxe générale de la procédure est de la forme suivante :

Syntaxe - options - commandes

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

Programme 4 :

```
/* Statistique descriptive */
proc univariate  data=biblio.voitures ; run ;
/* Options */
proc sort  data=biblio.voitures ; /* tri de la table selon les groupes de la variable pfis */
by pfis ; run ;
proc univariate  plot freq data=biblio.voitures ;
var prix ; by pfis ; id nom ; run ;
```

*En utilisant la procédure **proc univariate** sans options et sans la commande **var**, première illustration du programme, on obtient pour chaque variable continue de la table SAS biblio.voitures, une fiche détaillée d'indices statistiques.*

Dans la deuxième illustration du programme, en plus des statistiques précédentes, on obtient avec les options :

***Plot** : Un diagramme en boîte (**Boxplot**) représentant les quartiles. Le deuxième quartile ''médiane'' est représentée par un trait à l'intérieur de la boîte tandis que la moyenne est représentée par le signe '**+**'. Un tel diagramme permet de visualiser, principalement, la forme d'une distribution (symétrique ou asymétrique). Un diagramme en feuilles (**Stem Leaf**) constitué d'un axe vertical dont les valeurs sont appelées tiges (Stem) qui représentent le premier (ou les deux premiers) chiffre de chaque donnée. A droite de chaque tige, à l'horizontale, se trouvent les feuilles (Leaf) lesquelles représentent le chiffre de la donnée suivant le ou les chiffres de la tige. Ainsi, par exemple, la donnée ?? se situera sur la tige ? avec ? comme feuille. Une courbe de probabilité normale cumulée (**Normal Probability Plot**) qui permet de vérifier graphiquement si les données peuvent s'apparenter à une distribution normale ou pas. En effet, si les points de cette courbe, représentés par le signe ''**+**'', s'écartent de ceux de la courbe cumulée, représentés par le symbole '*', sont importants, il y a de grandes chances que les données ne soient pas distribuées selon une loi normale.*

***freq** : permet la discrétisation d'une variable, on obtient ainsi, la liste des valeurs de la variable, les fréquences absolues et relatives de ces valeurs. La commande **id**, suivie de la variable nom, a été utilisée afin de faire figurer vis-à-vis de chaque valeur, le nom de la voiture. Ainsi, pour chaque groupe de la variable pfis, qui suit la commande **by**, on obtient les statistiques de la variable prix ainsi que les noms des 5 voitures les plus chères et les noms des 5 voitures les moins chères. A noter que les données de la table SAS biblio.voitures ont été préalablement ordonnées par la procédure **proc sort**.*

---------- puissance fiscale=4CV ----------

The UNIVARIATE Procedure

Variable: PRIX (prix en francs belges)

Moments

N	13	Sum Weights	13
Mean	257.369231	Sum Observations	3345.8
Std Deviation	19.0564243	Variance	363.147308
Skewness	-0.149728	Kurtosis	0.38466355
Uncorrected SS	865463.74	Corrected SS	4357.76769
Coeff Variation	7.40431335	Std Error Mean	5.28530115

Basic Statistical Measures

Location		Variability	
Mean	257.3692	Std Deviation	19.05642
Median	261.0000	Variance	363.14731
Mode	261.0000	Range	73.20000
		Interquartile Range	23.20000

Tests for Location: Mu0=0

Test	-Statistic-		-----p Value------	
Student's t	t	48.69528	Pr > |t|	<.0001
Sign	M	6.5	Pr >= |M|	0.0002
Signed Rank	S	45.5	Pr >= |S|	0.0002

Quantiles (Definition 5)

Quantile	Estimate
100% Max	292.5
99%	292.5
95%	292.5
90%	280.0
75% Q3	265.5
50% Median	261.0
25% Q1	242.3
10%	239.9
5%	219.3
1%	219.3
0% Min	219.3

Frequency Counts

		Percents				Percents				Percents	
Value	Count	Cell	Cum	Value	Count	Cell	Cum	Value	Count	Cell	Cum
219.3	1	7.7	7.7	248.0	1	7.7	38.5	265.5	1	7.7	76.9
239.9	1	7.7	15.4	259.6	1	7.7	46.2	269.5	1	7.7	84.6
242.0	1	7.7	23.1	261.0	2	15.4	61.5	280.0	1	7.7	92.3
242.3	1	7.7	30.8	265.2	1	7.7	69.2	292.5	1	7.7	100.0

---------- puissance fiscale=4CV ----------

The UNIVARIATE Procedure

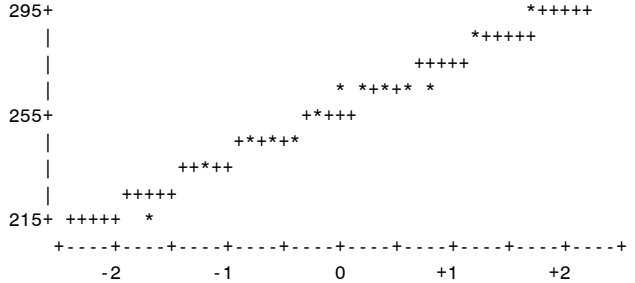
Variable: PRIX (prix en francs belges)

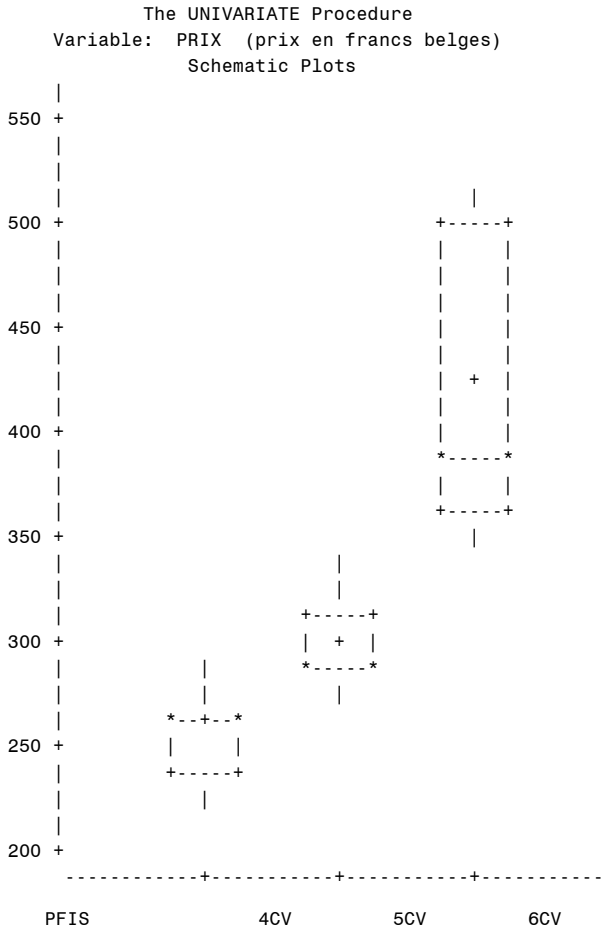
Stem	Leaf	#	Boxplot
29	2	1	|
28	0	1	|
27	0	1	|
26	01156	5	+-----+
25			| + |
24	0228	4	+-----+
23			|
22			|
21	9	1	|

----+----+----+----+

Multiply Stem.Leaf by 10**+1

Normal Probability Plot





5 PROC CHART

*La procédure **proc chart** permet de représenter la distribution d'une variable sous forme d'un diagramme en bâtons, d'un histogramme ou d'un diagramme circulaire.*

La syntaxe générale de la procédure est de la forme suivante :

Syntaxe - options - commandes

proc chart *<options > ;*
by *<descending> variable;*
hbar *liste de variables / <options> ;*
vbar *liste de variables / <options> ;*
pie *liste de variables / <options> ;*
star *liste de variables / <options> ;*
block *variable /* ***group=****variable <options> ;*

Programme 5.1 :

```
/* représentations graphiques - options */
proc chart  data=biblio.voitures ;
vbar prix; /type=mean sumvar=vite; /* histogramme */
vbar prix /group=marq subgroup=pfis; /* histogramme par groupe et sous groupe*/
pie pfis / type =mean sumvar=prix; /* diagramme circulaire */
block marq / group=pfis type =mean sumvar=prix; /* graphique en 3D '
run ;
```

*La commande **hbar** permet d'obtenir un diagramme en bâtons "à barres horizontales" ; par défaut, la procédure imprime la fréquence absolue, la fréquence absolue cumulée, la fréquence relative et la fréquence cumulée.*

*La commande **vbar** permet d'obtenir un histogramme. L'exemple ci-dessus représente la variable prix. On a utilisé, pour cet histogramme, l'option **type = mean sumvar =** vite, c'est-à-dire la vitesse moyenne comme hauteur des barres. La deuxième illustration présente l'histogramme de la variable prix selon les groupes de la variable marq et ceux de la variable pfis.*

*La commande **pie** permet d'obtenir un diagramme circulaire. L'exemple illustré représente la variable pfis. On a utilisé, pour ce diagramme, l'option prix moyen i.e. les pourcentages d'un secteur quelconque représentent donc le prix moyen de ce secteur divisé par la somme des prix moyens.*

*La commande **star** permet de réaliser un diagramme en étoile semblable à un diagramme à barres verticales (diagramme en bâtons) sauf que les barres débutent au centre du graphique (comme les rayons d'une roue) et que les points de chaque barre adjacente sont reliées entre elles. Ce diagramme en toile d'araignée est approprié pour des données cycliques.*

*La commande **block** permet d'obtenir un graphique à trois dimensions (stéréogramme). L'exemple donné par le programme représente le prix moyen par marq et par pfis.*

*Il est important de souligner que si les valeurs de la variable sont numériques, SAS considère la variable comme continue. Pour indiquer à SAS de considérer une variable numérique comme une variable discrète, il faut spécifier l'option **discrete**.*

*Par défaut, l'axe vertical représente la fréquence en nombre absolue pour chacune des variables. Pour obtenir les fréquences en pourcentage, on précise l'option **type = percent**.*

*L'option **missing** demande que les valeurs manquantes apparaissent aussi dans le diagramme.*

*L'option **symbol =** '+' permet de spécifier le caractère d'impression, ici le symbole '+'.*

*Pour l'analyse de données discrètes, on utilise la procédure **chart** pour construire généralement des diagrammes en bâtons et histogrammes, alors que pour l'analyse des données continues, il est préférable d'utiliser la procédure **plot** suivante.*

6 PROC PLOT

*La procédure **proc plot** permet de tracer des nuages de points en deux dimensions. Très utile dans le cas d'une analyse bi-variée (régression simple, analyse d'une fonction, représentations factorielles en analyse des données, etc.). La représentation graphique permet de visualiser la relation entre deux variables continues. La syntaxe générale de la procédure est la suivante :*

Syntaxe - options - commandes

```
proc plot  <options > ;
by <descending> variable;
plot  liste de graphiques  / <options> ;
```

Dans un premier temps, la procédure produit le nuage de points-voitures de la variable prix (axe des ordonnées) en fonction de la variable cons (axe des abscisses) à partir de l'ensemble des valeurs contenues dans la table SAS biblio.voitures.

*Plusieurs commandes **plot** peuvent être définies dans une même procédure **plot** et on peut demander l'affichage de plusieurs graphiques à partir d'un même commande **plot**. Il est également possible de spécifier le caractère à utiliser pour marquer les points sur le graphique.*

Programme 6.1 :

```
/* représentations graphiques */
proc plot  data=biblio.voitures ; plot prix*vite; run ;
/* représentations graphiques par groupe  */
proc sort data=biblio.voitures;  by pfis;  run;
proc plot  data=biblio.voitures ; by pfis ; plot prix*vite; run ;
```

Programme 6.2 :

```
/* regression linéaire */
proc reg data=biblio.voitures;
model prix = vite;  output out=biblio.graphe p=valajust ;  run;
proc plot data=graphe;  plot valajust*vite='*' prix*vite='+' / overlay; run ;
```

*Dans l'exemple ci-dessus, l'option **overlay** permet de représenter deux courbe sur le même graphique ; le nuage des valeurs observées "**+**" et le nuage des valeurs ajustées "*****" par le modèle de régression linéaire.*

Programme 6.3 :

```
/* tracer la courbe d'une fonction */
data expoX ;
do X=-8 to 8 by 0.05;
Y = exp(X) ;
output ;
end ;
run ;
proc plot ;
plot Y*X='+' / vzero hzero;
Title 'courbe y = exponentiel(x)';
label Y='axe vertical' X='axe horizontal';
run;
```

*Les options **vzero** et **hzero** spécifient respectivement que l'axe des ordonnées et l'axe des abscisses soient gradués à partir de zéro.*

Résultats du programme 6.2 :

The REG Procedure : Model: MODEL1
Dependent Variable: PRIX prix en francs belges

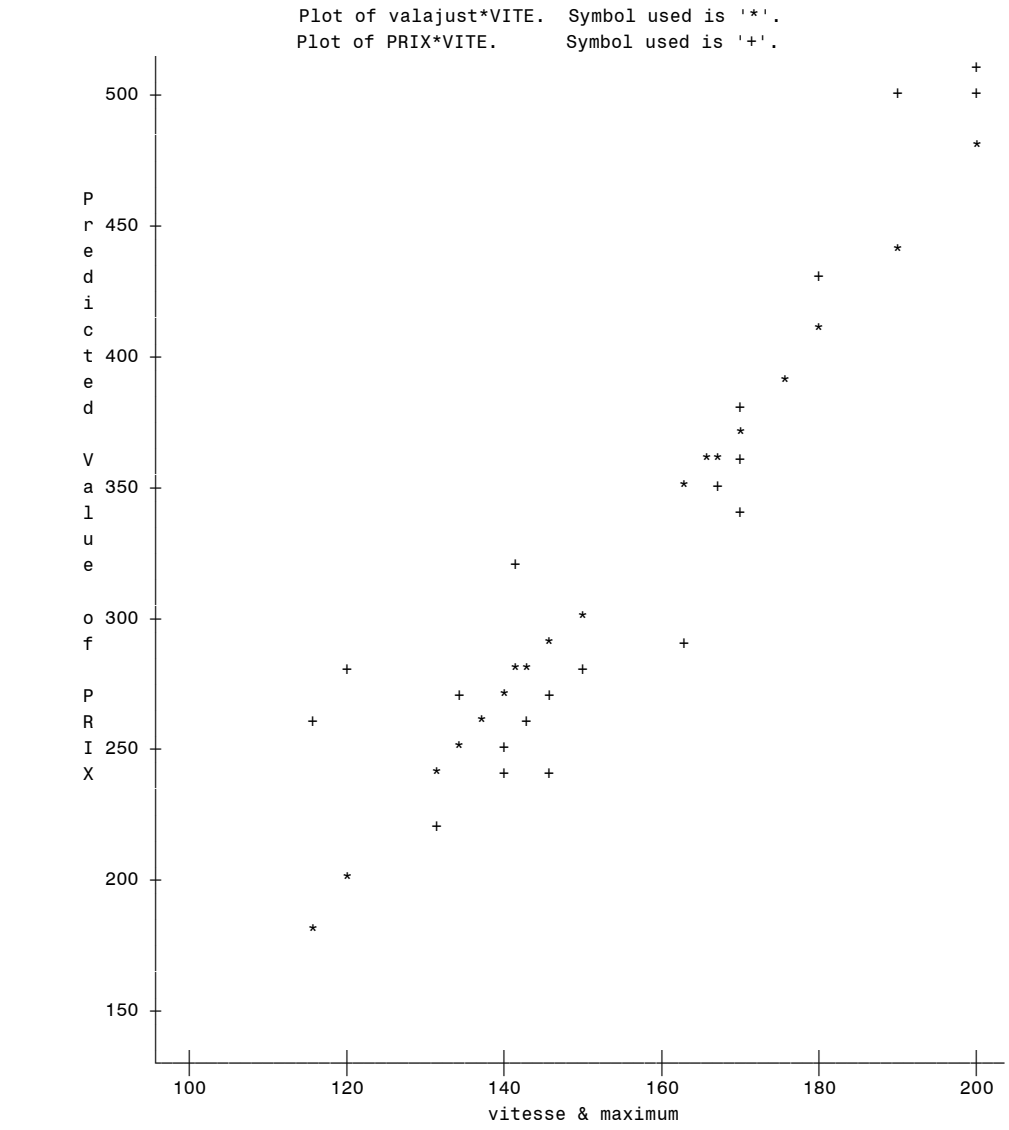
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	154737	154737	127.25	<.0001
Error	25	30400	1215.98568		
Corrected Total	26	185136			

Root MSE	34.87099	R-Square	0.8358
Dependent Mean	319.37407	Adj R-Sq	0.8292
Coeff Var	10.91854		

Parameter Estimates

Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	1	-213.00789	47.66922	-4.47	0.0001
VITE	vitesse & maximum	1	3.45122	0.30594	11.28	<.0001



NOTE: 16 obs hidden.

Il existe encore d'autres options :

***box**, qui permet de tracer une boîte tout autour du diagramme de dispersion,*
***hreverse**, qui permet d'inverser l'ordre des valeurs de l'axe horizontal,*
***vreverse**, qui permet d'inverser l'ordre des valeurs de l'axe vertical,*
***vaxis**, qui permet de spécifier les valeurs de la variable se situant sur l'axe vertical, ect.*

Chapitre 3 : Tests statistiques

Les principales procédures SAS de la théorie des tests paramétriques et non-paramétriques, utilisés pour vérifier certaines hypothèses ou affirmations, sont présentées et commentées par procédure SAS.

1 PROC FREQ

*La procédure **proc freq** présentée dans le chapitre 2, est également utilisée avec d'autres options pour effectuer trois tests statistiques. Le test non paramétrique d'indépendance, d'homogénéité et de comparaison de deux proportions.*

procédure – commandes - options

```
proc freq  ;
tables  variable_ligne * variable_colonne  / <options > ;
weight  variable ;
```

1.1 Test d'indépendance

Un tableau croisé (tableau de contingence) permet d'analyser la relation entre deux variables qualitatives ayant respectivement p et q modalités. On peut vérifier qu'il existe une relation ou non en effectuant un test d'indépendance. Les hypothèses statistiques sont les suivantes :

Exemple 1.1:

Existe-t-il un lien entre la puissance fiscale et la marque du constructeur au seuil de signification $\alpha = 5\%$?

Hypothèse statistiques :

H_0 : les deux variables sont indépendantes.

H_1 : les deux variables sont dépendantes.

Programme 1.1:

```
/* Test d'indépendance */
/* tri croisé */
proc freq  data=biblio.voitures ;
tables marq*pfis / chisq;
run;
```

*La procédure **proc freq** permet d'obtenir la table de contingence (tableau croisé : effectifs, pourcentages, pourcentages lignes et colonnes) selon les modalités des variables spécifiées dans la commande **tables**. L'option **chisq** doit absolument être utilisée pour obtenir les résultats du test d'hypothèse.*

Résultats du programme 1.1 :

The FREQ Procedure
Table of MARQ by PFIS
MARQ(marque du constructeur)
PFIS(puissance fiscale)

Frequency / Percent / Row Pct / Col Pct	4CV	5CV	6CV	Total
E	9	2	6	17
	33.33	7.41	22.22	62.96
	52.94	11.76	35.29	
	69.23	40.00	66.67	
F	4	3	3	10
	14.81	11.11	11.11	37.04
	40.00	30.00	30.00	
	30.77	60.00	33.33	
Total	13	5	9	27
	48.15	18.52	33.33	100.00

Statistics for Table of MARQ by PFIS

Statistic	DF	Value	Prob
Chi-Square	2	1.4025	0.4960
Likelihood Ratio Chi-Square	2	1.3586	0.5070
Mantel-Haenszel Chi-Square	1	0.0447	0.8325
Phi Coefficient		0.2279	
Contingency Coefficient		0.2222	
Cramer's V		0.2279	

WARNING: 67% of the cells have expected counts less than 5. Chi-Square may not be a valid test.

Sample Size = 27

---

*La règle de décision est la suivante : l'hypothèse alternative H_1 est favorisée si la probabilité **Prob** qui se situe sur la ligne **Chi-square** est inférieure au risque α choisi. Le programme 1.1 effectue le test d'indépendance. Pour notre exemple, les résultats s'analysent ainsi : la probabilité **Prob** est égale à **0.4960 (> α = 0.05)**, nous favorisons l'hypothèse nulle H_0 et on peut conclure qu'il n'y a pas de lien entre la puissance fiscale et la marque du constructeur des voitures.*

Quelques remarques :

*La statistique **Likelihood Ratio Chi-square** permet également de vérifier s'il y a indépendance ou non entre les deux variables. Le résultat s'analyse comme pour la statistique du **Chi-Square**. Pour cet exemple, les probabilités sont légèrement différentes vu le nombre d'effectifs théoriques inférieurs à 5.*

*La statistique **Mantel-Haenszel Chi-square** permet de tester la présence ou non d'une relation linéaire entre les deux variables qualitatives. Cette statistique est appropriée seulement lorsque les deux variables sont ordonnées (échelle ordinale). Si la probabilité **Prob** est inférieure au risque α choisi, la relation est significative et linéaire.*

*Lorsque le tableau est de dimension 2x2 et que la taille de l'échantillon est faible (affectant le respect des conditions d'applications), il est préférable d'effectuer le test de Fisher, **Fisher's exact test (2-tail)**. si la probabilité **Prob** est inférieure à α, la relation est significative.*

*Les résultats contiennent également certaines mesures d'association qui tentent de mesurer l'intensité de la relation (**Phi coefficient**, **Contingency Coefficient** et **Cramer's V**).*

1.2 Test d'homogénéité

Ce test non paramétrique permet de comparer k échantillons indépendants prélevés au hasard de k populations selon un caractère à r modalités. On vérifie ainsi si les populations sont homogènes (semblables) ou hétérogènes (différentes) sur la base des r modalités du caractère.

A ne pas confondre avec le test d'indépendance précédent où l'analyse s'effectue sur deux caractères..

Exemple 1.1: ———————————————————————————————

Le service marketing d'une entreprise a effectué un pré-test pour étudier la réaction en ce qui a trait à une nouvelle version de son produit. Des ménagères ont été choisies au hasard dans chacun des 3 départements bas-normands. On a remis à chaque personne les trois produits (ayant une couleur différente mais de version non identifiée). A l'issue d'une période d'essai d'un mois, les préférences se présentent comme suit :

Versions du produit	Calvados	Orne	Manche
Nouvelle	105	90	80
Actuelle	60	50	45
Ancienne	35	60	55

Peut-on conclure, au seuil de signification α = 5%, que les consommateurs de chaque département apprécient de manière identique les diverses versions du produit ?

