

# Guide d'introduction à



---

*Les notions de base dans le cadre de la M.Sc.  
option intelligence d'affaires*

Eng Seng Tang

2006

## Nos remerciements

*La création de ce guide a été rendue possible grâce aux subventions accordées par le Fonds leadership, volet initiatives pédagogiques étudiants (FLIPE). Nous tenons grandement à les remercier.*

*Nous voulons également remercier M. François Bellavance, professeur titulaire à HEC Montréal, pour avoir dirigé la rédaction de cet ouvrage.*

# Table des matières

|   |           |
|---|-----------|
| <b>Introduction.....</b>  | <b>1</b>  |
| Objectif de ce guide .....  | 1         |
| Organisation du guide .....   | 1         |
| Les pré-requis à la compréhension optimale de ce guide .....                    | 2         |
| Les icônes utilisés dans le guide .....   | 2         |
| Conventions pour les syntaxes des programmes SAS .....                          | 2         |
| Exécution des programmes dans les exemples du guide .....                       | 3         |
| Le fichier utilisé dans cet ouvrage .....                                       | 3         |
| Mise en situation pour les sous-sections El Nino! .....                         | 4         |
| <b>Présentation du système SAS.....</b>   | <b>5</b>  |
| Qu'est-ce que SAS? .....  | 5         |
| Aperçu de la structure du système SAS et SAS BASE .....                         | 5         |
| L'environnement de travail – l'interface graphique SAS .....                    | 6         |
| <b>Les concepts fondamentaux de SAS.....</b>                                    | <b>8</b>  |
| Les fichiers de données SAS (Data set) .....                                    | 8         |
| Les tables, variables et observations .....                                     | 10        |
| Les valeurs manquantes dans SAS .....   | 11        |
| Les librairies SAS .....  | 12        |
| Navigation dans les librairies et fichiers de données SAS.....                  | 13        |
| Les programmes SAS .....  | 14        |
| Comment exécuter, examiner les résultats et sauvegarder un programme SAS.....   | 16        |
| Aperçu du débogage des programmes SAS – Fenêtre Log .....                       | 17        |
| <b>Étapes DATA - Traitement des données.....</b>                                | <b>22</b> |
| Qu'est-ce qu'une étape Data? .....  | 22        |
| Création de fichier de données SAS .....  | 23        |
| Instructions DROP et KEEP .....   | 27        |
| Création de nouvelles variables à l'aide d'expressions .....                    | 29        |
| Les opérateurs dans les expressions.....  | 31        |
| Les opérandes dans les expressions: constantes et variables .....               | 32        |
| Le traitement conditionnel des données.....                                     | 33        |
| <b>Étapes PROC - Analyses de données .....</b>                                  | <b>37</b> |
| Qu'est-ce qu'une étape PROC? .....  | 37        |
| Procédure CONTENTS – Examiner la zone descriptive d'un fichier de données ..... | 37        |
| PROC PRINT - Examiner la zone de données d'un fichier SAS .....                 | 38        |
| PROC SORT - Tri des données .....   | 41        |
| PROC TRANSPOSE -Transposition des données d'une table SAS .....                 | 42        |
| PROC MEANS, FREQ et UNIVARIATE - Statistiques descriptives .....                | 46        |
| PROC UNIVARIATE - Statistiques descriptives plus détaillées.....                | 50        |

|   |               |
|---|---------------|
| <b>Les procédures SAS – Création de graphiques.....</b>   | <b>53</b>     |
| Création de graphiques dans SAS.....  | 53            |
| PROC GCHART - Diagramme à barres et en pointes de tarte.....                                    | 53            |
| PROC GPLOT - Diagramme en nuage de points (ou de dispersion) .....                              | 55            |
| <br><b>Autres notions utiles .....</b>  | <br><b>58</b> |
| Importation de fichiers de données Excel dans SAS .....   | 58            |
| Importation de fichiers de données brutes dans SAS.....   | 59            |
| Utilisation des listes de variables.....  | 60            |
| Concaténation et fusion de tables .....   | 61            |
| Introduction à la manipulation des fichiers de données dans le module SAS Entreprise Miner..... | 64            |
| <br><b>Quelques documents à consulter .....</b>   | <br><b>73</b> |
| Livres à consulter.....   | 73            |
| Liens Internet .....  | 73            |