Hypothèse statistiques :

H_0 : la préférence du produit suivant ses trois versions se répartit de façon identique dans les trois départements. (comportement homogène des départements en ce qui concerne la préférence des versions du produit).

H_1 : la préférence ne se répartit pas de façon identique dans les trois départements.

Programme 1.2:

```
/* Test d'homogénéité */
proc format ;
value $a 'NOUV' = 'Nouvelle'    'ACTU' = 'Actuelle' 'ANCI' = 'Ancienne';
value $b 'C' = 'Calvados'  'O' = 'Orne'    'M' = 'Manche';
data homog1;
options pagesize = 60 linesize = 80;
options nodate;
title 'Préférences du produit - Départements Bas-Normands';
input dept$ version$ nb @@;
label dept = 'Département'
      version = 'Versions du produit'
      nb = 'Effectif';
format version a.;
format dept b.;
cards;   /* Saisie des données */
C NOUV 105 C ACTU 60 C ANCI 35 O NOUV 90 O ACTU 50 O ANCI 60 M NOUV 80 M ACTU 45 M ANCI 55
run;
proc freq ; tables version * dept / chisq;  weight nb;  run;
```

---

*La commande **weight** indique la variable, ici nb, qui contient les effectifs de chaque couple de modalités des variables croisées : en ligne la version et colonne le département.*

Résultats du programme 1.2 :

Préférences du produit - Départements Bas-Normands 1

The FREQ Procedure

Table of version by dept

version(Versions du produit) dept(Département)

Frequency Percent Row Pct Col Pct	Calvados	Manche	Orne	Total
Actuelle	60	45	50	155
	10.34	7.76	8.62	26.72
	38.71	29.03	32.26	
	30.00	25.00	25.00	
Ancienne	35	55	60	150
	6.03	9.48	10.34	25.86
	23.33	36.67	40.00	
	17.50	30.56	30.00	
Nouvelle	105	80	90	275
	18.10	13.79	15.52	47.41
	38.18	29.09	32.73	
	52.50	44.44	45.00	
Total	200	180	200	580
	34.48	31.03	34.48	100.00

Statistics for Table of version by dept

Statistic	DF	Value	Prob
Chi-Square	4	11.1623	0.0248
Likelihood Ratio Chi-Square	4	11.6651	0.0200
Mantel-Haenszel Chi-Square	1	0.0893	0.7651
Phi Coefficient		0.1387	
Contingency Coefficient		0.1374	
Cramer's V		0.0981	

Sample Size = 580

---

*A noter que l'ordre de la saisie des données par la commande **cards** doit respecter celui des variables spécifiées dans la commande **input**. Il s'agit ici de la saisie de données d'une table et non d'un tableau individus/variables.*

*L'option **chisq** permet de fournir plusieurs statistiques. Celle qui nous intéresse pour effectuer ce test est la statistique **Chi-Square**. Si pour cette statistique, la valeur **Prob** > α (risque d'erreur choisi), alors nous favorisons l'hypothèse nulle H_0 d'homogénéité des populations sinon nous favorisons l'hypothèse alternative H_1 d'hétérogénéité.*

*Ainsi, pour l'exemple ci-dessus, **Prob** = 0.0248 < 5% nous favorisons donc l'hypothèse alternative H_1. On en conclut que les trois départements n'ont pas un comportement homogène en ce qui concerne la préférence du produit.*

*Enfin, les options d'édition du tableau croisé **nocol** , **norow** et **nopercent** permettent respectivement d'éliminer l'impression des pourcentages colonnes, des pourcentages lignes et des pourcentages.*

1.3 Test sur 2 proportions

Ce test paramétrique permet de comparer deux proportions. Les conditions qui sont requises ici sont que les deux échantillons soient indépendants et que leurs tailles respectives soient suffisamment grandes (théorème central limite – approximation d'une loi binomiale par une loi normale). La procédure utilisée pour résoudre ce test de comparaison de deux proportions est la même que celle utilisée pour un test d'homogénéité dans le cas particulier où k = r = 2.

Exemple 1.3:

Sur 125 personnes interrogées en basse-Normandie, 44 disent aller voter aux prochaines élections municipales. Sur 100 personnes interrogées d'autres régions, 32 seulement comptent aller voter.

Peut-on affirmer, au seuil de signification α = 5%, que les intentions de vote sont les mêmes en Basse-Normandie que dans les autres régions de France ?

Hypothèses statistiques :

H_0 : $p_{BN} = p_{AR}$ (Les intentions de vote sont les mêmes)

H_1 : $p_{BN} \neq p_{AR}$ (Les intentions de vote sont les différentes).

Programme 1.3:

```
/* Test sur 2 proportions */
proc format ;
value $a 'O' = 'Oui'   'N' = 'Non';
value $b 'BN' = 'Basse_Normandie'  'AR' = 'Autres régions' ;
data proport;
options pagesize = 60 linesize = 80;
options nodate;
title 'Intentions de vote';
input vote$ region$ nb @@;
label region = 'Localisation de la région'
      vote = 'Intention de vote'
      nb = 'Effectif';
format vote a.;
format region b.;
cards;
O BN 44 N BN 81 O AR 32 N AR 68
run;
proc freq ;  tables vote * region / chisq norow nopercent;  weight nb;  run;
```

*Conclusion du test à partir de la statistique **Chi-Square.** On a **Prob** = 0.614 > α = 5% nous favorisons donc l'hypothèse nulle H_0. Il n'y a pas de différence significative entre les intentions de vote en Basse-Normandie que dans les autres régions de France. Il semble que les intentions de vote sont identiques.*

A noter que les autres statistiques, variantes du Chi-Square, aboutissent à la même conclusion.

Résultats du programme 1.3:

The FREQ Procedure

Table of vote by region

vote(Intention de vote) region(Localisation de la région)

Frequency Col Pct	Autres régions	Basse_Normandie	Total
Non	68 68.00	81 64.80	149
Oui	32 32.00	44 35.20	76
Total	100	125	225

Statistics for Table of vote by region

Statistic	DF	Value	Prob
Chi-Square	1	0.2543	0.6140
Likelihood Ratio Chi-Square	1	0.2548	0.6137
Continuity Adj. Chi-Square	1	0.1314	0.7170
Mantel-Haenszel Chi-Square	1	0.2532	0.6148
Phi Coefficient		0.0336	
Contingency Coefficient		0.0336	
Cramer's V		0.0336	

2 PROC TTEST

*La procédure **proc ttest** est utilisée pour effectuer deux tests statistiques paramétriques. Le test de comparaison de deux moyennes et le test de comparaison de deux variances.*

procédure – commandes - options

```
proc ttest ;
class variable_1 ;
var variable_2 ;
```

*La variable_1 (nominale ou de groupe) de la commande **class** identifie les deux populations et la variable_2 continue de la commande **var** est la variable de comparaison.*

*Cette procédure fournit également les **intervalles de confiance** (moyenne, différence de moyennes et l'écart-type) par groupe de la variable_1 de la variable_2 considérée.*

2.1 Test sur 2 moyennes

Ce test permet de comparer les moyennes de deux populations indépendantes. Ce test nécessite la condition de normalité des données lorsque au moins un des deux échantillons est de petite taille (< 30).

Hypothèses statistiques :

H_0 : $m_1 = m_2$ (les moyennes sont identiques)

H_1 : $m_1 \neq m_2$ (les moyennes sont différentes)

$m_1 > m_2$ ou $m_1 < m_2$ (test unilatéral).

Programme 2.1:

```
/* Test de comparaison de 2 moyennes et de 2 variances - prix selon la marque du constructeur */
data comp2moy;
set biblio.voitures;
proc ttest;  class marq;  var prix ;  run;
```

*La procédure **proc ttest** fournit pour chaque échantillon les statistiques descriptives : la taille (**N**), la moyenne (**Mean**), l'écart-type (**Std Dev**), l'erreur-type de la moyenne (**Std Error**).*

*On y retrouve l'intervalle de confiance de niveau 95% de la moyenne (**LOWER CL** et **UPPER CL** pour **mean**) et l'intervalle de confiance de niveau 95% de l'ecart-type (**LOWER CL** et **UPPER CL** pour **Std Dev**), ainsi que le minimum et le maximum.*

*La procédure **ttest** fournit les résultats d'un test d'égalité des moyennes selon deux hypothèses : les variances sont égales (**Equal**) ou inégales (**Unequal**), on a l'écart-réduit (**T**), les degrés de liberté (**DF**) et la probabilité **Pr > |t|**.*

*Pour conclure le test, il suffit de comparer **Pr > |t|**, selon que l'on suppose l'égalité des variances (**Equal**) ou pas (**Unequal**), au risque d'erreur α (test bilatéral) ou 2α (test unilatéral).*

*Si la valeur **Pr > |t|** > α (ou 2α), alors nous favorisons l'hypothèse nulle H_0 d'égalité des moyennes sinon nous favorisons l'hypothèse alternative H_1 de la différence ou de l'inégalité des moyennes.*

*Ainsi, pour l'exemple ci-dessus, en supposant l'égalité des variances, on accepte l'hypothèse nulle H_0 car la probabilité **Pr > |t| =** 0.3346 > 5%. Donc, les moyennes ne diffèrent pas de façon significative.*

*On aboutit à la même conclusion, la probabilité **Pr > |t| =** 0.3791 > 5%, si on suppose que les variances sont inégales .*

*A noter que lorsque les variances sont inégales, pour l'approximation de la probabilité **Pr > |t|**, SAS utilise l'approximation de Satterthwaite pour déterminer les degrés de liberté associés à l'écart-réduit. On peut effectuer l'approximation de Cochran et Cox en ajoutant l'option cochran dans la procédure proc ttest.*

Résultats du programme 2.1:

The TTEST Procedure

Statistics

Variable	MARQ	N	Lower CL Mean	Mean	Upper CL Mean
PRIX	E	17	269.18	307.11	345.04
PRIX	F	10	268.23	340.22	412.21
PRIX	Diff (1-2)		-102.4	-33.11	36.195

Statistics

Variable	MARQ	Lower CL Std Dev	Std Dev	Upper CL Std Dev	Std Err	Minimum	Maximum
PRIX	E	54.944	73.773	112.28	17.892	219.3	500.1
PRIX	F	69.224	100.64	183.73	31.825	259.6	506.3
PRIX	Diff (1-2)	66.219	84.436	116.56	33.65		

T-Tests

Variable	Method	Variances	DF	t Value	Pr > |t|
PRIX	Pooled	Equal	25	-0.98	0.3346
PRIX	Satterthwaite	Unequal	14.8	-0.91	0.3791

Equality of Variances

Variable	Method	Num DF	Den DF	F Value	Pr > F
PRIX	Folded F	9	16	1.86	0.2668

2.2 Test sur 2 variances

Ce test permet de comparer les variances de deux populations indépendantes. La condition de normalité des deux populations est nécessaire.

Hypothèses statistiques :

H_0 : $\sigma_1^2 = \sigma_2^2$ (les variances sont égales)
H_1 : $\sigma_1^2 \neq \sigma_2^2$ (les variances sont différentes).

*La procédure **proc ttest** utilisée pour la comparaison de deux moyennes effectue automatiquement le test de comparaison des variances.*

*Les résultats sont fournis dans la dernière ligne des résultats du programme. Pour conclure le test, il suffit également de comparer **Pr > F** au risque d'erreur α (test bilatéral) ou 2α (test unilatéral). Si la valeur de la probabilité **Pr > F** > α (ou 2α), alors nous favorisons l'hypothèse nulle H_0 d'égalité des variances sinon nous favorisons l'hypothèse alternative H_1 de différence ou d'inégalité des variances.*

*Ainsi, pour l'exemple 2.1 précédent, la probabilité **Pr > F =** 0.2668 > α = 5%, on accepte donc l'hypothèse nulle H_0. On peut donc conclure que les variances sont égales. Ce qui nous permet d'utiliser, pour le test de comparaison de deux moyennes, la probabilité **Pr > |t|** de la ligne **Equal.***

3 PROC MEANS

*Avec la procédure **proc means,** on peut effectuer deux tests statistiques paramétriques. Le test sur une moyenne ainsi que le test de comparaison de deux moyennes d'échantillons dépendants (données appariées).*

———— *procédure – commandes - options* ————
proc means <options> ;
var variable ;
———————————————————————

*La variable de la commande **var** est la variable de comparaison.*

3.1 Test sur une moyenne

Ce test permet de vérifier si la moyenne d'une population est égale à une valeur moyenne quelconque spécifiée. Ce test nécessite la condition de normalité des données lorsque l'échantillon prélevé est de petite taille (< 30).

Hypothèses statistiques : test unilatéral

H_0 : $m = m_0$ ou $m \leq m_0$ (la moyenne n'est pas supérieure à la valeur spécifiée m_0)
H_1 : $m > m_0$ (la moyenne est supérieure à la valeur spécifiée m_0).

Programme 3.1 : ————————————————————————————————

```
/* Test de comparaison d'une moyenne à une valeur spécifiée - Hypothèses statistiques */
/* H0 : la vitesse moyenne est ( ≤ ) = 150 km/h  contre  H1 : la vitesse moyenne > 150 km/h */
data t1moy;
set biblio.voitures;
vite0 = vite - 150;    /*transformation des données */
proc means n mean t prt;  var vite0;  run;
```

———

*La procédure **proc means** utilisée avec les options **n**, **mean**, **t** et **prt** fournit la taille de l'échantillon (**n**), la moyenne (**mean**), la valeur de statistique de test de Student (**t**) et la probabilité **Pr > |t|.***

*A noter qu'une transformation des données doit être effectuée avant d'exécuter la procédure **proc means**. En effet, cette procédure ne test que l'hypothèse : H_0 : $m = 0$ (moyenne nulle). Pour cela, il suffit de soustraire aux données de l'échantillon la valeur spécifiée m_0. Dans notre exemple, $m_0 = 150$ ce qui revient à créer une nouvelle variable centrée en m_0, ici vite0, qui sera spécifiée dans la commande **var**.*

*Pour conclure le test, il suffit de comparer la probabilité **Pr > |t|** au risque d'erreur α (test bilatéral) ou 2α (test unilatéral). Si la valeur **Pr > |t|** > α (ou 2α), alors nous favorisons l'hypothèse nulle H_0 sinon nous favorisons l'hypothèse alternative H_1.*

Résultats du programme 3.1: ————————————————————————

The MEANS Procedure
Analysis Variable : vite0

N	Mean	t Value	Pr > |t|
27	4.2592593	0.99	0.3313

———

*Ainsi, pour l'exemple ci-dessus, en supposant que la vitesse est normalement distribuée (petite taille d'échantillon : n = 27 voitures), on accepte l'hypothèse nulle H_0 car la probabilité **Pr > |t| =** 0.3313 > 2α = 10% (test unilatéral). On peut donc conclure que la vitesse moyenne des voitures n'est pas significativement supérieure à 150 km/h.*

3.2 Test sur la différence de 2 moyennes (Données appariées – Echantillons dépendants))

Ce test permet de comparer les moyennes de 2 échantillons dépendants (méthode des couples, échantillons appariés). Ce test de Student sur 2 échantillons appariés nécessite la condition de normalité des différences lorsque la taille de l'échantillon est inférieure à 30 observations.

Exemple 3.2: ———————————————————————————————————

On expérimente en laboratoire les effets d'un nouveau traitement de l'excès de cholestérol dans le sang. Sur un groupe de 10 lapins nourris avec un régime enrichi en cholestérol, on a observé les taux de cholestérol (en dg/l) suivants :

lapin n°	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Avant traitement	*21*	*24*	*33*	*36*	*23*	*15*	*26*	*31*	*35*	*28*
Après traitement	*18*	*22*	*33*	*34*	*19*	*12*	*27*	*32*	*31*	*30*

Peut-on affirmer que le traitement à contribué à faire baisser, de façon significative (risque d'erreur $\alpha = 5\%$), le taux de cholestérol ?

Hypothèses statistiques : test unilatéral

H_0 : $m_d = 0$ ou $m_d \geq 0$ (le traitement n'est pas efficace)

H_1 : $m_d < 0$ (le traitement est efficace – baisse du taux de cholestérol).

———

Programme 3.2: ———————————————————————————————————

```
/* Test différence de 2 moyennes - Echantillons appariés */
data tmoyappa;
options nodate;
title 'Efficacité du traitement';
input avant apres @@;
differen = apres - avant ;
cards;
21 18 24 22 33 33 36 34 23 19 15 12 26 27 31 32 35 31 28 30
run;
proc means n mean t prt; var differen;  run;
```

———

*A noter qu'avant d'exécuter la procédure **proc means** avec les options **n**, **mean**, **t** et **prt,** il faut au préalable créer, sous la commande **input**, la variable des différences, ici differen, qui sera spécifiée dans la commande **var**. Le test s'effectue sur les différences des observations.*

*Comme précédemment, la règle de décision s'effectue à partir de la comparaison de la valeur de la probabilité **Pr > |t|** au risque d'erreur α (test bilatéral) ou 2α (test unilatéral). Si la valeur **Pr > |t|**.> α (ou 2α), alors nous favorisons l'hypothèse nulle H_0 sinon nous favorisons l'hypothèse alternative H_1.*

Résultats du programme 3.2: ———————————————————————————————

```
                    Efficacité du traitement                          1
                  The MEANS Procedure
                Analysis Variable : differen

 N            Mean     t Value     Pr > |t|
 ------------------------------------------
10      -1.4000000       -1.99        0.0774
 ------------------------------------------
```

———

*Pour l'exemple 3.2, en supposant que la variable créée des différences est normalement distribuée (faible taille d'échantillon n = 10 < 30 observations), la valeur de la probabilité **Pr > |t|** = 0.0774 est inférieure à 2α = 10% (test unilatéral); on doit donc favoriser l'hypothèse alternative H_1. On peut donc conclure que le traitement est efficace c'est-à-dire qu'il contribue à faire baisser significativement le taux de cholestérol.*

4 PROC UNIVARIATE

*La procédure **proc univariate** permet d'effectuer deux tests paramétriques : le test sur une moyenne (paragraphe 3.1) et le test sur la différence de deux moyennes d'échantillons dépendants (paragraphe 3.2). Pour ces deux tests, la façon de procéder est sensiblement la même que celle utilisée avec la procédure **proc means**. Les hypothèses statistiques et la règle de décision restent inchangées. L'avantage d'utiliser la procédure **proc univariate** est qu'un test de normalité est automatiquement effectué pour vérifier les conditions d'application.*

Cette procédure permet aussi d'effectuer trois tests non-paramétriques : le test de normalité, le test de Wilcoxon et le test des signes.

———— *procédure – commandes - options* ————
proc univariate *<options> ;*
var *variable ;*
———————————————————————————

4.1 Test de normalité

*Les conditions d'application des tests paramétriques sur des échantillons de petites tailles, la procédure **proc univariate** suivie de l'option **normal** permet de vérifier si la condition de normalité est respectée. Cette procédure vérifie si la variable spécifiée dans la commande **var** est normalement distribuée.*

Hypothèses statistiques :

H_0 : la distribution observée est normalement distribuée
H_1 : la distribution observée n'est pas normalement distribuée.

Programme 4.1.1: ————————————————————————————————————

```
/* Test de normalité : Exemple 3.1 */
data tnormal1;
set biblio.voitures;
vite0 = vite - 150 ;
proc univariate normal ; var vite0;  run;
```

———

*La procédure **proc univariate** utilisée avec les options **normal** fournit par défaut des indices statistiques de la variable dont on analyse la normalité, ici vite.*

*Pour conclure ce test, les résultats s'analysent ainsi : si la valeur de la probabilité **Pr < W** est supérieure au risque α choisi alors nous favorisons l'hypothèse nulle H_0 de normalité sinon nous favorisons l'hypothèse alternative H_1 et on conclut que la distribution observée n'est pas normalement distribuée. Ce test est celui de Shapiro-Wilk, il est utilisé par le système SAS lorsque la taille de l'échantillon est relativement petite (inférieure ou égale à 2000 observations). Si la taille est encore plus grande, SAS utilise le test de Kolmogorov-Smirnov et indique la probabilité **Pr > D** (la règle de décision demeure la même).*

Résultats du programme 4.1.1: ————————————————————————————

The UNIVARIATE Procedure
Variable: vite0

Moments

N	27	Sum Weights	27
Mean	4.25925926	Sum Observations	115
Std Deviation	22.3530975	Variance	499.660969
Skewness	0.50319457	Kurtosis	-0.3224465
Uncorrected SS	13481	Corrected SS	12991.1852
Coeff Variation	524.811855	Std Error Mean	4.30185562

Basic Statistical Measures

Location		Variability	
Mean	4.25926	Std Deviation	22.35310
Median	-5.00000	Variance	499.66097
Mode	-5.00000	Range	85.00000
		Interquartile Range	30.00000

```
                Tests for Location: Mu0=0
     Test           -Statistic-    -----p Value------
     Student's t    t  0.990098    Pr > |t|    0.3313
     Sign           M        -2    Pr >= |M|   0.5572
     Signed Rank    S      34.5    Pr >= |S|   0.3910

                  Tests for Normality
Test                  --Statistic---    -----p Value------
Shapiro-Wilk          W     0.940214    Pr < W      0.1233
Kolmogorov-Smirnov    D     0.216202    Pr > D     <0.0100
Cramer-von Mises      W-Sq  0.145172    Pr > W-Sq   0.0252
Anderson-Darling      A-Sq  0.739586    Pr > A-Sq   0.0478
```

Ainsi, pour l'exemple 4.1.1 ci-dessus, les deux tests de normalité (Shapiro-Wilk et Kolmogorov-Smirnov) conduisent à accepter l'hypothèse de normalité. En effet, la probabilité ***Pr < W =*** *0.1233 >* α *= 5% (ou encore* ***Pr > D*** *< 0,01* $\Rightarrow$ ***Pr < D*** *> 0,99). On accepte donc l'hypothèse nulle* H_0*, on peut donc conclure que la variable vite0 est normalement distribuée.*

On vérifie ainsi la condition d'application nécessaire du test de l'exemple du programme 3.1 et on retrouve également la statistique de décision du test ***Pr > |t| =*** *0.3313.*

Programme 4.1.2: ———————————————————————————————————

```
/* Test de normalité : Exemples 3.2 Echantillons appariés */
data tmormal2;
title 'Efficacité du traitement';
input avant apres @@;
differen = apres - avant ;
cards;
21 18 24 22 33 33 36 34 23 19 15 12 26 27 31 32 35 31 28 30
run;
proc univariate normal; var differen ; run;
```

Résultats du programme 4.1.2: ———————————————————————————————

```
                          Efficacité du traitement                          1
                          The UNIVARIATE Procedure
                            Variable:  differen
                                 Moments
N                          10    Sum Weights                10
Mean                     -1.4    Sum Observations          -14
Std Deviation      2.22111083    Variance            4.93333333
Skewness           0.32854269    Kurtosis            -1.5719242
Uncorrected SS             64    Corrected SS              44.4
Coeff Variation    -158.65077    Std Error Mean      0.70237692

                    Basic Statistical Measures
          Location                     Variability
     Mean     -1.40000     Std Deviation            2.22111
     Median   -2.00000     Variance                 4.93333
     Mode     -4.00000     Range                    6.00000
                           Interquartile Range      4.00000
NOTE: The mode displayed is the smallest of 4 modes with a count of 2.

                Tests for Location: Mu0=0
     Test           -Statistic-    -----p Value------
     Student's t    t  -1.99323    Pr > |t|    0.0774
     Sign           M      -1.5    Pr >= |M|   0.5078
     Signed Rank    S     -15.5    Pr >= |S|   0.0703

                  Tests for Normality
Test                  --Statistic---    -----p Value------
Shapiro-Wilk          W     0.897585    Pr < W      0.2061
Kolmogorov-Smirnov    D     0.206472    Pr > D     >0.1500
Cramer-von Mises      W-Sq  0.074892    Pr > W-Sq   0.2227
Anderson-Darling      A-Sq  0.447529    Pr > A-Sq   0.2267
```

De même pour l'exemple 4.1.2, on accepte l'hypothèse nulle H_0 *de normalité des différences car la probabilité* ***Pr < W =*** *0.2209 >* α *= 5%. Condition nécessaire du test de comparaison de deux échantillons appariés de l'exemple du programme 3.2.*

4.2 Test de Wilcoxon

Ce test non-paramétrique permet de savoir si la médiane d'une population est nulle. Il est également utilisé pour vérifier si les médianes de 2 échantillons appariés 'dépendants' sont comparables. Dans ce dernier cas, il suffit de vérifier si la médiane des différences est nulle.

Le test de Wilcoxon sur 2 échantillons appariés ne nécessite aucune condition si ce n'est qu'un échantillon aléatoire de n paires d'observations indépendantes.

Les individus peuvent être identiques dans les deux échantillons ou appariés en fonction de variables externes.

Exemple 4.2:

Efficacité du traitement contre le cholestérol (exemple 3.2).

lapin n°	1	2	3	4	5	6	7	8	9	10
Avant traitement	21	24	33	36	23	15	26	31	35	28
Après traitement	18	22	33	34	19	12	27	32	31	30

Hypothèses statistiques : Test bilatéral - Unilatéral

H_0 : la médiane des différences est nulle : les deux médianes sont égales (le traitement n'est pas efficace)

H_1 : la médiane des différences n'est pas nulle (le traitement est efficace).
la médiane ''avant'' est supérieure à la médiane ''après''.

Programme 4.2:

```
/* Test de Wilcoxon: Echantillons appariés */
data twilcox;
options nodate;
title 'Efficacité du traitement';
input avant apres @@;
differen = apres - avant ;
cards;
21 18 24 22 33 33 36 34 23 19 15 12 26 27 31 32 35 31 28 30
run;
proc univariate normal; var differen; run;
```

Pour conclure ce test, il suffit de comparer la probabilité ***Pr > |S|*** *au risque d'erreur α pour un test bilatéral (2α pour un test unilatéral). On favorise alors l'hypothèse nulle H_0 si* ***Pr > |S|*** *> α (ou 2α) sinon l'hypothèse alternative H_1 doit être retenue.*

Pour l'exemple 4.2 (mêmes résultats que l'exemple 4.1.2), la probabilité ***Pr >= |S|*** *= 0,0703 < 2α = 10%. On accepte l'hypothèse alternative H_1. On peut conclure qu'il y a une différence significative entre les taux de cholestérol avant et après le traitement. La médiane des différences est significativement inférieure à zéro, le traitement semble efficace.*

Résultats du programme 4.2:

```
                Tests for Location: Mu0=0
Test           -Statistic-    -----p Value------
Student's t    t  -1.99323    Pr > |t|    0.0774
Sign           M      -1.5    Pr >= |M|   0.5078
Signed Rank    S     -15.5    Pr >= |S|   0.0703

                Tests for Normality
Test                 --Statistic---    -----p Value------
Shapiro-Wilk         W     0.897585    Pr < W      0.2061
Kolmogorov-Smirnov   D     0.206472    Pr > D     >0.1500
Cramer-von Mises     W-Sq  0.074892    Pr > W-Sq   0.2227
Anderson-Darling     A-Sq  0.447529    Pr > A-Sq   0.2267
```

A noter qu'on a ajouté l'option ***normal*** *à la procédure* ***proc univariate*** *afin d'effectuer un test de normalité sur la variable différence, ici differen. Vu que la variable différence est normalement distribuée (la probabilité* ***Pr < W*** *= 0,2061 > 5%); on peut appliquer le test paramétrique sur la différence de deux moyennes avec la procédure* ***proc means*** *(paragraphe 3.2 - échantillons appariés).*

5 PROC CORR

*La procédure **proc corr** permet d'effectuer deux tests statistiques : le test (paramétrique) sur le coefficient de corrélation linéaire et le test (non-paramétrique) sur le coefficient de corrélation de rang de Spearman. Les commandes et options requises pour effectuer ces tests :*

```
———————— procédure – commandes - options ————————
      proc corr  < options > ;
      var  variable_1  variable_2 …;
_____________________________________________________
```

*L'option de la procédure **proc corr** spécifie le test à effectuer sur les variables variable_1 et variable_2 de la commande **var**.*

5.1 Test sur le coefficient de corrélation linéaire

Ce test permet de savoir si le coefficient de corrélation linéaire de Bravais-Pearson est significativement différent de zéro. En d'autres termes, il permet de vérifier s'il existe une dépendance linéaire significative entre deux variables (quantitatives).

Hypothèses statistiques :

H_0 : $\rho = 0$ Absence de dépendance linéaire (le coefficient de corrélation linéaire est nul)
H_1 : $\rho \neq 0$ Présence d'une dépendance linéaire (le coefficient de corrélation linéaire est différent de zéro).
$\rho > 0$ Présence d'une dépendance linéaire positive (même sens)
$\rho < 0$ Présence d'une dépendance linéaire négative (sens inverse)

On suppose que l'échantillon aléatoire est prélevé d'une population normale à deux dimensions dans le coefficient de corrélation linéaire est nul sous l'hypothèse H_0.

Programme 5.1: ————————————————————————————————

```
/* Test sur le coefficient de corrélation linéaire */
data coefcor;
set biblio.voitures;
title ' Dépendance linéaire entre le prix et le volume ';
proc corr nosimple ;
var prix volu ;
run;
```

_____

*La procédure **proc corr** avec l'option **nosimple,** présente les statistiques sommaires des variables spécifiées dans la commande **var** ainsi que la matrice de corrélations. La procédure effectue ensuite le test d'analyse du coefficient de corrélation linéaire, entre les variables spécifiées prix et volu .*

Résultats du programme 5.1: ————————————————————————————————

```
Dépendance linéaire entre le prix et le volume                    1
                The CORR Procedure
        2  Variables:    PRIX     VOLU
  Pearson Correlation Coefficients, N = 27
          Prob > |r| under HO: Rho=0

                                   PRIX          VOLU

PRIX                            1.00000       0.21854
prix en francs belges                          0.2735

VOLU                            0.21854       1.00000

volume maximum du coffre         0.2735
```

_____

*Pour conclure à l'existence ou non d'une dépendance linéaire entre les deux variables, il suffit de comparer la probabilité **Prob > |r|** au risque d'erreur α pour un test bilatéral (2α pour un test unilatéral). On favorise alors l'hypothèse nulle H_0 d'absence de dépendance linéaire si **Prob > |r|** > α (ou 2α) sinon on doit favoriser l'hypothèse alternative H_1 d'existence d'une dépendance linéaire.*

*Pour l'exemple 5.1, le coefficient de corrélation linéaire observée entre les deux variables est de 0,21854. La probabilité **Prob > |r| =** 0.2735 est supérieure à $\alpha = 5\%$. On peut conclure que le coefficient de corrélation linéaire n'est pas significativement différent de zéro ; il n'existe pas de dépendance linéaire entre le prix et le volume des voitures.*

5.2 Test sur le coefficient de corrélation de rangs de Spearman

Le coefficient de corrélation de rangs de Spearman permet de mesurer le degré d'association entre 2 variables ordinales. Il varie entre -1 et +1. Si les 2 variables sont quantitatives, elles sont alors transformées en rangs dans les calculs. En effet, ce coefficient est calculé à partir des rangs des observations et non pas à partir des valeurs des observations comme c'est le cas du coefficient de corrélation linéaire. A noter qu'il est égal au coefficient de corrélation linéaire dans le cas où il n'y pas d'ex-æquo.

Ce test, relatif au cas de 2 échantillons appariés indépendants, permet de mesurer le degré d'une éventuelle dépendance ''monotone'' entre les deux variables. Il teste alors si le coefficient de corrélation de rangs est significativement différent de zéro.

La seule condition d'application requise pour exécuter ce test est que l'on ait prélevé un échantillon aléatoire d'une population dont les deux variables concernées sont indépendantes.

Hypothèses statistiques :

H_0 : $\rho_S = 0$ *Absence de dépendance (le coefficient de corrélation de rangs est nul)*

H_1 : $\rho_S \neq 0$ *Existence d'une dépendance (le coefficient de corrélation de rangs est différent de zéro).*

Programme 5.2:

```
/* Test sur le coefficient de corrélation de rang */
data coefcor;
set biblio.voitures;
title ' Dépendance entre le prix et le volume ';
proc corr spearman ; var prix volu ; run;
```

Avec l'option ***spearman*** *la procédure* ***proc corr*** *effectue le test non-paramétrique du coefficient de corrélation de rangs entre les variables prix et volu spécifiées dans la commande* ***var.****.*

La règle de décision s'effectue de la même manière que celle du test précédent. Si la probabilité ***Prob > |r|*** *est supérieure au risque d'erreur α pour un test bilatéral (ou 2α pour un test unilatéral) alors on favorise l'hypothèse nulle H_0 d'absence de dépendance dans le cas contraire,* ***Prob > |r|*** *< α (2α), on doit favoriser l'hypothèse alternative H_1 d'existence d'une dépendance entre les deux variables.*

Résultats du programme 5.2:

Spearman Correlation Coefficients, N = 27
Prob > |r| under H0: Rho=0

	PRIX	VOLU
PRIX	1.00000	0.24176
prix en francs belges		0.2244
VOLU	0.24176	1.00000
volume maximum du coffre	0.2244	

Pour notre exemple 5.2, mêmes données que l'exemple 5.1, on obtient un coefficient de corrélation de rangs de Spearman de 0,24176. La probabilité ***Prob > |r|*** *= 0.2244 supérieure à α = 5%, on peut donc conclure que le coefficient de corrélation de rangs est significativement nul c'est-à-dire qu'il n'existe pas de dépendance entre le prix et le volume des voitures.*

6 PROC ANOVA : Analyse de la variance

La procédure ***proc anova*** *est utilisée pour effectuer une analyse de la variance à un facteur contrôlé (test de comparaison des moyennes de plusieurs populations). Cette analyse nécessite les conditions que les données de chaque population doivent suivre une loi normale et doivent avoir la même variance.*

——— procédure – commandes - options ———

```
proc anova ;
class variable_1
model  variable_2 = variable_1 ;
```

*La variable_1 (variable indépendante) de la commande **class** identifie les différentes populations et la variable_2 (variable dépendante) de la commande **model** permet l'identification des variables.*

Hypothèses statistiques :

H_0 : $m_1 = m_2 = m_3$

H_1 : *au moins 2 moyennes sont différentes.*

Programme 6.1:

```
/* Analyse de la variance */
data anova;
set biblio.voitures;
title 'Effet de la puissance fiscale sur la consommation';
proc anova;
class pfis ; /* puissance fiscale ( 3 modalités : populations 4CV, 5CV et 6CV ) */
model cons = pfis;  run;
```

*La procédure **proc anova** fournit la somme des carrés (**Sum of squares**) : la somme des carrés expliquée (**Model**), la somme des carrés résiduelle (**Error**) et la somme des carrés totale (**Corrected Total**). Ainsi que les degrés de liberté (**DF**), la valeur de la statistique de Fisher (**F Value**) et la probabilité **Pr > F.***

Résultats du programme 6.1:

Effet de la puissance fiscale sur la consommation
The ANOVA Procedure
Class Level Information

Class	Levels	Values
PFIS	3	4CV 5CV 6CV

Number of observations 27

Effet de la puissance fiscale sur la consommation
The ANOVA Procedure
Dependent Variable: CONS consommation & urbaine

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	27.40874074	13.70437037	50.96	<.0001
Error	24	6.45422222	0.26892593		
Corrected Total	26	33.86296296			

R-Square	Coeff Var	Root MSE	CONS Mean
0.809402	7.266050	0.518581	7.137037

Source	DF	Anova SS	Mean Square	F Value	Pr > F
PFIS	2	27.40874074	13.70437037	50.96	<.0001

*Pour conclure, il suffit de comparer la probabilité **Pr > F** au risque d'erreur α. Si la valeur **Pr > F**. est supérieure à α, alors l'hypothèse nulle H_0 est favorisée sinon nous favorisons l'hypothèse alternative H_1.*

*Pour l'exemple présenté, en supposant que les conditions d'application de la méthode sont respectées, la valeur de la probabilité **Pr > F** = 0.0001 est inférieure à α = 5% ; on doit donc favoriser l'hypothèse alternative H_1. On peut conclure que les différences de prix observées semblent significatives. Il y a au moins 2 prix moyens qui diffèrent, les prix moyens ne sont pas les mêmes selon que la puissance fiscale est de 4cv, 5cv ou 6cv.*

Lorsque l'hypothèse alternative est acceptée, c'est-à-dire qu'il y a au moins une différence, on peut poursuivre l'analyse pour localiser les différences entre les moyennes des populations et savoir ainsi celles qui diffèrent.

Programme 6.2:

```
/* Analyse de la variance */
data anova;
set biblio.voitures;
title 'Analyse des moyennes qui diffèrent significativement';
proc anova;
class pfis ; /* puissance fiscale ( 3 modalités : populations 4CV, 5CV et 6CV ) */
model cons = pfis;  means pfis / tukey lines ; run;
```

*Pour cela, il suffit d'ajouter la commande **means** variable_1 **/** <**options**> qui indique les moyennes de chaque groupe c'est-à-dire la variable indépendante, avec l'option **tukey** (Tukey's studentized range test) pour le test de comparaison des moyennes et l'option **lines** pour une présentation des résultats par ordre descendant de la moyenne et une indication des ''paires'' non significatives.*

Résultats du programme 6.2: ——————————————————————————————

```
                    Analyse des moyennes qui diffèrent significativement
                                The ANOVA Procedure
                        Tukey's Studentized Range (HSD) Test for CONS
NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher
                                Type II error rate than REGWQ

                                Alpha                                   0.05
                                Error Degrees of Freedom                  24
                                Error Mean Square                   0.268926
                                Critical Value of Studentized Range  3.53170
                                Minimum Significant Difference        0.6587
                                Harmonic Mean of Cell Sizes         7.731278
                                      NOTE: Cell sizes are not equal.

                          Means with the same letter are not significantly different.
                           Tukey Grouping          Mean      N    PFIS
                                        A        8.5556      9    6CV
                                        B        6.5000     13    4CV
                                        B
                                        B        6.2400      5    5CV
```

———

Ainsi, on peut affirmer que les moyennes des populations 4CV et 5CV, signalées par la lettre B, sont significativement différentes de celle de la population des 6CV signalée par la lettre A. Les moyennes des populations 4CV et 5CV ne sont pas significativement différentes l'une de l'autre.

*L'exemple d'application 6.3, montre d'autres <**options**> possibles de la commande **means** qui permettent notamment de tester l'égalité des variances dans le cas de plus de 2 groupes.*

*L'homoscédasticité (égalité des variances d'une variable dans plusieurs échantillons) peut être vérifiée par les tests de **Levene** (le meilleur car peu sensible à la non-normalité), de **Bartlett** (le meilleur si la distribution est normale) ou de **Fisher** (le moins robuste en l'absence de normalité). Ils testent l'hypothèse nulle de l'égalité des variances :*

Homoscédasticité $H_0 : \sigma_1^2 = \sigma_2^2 = \ldots = \sigma_k^2$ *(les variances de la variable dans les k populations sont égales)*

Hétéroscédasticité $H_1 : \sigma_i^2 \neq \sigma_j^2$ *pour au moins une paire (i,j).*

Programme 6.3: ——————————————————————————————

```
/* Analyse de la variance */
data anova;
set biblio.voitures;
title 'Effet de la puissance fiscale sur la consommation';
proc sort; by pfis ; run;
title 'Test de normalité des trois groupes de consommation';
proc univariate normal; var cons; by pfis; run;
title 'Test d''égalité des variances : Bartlett's Test for Homogeneity of Variance';
proc anova; class pfis ; model cons = pfis; run;
means pfis / hovtest= bartlett ; /* hovtest= levene */ run;
```

———

Le test de Bartlett (ou de Levene) demandé dans l'exemple d'application 6.3 montre qu'il n'y a pas de différence significative entre les variances dans les 3 groupes ; la probabilité calculée (Bartlett : 70.31%, Levene : 42.22%) étant supérieure à 5% on accepte l'hypothèse nulle d'égalité des variances.

Résultats du programme 6.3:

```
                    Test de normalité des trois groupes de consommation                    11

---------------------------------- puissance fiscale=4CV ----------------------------------
                              The UNIVARIATE Procedure
                        Variable:  CONS  (consommation & urbaine)

                                 Tests for Normality
Test                        --Statistic---    -----p Value------
Shapiro-Wilk                W     0.952569    Pr < W       0.6377

                                 Tests for Normality
Test                        --Statistic---    -----p Value------
Shapiro-Wilk                W     0.900096    Pr < W       0.4104

                                 Tests for Normality
Test                        --Statistic---    -----p Value------
Shapiro-Wilk                W     0.879916    Pr < W       0.1567

                                    ---------------

                                The ANOVA Procedure
                              Test d'égalité des variances
                      Bartlett's Test for Homogeneity of CONS Variance

                          Source        DF       Khi 2    Pr > Khi 2
                          PFIS           2      0.7046        0.7031

                                    ---------------
                                The ANOVA Procedure
                          Levene's Test for Homogeneity of CONS Variance
                           ANOVA of Squared Deviations from Group Means

                        Somme
                        des       Carré     Valeur
                        Source      DF      carrés       moyen          F     Pr > F
                        PFIS         2      0.1111      0.0556       0.89     0.4222
                        Error       24      1.4916      0.0621

                        Niveau de           -------------CONS------------
                        PFIS          Nb          Moyenne       Écart-type
                        4CV           13       6.50000000       0.47958315
                        5CV            5       6.24000000       0.43931765

                        6CV            9       8.55555556       0.60438215
```

7 PROC NPAR1WAY

*La procédure **proc npar1way** est utilisée pour vérifier l'égalité des médianes de deux (test U de mann-Whitney) ou plusieurs populations (test de Kruskal-Wallis). Les commandes et options requises pour effectuer ces deux tests statistiques non-paramétriques sont :*

——— procédure – commandes - options ———

```
proc npar1way  <wilcoxon> ;
class variable_1
var  variable_2  ;
```

*L'option **wilcoxon** doit être spécifiée pour effectuer ces deux tests. La variable_1 de la commande **class** identifie les différentes populations et la variable_2 de la commande **var** indique la variable de comparaison des populations.*

7.1 Test U de Mann-Whitney

Dans le cas de deux populations, ce test permet de vérifier si les distributions des populations sont semblables. Si on suppose que la différence ne se situe qu'au niveau de la tendance centrale, cela revient à vérifier l'égalité des médianes.

Hypothèses statistiques :

H_0 : *les médianes sont égales*

H_1 : *les médianes sont différentes .*

Programme 7.1:

```
/* Test de Mann-whitney - prix selon la marque du constructeur : 2 populations */
data manwhit1;
set biblio.voitures;
proc npar1way wilcoxon;  class marq;   var prix ;  run;
```

Pour conclure ce test, il suffit de comparer la probabilité ***Pr > |Z|*** *au risque d'erreur* α *pour un test bilatéral (* 2α *pour un test unilatéral). Si la probabilité* ***Pr > |Z|*** *est supérieure à* α *(ou à* 2α *), on rejette l'hypothèse alternative* H_1 *et on favorise l'hypothèse nulle* H_0 *d'égalité des médianes.*

Résultats du programme 7.1:

The NPAR1WAY Procedure

Wilcoxon Scores (Rank Sums) for Variable PRIX

Classified by Variable MARQ

MARQ	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
E	17	218.0	238.0	19.913452	12.823529
F	10	160.0	140.0	19.913452	16.000000

Average scores were used for ties.

Wilcoxon Two-Sample Test

Statistic	160.0000
Normal Approximation	
Z	0.9792
One-Sided Pr > Z	0.1637
Two-Sided Pr > |Z|	0.3275
t Approximation	
One-Sided Pr > Z	0.1682
Two-Sided Pr > |Z|	0.3365

Z includes a continuity correction of 0.5.

Kruskal-Wallis Test

Chi-Square	1.0087
DF	1
Pr > Chi-Square	0.3152

La procédure ***proc npar1way*** *fournit pour chaque population, la somme des rangs (****Sum of scores****) : la somme des rangs sous l'hypothèse nulle* H_0 *(****Expected Under H0****), l'écart-type sous* H_0 *(****Std Dev Under H0****), le score moyen (****Mean score****) et la valeur de probabilité de la statistique de test sous* H_0 *(****Pr > |Z|****).*

Dans l'exemple 7.1, la valeur de la probabilité ***Pr > |Z|*** *= 0.3275 est supérieure à* α *= 5%; on doit donc favoriser l'hypothèse nulle* H_0 *On peut conclure que les prix sont identiques et qu'il ne semble pas y avoir de différence significative entre les prix des deux marques (les deux médianes semblent égales). De même, la probabilité* ***Pr > CHI-Square*** *= 0.3152 est supérieure à* α *= 5% ; on doit donc favoriser l'hypothèse nulle* H_0

7.2 Test de Kruskal-Wallis

Ce test est une généralisation du test U de Mann-Whitney dans le cas de plus de deux populations. Il permet de comparer les médianes de k populations. Les instructions requises sont les mêmes que celles du test précédant mais la règle décision est différente.