# Introduction

## Objectif de ce guide

Ce guide se veut une porte d'entrée à l'apprentissage du logiciel **SAS** et se donne comme objectif de simplifier sa compréhension et son utilisation. Vous utiliserez constamment ce logiciel durant vos études en intelligence d'affaires afin de traiter et analyser des données dans divers contextes d'application. Tout au long de cet ouvrage, vous apprendrez les bases d'utilisation de ce logiciel peu convivial et s'adressant encore à un nombre restreint d'utilisateurs spécialisés.

Il est très important de mentionner que les notions abordées dans ce guide ont été choisies afin de corroborer avec l'utilisation que vous faites de SAS dans vos cours. Elles vous permettront donc d'avoir une compréhension de base des manipulations qui y sont effectuées.

Notez également qu'il existe plusieurs versions de SAS et que celle que nous traitons dans ce guide est la version 8.2 tournant sous le système d'exploitation Windows XP. Les notions couvertes dans ce guide, par contre, ne changent pas ou peu d'une version à l'autre.

## Organisation du guide

Le guide se divise en six sections:

1. La première partie du guide se consacre à la **présentation et à l'explication du fonctionnement des concepts fondamentaux** nécessaires à une bonne utilisation du logiciel SAS.
2. et 3. Dans les deuxième et troisième sections, nous nous attardons aux éléments principaux de la programmation SAS c'est-à-dire les étapes **DATA** et **PROC**.
4. En quatrième partie, nous abordons la création de **graphiques** dans SAS.
5. En cinquième partie, nous traitons de quelques éléments très utiles.
6. Finalement, nous donnons des sources d'informations sur SAS.

## Les pré-requis à la compréhension optimale de ce guide

Lors de la rédaction de ce document, nous avons supposé que le lecteur est relativement à l'aise avec l'environnement de travail Windows. C'est-à-dire qu'il doit être capable de naviguer à l'intérieur de différentes fenêtres, de savoir comment copier ou déplacer des fichiers et de connaître les termes techniques reliés à l'utilisation de base de Windows. De plus, il est préférable que le lecteur ait des connaissances de base au niveau de la statistique descriptive et des opérateurs arithmétiques et logiques étant donné que certaines notions élémentaires de ces sujets seront abordées.

## Les icônes utilisés dans le guide

Vous noterez, à travers ce guide, la présence de différents icônes. Ceux-ci ont été utilisés afin de diriger votre attention sur des éléments ou informations précis. Voici ces icônes ainsi qu'une explication de leurs utilisations:



### **Remarque!**

L'icône **Remarque** apporte des précisions supplémentaires aux explications données.



### **El Nino!**

L'icône **El Nino!** identifie des sous-sections dédiées à la manipulation de l'interface de SAS et à la création de programme portant sur les éléments vu dans chacune des quatre premières sections du guide. Ces sous-sections sont réalisées à l'intérieur d'une mise en situation faisant utilisation du fichier de données **elnino.sas7bat** qui est fourni avec ce guide.



### **Important!**

L'icône **Important!** met l'accent sur des notions ou éléments que nous jugeons important de comprendre et retenir.

## Conventions pour les syntaxes des programmes SAS

Lors des présentations des syntaxes de programme SAS, nous utiliserons les normes d'écriture suivantes:

- Les mots clés de SAS sont en caractères **gras** et en majuscules.
- Les valeurs que vous devez inscrire sont en caractères *italiques*.
- Les éléments entre chevrons (< >) sont optionnels.

Voici un exemple :

```
DATA nom_librairie.nom_fichier;  
INPUT nom_variable-1 <$> nom_variable-2 <$> ... nom_variable-n;  
DATALINES;  
valeur_variable-1 valeur_variable-2...valeur_variable-n  
RUN;
```

*\*\* Bien que nous utilisons des majuscules et des minuscules pour la notation, le fait que vous écriviez en majuscule ou en minuscule n'a aucune incidence sur le programme.*

## Exécution des programmes dans les exemples du guide

Pour des fins d'apprentissage, les programmes des exemples dans les sections deux à quatre du guide ainsi que ceux dans les sous-sections **El Nino!** peuvent directement être copiés et exécutés dans la fenêtre **Editor** à la condition que ceux-ci soient exécutés dans l'ordre dans lesquels ils sont présentés. De plus, les manipulations de la sous-section **El Nino! 1** doivent être faites avant d'exécuter les programmes des sous-sections **El Nino!** suivantes.

## Le fichier utilisé dans cet ouvrage

Le fichier utilisé pour les sections **El Nino!** de ce guide provient des archives de l'**UCI** (*University of California, Irvine*) ayant pour but la découverte de connaissances dans les bases de données (*Knowledge discovery in database ou KDD*). Ce fichier en question contient des données portant sur le phénomène naturel **El Nino**. Ces données sont des mesures météorologiques océaniques et de surfaces collectées par une série de bouées placées dans l'océan Pacifique équatorial. La base de données originale contient **178 080** observations sur les **12** variables qui sont présentées dans la mise en situation qui suit un peu plus loin. Nous avons décidé de ne garder que **10 000** observations afin de réduire la taille du fichier et les temps de traitement. Ce nombre d'observations comble amplement les besoins de notre contexte d'utilisation.

Pour plus de détails, vous pouvez visiter le site de l'UCI en cliquant sur le lien Internet suivant: <http://kdd.ics.uci.edu/>

## Mise en situation pour les sous-sections El Nino!

Les sous-sections **El Nino!** de ce guide sont réalisées à l'intérieur du contexte que voici :

### *Les mystères d'El-Nino*

Baptisé en référence à Jésus dû au fait que celui-ci survient à Noël, El-Nino est un phénomène climatique qui perturbe, tous les trois à onze ans, le régime mondial des précipitations causant ainsi des sécheresses et des inondations dans certaines régions du monde.

Afin d'étudier ce phénomène, l'entreprise **ELC** a posé une série de bouées, qui effectuent des observations météorologiques, dans l'océan Pacifique.

De 1980 à 1997, ces bouées ont effectué 10 000 observations sur les variables suivantes :

| Variable         | Description des variables            |
|------------------|--------------------------------------|
| <b>Year</b>      | Année de l'observation               |
| <b>Month</b>     | Mois de l'observation                |
| <b>Day</b>       | Jour de l'observation                |
| <b>Date</b>      | Date complète de l'observation       |
| <b>Latitude</b>  | Latitude de l'observation            |
| <b>Longitude</b> | Longitude de l'observation           |
| <b>Zon_wind</b>  | Vent zonal (<0: Ouest et >0:Est)     |
| <b>Mer_wind</b>  | Vent méridional (<0: Sud et >0:Nord) |
| <b>Air_temp</b>  | Température de l'air                 |
| <b>S_S_Temp</b>  | Température à la surface de la mer   |

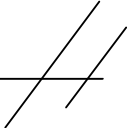
L'analyste en chef d'**ELC** a été chargé de préparer, à l'aide du logiciel SAS, la base de données contenant ces observations à des fins d'analyses statistiques. Par contre, il connaît peu de choses sur l'utilisation de ce logiciel et il se rapportera donc à vous afin de régler ses problèmes.