Hypothèses statistiques :

H_0 : *les médianes sont les mêmes (pas de différence)*

H_1 : *les médianes sont différentes (il y a une différence).*

Programme 7.2:

```
/* Test de Kruskal-Wallis */
data compar;
set biblio.voitures;
title 'Comparaison des médianes - Distributions de 3 populations';
proc npar1way wilcoxon ; class pfis ; var prix ; run;
```

*La règle de décision de ce test est telle que si la probabilité **Pr > CHI-Square** est supérieure au risque d'erreur α, alors on accepte l'hypothèse nulle H_0 d'égalité des médianes sinon on favorise l'hypothèse alternative H_1.*

*Dans l'exemple 7.2, où l'on compare les médianes des distributions du prix selon les trois puissances fiscales, la valeur de la probabilité **Pr > CHI-Square** = 0.0001 est inférieure à α = 5% ; on doit donc favoriser l'hypothèse alternative H_1. On peut donc conclure qu'il y a une différence significative du prix des voitures selon leur puissance fiscale (les médianes sont différentes).*

Résultats du programme 7.2 :

Comparaison des médianes - Distributions de 3 populations 1

The NPAR1WAY Procedure

Wilcoxon Scores (Rank Sums) for Variable PRIX

Classified by Variable PFIS

PFIS	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
4CV	13	94.0	182.0	20.604297	7.230769
5CV	5	77.0	70.0	16.018374	15.400000
6CV	9	207.0	126.0	19.439254	23.000000

Average scores were used for ties.

Kruskal-Wallis Test	
Chi-Square	21.1889
DF	2
Pr > Chi-Square	<.0001

Quelques remarques

Dans ce chapitre, différents tests statistiques ont été utilisés avec les mêmes exemples. Comment déterminer le test approprié ?

Entre le test U de Mann-Whitney (PROC NPAR1WAY wilcoxon) et le test sur 2 moyennes (échantillons indépendants, PROC TTEST), ce dernier est à privilégier à condition qu'il soit valide c'est-à-dire que dans le cas de petits échantillons ($n_1 < 30$ et/ou $n_2 < 30$) il faut que les deux distributions soient normalement distribuées. Si cette condition n'est pas respectée, le test U de Mann-Whitney sera utilisé.

Le test d'analyse de la variance (PROC ANOVA) est préférable au test de Kruskal-Wallis (PROC NPAR1WAY wilcoxon). Cependant, pour utiliser le test d'analyse de la variance, les k populations doivent être normalement distribuées avec des variances identiques. Si cette condition n'est pas respectée, le test de Kruskal-Wallis sera utilisé.

Entre le test sur la différence de 2 moyennes (échantillons appariés, PROC MEANS) et le test de Wilcoxon (PROC UNIVARIATE), on devrait, si possible, effectuer le test de la différence de 2 moyennes à condition que les différences soient normalement distribuées lorsque la taille de l'échantillon est inférieure à 30. Si cette condition n'est pas respectée, on pourra utiliser le test de Wilcoxon.

Enfin, le test sur le coefficient de corrélation linéaire (PROC CORR nosimple) exige que les deux populations soient normalement distribuées et que la dépendance soit linéaire. Si ces conditions ne sont pas respectées, le test sur le coefficient de corrélation de rangs de Spearman (PROC CORR spearman) sera utilisé.

Chapitre 4 : Régression linéaire

La régression linéaire est une méthode d'analyse permettant de décrire les relations linéaires entre une variable dépendante ou à expliquer Y et une ou plusieurs autres variables X_1, X_2, ..X_p dites indépendantes ou explicatives.

La théorie des modèles de régressions sont présentées et commentées par procédure SAS. L'accent est mis sur l'interprétation des résultats et surtout les moyens de vérifier les conditions d'application de base de tels modèles

1 PROC REG

Cette méthode de prévision est dite 'simple' lorsqu'une seule variable indépendante intervient dans le modèle linéaire supposé de la forme $Y = \beta_0 + \beta_1 X + \varepsilon$, elle est dite 'multiple' lorsque le modèle linéaire, supposé de la forme $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ..+ \beta_p X_p + \varepsilon$, contient plusieurs variables indépendantes.

*La procédure **proc reg** permet d'effectuer des analyses de régression par la méthode des moindres carrés ordinaires (MCO). Certaines commandes et options de cette procédure sont très utiles notamment pour effectuer et comparer différentes méthodes de sélection de variables indépendantes du modèle ou encore pour vérifier les hypothèses de base que doit respecter un modèle de régression linéaire à savoir,*

- *$E(\varepsilon) = 0$,*
- *Homoscédasticité (variance constante),*
- *Normalité des erreurs,*
- *Absence d'autocorrélation,*
- *Absence de colinéarité.*

1.1 Régression linéaire simple

Dans le cas d'une régression linéaire simple (une seule variable explicative), en supposant que le modèle linéaire de la forme $Y = \beta_0 + \beta_1 X + \varepsilon$ peut être approprié, la syntaxe de la procédure se présente comme suit :

```
———————— procédure – commandes - options ————————
      proc reg  ;
      model  variable_Y = variable_X  /  <options >;
_____________________________________________________
```

*La variable_X (variable indépendante) et la variable_Y (variable dépendante) de la commande **model** permettent l'identification des variables du modèle.*

Programme 1.1.1: ——————————————————————————————

```
/* Régression linéaire simple - options */
proc reg data=biblio.voitures;
title 'Régression linéaire simple';
model prix = vite / clm;      /* options : intervalles de confiance (clm ) de prévision (cli) */
output out=resultat  r=residus  p=prixest  stdp=etyest  stdi=etiest ;  /* sauvegarde des résultats */
run;
proc plot data=resultat;        /* Représentations graphiques */
title 'Vérification des hypothèses de base';
options ps=30 ls=65;
plot prixest*vite='*' prix*vite='+' / overlay  box;
plot residus*vite='*' / box;             /* E(ε) = 0 */
proc univariate normal plot;                /* Normalité des résidus : test */
var residus;
QQPLOT residus / normal(MU=EST SIGMA=EST COLOR=RED L=1);  /* Normalité des résidus : graphique */
run;
```

———

*Les résultats de la procédure **proc reg** sont présentés en deux parties :*

1) l'analyse de la variance (Analysis of variance)

*Pour chacune des trois sources de variations : expliquée (**Model**) attribuable au modèle de régression, non-expliquée ou résiduelle (**Error**) et totale à expliquer (**C Total**), sont présentés les degrés de libertés (**DF**), les sommes de carrés (**Sum of Squares**). Ainsi que le carré moyen (ou variance) (**Mean Square**) expliqué et résiduel, leur rapport (**F value**) et le test de Fisher (**Prob>F**).*

*De plus, d'autres résultats sont présentés comme l'écart-type des résidus (**Root MSE**) qui correspond à la racine carrée de la variance résiduelle (carré moyen résiduel), la moyenne de la variable dépendante (**Dep mean**), le coefficient de variation (en %) de la variable dépendante (**C.V.** = Root MSE$_{x100}$ / Dep mean), le coefficient de détermination (**R-square** = R^2 carré du coefficient de corrélation linéaire) et le coefficient de détermination ajusté (**Adj R-sq** = $1 - (n-1)(1-R^2)/(n-p-1)$ où p est le nombre de variables explicatives et n le nombre d'observations).*

*La probabilité **Pr > F** permet de savoir si la régression est significative ou pas cela revient à tester les hypothèses statistiques suivantes :*

$H_0 : \beta_1 = 0$ *la régression n'est pas significative*
$H_1 : \beta_1 \neq 0$ *la régression est significative.*

Résultats du programme 1.1.1:

Régression linéaire simple *1*
The REG Procedure
Model: MODEL1
Dependent Variable: PRIX prix en francs belges

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	154737	154737	127.25	<.0001
Error	25	30400	1215.98568		
Corrected Total	26	185136			

Root MSE	34.87099	R-Square	0.8358
Dependent Mean	319.37407	Adj R-Sq	0.8292
Coeff Var	10.91854		

Parameter Estimates

Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	1	-213.00789	47.66922	-4.47	0.0001
VITE	vitesse & maximum	1	3.45122	0.30594	11.28	<.0001

*La règle de décision de ce test est telle que si la probabilité **Pr > F** est inférieure risque α, alors on accepte l'hypothèse alternative $H_1 : \beta_1 \neq 0$ et on conclut que la régression est significative c'est-à-dire que la variable X a un effet sur la variable Y. C'est le cas de l'exemple où la probabilité **Pr > F** <0.0001 est inférieure à $\alpha = 5\%$ on peut donc conclure qu'il y a un effet de la vitesse sur le prix des voitures.*

2) l'estimation des paramètres du modèle linéaire (Parameter Estimates)

*Présente sous la colonne **Parameter Estimate** les paramètres estimés de la droite empirique ou ajustée : $\hat{Y} = b_0 + b_1 X$ avec sur la ligne **INTERCEPT** $\rightarrow b_0$ l'estimation de β_0 et sur la ligne **Variable_X**, ici vite, $\rightarrow b_1$ l'estimation de β_1. Ainsi, pour l'exemple 1.1.1, la droite de régression empirique s'écrit :*

$prix_{ajuste} = -213.007894 + 3.451216\ vite$.

*Sous la colonne **Standard Error** sont présentés les écart-types de chaque estimateur soit $s(b_0)$ et $s(b_1)$. Sous la colonne **t Value** (test T sous H_0) **,** on retrouve les écarts-réduits $t = b_j/s(b_j)$; la valeur de la statistique de test en T de Student sous l'hypothèse nulle $H_0 : \beta_j = 0$ (j = 0, 1). On peut en déduire un intervalle de confiance de niveau 1-α du paramètre $\beta_j : b_j \pm t_{\alpha/2\ ;n-2}\ s(bj)$, où $t_{\alpha/2\ ;n-2}$ est le fractile de la loi de Student à n-2 degrés de liberté avec un risque d'erreur α.*

*Enfin sous la dernière colonne, **Pr > | t |**, on retrouve la probabilité qui permet de tester si la valeur du paramètre β_j est nulle. La règle de décision de ce test de Student est telle que si la probabilité **Pr > |t|** est inférieure risque α (ou 2α pour un test unilatéral), alors on accepte l'hypothèse alternative $H_1 : \beta_j \neq 0$ et on conclut que la valeur du paramètre est significativement différente de zéro. Pour l'exemple 1.1.1, les paramètres β_0 et β_1 sont significativement différents de zéro puisque les probabilités **Pr > |t|** sont inférieures à $\alpha = 5\%$.*

A noter qu'on peut ajuster un modèle linéaire sans ordonnée à l'origine (β_0), il suffit d'ajouter l'option ***noint*** *à la commande* ***model****.*

L'option ***clm*** *ajoutée à la commande* ***model*** *permet d'établir des intervalles de confiance (de niveau 95% par défaut) pour les moyennes $E(Y_i)$ de la distribution conditionnelle de Y lorsque $X = X_i$. Ainsi, $\hat{Y}$* ***(Predict Value)*** *est l'estimation de $E(Y_i)$ obtenue à partir de l'équation de régression estimée,* ***Lower*** *et* ***Upper 95% Mean*** *représentent les bornes respectivement inférieure et supérieure de l'intervalle de confiance,* ***STd Err Predict*** *est l'écart-type $s(\hat{Y}_i)$ estimation de $\sigma(\hat{Y}_i)$ et* ***Residual*** *représente les résidus (erreurs $e_i = Y_i - \hat{Y}_i$).*

Chaque observation de l'échantillon est identifiée dans la colonne ***Obs****, ainsi pour la première observation des données de l'exemple 1.1, la variable indépendante, ici vite, vaut 140 et la variable dépendante Y (****Dep Var****), ici prix, vaut 239.9. On obtient l'intervalle de confiance : $253.7 \leq E(Y_1) \leq 286.6$ c'est-à-dire qu'on a 95% des chances de contenir le vrai prix moyen des voitures dont la vitesse est de 140 km/h.*

———— *Résultats de l'option clm - Intervalles de confiances* ————

Régression linéaire simple 2

Obs	Dep Var PRIX	Predict Value	Std Err Predict	Lower95% Mean	Upper95% Mean	Residual
1	239.9	270.2	8.004	253.7	286.6	-30.2623
2	242.0	270.2	8.004	253.7	286.6	-28.1623
...	..	..	..	..	..	..
...	..	..	..	..	..	..
26	503.5	442.7	12.830	416.3	469.1	60.7769
27	506.3	477.2	15.520	445.3	509.2	29.0648

Sum of Residuals	0
Sum of Squared Residuals	30399.6421
Predicted Resid SS (Press)	38010.6497

On peut en déduire la marge d'erreur (***(Upper - Lower)/2*** *) en valeur absolue, au niveau 95%, dans l'estimation de ce prix moyen.*

L'option ***clm*** *fournit également la somme des résidus (****Sum of Residuals****), la somme des résidus au carré (****Sum of Squared Residuals****) et la statistique (****Press****) Predicted Resid SS.*

De même, pour obtenir des intervalles de prévision des Y_i à $X = X_i$, il suffit de remplacer l'option ***clm*** *par l'option* ***cli****. A noter que les résultats obtenus ne fournissent pas les écarts-types $s(d_i)$ des erreurs de prévision et que les bornes inférieures et supérieures de l'intervalle de prévision (de niveau 95% par défaut) sont indiquées respectivement par* ***Lower*** *et* ***Upper 95% Predict****.*

———— *Résultats de l'option cli - Intervalles de prévision* ————

Régression linéaire simple 2

Obs	Dep Var PRIX	Predict Value	Std Err Predict	Lower95% Predict	Upper95% Predict	Residual
1	239.9	270.2	8.004	196.5	343.8	-30.2623
2	242.0	270.2	8.004	196.5	343.8	-28.1623
...	..	..	..	..	..	..
...	..	..	..	..	..	..
26	503.5	442.7	12.830	366.2	519.2	60.7769
27	506.3	477.2	15.520	398.6	555.8	29.0648

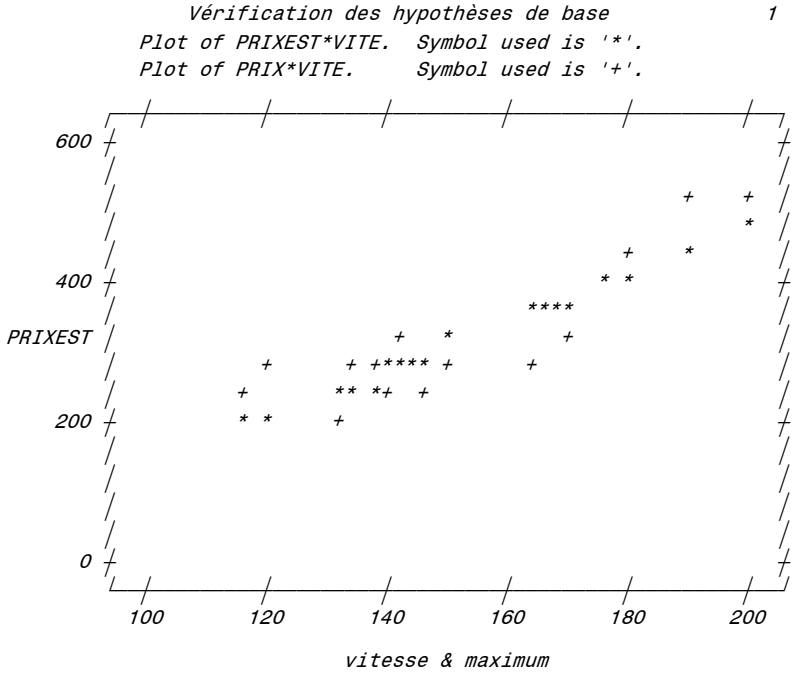
Par exemple, l'intervalle de prévision de Y_1 à $X = X_1$, est : $196.5 \leq Y_1 \leq 343.8$ c'est-à-dire qu'on a 95% des chances de contenir le vrai prix des voitures dont la vitesse est de 140 km/h.

*A noter que si on veut obtenir des estimations, des intervalles de confiance ou de prévision pour des valeurs de la variable indépendante différentes de celles observées dans l'échantillon, les commandes sont exactement les mêmes, il suffit d'ajouter, à la suite des données de l'échantillon, les valeurs des variables indépendantes et mettre des points "****.****" à l'endroit des valeurs de la variable dépendante Y, indiquant ainsi qu'il s'agit de données manquantes.*

De plus, dans l'exemple 1.1.1, on a utilisé la commande ***output out=****nom_table après la commande* ***model*** *afin de sauvegarder les résultats de l'analyse dans une table SAS nommée resultat. Cette table contient les variables du modèle (prix, vite) ainsi que les variables spécifiées par l'utilisateur, indiquées par les options suivantes :*

r = *residus variable des résidus (**Residual**),*
p = *prixest variable des valeurs estimées par le modèle (Ŷ = **Predict Value**),*
stdp = *etyest variable des écarts-types estimés (s(Ŷ$_i$) = **Std Err Predict**).*
stdi = *etiest variable des écarts-types des prévisions s(di).*

*On a ensuite représenté, à l'aide de la procédure **proc plot** (cf.chapitre 2), sur un même graphique les diagrammes de dispersion, les variables dépendante (prix) et prédite (prixest) en fonction de la variable indépendante (vite) afin de visualiser si le modèle linéaire ajusté est plausible.*

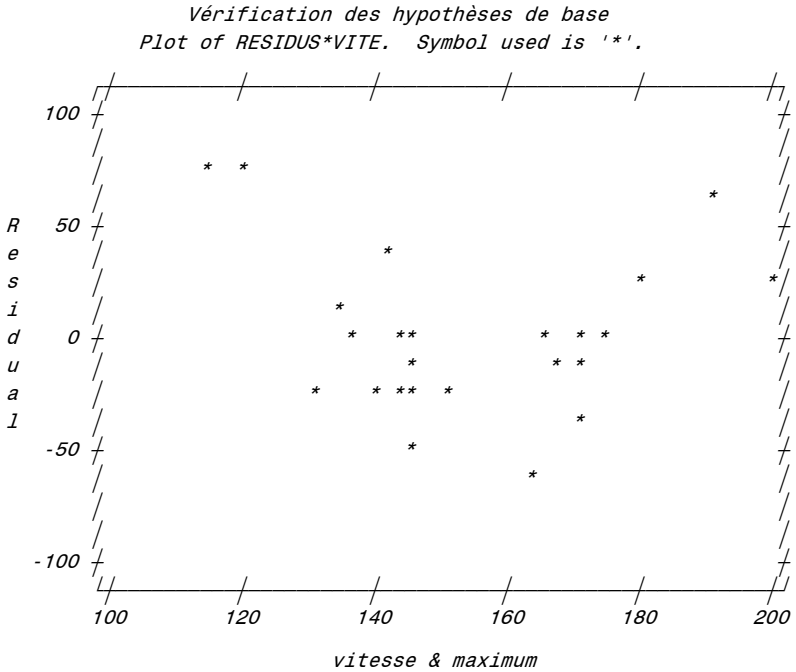


NOTE: 22 obs hidden.

Des procédures sont ensuite utilisées pour vérifier si les hypothèses de base du modèle de régression linéaire sont respectées :

a) *L'hypothèse, **E(ε) = 0**, est vérifiée graphiquement avec la procédure **proc plot**. Pour conclure que cette hypothèse est respectée, le graphique des résidus en fonction de la variable indépendante (vite) doit présenter une certaine symétrie par rapport à l'axe e$_i$ = 0.*

Selon le diagramme de dispersion ci-dessous, il ne semble pas y avoir un problème concernant cette hypothèse de base car les résidus semblent répartis aléatoirement autour de l'axe e$_i$ = 0.



NOTE: 3 obs hidden.

b) *C'est sur l'hypothèse de* ***normalité des résidus*** *que sont basés les tests d'hypothèses et les intervalles de confiance et de prévision. Pour vérifier cette hypothèse on utilise un test de normalité des résidus à l'aide de procédure* ***proc univariate*** *(cf.chapitre 3).*

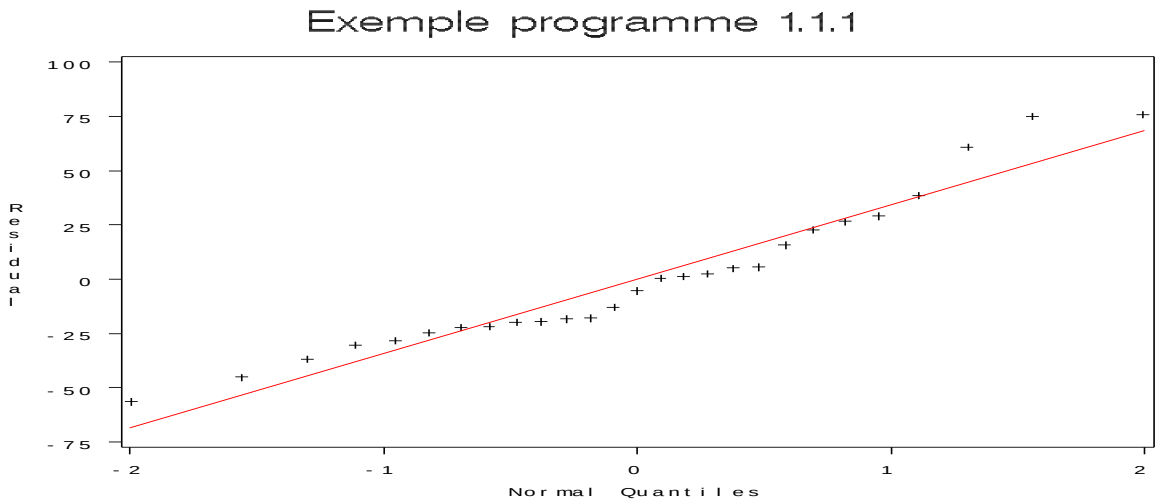
Résultats du test de normalité des résidus

Vérification des hypothèses de base 3
Tests for Normality

Test	--Statistic---		-----p Value------	
Shapiro-Wilk	W	0.935912	Pr < W	0.0966
Kolmogorov-Smirnov	D	0.144313	Pr > D	>0.1500
Cramer-von Mises	W-Sq	0.101908	Pr > W-Sq	0.1008
Anderson-Darling	A-Sq	0.636952	Pr > A-Sq	0.0894
W:Normal		0.934101	Pr<W	0.0957

La probabilité ***Pr < W*** *étant égale à 0,0957, ce qui est supérieur au risque d'erreur* α *= 5%, il semble que les résidus possèdent une distribution normale (du moins, le test statistique ne nous permet pas de penser le contraire).*

On peut aussi vérifier graphiquement cette hypothèse de normalité à partir des quantiles normalisés des résidus : option ***QQPLOT****. Elle consiste à renomer les résidus par ordre décroissant, d'associer à un résidu* e_i *le (i / (n + 1))-quantile* q_i *d'une loi normale centrée-réduite puis de représenter les* e_i *en fonction des* q_i*. Si les erreurs sont normalement distribuées, les points sur le graphique doivent être à peu près alignés.*



*c) L'hypothèse concernant l'***homoscédasticité (variance constante)** *: si la variance n'est pas constante (hétéroscédasticité), les estimations des écart-types sont biaisées. Il en résulte que les résultats des tests statistiques sont biaisés et que le calcul des intervalles de confiance et de prévision est également biaisé. Cette hypothèse peut être décelée à l'aide du graphique des résidus en fonction de la variable indépendante (vite) ou en fonction des valeurs prédites par le modèle. L'hypothèse d'homoscédasticité est confirmée si les résidus sont distribués aléatoirement à l'intérieur d'une bande.*

On peut tester cette hypothèse en ajoutant l'option ***spec*** *à la commande* ***model****.*

Résultats de l'option spec - Homoscédasticité

Régression linéaire simple 9
The REG Procedure
Model: MODEL1
Dependent Variable: PRIX prix en francs belges
Test of First and Second
Moment Specification

DF	Chi-Square	Pr > ChiSq
2	2.97	0.2260

*La règle de décision de ce test est telle que si la probabilité **Pr > Chisq** est supérieure au risque d'erreur α, on peut conclure que les variances sont constantes (homoscédasticité) sinon on conclut que les variances ne sont pas constantes (hétéroscédasticité).*

*Pour l'exemple 1.1, la probabilité **Pr > Chisq** = 0.2260 est supérieure à α = 5%, on peut conclure que les variances sont égales ; l'hypothèse d'homoscédasticité semble donc vérifiée.*

L'exemple 1.1.2 suivant, illustre comment déterminer des intervalles de confiance et de prévision pour un niveau de confiance différent de 95%.