# Section 1

## Les notions essentielles

1. Présentation du système SAS
  2. Les éléments et concepts  
fondamentaux de SAS
- 

# Présentation du système SAS

## Qu'est-ce que SAS?

**SAS** (auparavant connu comme étant l'acronyme de *Statistical Analysis System*) est un système intégré de logiciels d'analyse de données produit par l'entreprise **SAS Institute**. Ce système a depuis dépassé les frontières de l'analyse statistique et offre désormais une gamme d'outils permettant de **gérer, stocker, modifier, extraire et analyser des données**.

## Aperçu de la structure du système SAS et SAS BASE

SAS est composé d'une multitude de modules ayant chacun leurs applications et spécialités spécifiques. Par exemple, le module **SAS STATS** se concentre sur les analyses de nature statistique, **SAS ENTREPRISE MINER** permet de réaliser du forage de données et **SAS OR** traite de recherche opérationnelle.

Au premier niveau de ces différentes composantes se trouve un module appelé **SAS BASE** et c'est précisément à celui-ci que nous nous attardons dans ce guide.

Le module **SAS BASE**, qui permet de gérer, de préparer les données ainsi que d'effectuer des analyses simples, comporte plusieurs éléments dont le **langage de programmation**, l'**interface graphique** et des **procédures** de SAS.

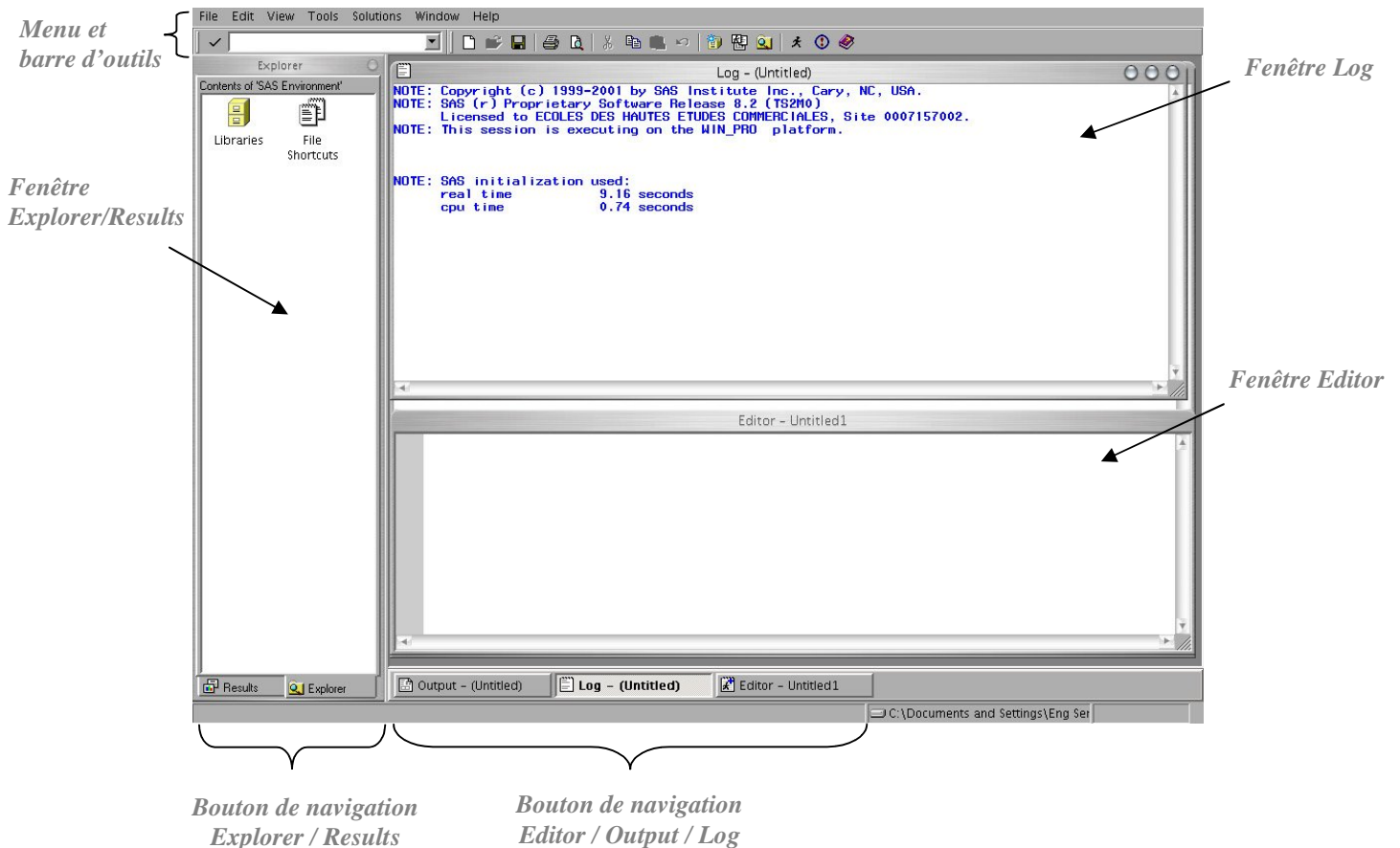
Présentons ce module en abordant l'environnement de travail de SAS c'est-à-dire son interface graphique.