Programme 1.1.2:

```
data interval;
title 'Intervalles de confiance et de prévision de niveau 99% ';
set resultat;
/* Intervalles de confiance */
bcinf = prixest - tinv(0.995,25)*etyest ;
bcsup = prixest + tinv(0.995,25)*etyest ;
/* Intervalles de prévision */
bpinf = prixest - tinv(0.995,25)*etiest ;
bpsup = prixest + tinv(0.995,25)*etiest ;
proc print;
var prixest bcinf bcsup bpinf bpsup ;
run;
```

*A partir des données de la table SAS resultat, résultats de l'analyse de l'exemple 1.1.1, on calcule les intervalles en utilisant la formule **tinv(1-α/2 , n-p-1)** correspondant au fractile $t_{\alpha/2\,;\,n-p-1}$ de la loi de Student à n-p-1 degrés de liberté avec un risque d'erreur α, où p désigne le nombre de variables indépendantes et n la taille de l'échantillon observé.*

*Seul l'écart-type diffère dans le calcul de la marge d'erreur, $t_{\alpha/2\,;\,n-p-1} \times s(\hat{Y}_i)$ s(y) pour un intervalle de confiance et $t_{\alpha/2\,;\,n-p-1} \times s(d_i)$ pour un intervalle de prévision. Ces écarts-types ont été sauvegardés (programme 1.1.1) respectivement par l'option **stdp** et **stdi** de la commande **output out=**.*

Résultats du programme 1.1.2 :

Intervalles de confiance et de prévision de niveau 99%   1

OBS	PRIXEST	BCINF	BCSUP	BPINF	BPSUP
1	270.162	247.851	292.474	170.434	369.891
2	270.162	247.851	292.474	170.434	369.891
...	..	..	..	..	..
...	..	..	..	..	..
26	442.723	406.961	478.485	339.152	546.294
27	477.235	433.974	520.496	370.842	583.628

*L'exemple 1.1.3, illustre la commande **test** de la procédure **proc reg**, particulièrement utile pour effectuer un test d'hypothèse sur un paramètre par exemple : $H_0 : \beta_1 = 4$ contre $H_1 : \beta_1 \neq 4$.*

*Les résultats du test pour vérifier si le paramètre est nul sont donnés dans la première partie **Parameter Estimates**. Elle s'utilise après la commande **model** avec la syntaxe : **test** variable_X **=** valeur testée en H_0 .*

Reprenons l'exemple 1.1.1, peut-on affirmer que chaque km/h de la vitesse contribue à faire augmenter le prix moyen de 4 MFB ?

Programme 1.1.3:

```
proc reg data=biblio.voitures;
model prix = vite ;
test prix = 4 ;
run;
```

Résultats du programme 1.1.3:

Test 1 Results for Dependent Variable PRIX

Source	DF	Mean Square	F Value	Pr > F
Numerator	1	3912.48086	3.22	0.0850
Denominator	25	1215.98568		

*La règle de décision est telle que si la probabilité **Pr > F** est supérieure au risque d'erreur α, on doit favoriser l'hypothèse nulle H_0. C'est le cas pour l'exemple 1.1.3, la probabilité **Pr > F** = 0,0850 > 5%, on conclut que β_1 = 4 c'est-à-dire pour chaque km/h supplémentaire de la vitesse, le prix augmente de 4 MFB en moyenne.*

1.2 Régression linéaire multiple

Dans le cas ou le modèle linéaire multiple de la forme $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \varepsilon$ peut être approprié, la syntaxe de la procédure se présente comme suit :

procédure – commandes - options

```
proc reg ;
model variable_Y = variable_X1  variable_X2  …  variable_Xp  / <options >;
```

*Les variables_X_j (variables indépendantes) et la variable_Y (variable dépendante) de la commande **model** permettent l'identification des variables du modèle.*

Programme 1.2:

```
/* Régression linéaire multiple - options */
proc reg data=biblio.voitures;
options ps=60 ls=80;
title 'Régression linéaire multiple';
model prix = vite cons cylin volu rpp / dw tol vif collin ; /* options autocorrélation - colinéarité */
/* Syntaxe de la commande test */
b2b4 : test cons = volu = 0 ;
run;
```

Les résultats de la procédure proc reg pour une analyse de régression multiple sont également présentés en deux parties, l'analyse de la variance et l'estimation des paramètres du modèle. Ces résultats s'analysent exactement de la même façon qu'en régression simple.

*La probabilité **Prob>F** permet de vérifier si la régression est significative dans son ensemble ce qui est équivalent à tester les hypothèses statistiques suivantes :*

H_0 : $\beta_1 = \beta_2 = \ldots = \beta_p = 0$ la régression n'est pas significative
(aucune contribution significative des X_j)

H_1 : au moins un des $\beta_j \neq 0$ la régression est significative dans son ensemble.
(au moins une variable indépendante apporte une contribution significative)

Résultats du programme 1.2:

Régression linéaire multiple 1
The REG Procedure
Dependent Variable: PRIX prix en francs belges

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	173418	34684	62.15	<.0001
Error	21	11719	558.03035		
Corrected Total	26	185136			

Root MSE	23.62267	R-Square	0.9367
Dependent Mean	319.37407	Adj R-Sq	0.9216
Coeff Var	7.39655		

Parameter Estimates

Variable	Label	DF	Parameter Estimate	Standard Error	t Value
Intercept	Intercept	1	-761.52182	140.40569	-5.42
VITE	vitesse & maximum	1	4.83273	0.73558	6.57
CONS	consommation & urbaine	1	0.83389	7.94025	0.11
CYLIN	cylindrée	1	0.10746	0.04457	2.41
VOLU	volume maximum du coffre	1	0.02534	0.01757	1.44
RPP	rapport & poids-puissance	1	9.72485	2.59732	3.74

Parameter Estimates

Variable	Label	DF	Pr > |t|	Tolerance	Variance Inflation
Intercept	Intercept	1	<.0001	.	0
VITE	vitesse & maximum	1	<.0001	0.07939	12.59664
CONS	consommation & urbaine	1	0.9174	0.26137	3.82592
CYLIN	cylindrée	1	0.0252	0.24954	4.00733
VOLU	volume maximum du coffre	1	0.1640	0.73562	1.35940
RPP	rapport & poids-puissance	1	0.0012	0.10412	9.60416

---

*La règle de décision de ce test est telle que si la probabilité **Prob>F** est supérieure au risque α, alors on accepte l'hypothèse nulle H_0 et on conclut que le modèle de régression multiple n'est pas significative.*

*Pour l'exemple 1.2, la probabilité **Prob>F** = 0.0001 est inférieure à α = 5% (rejet de l'hypothèse nulle H_0) , il y a donc au moins une variable indépendante qui a un effet significatif sur le prix des voitures.*

*En effet, les variables indépendantes significatives sont présentées dans la partie estimation des paramètres où chaque paramètre est testé à zéro. On peut affirmer que $\beta_1 \neq 0$, $\beta_3 \neq 0$ et $\beta_5 \neq 0$ car les probabilités **Pr > \|t\|** pour ces paramètres sont inférieures au risque α = 5%.*

*Ce test, lorsqu'il est effectué sur un paramètre associé à une variable indépendante, permet de tester la contribution marginale de cette variable indépendante ; c'est le cas de la vitesse, de la cylindrée et du rapport poids/puissance qui ont un effet significatif sur le prix des voitures, les probabilités **Pr > \|t\|** pour les paramètres de ces variables sont inférieures au risque α = 5%.*

*La commande **test** est également valide en régression multiple. Il est même possible de **tester** plusieurs variables simultanément c'est-à-dire faire un test sur un sous-ensemble de paramètres du modèle. Le programme 1.2 effectue ce test sur les variables consommation et volume afin de vérifier qu'elles n'ont aucun effet significatif sur le prix.*

Résultats de la commande test: ———————————————————————————

The REG Procedure
Test b2b4 Results for Dependent Variable PRIX

Source	DF	Mean Square	F Value	Pr > F
Numerator	2	707.36886	1.27	0.3022
Denominator	21	558.0303		

---

*On peut conclure que les variables indépendantes consommation et volume n'ont aucun effet **sur** le prix vu que la probabilité **Pr > F** = 0.3022 est supérieure à α = 5%, on favorise l'hypothèse nulle H_0.*

*Avant la commande **test**, on a donné un nom, b2b4 (2ème et 4ème variables explicatives) suivi du double-point "**:**", au test effectué. Ce nom est facultatif, il peut être utile si on effectue plusieurs commandes **test** à l'intérieur d'un même programme, cela permet de les identifier.*

<u>Options de Vérification des hypothèses de base</u> (suite)

d) <u>L'hypothèse d'absence d'autocorrélation</u *peut ne pas être vérifiée si les données sont d'ordre chronologiques. Il s'agit alors de régression temporelle. Les inconvénients d'une autocorrélation sont importants : les variances des paramètres et des résidus sont biaisées, ce qui engendre des biais lorsqu'on effectue des tests paramétriques ou lorsqu'on calcule des intervalles de confiance ou de prévision.*

Cette hypothèse peut se vérifier graphiquement à partir du diagramme de dispersion des résidus en fonction du temps. Le nuage de points va présenter une certaine particularité selon que l'autocorrélation est positive ou négative : si les résidus forment une pente de droite positive ou négative ou si les résidus montent et descendent de façon non aléatoire, alors on est probablement en présence d'autocorrélation.

*Le système SAS dispose d'un test statistique qui s'utilise avec l'option **dw** de la commande **model**. Ce test de Durbin-Watson est le plus utilisé pour déceler l'autocorrélation. Les hypothèses statistiques sont les suivantes :*

H_0 : Absence d'autocorrélation positive (négative)
H_1 : présence d'autocorrélation positive (négative).

Résultats de l'option dw - test d'autocorrélation :

Durbin-Watson D	1.342
Number of Observations	27
1st Order Autocorrelation	0.250

*Pour vérifier s'il y a autocorrélation ou non, il faut comparer la valeur la statistique **D** de Durbin-Watson avec des valeurs (d_1 et d_2) obtenues de la table statistique de Durbin-Watson. Ces valeurs dépendent de la taille de l'échantillon, du nombre de variables indépendantes du modèle et du risque d'erreur α choisi.*

Règle de décision :
- *Si $D < d_1$ alors il y a autocorrélation positive*
- *Si $d_1 < D < d_2$ on ne peut rien conclure*
- *Si $D > d_2$ il n'y a pas d'autocorrélation positive*

Règle de décision :
- *Si $4 - D < d_1$ alors il y a autocorrélation négative*
- *Si $d_1 < 4 - D < d_2$ on ne peut rien conclure*
- *Si $4 - D > d_2$ il n'y a pas d'autocorrélation négative*

Pour l'exemple 1.2, le modèle de régression utilisé a 5 variables indépendantes, n = 27 observations avec un risque d'erreur α = 5%, la statistique D = 1.342 et les valeurs de la tables (d_1 = 1.01 , d_2 = 1.86). On peut seulement conclure qu'il n'y a pas d'autocorrélation négative puisque 4 – D = 2.658 > d_2 = 1.86.

La valeur de la statistique D est comprise entre 0 et 4. D'une façon générale, D < 2 pour des corrélations positives, D > 2 pour des corrélations négatives, et doit être proche de 2 (disons entre 1.5 et 2.5) pour que l'autocorrélation soit acceptable.

***e) <u>L'hypothèse d'absence de colinéarité</u>** : Il y a colinéarité lorsque plusieurs variables indépendantes sont liées linéairement, cela affecte la précision des estimateurs des paramètres du modèle. Les options SAS **collin**, **collinoint**, **tol** et **vif** de la commande **model**, permettent de détecter la colinéarité.*

*L'option **vif** fournit les facteurs d'inflation de la variance (**Variance Inflation**). Ils mesurent l'augmentation de la variance des estimateurs due à la colinéarité qui existe entre les variables indépendantes. Il n'y a pas de critère formel pour décider si un facteur est trop élevé pour qu'il y ait colinéarité. En pratique, un facteur excédant 10 est souvent considéré comme étant une indication de la présence de colinéarité.*

*L'option **tol** permet d'obtenir, pour chaque variable, le facteur de tolérance (**Tolérance**). Un facteur de tolérance est l'inverse mathématique du facteur d'inflation de la variance. De même, le facteur de tolérance n'admet pas de règle formelle. Cependant, en pratique, lorsque le facteur de tolérance est inférieur à 10%, on peut considérer qu'il y a colinéarité.*

*L'option **collin** (**collinoint** si le modèle ne contient pas d'ordonnée à l'origine) permet aussi de déceler la colinéarité : si la valeur propre (**Eigenvalue**) tend vers zéro (ou si le CI (**Condition Index**) est supérieur à 100), cela indique la présence probable d'un problème de colinéarité. Les variables qui ont des valeurs élevées sur cette ligne sont les variables qui contribuent à la colinéarité. De plus, d'un point de vue pratique, si le CI < 100, la colinéarité n'est pas un problème, si CI > 1000, la colinéarité est sévère.*

Résultats des options tol, vif et collin - test de colinéarité :

Parameter Estimates

Variable	Label	DF	Pr > |t|	Tolerance	Variance Inflation
Intercept	Intercept	1	<.0001	.	0
VITE	vitesse & maximum	1	<.0001	0.07939	12.59664
CONS	consommation & urbaine	1	0.9174	0.26137	3.82592
CYLIN	cylindrée	1	0.0252	0.24954	4.00733
VOLU	volume maximum du coffre	1	0.1640	0.73562	1.35940
RPP	rapport & poids-puissance	1	0.0012	0.10412	9.60416

Collinearity Diagnostics

		Condition	Proportion of Variation		
Number	Eigenvalue	Index	Intercept	VITE	CONS
1	5.78428	1.00000	0.00003113	0.00004616	0.00018510
2	0.12926	6.68961	0.00001184	0.00136	0.00385
3	0.07562	8.74604	0.00088631	0.00053883	0.00008014
4	0.00536	32.84643	0.03497	0.09714	0.10125
5	0.00486	34.50267	0.00286	0.00184	0.85129
6	0.00062251	96.39428	0.96124	0.89908	0.04335

*Pour l'exemple 1.2, les résultats des options **tol** et **vif** sont présentés dans la partie estimation des paramètres (**Parameter Estimates**). Selon les résultats obtenus, il n'y a pas de problème de colinéarité. Dans la partie **Collinearity Diagnostics** se trouvent les résultats de l'option **collin**. Selon ces résultats, il n'y a pas de colinéarité.*

1.3 Méthodes de sélection de variables indépendantes

*Trois méthodes de sélection de variables indépendantes sont présentées et comparées à l'aide du coefficient de détermination ajusté et de la statistique C_p de Mallows. Ces divers modèles de régression multiple s'obtiennent à l'aide de l'option **selection =** méthode de la commande **model** suivie des seuils **sle** et/ou **sls** appropriés pour l'introduction et/ou le retrait de variables indépendantes du modèle.*

*L'option **details** de la commande **model**, que l'on peut utiliser après l'option **selection,** permet de présenter des résultats plus détaillés pour chacune des étapes de la méthode choisie.*

1.3.1 Régression pas à pas

*Le modèle de régression pas à pas s'effectue à l'aide de l'option : **selection = stepwise** et des seuils **sle=**seuil_introduction **sls=**seuil_retrait.*

Programme 1.3.1 :

```
proc reg data=biblio.voitures;
title 'Régression pas à pas';
/* Syntaxe de l'option stepwise de la commande model */
model prix = vite cons cylin volu rpp / selection = stepwise  sle = 0.05  sls = 0.05  details ;
run;
```

Le programme 1.3.1 utilise le modèle de régression pas à pas avec un seuil de 5% pour l'introduction et le retrait des variables indépendantes.

*Le premier résultat représente la première étape (**Step 1**). L'option **details** permet d'obtenir, pour chaque variable indépendante, le facteur de tolérance (**Tolerance**), le coefficient de détermination du modèle si cette variable est introduite (**Model R-Square**), le rapport **F** (''F partiel'') et la probabilité **Pr > F**.*

*La variable dont le rapport **F** est maximal est introduite dans le modèle si la probabilité **Pr >F** correspondante est inférieure au seuil d'introduction.*

*Pour l'étape 1 ci-dessus, la vitesse VITE est la première variable introduite dans le modèle car c'est la variable indépendante dont le rapport **F** = 127.252 est maximal et dont la probabilité **Pr > F** = 0.0001 est inférieure au seuil d'introduction de 5%.*

Résultats du programme 1.3.1:

Méthode pas à pas - Etape 1

Régression pas à pas 1

The REG Procedure

Dependent Variable: PRIX prix en francs belges

Stepwise Selection: Step 1

Statistics for Entry

DF = 1,25

Variable	Tolerance	Model R-Square	F Value	Pr > F
VITE	1.000000	0.8358	127.25	<.0001
CONS	1.000000	0.6624	49.05	<.0001
CYLIN	1.000000	0.7286	67.11	<.0001
VOLU	1.000000	0.0478	1.25	0.2735
RPP	1.000000	0.5897	35.92	<.0001

Variable VITE Entered: R-Square = 0.8358 and C(p) = 31.4767

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	154737	154737	127.25	<.0001
Error	25	30400	1215.98568		
Corrected Total	26	185136			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	-213.00789	47.66922	24280	19.97	0.0001
VITE	3.45122	0.30594	154737	127.25	<.0001

Bounds on condition number: 1, 1

---

*Le coefficient de détermination du modèle **R-square**= 0.8358 et la statistique **C(p)** = 31.4767 Mallows sont indiqués.*

*Une analyse de la variance est ensuite effectuée avec cette première variable suivie de l'estimation des paramètres du modèle. La présentation de la partie estimation des paramètres diffère de celle de la régression multiple (paragraphe 1.2) ; les écarts-réduits **t** et les probabilités **Pr > |t|** sont remplacés par les rapports **F** et les probabilités **Pr > F**.*

Méthode pas à pas - Etape 2

Stepwise Selection: Step 2

Statistics for Entry

DF = 1,24

Variable	Tolerance	Model R-Square	F Value	Pr > F
CONS	0.390908	0.8616	4.47	0.0451
CYLIN	0.307405	0.8638	4.93	0.0361
VOLU	0.999517	0.8752	7.58	0.0111
RPP	0.120906	0.9017	16.10	0.0005

Variable RPP Entered: R-Square = 0.9017 and C(p) = 11.6012

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	166944	83472	110.12	<.0001
Error	24	18192	758.01893		
Corrected Total	26	185136			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	-826.44267	157.42758	20890	27.56	<.0001
VITE	6.06505	0.69469	57778	76.22	<.0001
RPP	11.27333	2.80921	12207	16.10	0.0005

Bounds on condition number: 8.2709, 33.084

L'étape 2, indique que c'est la variable rapport poids/puissance RPP qui est ensuite introduite comme deuxième variable indépendante du modèle. Tout comme l'étape 1, on obtient les résultats de l'analyse de variance et l'estimation des paramètres du modèle qui contient maintenant 2 variables indépendantes VITE et RPP.

*Les rapports F associés aux estimateurs des paramètres nous indiquent si une variable indépendante doit être retranchée ou pas du modèle. Si la probabilité **Pr > F** est supérieure au seuil de retrait (5%), la variable correspondante sera retranchée du modèle. Vu que les deux variables introduites VITE et RPP ont des probabilités **Pr > F** > 5%, aucune variable n'est retranchée du modèle.*

Méthode pas à pas - Etape 3

Stepwise Selection: Step 3

Statistics for Removal

DF = 1,24

Variable	Partial R-Square	Model R-Square	F Value	Pr > F
VITE	0.3121	0.5897	76.22	<.0001
RPP	0.0659	0.8358	16.10	0.0005

Statistics for Entry
DF = 1,23

Variable	Tolerance	Model R-Square	F Value	Pr > F
CONS	0.370826	0.9128	2.91	0.1014
CYLIN	0.307387	0.9291	8.86	0.0068
VOLU	0.875227	0.9150	3.60	0.0703

Variable CYLIN Entered: R-Square = 0.9291 and C(p) = 4.5352

Stepwise Selection: Step 3
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	172003	57334	100.41	<.0001
Error	23	13133	571.01631		
Corrected Total	26	185136			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	-819.87471	136.65379	20554	36.00	<.0001
VITE	5.11544	0.68214	32112	56.24	<.0001
CYLIN	0.12093	0.04063	5059.07926	8.86	0.0068
RPP	11.21773	2.43827	12086	21.17	0.0001

Bounds on condition number: 10.587, 66.333

---

*A l'étape 3, la variable cylindrée CYLIN est introduite, on obtient également, grâce à l'option details les statistiques pour le retrait (**Statistics for Removal**).*

Méthode pas à pas - Etape 4

Stepwise Selection: Step 4
Statistics for Removal
DF = 1,23

Variable	Partial R-Square	Model R-Square	F Value	Pr > F
VITE	0.1734	0.7556	56.24	<.0001
CYLIN	0.0273	0.9017	8.86	0.0068
RPP	0.0653	0.8638	21.17	0.0001

Statistics for Entry
DF = 1,22

Variable	Tolerance	Model R-Square	F Value	Pr > F
CONS	0.301056	0.9304	0.43	0.5169
VOLU	0.847294	0.9367	2.64	0.1182

All variables left in the model are significant at the 0.0500 level.
No other variable met the 0.0500 significance level for entry into the model.

The REG Procedure
Dependent Variable: PRIX prix en francs belges

Summary of Stepwise Selection

Step	Variable Entered	Variable Removed	Label	Number Vars In	Partial R-Square	Model R-Square
1	VITE		vitesse & maximum	1	0.8358	0.8358
2	RPP		rapport & poids-puissance	2	0.0659	0.9017
3	CYLIN		cylindrée	3	0.0273	0.9291

Summary of Stepwise Selection

Step	C(p)	F Value	Pr > F
1	31.4767	127.25	<.0001
2	11.6012	16.10	0.0005
3	4.5352	8.86	0.0068

A l'étape 4, les résultats indiquent qu'aucune autre variable ne peut être introduite (les probabilités Pr>F des variables CONS et VOLU sont supérieures à 5%) ou retranchée (les probabilités Pr>F des variables introduite VITE, RPP et CYLIN sont inférieures à 5%).

*Un sommaire du modèle est également présenté où l'on retrouve, pour chaque étape, la variable introduite (**Entered**) ou retranchée (**Removed**), le nombre de variables dans le modèle (**Number In**), le coefficient de détermination partiel (**Partial R-square**), le coefficient de détermination du modèle (**Model R-square**), la*

*statistique de Mallows **C(p)**, la valeur **F** pour tester si la variable apporte une contribution marginale significative, la probabilité **Pr > F** qui permet d'effectuer le test en F précédent.*

Enfin, $Prix_{ajusté}$ = -819.875 + 5.115 VITE + 0.121 CYLIN + 11.218 RPP est la droite de régression empirique obtenue avec la méthode de régression pas à pas.

1.3.2 Introduction progressive

*Le modèle d'introduction progressive s'utilise avec l'option : **selection = forward** avec le seuil d'introduction **sle=**seuil_introduction uniquement, aucune variable ne peut être retranchée du modèle.*

Programme 1.3.2:

```
proc reg data=biblio.voitures;
title 'Introduction progressive';
/* Syntaxe de l'option forward de la commande model */
model prix = vite cons cylin volu rpp / selection = forward  sle = 0.05  details ; run;
```

Les résultats du modèle d'introduction progressive sont présentés par étape d'introduction de variables indépendantes. Cette méthode présente également un sommaire de toutes les étapes. La droite de régression empirique est identique à celle obtenue avec la méthode de régression pas à pas.

Résultats du programme 1.3.2:

Méthode d'introduction progressive - Etape 1

Introduction progressive 1

The REG Procedure

Dependent Variable: PRIX prix en francs belges

Forward Selection: Step 1

Statistics for Entry

DF = 1,25

Variable	Tolerance	Model R-Square	F Value	Pr > F
VITE	1.000000	0.8358	127.25	<.0001
CONS	1.000000	0.6624	49.05	<.0001
CYLIN	1.000000	0.7286	67.11	<.0001
VOLU	1.000000	0.0478	1.25	0.2735
RPP	1.000000	0.5897	35.92	<.0001

Variable VITE Entered: R-Square = 0.8358 and C(p) = 31.4767

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	154737	154737	127.25	<.0001
Error	25	30400	1215.98568		
Corrected Total	26	185136			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	-213.00789	47.66922	24280	19.97	0.0001
VITE	3.45122	0.30594	154737	127.25	<.0001

Bounds on condition number: 1, 1

Méthode d'introduction progressive - Etape 2

Forward Selection: Step 2

Statistics for Entry

DF = 1,24

Variable	Tolerance	Model R-Square	F Value	Pr > F
CONS	0.390908	0.8616	4.47	0.0451
CYLIN	0.307405	0.8638	4.93	0.0361
VOLU	0.999517	0.8752	7.58	0.0111
RPP	0.120906	0.9017	16.10	0.0005

Forward Selection: Step 2

Variable RPP Entered: R-Square = 0.9017 and C(p) = 11.6012

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	166944	83472	110.12	<.0001
Error	24	18192	758.01893		
Corrected Total	26	185136			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	-826.44267	157.42758	20890	27.56	<.0001
VITE	6.06505	0.69469	57778	76.22	<.0001
RPP	11.27333	2.80921	12207	16.10	0.0005

Bounds on condition number: 8.2709, 33.084

Méthode d'introduction progressive - Etape 3

Forward Selection: Step 3

Statistics for Entry

DF = 1,23

Variable	Tolerance	Model R-Square	F Value	Pr > F
CONS	0.370826	0.9128	2.91	0.1014
CYLIN	0.307387	0.9291	8.86	0.0068
VOLU	0.875227	0.9150	3.60	0.0703

Variable CYLIN Entered: R-Square = 0.9291 and C(p) = 4.5352

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	172003	57334	100.41	<.0001
Error	23	13133	571.01631		
Corrected Total	26	185136			

Forward Selection: Step 3

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	-819.87471	136.65379	20554	36.00	<.0001
VITE	5.11544	0.68214	32112	56.24	<.0001
CYLIN	0.12093	0.04063	5059.07926	8.86	0.0068
RPP	11.21773	2.43827	12086	21.17	0.0001

Bounds on condition number: 10.587, 66.333

Méthode d'introduction progressive - Etape 4

Forward Selection: Step 4

Statistics for Entry

DF = 1,22

Variable	Tolerance	Model R-Square	F Value	Pr > F
CONS	0.301056	0.9304	0.43	0.5169
VOLU	0.847294	0.9367	2.64	0.1182

No other variable met the 0.0500 significance level for entry into the model.

Summary of Forward Selection

Step	Variable Entered	Label	Number Vars In	Partial R-Square	Model R-Square	C(p)
1	VITE	vitesse & maximum	1	0.8358	0.8358	31.4767
2	RPP	rapport & poids-puissance	2	0.0659	0.9017	11.6012
3	CYLIN	cylindrée	3	0.0273	0.9291	4.5352

Summary of Forward Selection

Step	F Value	Pr > F
1	127.25	<.0001
2	16.10	0.0005
3	8.86	0.0068

1.3.3 Elimination progressive

*Le modèle d'élimination progressive s'utilise avec l'option : **selection = backward** avec le seuil d'élimination **sls=**seuil_retrait uniquement, aucune variable ne peut être introduite dans modèle.*

La méthode débute (Step 0) la régression avec toutes les variables indépendantes dans le modèle. A chaque étape, la variable retranchée est indiquée ainsi que l'analyse de variance et l'estimation des paramètres du modèle.

Programme 1.3.3 :

```
proc reg data=biblio.voitures;
title 'Elimination progressive';
/* Syntaxe de l'option backward de la commande model */
model prix = vite cons cylin volu rpp / selection = backward  sls = 0.05 details ;
run;
```

A l'étape 1, la variable consommation CONS est retirée du modèle, puis la variable volume VOLU à l'étape 2. Le sommaire des étapes de cette méthode est présenté à la fin de la dernière étape.

Résultats du programme 1.3.3 :

Méthode d'élimination progressive - Etape 0

Elimination progressive 3

The REG Procedure

Dependent Variable: PRIX prix en francs belges

Backward Elimination: Step 0

All Variables Entered: R-Square = 0.9367 and C(p) = 6.0000

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	173418	34684	62.15	<.0001
Error	21	11719	558.03035		
Corrected Total	26	185136			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	-761.52182	140.40569	16415	29.42	<.0001
VITE	4.83273	0.73558	24087	43.16	<.0001
CONS	0.83389	7.94025	6.15472	0.01	0.9174
CYLIN	0.10746	0.04457	3243.27000	5.81	0.0252
VOLU	0.02534	0.01757	1160.74130	2.08	0.1640
RPP	9.72485	2.59732	7822.96191	14.02	0.0012

Bounds on condition number: 12.597, 156.97

Méthode d'élimination progressive - Etape 1

Backward Elimination: Step 1

Statistics for Removal

DF = 1,21

Variable	Partial R-Square	Model R-Square	F Value	Pr > F
VITE	0.1301	0.8066	43.16	<.0001
CONS	0.0000	0.9367	0.01	0.9174
CYLIN	0.0175	0.9192	5.81	0.0252
VOLU	0.0063	0.9304	2.08	0.1640
RPP	0.0423	0.8944	14.02	0.0012

Variable CONS Removed: R-Square = 0.9367 and C(p) = 4.0110

Backward Elimination: Step 1

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	173411	43353	81.35	<.0001
Error	22	11725	532.94509		
Corrected Total	26	185136			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	-762.92883	136.58747	16628	31.20	<.0001
VITE	4.85849	0.67770	27391	51.40	<.0001
CYLIN	0.10934	0.03989	4004.13879	7.51	0.0119
VOLU	0.02601	0.01600	1408.58300	2.64	0.1182
RPP	9.75637	2.52126	7980.35687	14.97	0.0008

Bounds on condition number: 11.195, 100.85

Méthode d'élimination progressive - Etape 2

Backward Elimination: Step 2

Statistics for Removal

DF = 1,22

Variable	Partial R-Square	Model R-Square	F Value	Pr > F
VITE	0.1480	0.7887	51.40	<.0001
CYLIN	0.0216	0.9150	7.51	0.0119
VOLU	0.0076	0.9291	2.64	0.1182
RPP	0.0431	0.8936	14.97	0.0008

Variable VOLU Removed: R-Square = 0.9291 and C(p) = 4.5352

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	172003	57334	100.41	<.0001
Error	23	13133	571.01631		
Corrected Total	26	185136			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	-819.87471	136.65379	20554	36.00	<.0001
VITE	5.11544	0.68214	32112	56.24	<.0001
CYLIN	0.12093	0.04063	5059.07926	8.86	0.0068
RPP	11.21773	2.43827	12086	21.17	0.0001

Bounds on condition number: 10.587, 66.333

Méthode d'élimination progressive - Etape 3

Backward Elimination: Step 3

Statistics for Removal

DF = 1,23

Variable	Partial R-Square	Model R-Square	F Value	Pr > F
VITE	0.1734	0.7556	56.24	<.0001
CYLIN	0.0273	0.9017	8.86	0.0068
RPP	0.0653	0.8638	21.17	0.0001

All variables left in the model are significant at the 0.0500 level.

Summary of Backward Elimination

Step	Variable Removed	Label	Number Vars In	Partial R-Square	Model R-Square	C(p)
1	CONS	consommation & urbaine	4	0.0000	0.9367	4.0110
2	VOLU	volume maximum du coffre	3	0.0076	0.9291	4.5352

La droite de régression empirique obtenue selon la méthode d'élimination progressive est identique à celle obtenue avec les deux méthodes précédentes.

1.4 Choix d'un modèle

*Le système SAS dispose de deux autres méthodes **adjrsq** et **cp** pour l'option **selection** de la commande **model**. Ces méthodes permettent de choisir un modèle, parmi tous les modèles possibles, qui possède respectivement le meilleur coefficient de détermination ajusté ou la plus faible valeur de la statistique cp de Mallows.*

*Il est conseillé d'ajouter l'option **best =** n (n entier) pour réduire le temps de calcul, seuls les n meilleurs modèles seront présenter.*

Programme 1.4:

```
/* Choix d'un modèle - options */
data select;
set biblio.voitures;
title 'Les 5 meilleurs modèle - R² ajusté - Cp Mallows';
proc reg;
model prix = vite cons cylin volu rpp / selection = adjrsq best = 10;
model prix = vite cons cylin volu rpp / selection = cp best = 10;
run;
```

Les options des deux méthodes de sélection sont utilisées dans le programme 1.4. Les cinq meilleurs modèles de chaque méthode sont présentés.

Résultats du programme 1.4:

```
Les 5 meilleurs modèle - R² ajusté - Cp Mallows              1
                    The REG Procedure
                  Dependent Variable: PRIX

                Adjusted R-Square Selection Method
Number in    Adjusted
  Model      R-Square    R-Square    Variables in Model
      4        0.9252      0.9367    VITE CYLIN VOLU RPP
      5        0.9216      0.9367    VITE CONS CYLIN VOLU RPP
      3        0.9198      0.9291    VITE CYLIN RPP
      4        0.9178      0.9304    VITE CONS CYLIN RPP
      4        0.9045      0.9192    VITE CONS VOLU RPP
      3        0.9040      0.9150    VITE VOLU RPP
      3        0.9014      0.9128    VITE CONS RPP
      2        0.8935      0.9017    VITE RPP
      3        0.8797      0.8936    VITE CYLIN VOLU
      4        0.8753      0.8944    VITE CONS CYLIN VOLU

                      C(p) Selection Method

Number in
  Model          C(p)    R-Square    Variables in Model
      4        4.0110      0.9367    VITE CYLIN VOLU RPP
      3        4.5352      0.9291    VITE CYLIN RPP
      5        6.0000      0.9367    VITE CONS CYLIN VOLU RPP
      4        6.0801      0.9304    VITE CONS CYLIN RPP
      3        9.1865      0.9150    VITE VOLU RPP
      4        9.8120      0.9192    VITE CONS VOLU RPP
      3        9.9381      0.9128    VITE CONS RPP
      2       11.6012      0.9017    VITE RPP
      3       16.3120      0.8936    VITE CYLIN VOLU
      4       18.0189      0.8944    VITE CONS CYLIN VOLU
```

Le modèle obtenu avec les méthodes pas à pas, introduction et élimination progressive composé des variables indépendantes VITE, CYLIN et RPP, se classe 3ème selon le critère du meilleur R² ajusté, 2ème selon la statistique cp la plus faible.

2 Régression logistique binaire

Dans le cas de la régression logistique binaire, on considère une variable cible binaire (Y = 0 ou 1) ou nominale à k = 2 modalités et p variables explicatives (X_1 , X_2 , …, X_p) continues, binaires ou nominales. Ici, si p = 1 la régression logistique est simple ; si p > 1 elle est multiple.

L'objectif de la régression logistique est celui de toute régression : modéliser l'espérance conditionnelle E(Y/X=x). Elle consiste, avec la fonction de lien logit, à chercher $\pi(x) = Prob(Y=1/X=x)$ sous la forme :

$$Log[\ \pi(x) / (1 - \pi(x)\ ] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p.$$

La syntaxe de la procédure se présente comme suit :

procédure – commandes - options

```
proc logistic ;
class  variables_Xi ;    (indépendantes nominales);
model  variable_Y = variable_X1  variable_X2  …  variable_Xp  / <options >;
```

*Les variables_X_j (variables indépendantes) et la variable_Y (variable dépendante binaire ou nominale à 2 groupes) de la commande **model** permettent l'identification des variables du modèle.*

Exemple d'application :

Les données de cet exemple (table : neuralgia) proviennent d'une étude sur les effets analgésiques de traitements testés auprès de n = 60 patients âgés souffrants de névralgies. Elles concernent l'atténuation ou non de la douleur après avoir reçu un traitement. On dispose de p = 2 variables quantitatives explicatives : l'âge du patient et la durée pendant laquelle le patient s'est plaint de douleurs avant le début du traitement ainsi que de m = 2 variables explicatives qualitatives : le sexe (F, M) et le traitement testé (A, B, Placebo). On dispose de ces mesures selon la variable cible ou de groupe à expliquer - douleur après traitement (Yes_{25} , No_{35}).

Outre le fait de comparer les deux traitements tests et un placebo, l'objectif est de mettre en évidence les caractéristiques mixtes qui différencient au mieux ces deux groupes de patients.

Programme 2.1:

```
Data exlogistic;  /* Régression logistique binaire - options */
proc logistic data=biblio.Neuralgia DESCENDING;
       class treatment sex;
       model pain= treatment sex age duration  / OUTROC=resroc  roceps=0   LACKFIT  LINK=logit;
       output out=outp p=pred; ods output association=assoc;
       run;
/* Courbe de ROC - Calcul de l'aire sous la courbe de ROC */
        data _null_;
       set assoc;
       if label2='c' then call symput("area",cvalue2);
       run;

title  "Courbe de ROC - Model : pain = treatment sex age duration";
title1 "Aire sous la courbe de ROC = &area";
SYMBOL1  I=JOIN C=RED V=NONE;
PROC GPLOT data=resroc;
       plot _sensit_*_1mspec_=1 / vaxis=0 to 1 by .1;
    run;
title2 "Matrice de confusion";  /* Estimation optimiste du taux de mal classés */
           data prev;
           set outp (keep=pain pred);
           if pred ge 0.5 then predy='Yes';
                          else predy='No';
                                          run;
proc freq data=prev;
tables pain*predy / nocol norow nopercent;
run;
```

Quelques options et instructions du modèle :

- *DESCENDING précise de modéliser la probabilité que $Y_i = 1$ (modalité de référence pour la variable à expliquer) et non pas la probabilité que $Y_i = 0$.*
- *OUTROC= resroc sauvegarde l'information nécessaire pour tracer la courbe ROC.*
- *LACKFIT demande d'effectuer le test d'adéquation de Hosmer-Lemeshow.*

- *LINK= logit , choix du modèle logit par défaut (lien avec l'analyse discriminante). Autres choix : probit, cloglog, glogit.*

On utilise la procédure GPLOT pour faire tracer la courbe ROC à partir des informations de l'option OUTROC= de la procédure LOGISTIC.

Les principaux résultats de la procédure PROC LOGISTIC pour une analyse de régression logistique binaire sont présentés en plusieurs parties.

Résultats du programme 2.1:

The LOGISTIC Procedure

Model Information

Data Set	BIBLIO.NEURALGIA
Response Variable	Pain
Number of Response Levels	2
Model	binary logit
Optimization Technique	Fisher's scoring
Number of Observations Read	60
Number of Observations Used	60

Response Profile

Ordered Value	Pain	Total Frequency
1	Yes	25
2	No	35

Probability modeled is Pain='Yes'.

Class Level Information

Class	Value	Design Variables	
Treatment	A	1	0
	B	0	1
	P	-1	-1
Sex	F	1	
	M	-1	

Model Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Les variables explicatives qualitatives sexe et traitement ne peuvent pas être introduites telles quelles dans le modèle logistique. Un codage de ces variables est nécessaire afin de simplifier l'interprétation des résultats. Une des modalités de la variable qualitative (généralement la plus fréquente, mais on peut également la préciser dans le programme) sera considérée comme une modalité de référence. Ainsi, il y aura création d'autant de variables que de modalités moins une. Le profil de réponse ci-dessus, montre les modalités de référence des variables : à expliquer "douleur" et explicatives "traitement" et "sexe" qui sont respectivement 'Yes', 'P' et 'H'.

On a ensuite, ci-dessous, les statistiques d'ajustement du modèle, le test de l'hypothèse nulle globale et l'analyse des estimations de la vraisemblance maximum, mis en oeuvre à l'aide de la procédure LOGISTIC de SAS.

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	83.503	60.736
SC	85.598	73.302
-2 Log L	81.503	48.736

Testing Global Null Hypothesis: BETA=0

Test	Khi 2	DF	Pr > Khi 2
Likelihood Ratio	32.7675	5	<.0001
Score	25.6666	5	0.0001
Wald	14.4512	5	0.0130

Les tests du rapport de vraisemblance, du score ou de Wald présentés conduisent tous au rejet de l'hypothèse de nullité de l'ensemble des coefficients.

Type 3 Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
Treatment	2	12.5310	0.0019
Sex	1	5.2946	0.0214
Age	1	7.2977	0.0069
Duration	1	0.0315	0.8591

Ainsi, parmi les quatre variables explicatives mixtes (qualitatives : Traitement et Sexe, continues : Age et Durée) introduites dans le modèle et avec un risque d'erreur classique de 5%, on constate seule la variable continue Durée n'a pas d'apport marginal significatif dans ce modèle (Pr > Khi2 = 0.8591 > 5%). Elle ne différencie donc pas les deux groupes de patients.

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	18.7872	6.9653	7.2752	0.0070
Treatment A	1	0.8849	0.5291	2.7969	0.0944
Treatment B	1	1.4118	0.6079	5.3933	0.0202
Sex F	1	0.9161	0.3981	5.2946	0.0214
Age	1	-0.2621	0.0970	7.2977	0.0069
Duration	1	0.00586	0.0330	0.0315	0.8591

Les résultats ci-dessus présentent les effets des variables continues et des modalités des variables qualitatives sur la variable cible 'douleur' : on constate que ni la Durée ni le Traitement-A ne différencient significativement les deux groupes de patients (Pr > Khi2 > 5%).

<u>Conclusion</u> : les patients n'ayant pas ressenti de douleur sont des femmes moins âgées qui ont bénéficié du traitement B. En revanche, le groupe de patients qui ont ressenti des douleurs sont des hommes plus âgés ayant reçu un placebo.