## L'environnement de travail – l'interface graphique SAS

L'interface graphique de SAS est principalement composée des quatre fenêtres suivantes:

1. La fenêtre **Editor**: Écriture/Codage/Édition des programmes SAS
2. La fenêtre **Output**: Fenêtre d'affichage des résultats des programmes SAS
3. La fenêtre **Log**: journal rapportant des renseignements sur les manipulations effectuées
4. Les fenêtres **Explorer/Results**: Permet la navigation dans les fichiers de données et les sorties (Output) des programmes SAS.

### Interface graphique de SAS



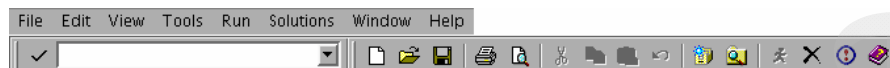
**\*\* Dans l'illustration, la fenêtre *Output* est cachée derrière les fenêtres *Editor* et *Log*.**

Bien sûr, nous retrouvons également un **menu** et une **barre d'outil** similairement à la majorité des logiciels sous **Windows**. Pour naviguer d'une fenêtre à l'autre, il suffit simplement d'utiliser les boutons de navigations qui se trouvent dans la partie inférieure de la fenêtre principale de SAS.

Il est important de noter que le menu de commande ainsi que les barres d'outils SAS sont **interactifs** et qu'ils changent selon la fenêtre qui est active, c'est-à-dire la fenêtre à l'intérieur de laquelle vous vous trouvez. Donc, si vous ne voyez pas la commande que vous utilisez habituellement, il est probable que vous ne soyez pas dans la bonne fenêtre.

Voici une illustration montrant le changement dans le menu et la barre d'outil lorsqu'on est respectivement dans la fenêtre **Editor** et la fenêtre **Explorer**.

*Menu - Fenêtre Editor*



*Menu - Fenêtre Explorer*



**Remarque!**

Il est également possible d'utiliser le menu **View** pour naviguer d'une fenêtre à l'autre ou d'ouvrir une fenêtre qui aurait été fermée. Il suffit de cliquer sur le menu **View** et de sélectionner la fenêtre à laquelle vous voulez aller. Par contre, lorsque vous sélectionnez la fenêtre **Editor** dans le menu **View**, SAS ouvrira une nouvelle fenêtre supplémentaire **Editor** afin d'éditer un nouveau programme SAS.

# Les concepts fondamentaux de SAS

Afin de faciliter la compréhension des éléments techniques de SAS, il nous est important de présenter les concepts clés se trouvant à la base du fonctionnement de ce logiciel. Ces concepts sont les **fichiers de données SAS (Dataset)**, les **tables**, les **variables**, les **observations**, les **librairies SAS** et les **programmes SAS**.