Association of Predicted Probabilities and Observed Responses

Percent Concordant	90.5	Somers' D	0.810
Percent Discordant	9.5	Gamma	0.810
Percent Tied	0.0	Tau-a	0.401
Pairs	875	c	0.905

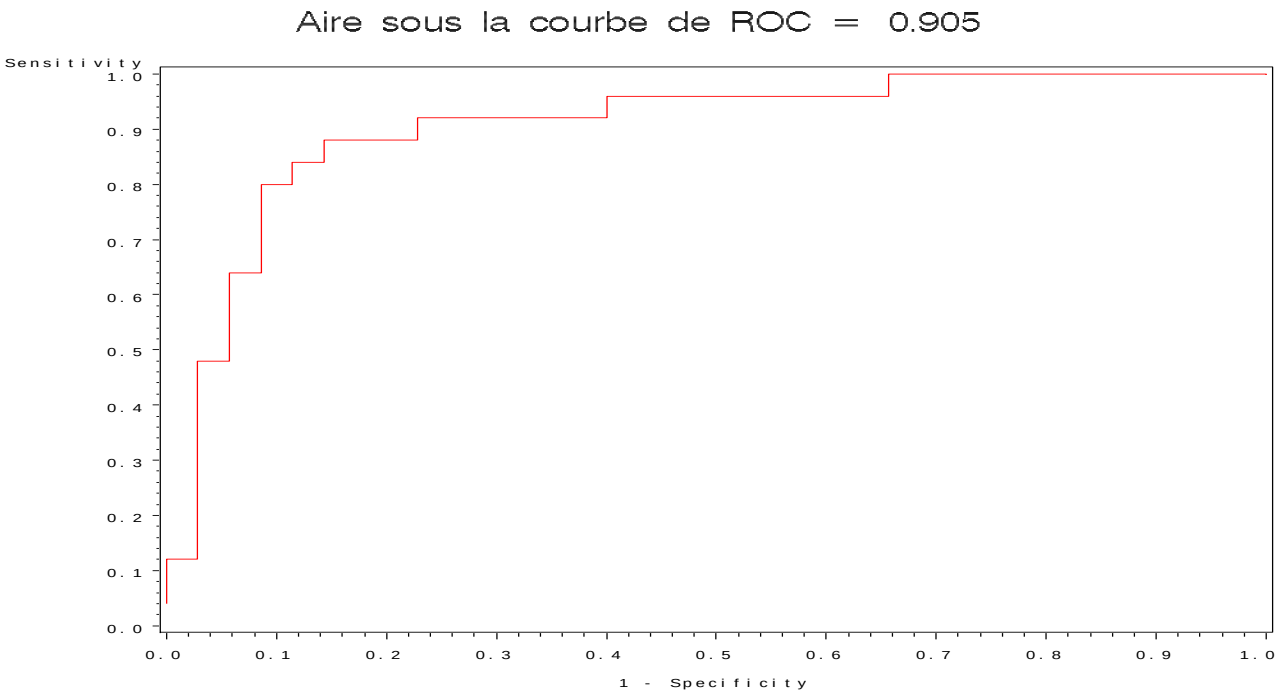
Les tests de concordance ci-dessus supposent que la variable cible Y prend les valeurs 0 et 1, et on note n_1 (resp. n_2) le nombre d'observations où Y=0 (resp. Y=1) , et $n = n_1 + n_2$ le nombre total d'observations. On note $t = n_1 n_2$ le nombre de paires formées d'une observation où Y=1 et d'une observation où Y=0. Parmi ces t paires : on a concordance si la probabilité estimée que Y=1 est plus grande quand Y=1 que quand Y=0. On note également nc le nombre de paires concordantes (ici 90.5% des paires) ; nd le nombre de paires discordantes ; t –nc-nd le nombre d'ex-aequo (tied) :

- D de Somers = (nc - nd) / t
- Gamma= (nc - nd) / (nc + nd)
- Tau-a = 2(nc - nd) / n(n - 1)
- c = (nc + 0.5 (t - nc - nd)) / t

Un modèle est d'autant meilleur que ces indices sont plus proches de 1. La quantité c n'est autre que l'aire sous la courbe ROC et que le D de Somers n'est autre que l'indice de Gini.

La courbe ROC (Receiver Operating Characteristic) permet d'étudier les variations de la spécificité et de la sensibilité d'un test pour différentes valeurs du seuil de discrimination. Le terme de courbe ROC peut être envisagé comme une "courbe de caractéristiques d'efficacité".

Courbe de ROC - Model : pain = treatment sex age duration
Aire sous la courbe de ROC = &area



L'aire sous la courbe ROC = 0.905 est bon indicateur global de performance, le modèle étant d'autant précis que l'air est proche de 1.

La matrice de confusion suivante permet de déterminer le pourcentage de bien classés par le modèle des observations de l'échantillon de base ou d'apprentissage.

La procédure FREQ
Table of Pain by predy

Pain Frequency	predy No	Yes	Total
No	31	4	35
Yes	5	20	25
Total	36	24	60

Pour l'exemple considéré, on a un bon pourcentage de bien classés par le modèle : (31 + 20) / 60 = 85%.

Le test de Hosmer et Lemeshow consiste à tester l'hypothèse nulle H_0 est que les fréquences observées sont celles attendues. La règle de décision de ce test est telle que si la probabilité ***Pr > Khi2*** *est supérieure au risque α, alors on accepte l'hypothèse nulle H_0 et on conclut que le modèle s'ajuste bien aux données.*

Hosmer and Lemeshow Goodness-of-Fit Test

Khi 2	DF	Pr > Khi 2
8.4981	8	0.3864

Pour l'exemple considéré, la probabilité ***Pr > Khi2*** *= 0.3864 est inférieure à α = 5% (non-rejet de l'hypothèse nulle H_0) ; le modèle s'ajuste donc bien aux données.*

L'odds-ratios 'rapports de cotes' d'une variable explicative mesure l'évolution des probabilités d'apparition de l'événement Y = 1 contre Y = 0. L'examen du tableau ci-dessous montre que :

- *Une variable explicative binaire a un seul odds-ratio. Si l'on s'intéresse à l'apparition de la douleur après traitement (Y=1 'Yes'), un odds-ratio de 0.160 pour la variable « sexe » (=1 pour Femme) signifie que le rapport des patients qui ont ressenti des douleurs aux patients qui n'ont pas ressenti de douleurs après traitement est de 0.160 moins important pour les femmes que pour les hommes.*
- *Regardant maintenant l'influence d'une variable explicative continue, l'Age par exemple, un odds-ratio de 1.30 signifie que le ratio des patients qui ont ressenti des douleurs après traitement sur ceux qui n'ont pas ressenti de douleur est multiplié par 1.30 chaque fois que l'âge augmente d'une année.*
- *Une variable explicative qualitative a autant d'odds-ratios que de modalités moins une. C'est le cas dans cet exemple, de la variable Treatment. Si on veut comparer la probabilité de ressentir des douleurs après traitement avec le Treatment-A, le Treatment-B et le Placebo. Nous avons choisi 'P = Placebo' comme modalité de référence et obtenu les résultats consignés dans le tableau ci-dessous.*

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
Treatment A vs P	0.042	0.006	0.304
Treatment B vs P	0.025	0.003	0.229
Sex F vs M	0.160	0.034	0.762
Age	1.300	1.075	1.572
Duration	0.994	0.932	1.061

De façon générale, un odds-ratio < 1 indique une influence négative de la variable explicative sur la variable à prédire, et un odds-ratio > 1 indique une influence positive, c'est le cas de la variable Age.

Statistique exploratoire multivariée avec SAS

*L'objectif de cette partie est de donner aux étudiants des outils de la statistique exploratoire des données permettent d'utiliser les principales méthodes d'Analyse des Données « à la française » avec le logiciel SAS "**Statistical Analysis System**" ou "**Système d'Analyse Statistique**".*

Le module SAS/STAT inclut la plupart des techniques de statistique multidimensionnelles utilisables sur de volumineux fichiers de données dont les méthodes factorielles et les techniques de classification automatique de données.

Les procédures SAS de chaque méthode sont présentées puis illustrées par un, voire plusieurs exemples de programmes SAS dont la syntaxe et les options sont commentées et les résultats obtenus interprétés.

Des exercices d'application sont proposés à la fin du chapitre afin d'appliquer de nouveau les notions traitées sur des données réelles.

Plan détaillé du cours : Méthodes d'Analyse de données

Chapitre 5 – Méthodes factorielles "Mapping"

- *Procédure **princomp** : Analyse en composantes principales,*
- *Procédure **corresp** : analyse des correspondances simples et multiples*
- *Procédures **discrim**, **stepdisc** et **candisc** : Analyse discriminante (classement d'observations, efficacité, validité et conditions d'application de l'analyse),*

Chapitre 6 – Méthodes de Classifications – Typologie "Clustering"

- *Procédure **fastclust** : classification sur individus, nuées dynamiques,*
- *Procédure **cluster** : classification hiérarchique ascendante,*
- *Procédure **varclust** : classification (hiérarchique et partition) des variables,*
- *Procédure **tree** : l'arbre de classification – dendogramme*
- *Exercices d'application.*

Quelques références bibliographiques supplémentaires

1- *L'aide en ligne du logiciel SAS* http://support.sas.com/documentation/onlinedoc/

2- *Bouroche J.M., Saporta G. L'analyse des données, "Que sais-je?" N°1854 , PUF, 8ème édition (2002).*

Ce fascicule de poche constitue une excellente introduction à l'analyse statistique multidimensionnelle. Il met l'accent sur l'interprétation intuitive des idées et concepts en n'ayant presque aucun recours à la notation mathématique. Il accorde aussi beaucoup d'importance à l'interprétation correcte des résultats.

Chapitre 5 : Méthodes factorielles

5.1 PROC PRINCOMP : Analyse en composantes principales

La procédure PRINCOMP permet de réaliser une analyse en composantes principale (ACP) d'un ensemble de variables quantitatives (numériques).

La procédure ***proc princomp*** *permet d'éditer les valeurs propres, vecteurs propres et composantes principales (facteurs). Dans le cas d'une analyse en composantes principales en SAS, la syntaxe de la procédure de base* PROC PRINCOMP *se présente comme suit :*

```
———— procédure – commandes - options ————
   proc princomp  <options > ;
   by  <descending > variable;
   var     X1  X2  … Xj …. Xp    /  <options >;
   weight  variable ;
_____________________________________________
```

Les variables X_j sont toutes continues (numériques).

Programme 5.1.1:

```
data acp1;
/* Analyse en composantes principales - options */
set biblio.voitures;
title 'ACP';
proc princomp out=acpdata  outstat=statacp;
var prix cons vite cylin volu rpp;
run;
proc plot data=acpdata;        /* Représentations graphiques sur les premières composantes principales*/
title 'representation des individus-voitures';
plot prin1*prin2= name / overlay  box;
run;
title 'Maco-commande : representation des individus-voitures';
%plotit(data=acpdata, labelvar=no,
plotvars=prin2 prin1, color=black,colors=blue);
run;
```

Remarques et options de la procédure :

Par défaut, cette procédure effectue une ACP normée ou standardisée (les variables sont centrées et réduites) ; une analyse basée sur la matrice des corrélations. On peut ajouter l'option ***COV*** *à la ligne* ***proc princomp*** *si on préfère une analyse basée sur la matice de covariances (données non standardisés). Les résultats de la procédure* ***proc princomp*** *sont présentés en deux parties selon les options out=table SAS et outstat=table SAS.*

L'option ***out=acpdata*** *crée la table SAS ''acpdata'' contenant les données initiales et les composantes principales, L'option* ***outstat=statacp*** *crée la table SAS ''statacp'' contenant les moyennes, écart-types, corrélations ou covariances, valeurs et vecteurs propres. L'option prefix prefixe utilisé pour nommer les facteurs (PRIN par défaut).*

Résultats du programme 5.1.1 :

Observations	27
Variables	6

Simple Statistics

	PRIX	CONS	VITE	CYLIN	VOLU	RPP
Mean	319.3740741	7.137037037	154.2592593	1165.629630	901.4074074	18.64814815
StD	84.3837871	1.141236896	22.3530975	208.059745	307.4144358	5.52771829

Correlation Matrix

		PRIX	CONS	VITE	CYLIN	VOLU	RPP
PRIX	prix en francs belges	1.0000	0.8139	0.9142	0.8536	0.2185	-.7679
CONS	consommation & urbaine	0.8139	1.0000	0.7804	0.7966	0.2946	-.6825
VITE	vitesse & maximum	0.9142	0.7804	1.0000	0.8322	0.0220	-.9376
CYLIN	cylindrée	0.8536	0.7966	0.8322	1.0000	0.1124	-.7788
VOLU	volume maximum du coffre	0.2185	0.2946	0.0220	0.1124	1.0000	0.1020
RPP	rapport & poids-puissance	-.7679	-.6825	-.9376	-.7788	0.1020	1.0000

Eigenvalues of the Correlation Matrix

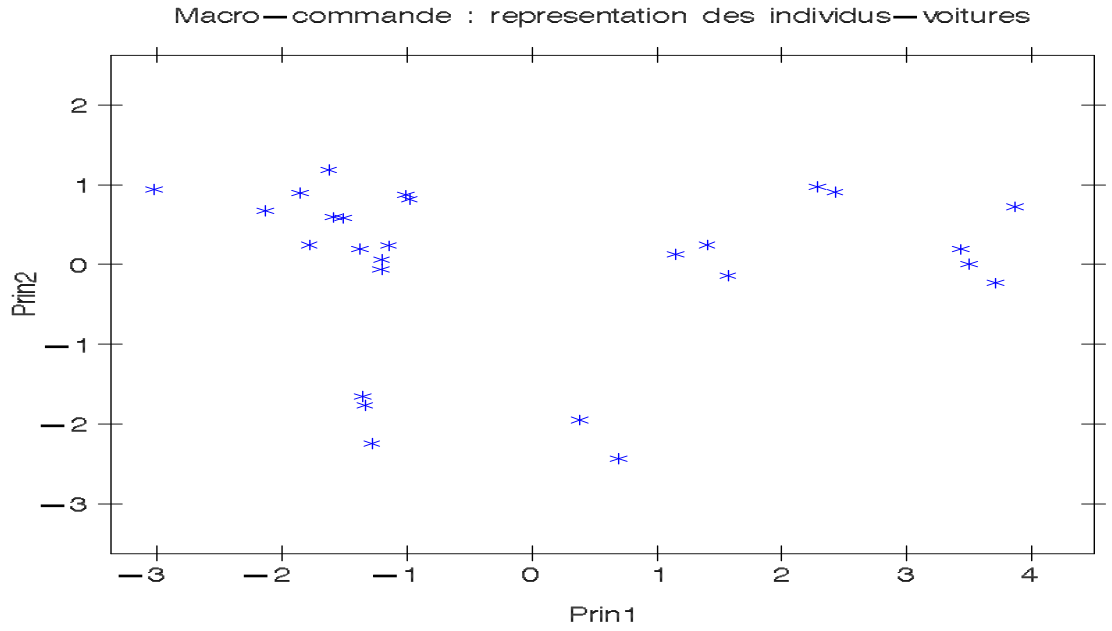
	Valeur propre	Différence	Proportion	Cumulée

1	4.28544731	3.17224886	0.7142	0.7142
2	1.11319845	0.86221378	0.1855	0.8998
3	0.25098467	0.06478271	0.0418	0.9416
4	0.18620196	0.03777827	0.0310	0.9726
5	0.14842369	0.13267976	0.0247	0.9974
6	0.01574393		0.0026	1.0000

Eigenvectors

		Prin1	Prin2	Prin3	Prin4	Prin5	Prin6
PRIX	prix en francs belges	0.457308	0.084400	-.055350	0.326208	0.698061	-.432414
CONS	consommation & urbaine	0.428281	0.207808	0.583801	-.656315	0.026693	-.033255
VITE	vitesse & maximum	0.465641	-.145239	-.349814	-.084124	0.200826	0.769614
CYLIN	cylindrée	0.445541	-.011247	0.407561	0.624587	-.483222	0.107926
VOLU	volume maximum du coffre	0.073776	0.920645	-.333362	0.034151	-.183886	0.029298
RPP	rapport & poids-puissance	-.432058	0.284401	0.506465	0.253941	0.452051	0.455081

Obs	NOM	PRIX	CONS	CYLIN	VITE	VOLU	RPP	LOGP	FAM	MAR	Prin1	Prin2	Prin3	Prin4	Prin5	Prin6
1	AS2	239.9	6.2	998	140	955	23.2	3.40	4CV	E	-1.78127	0.24629	-0.17349	-0.00284	-0.07796	0.23651
2	FI3	242.0	6.3	999	140	1088	21.8	3.64	4CV	E	-1.58888	0.59282	-0.39425	-0.09877	-0.25462	0.12077
3	FI5	269.5	6.2	999	145	968	21.5	3.64	4CV	E	-1.37856	0.19482	-0.43905	0.01912	0.06270	0.11878
4	FO1	261.0	7.0	1117	137	900	22.7	3.64	4CV	E	-1.14848	0.23569	0.51580	-0.04190	-0.19602	0.01711
5	NI1	248.0	6.4	988	140	375	17.0	3.64	4CV	E	-1.33832	-1.76463	-0.03519	-0.46582	-0.14314	-0.38172
6	OP1	261.0	7.2	993	143	845	22.4	3.62	4CV	E	-1.20373	0.05967	0.31346	-0.57163	0.15892	0.12360
7	SE9	219.3	7.3	903	131	1088	23.4	3.46	4CV	E	-1.85473	0.89820	0.23158	-0.94242	-0.14607	-0.01999
8	SZ2	242.3	6.4	993	145	400	18.4	3.58	4CV	E	-1.35777	-1.65618	0.00126	-0.42457	-0.05745	-0.06013
9	TO1	280.0	6.1	999	150	202	19.5	3.70	4CV	E	-1.28254	-2.24229	0.07207	-0.07857	0.48679	0.00238
10	CI4	265.5	5.6	954	145	1170	19.4	3.50	4CV	F	-1.50915	0.58090	-1.24296	0.13959	-0.17247	-0.02022
11	PE6	292.5	6.7	993	145	1151	20.8	3.61	4CV	F	-0.98050	0.82123	-0.47270	-0.20935	0.11189	-0.05695
12	PE1	265.2	6.8	954	134	1200	23.8	3.70	4CV	F	-1.62630	1.18681	-0.08615	-0.30481	0.09617	-0.06728
13	RE1	259.6	6.3	956	115	950	33.1	3.67	4CV	F	-3.02270	0.94329	1.08619	0.43806	0.77289	0.06467
14	SZ3	293.1	6.5	1324	163	400	14.0	3.58	5CV	E	0.38274	-1.94839	-0.01735	0.43808	-0.60173	0.10584
15	TO3	337.0	6.8	1295	170	202	15.0	3.70	5CV	E	0.69127	-2.43529	0.24730	0.34580	0.09890	0.16156
16	PE3	315.6	5.8	1124	142	1200	21.4	3.70	5CV	F	-1.01017	0.87048	-0.64285	0.83508	-0.02951	-0.13037
17	RE3	276.1	6.3	1108	120	950	28.4	3.67	5CV	F	-2.13627	0.67727	0.86425	0.72341	0.21692	-0.15582
18	RE4	283.1	5.8	1108	143	915	20.6	3.59	5CV	F	-1.20559	-0.06234	-0.43276	0.58924	-0.14717	-0.03072
19	FI8	500.1	8.9	1301	200	968	11.0	3.64	6CV	E	3.49752	0.00319	-0.44030	-0.42494	0.96754	0.04428
20	FID	356.9	7.7	1302	165	968	16.0	3.64	6CV	E	1.15337	0.12607	0.04757	0.07601	-0.15302	0.02017
21	DA2	379.3	9.2	1360	170	1200	13.9	3.70	6CV	E	2.28585	0.97273	0.39158	-0.61544	-0.33293	-0.08686
22	FO9	345.0	7.9	1397	167	915	13.8	3.59	6CV	E	1.56827	-0.13946	0.16838	0.08569	-0.59767	0.00729
23	SE4	385.6	8.8	1461	175	1200	14.7	3.63	6CV	E	2.42779	0.90941	0.37572	-0.03991	-0.41440	0.18291
24	VW3	360.9	7.8	1272	170	1040	14.3	3.65	6CV	E	1.40264	0.24557	-0.27475	-0.14501	-0.18509	0.02025
25	RE7	434.8	9.3	1597	180	973	12.0	3.64	6CV	F	3.43401	0.19109	0.78617	0.10293	-0.35166	-0.08500
26	PE9	503.5	8.7	1580	190	1200	11.2	3.70	6CV	F	3.87007	0.72515	-0.07508	0.61336	0.13074	-0.12831
27	RE8	506.3	8.7	1397	200	915	10.2	3.59	6CV	F	3.71145	-0.23210	-0.37445	-0.04040	0.75747	-0.00277



*La procédure **proc princomp** ne fournit que quelques résultats principaux de l'ACP sans aides à l'interprétation (contributions, corrélations variable-facteurs, représentations graphiques des variables, etc.). Des macros-commandes SAS ont donc été écrites pour compléter et aider à l'interprétation des résultats.*

Programme 5.1.2:

```
data acp2;
title ' Macros-commandes : ACP';
/* Macro-commande : Analyse en composantes principales  -  procédure princomp */
%acp(voitures, nom, prix cons vite cylin volu rpp,q=6);
%gacpsx;      /* Macro-commande : Représentation des valeurs propres */
%gacpvx;     /* Macro-commande : Représentation des variables - cercle de corrélations */
%gacpix;    /* Macro-commande : Représentation des individus */
run;
```

*Toutes les macros-commandes (programme précédé du symbole **%**) contenues dans le programme 5.1.2. doivent être compilées avant d'exécuter ce dernier. La macro-commande :*

***%acp (table, varind, liste de variables, <red=cov>, poids=,q=)** : lance l'exécution de la procédure princomp avec les paramètres : le nom de la table SAS (sasuser. est pris par défaut), le nom de la variable contenant les identificateurs des individus ''varind'', la liste des variables à analyser, la non réduction éventuelle des variables (par défaut, ACP normée, les variables sont centrées et réduites), **poids=**la variable de pondération (par défaut, tous égaux à 1/n), le nombre de composantes calculées (par défaut 3). On obtient ainsi les valeurs propres et les statistiques élémentaires. Calcule et affiche les composantes principales, les qualités de représentation des individus et leurs contributions pour les axes factoriels ainsi que les coordonnées des variables, les corrélations variable-facteur, les qualités de représentation.*

***%gacpsx** : trace l'éboulis des valeurs propres.*

***%gacpvx (x=axeh, y=axev, nc=nbcar, coeff=1)** : trace la représentation des variables centrées-réduites avec le cercle des corrélations. Les paramètres indiquent les numéros des axes factoriels (le premier plan principal par défaut), le nombre max de caractères pour afficher les identificateurs (4 par défaut) et un coefficient pour modifier la taille des caractères.*

***%gacpix (x=axeh, y=axev, nc=nbcar, coeff=1)** : trace la représentation des individus.*

Le programme suivant fournit en plus les résultats numériques et graphiques suivants,

Résultats du programme 5.1.2 :

Coordonnees des individus contributions et cosinus carres

ident	poids	Prin1	Prin2	Prin3	Prin4	Prin5	Prin6	contg	cont1	cont2
AS2	0.04	-1.82	0.25	-0.18	0.00	-0.08	0.24	2.13	2.85	0.21
FI3	0.04	-1.62	0.60	-0.40	-0.10	-0.26	0.12	2.00	2.27	1.21
FI5	0.04	-1.40	0.20	-0.45	0.02	0.06	0.12	1.38	1.71	0.13
					:					
					:					
PE9	0.04	3.94	0.74	-0.08	0.63	0.13	-0.13	10.20	13.44	1.82
RE8	0.04	3.78	-0.24	-0.38	-0.04	0.77	0.00	9.32	12.36	0.19

cont3	cont4	cont5	cont6	cosca1	cosca2	cosca3	cosca4	cosca5	cosca6
0.46	0.00	0.16	13.67	0.95	0.02	0.01	0.00	0.00	0.02
2.38	0.20	1.68	3.56	0.81	0.11	0.05	0.00	0.02	0.00
2.95	0.01	0.10	3.45	0.88	0.02	0.09	0.00	0.00	0.01
					:				
					:				
0.09	7.77	0.44	4.02	0.94	0.03	0.00	0.02	0.00	0.00
2.15	0.03	14.87	0.00	0.95	0.00	0.01	0.00	0.04	0.00

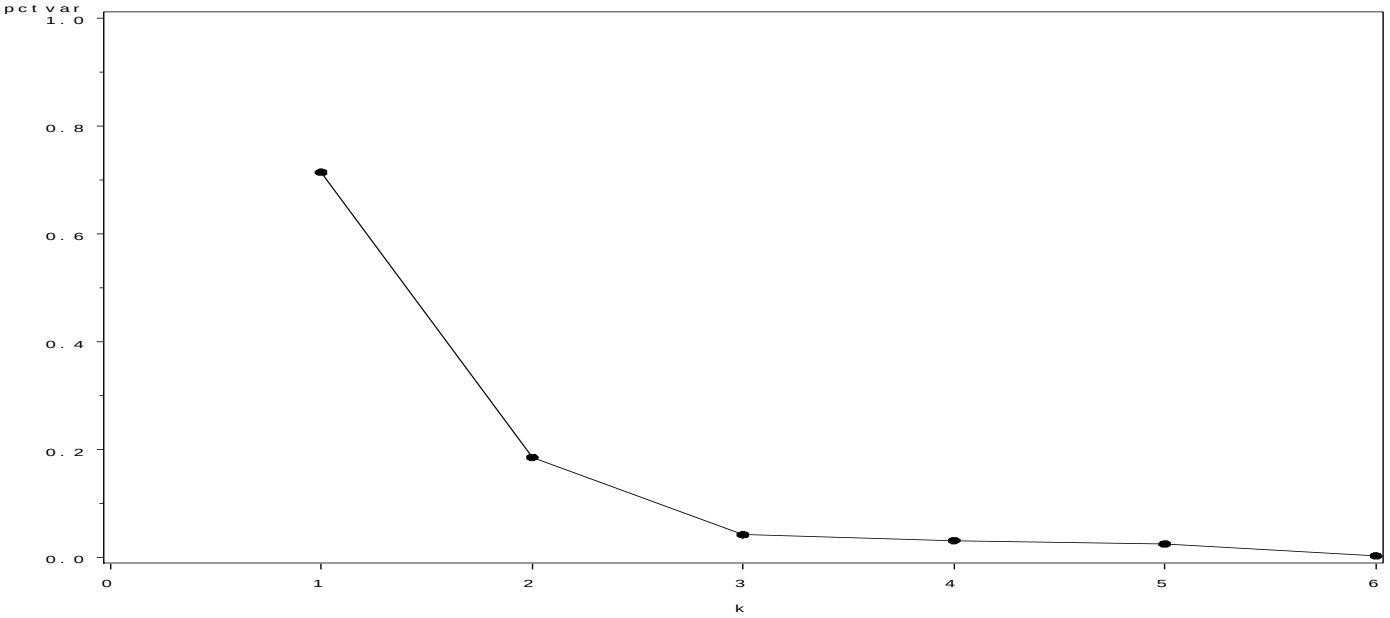
Coordonnees des variables (isométrique colonnes)

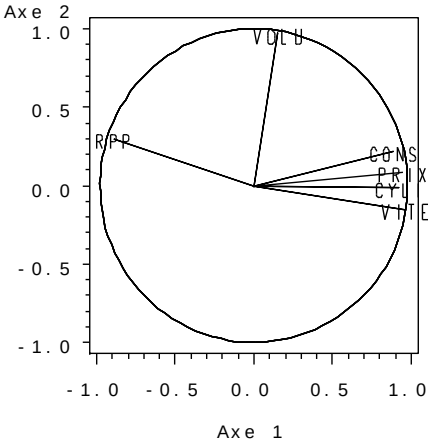
NAME	_LABEL_	v1	v2	v3	v4	v5	v6
PRIX	prix en francs belges	0.95	0.09	-0.03	0.14	0.27	-0.05
CONS	consommation & urbaine	0.89	0.22	0.29	-0.28	0.01	0.00
VITE	vitesse & maximum	0.96	-0.15	-0.18	-0.04	0.08	0.10
CYLIN	cylindrée	0.92	-0.01	0.20	0.27	-0.19	0.01
VOLU	volume maximum du coffre	0.15	0.97	-0.17	0.01	-0.07	0.00
RPP	rapport & poids-puissance	-0.89	0.30	0.25	0.11	0.17	0.06

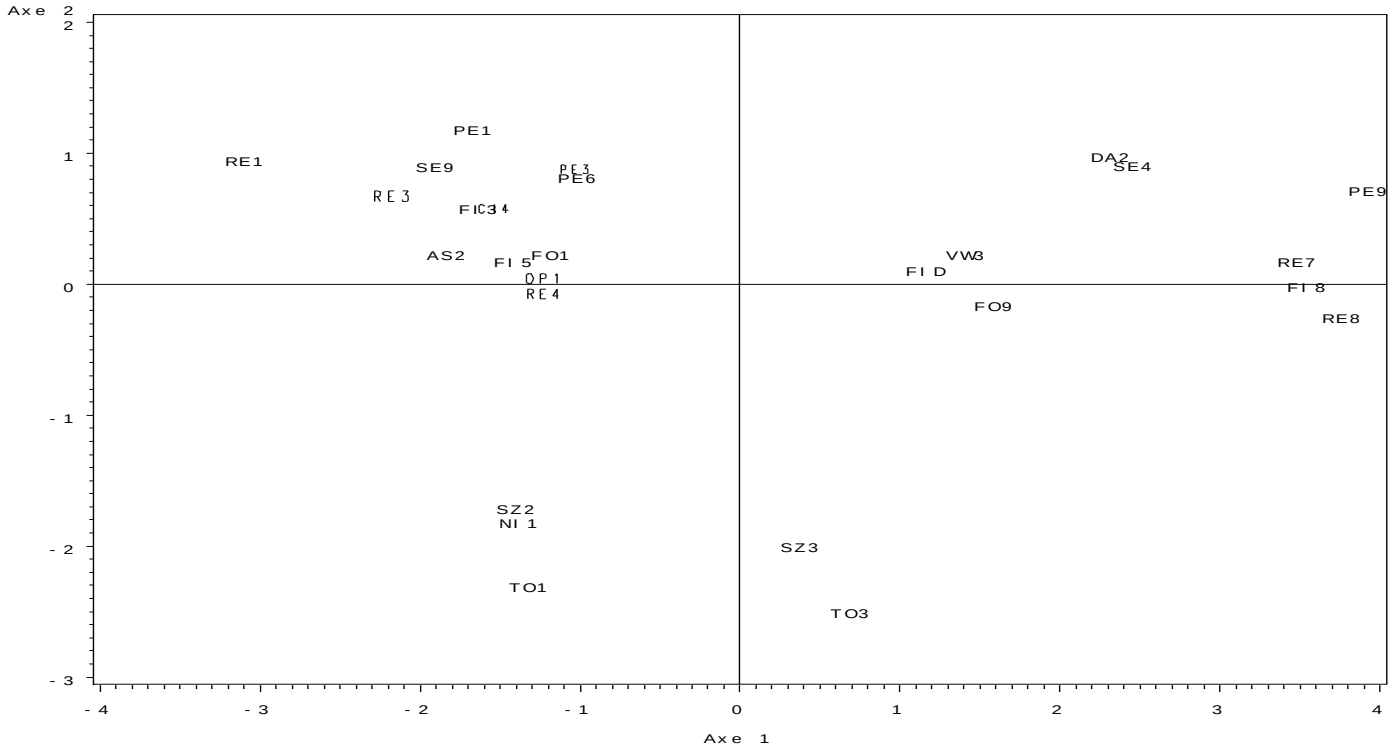
Correlations variables x facteurs

NAME	v1	v2	v3	v4	v5	v6

PRIX	0.95	0.09	-0.03	0.14	0.27	-0.05
CONS	0.89	0.22	0.29	-0.28	0.01	0.00
VITE	0.96	-0.15	-0.18	-0.04	0.08	0.10
CYLIN	0.92	-0.01	0.20	0.27	-0.19	0.01
VOLU	0.15	0.97	-0.17	0.01	-0.07	0.00
RPP	-0.89	0.30	0.25	0.11	0.17	0.06







5.2 PROC CORRESP : Analyse des Correspondances

La procédure PRINCORRESP réalise des analyses des correspondances simples (AFC) ou multiples (AFCM) à partir d'un tableau de variables qualitatives, d'un tableau disjonctif complet, d'une table de contingence (AFC) ou d'un tableau de Burt (AFCM).

Cette procédure édite la plupart des résultats usuels : différentes options de coordonnées pour les modalités, les contributions à la dispersion, individus et variables supplémentaires, … etc. Les options possibles pour cette procédure sont très nombreuses mais toutes leurs combinaisons ne sont pas intéressantes ni même autorisées. Les deux variables (AFC) ou l'ensemble des variables (AFCM) sont qualitatives (alphabétiques). La syntaxe de la procédure ***PROC CORRESP*** *se présente comme suit :*

```
———— procédure – commandes - options ————
     proc corresp  <options > ;
     /* Une des commandes suivantes */
     tables  variables lignes, <variables colonnes> ;
     var  liste des variables
     /* Commandes optionnelles */
     by  <descending > variable;
     id  variable ;  (contenant les identificateurs des modalités lignes lorsque la commande var est utilisée)
     supplementary  liste de variables (supplémentaires) ;
     weight  variable (pondération) ;
——————————————————————————————————
```

Remarques :

- *La commande* ***var*** *est utilisée lorsque les données sont sous la forme d'un tableau de contingence (AFC) ou d'un tableau de Burt (AFCM en précisant les options* ***mca et nvars=****). Elle est associée à la commande* ***id*** *(identificateur) qui spécifie les noms des modalités de la variable ligne du tableau.*
- *La commande* ***tables*** *permet de créer la table de contingence ou le tableau de Burt à partir des variables citées avant de lancer l'analyse. On construit,*
  - *la table de contingence (AFC) par croisement de deux variables : tables variable ligne, variable colonne ;*
  - *le tableau de Burt symétrique (AFCM avec l'option mca) en listant les variables (sans virgule) : tables var1 var2 var3 …. varp ;*

  cela est équivalent à lister sans l'option ***mca*** *: tables var1 var2 var3 …. varp , var1 var2 var3 …. varp ;*

- *L'AFCM du tableau disjonctif complet avec le calcul des coordonnées des individus en construisant une table de contingence croisant une variable d'identificateur de chaque individu (ident) avec la liste des variables :* `tables ident, var1 var2 var3 …. varp ;`
- *Une autre option de la procédure corresp :* ***cross=row/column/both*** *permet de construire les variables obtenues par croisement 2 à 2 des modalités des variables citées en (ligne/colonne/les deux).*

- *Les coordonnées des modalités supplémentaires sont calculées mais elles ne participent pas à la détermination des facteurs principaux.*

Les options :

data = table SAS *indique le nom de la table lue ou, par défaut, la dernière créée.*
out = table SAS *crée la table SAS contenant les coordonnées des modalités pour les représentations graphiques ainsi que les aides à l'interprétation (contributions, qualités).*
mca *pour analyse des corespondance multiple – multiple correspondence analysis.*
dimens = *spécifie le nombre de facteurs à retenir.*
cp *édition des profils colonnes.*
rp *édition des profils lignes.*
nocolumn *pas d'édition des coordonnées des modalités colonnes.*
norow *pas d'édition des coordonnées des modalités lignes.*
observed *édition du tableau de contingence ou de Burt .*
noprint *pas d'édition.*

Les graphiques :

Ils s'obtiennent comme pour l'ACP par la construction d'une table d'annotations et en utilisant les coordonnées listées dans la table SAS, sortie de l'option ***outc****.*

Programme 5.2.1:

```
data afc1;
/* Analyse des correspondances simples - options */
set biblio.voitures;
title 'AFC : analyse de la dépendance entre le niveau d'éducation et le statut professionnel';
proc corresp outc=resultats  observed  rp cp;
tables  nedu , spro ;
run;
```

Programme 5.2.1bis :

```
data afc2;
title 'AFC : analyse de la dépendance entre le niveau d'éducation et le statut professionnel';
/* Analyse des correspondances simples à partir d'un tableau de contingence */
input nedu$  Bureau Manager Securite;
cards;
                Niv1            216            1           26          243
                Niv2            117            4            1          122
                Niv3             30           79            0          109
;
title 'AFC';
proc corresp data=afc2 outc=resultats  observed  rp cp;
var Bureau Manager Securite;
id nedu;
run;
```

Les deux programmes ci-dessus présentent les mêmes résultats. La première analyse est obtenue à partir des deux variables qualitatives de la table SAS `biblio.voitures` *, la deuxième analyse est obtenue à partir de la table SAS, table de contingence, lue directement avec la commande cards.*

AFC
The CORRESP Procedure

Contingency Table

	Bureau	Manager	Sécurit	Sum
Niv1	216	1	26	243
Niv2	117	4	1	122
Niv3	30	79	0	109
Sum	363	84	27	474

Row Profiles

	Bureau	Manager	Sécurit
Niv1	0.888889	0.004115	0.106996
Niv2	0.959016	0.032787	0.008197
Niv3	0.275229	0.724771	0.000000

Column Profiles

	Bureau	Manager	Sécurit
Niv1	0.595041	0.011905	0.962963
Niv2	0.322314	0.047619	0.037037
Niv3	0.082645	0.940476	0.000000

Inertia and Chi-Square Decomposition

Singular Value	Principal Inertia	Chi-Square	Percent	Cumulative Percent	19 38 57 76 95
0.78604	0.61786	292.868	95.44	95.44	**********
0.17172	0.02949	13.978	4.56	100.00	*
Total	0.64735	306.845	100.00		

Degrees of Freedom = 4

Row Coordinates

	Dim1	Dim2
Niv1	-0.4637	0.1333
Niv2	-0.3596	-0.2809
Niv3	1.4362	0.0172

Summary Statistics for the Row Points

	Quality	Mass	Inertia
Niv1	1.0000	0.5127	0.1844
Niv2	1.0000	0.2574	0.0828
Niv3	1.0000	0.2300	0.7329

Partial Contributions to Inertia for the Row Points

	Dim1	Dim2
Niv1	0.1784	0.3089
Niv2	0.0539	0.6888
Niv3	0.7677	0.0023

Indices of the Coordinates that Contribute Most to Inertia for the Row Points

	Dim1	Dim2	Best
Niv1	2	2	2
Niv2	0	2	2
Niv3	1	0	1

Squared Cosines for the Row Points

	Dim1	Dim2
Niv1	0.9237	0.0763
Niv2	0.6210	0.3790
Niv3	0.9999	0.0001

Column Coordinates

	Dim1	Dim2
Bureau	-0.3475	-0.0571
Manager	1.6896	0.0257
Sécurit	-0.5850	0.6869

Summary Statistics for the Column Points

	Quality	Mass	Inertia
Bureau	1.0000	0.7658	0.1467
Manager	1.0000	0.1772	0.7817
Sécurit	1.0000	0.0570	0.0716

Partial Contributions to Inertia for the Column Points

	Dim1	Dim2
Bureau	0.1496	0.0845
Manager	0.8188	0.0040
Sécurit	0.0316	0.9115

Indices of the Coordinates that Contribute Most to Inertia for the Column Points

	Dim1	Dim2	Best
Bureau	0	0	1
Manager	1	0	1
Sécurit	0	2	2

Squared Cosines for the Column Points

	Dim1	Dim2
Bureau	0.9737	0.0263
Manager	0.9998	0.0002
Sécurit	0.4204	0.5796

---

*On utilise la même procédure proc corresp pour une analyse multiple (plus de deux variables qualitatives) avec l'option **mca**.*

Programme 5.2.2:

```
data afcm;
/* Analyse des correspondances multiple - options */
set biblio.employes;
title 'AFCM  : signalétique des employés';
proc corresp outc=resultats  mca  observed ;
tables  sexe nedu spro nati ;
run;
```

Résultats du programme 5.2.2 :

AFCM : signalétique des employés
The CORRESP Procedure

Burt Table

	F	H	Niv1	Niv2	Niv3	Bureau	Manager	Sécurit	A	F
F	216	0	158	33	25	206	10	0	40	176
H	0	258	85	89	84	157	74	27	64	194
Niv1	158	85	243	0	0	216	1	26	65	178
Niv2	33	89	0	122	0	117	4	1	27	95
Niv3	25	84	0	0	109	30	79	0	12	97
Bureau	206	157	216	117	30	363	0	0	87	276
Manager	10	74	1	4	79	0	84	0	4	80
Sécurit	0	27	26	1	0	0	0	27	13	14
A	40	64	65	27	12	87	4	13	104	0
F	176	194	178	95	97	276	80	14	0	370

Column Profiles

	F	H	Niv1	Niv2	Niv3	Bureau	Manager	Sécurit	A	F
F	0.250000	0.000000	0.162551	0.067623	0.057339	0.141873	0.029762	0.000000	0.096154	0.118919
H	0.000000	0.250000	0.087449	0.182377	0.192661	0.108127	0.220238	0.250000	0.153846	0.131081
Niv1	0.182870	0.082364	0.250000	0.000000	0.000000	0.148760	0.002976	0.240741	0.156250	0.120270
Niv2	0.038194	0.086240	0.000000	0.250000	0.000000	0.080579	0.011905	0.009259	0.064904	0.064189
Niv3	0.028935	0.081395	0.000000	0.000000	0.250000	0.020661	0.235119	0.000000	0.028846	0.065541
Bureau	0.238426	0.152132	0.222222	0.239754	0.068807	0.250000	0.000000	0.000000	0.209135	0.186486
Manager	0.011574	0.071705	0.001029	0.008197	0.181193	0.000000	0.250000	0.000000	0.009615	0.054054
Sécurit	0.000000	0.026163	0.026749	0.002049	0.000000	0.000000	0.000000	0.250000	0.031250	0.009459
A	0.046296	0.062016	0.066872	0.055328	0.027523	0.059917	0.011905	0.120370	0.250000	0.000000
F	0.203704	0.187984	0.183128	0.194672	0.222477	0.190083	0.238095	0.129630	0.000000	0.250000

Inertia and Chi-Square Decomposition

Singular Value	Principal Inertia	Chi-Square	Percent	Cumulative Percent	7 14 21 28 35
					----+----+----+----+----+---
0.70879	0.50239	1289.03	33.49	33.49	**********
0.57358	0.32900	844.15	21.93	55.43	**********
0.54411	0.29606	759.63	19.74	75.16	**********
0.45948	0.21112	541.69	14.07	89.24	**********
0.32997	0.10888	279.37	7.26	96.50	*****
0.22924	0.05255	134.83	3.50	100.00	***
Total	1.50000	3848.69	100.00		

Degrees of Freedom = 81

Column Coordinates

	Dim1	Dim2
F	-0.6027	-0.6904
H	0.5045	0.5780
Niv1	-0.6458	-0.0226
Niv2	-0.1468	0.3100
Niv3	1.6039	-0.2966
Bureau	-0.4315	-0.1624
Manager	1.9567	-0.2513
Sécurit	-0.2861	2.9647
A	-0.4947	1.0994
F	0.1390	-0.3090

Summary Statistics for the Column Points

	Quality	Mass	Inertia
F	0.7031	0.1139	0.0907
H	0.7031	0.1361	0.0759
Niv1	0.4392	0.1282	0.0812
Niv2	0.0408	0.0643	0.1238
Niv3	0.7945	0.0575	0.1283
Bureau	0.6951	0.1915	0.0390
Manager	0.8382	0.0443	0.1371
Sécurit	0.5358	0.0142	0.1572
A	0.4085	0.0549	0.1301
F	0.4085	0.1951	0.0366

Partial Contributions to Inertia for the Column Points

	Dim1	Dim2
F	0.0824	0.1650
H	0.0690	0.1382
Niv1	0.1064	0.0002
Niv2	0.0028	0.0188
Niv3	0.2944	0.0154
Bureau	0.0710	0.0153
Manager	0.3376	0.0085
Sécurit	0.0023	0.3804
A	0.0267	0.2015
F	0.0075	0.0566

Indices of the Coordinates that Contribute Most to Inertia for the Column Points

	Dim1	Dim2	Best
F	2	2	2
H	0	2	2
Niv1	1	0	1
Niv2	0	0	2
Niv3	1	0	1
Bureau	0	0	1
Manager	1	0	1
Sécurit	0	2	2
A	0	2	2
F	0	0	2

Squared Cosines for the Column Points

	Dim1	Dim2
F	0.3041	0.3990
H	0.3041	0.3990
Niv1	0.4387	0.0005
Niv2	0.0075	0.0333
Niv3	0.7683	0.0263
Bureau	0.6089	0.0862
Manager	0.8246	0.0136
Sécurit	0.0049	0.5309
A	0.0688	0.3397
F	0.0688	0.3397

5.3 PROC CANDISC : Analyse factorielle discriminante

La procédure CANDISC « Canonical discriminant analysis » cherche à décrire la liaison entre une variable qualitative (à discriminer) et un ensemble de variables numériques (discriminantes). La syntaxe de la procédure ***PROC CANDISC*** *se présente comme suit :*

```
———————— procédure – commandes - options ————————
        proc candisc   <options > ;
        by  <descending > variable;
        class   variable qualitative (à discriminer - cible) ;
        var   liste des variables numériques (discriminantes)
        freq  variable ;
_____________________________________________________
```

Remarques :