## Les fichiers de données SAS (Data set)

Les fichiers de données SAS sont simplement des fichiers de données qui sont sous un format propre à SAS. Nous les dénommons également par « **Data set** » dans le jargon de ce logiciel. Ces fichiers sont composés de deux parties. La première partie est la **zone descriptive** qui nous donne des informations sur les données contenues dans le fichier tel que le **nombre d'observations et de variables**, la **liste des variables** et la **date de création du fichier**.

### Zone descriptive du fichier de données El Nino

```

Data Set Name: ELC.ELNINO
Member Type: DATA
Engine: V8
Created: 17:54 Thursday, March 9, 2006
Last Modified: 17:54 Thursday, March 9, 2006
Protection:
Data Set Type:
Label:

Observations: 10000
Variables: 12
Indexes: 0
Observation Length: 96
Deleted Observations: 0
Compressed: NO
Sorted: NO

```

*Date de création /  
dernière modification*

```

-----Engine/Host Dependent Information-----
Data Set Page Size: 8192
Number of Data Set Pages: 120
First Data Page: 1
Max Obs per Page: 84
Obs in First Data Page: 62
Number of Data Set Repairs: 0
File Name: C:\Documents and Settings\Eng Seng\Desktop\FLIPE\Projet\EI
Nino Database\elnino.sas7bdat
Release Created: 8.0202M0
Host Created: WIN_PRO

```

*Nombre  
d'observations  
et de variables*

### -----Alphabetic List of Variables and Attributes-----

| #  | Variable  | Type | Len | Pos |
|----|-----------|------|-----|-----|
| 11 | air_temp  | Num  | 8   | 80  |
| 5  | date      | Num  | 8   | 32  |
| 4  | day       | Num  | 8   | 24  |
| 10 | humidity  | Num  | 8   | 72  |
| 6  | latitude  | Num  | 8   | 40  |
| 7  | longitude | Num  | 8   | 48  |
| 9  | mer_winds | Num  | 8   | 64  |
| 3  | month     | Num  | 8   | 16  |
| 1  | obs       | Num  | 8   | 0   |
| 12 | s_s_temp  | Num  | 8   | 88  |
| 2  | year      | Num  | 8   | 8   |
| 8  | zon_winds | Num  | 8   | 56  |

*Liste des  
variables*

\*\* Ces informations sont obtenues à l'aide de la procédure **CONTENTS** que nous aborderons dans la section 3 de ce guide.

La deuxième partie d'un fichier de données SAS est la **zone de données** où, bien sûr, les données sont entreposées.

*Zone de données du fichier de données El Nino (Illustration partielle)*

|    | obs | year | month | day | date   | latitude | longitude | zon_winds | mer_winds |
|----|-----|------|-------|-----|--------|----------|-----------|-----------|-----------|
| 1  | 1   | 80   | 3     | 7   | 800307 | -0.02    | -109.46   | -6.8      | 0.7       |
| 2  | 2   | 80   | 3     | 8   | 800308 | -0.02    | -109.46   | -4.9      | 1.1       |
| 3  | 3   | 80   | 3     | 9   | 800309 | -0.02    | -109.46   | -4.5      | 2.2       |
| 4  | 4   | 80   | 3     | 10  | 800310 | -0.02    | -109.46   | -3.8      | 1.9       |
| 5  | 5   | 80   | 3     | 11  | 800311 | -0.02    | -109.46   | -4.2      | 1.5       |
| 6  | 6   | 80   | 3     | 12  | 800312 | -0.02    | -109.46   | -4.4      | 0.3       |
| 7  | 7   | 80   | 3     | 13  | 800313 | -0.02    | -109.46   | -3.2      | 0.1       |
| 8  | 8   | 80   | 3     | 14  | 800314 | -0.02    | -109.46   | -3.1      | 0.6       |
| 9  | 9   | 80   | 3     | 15  | 800315 | -0.02    | -109.46   | -3        | 1         |
| 10 | 10  | 80   | 3     | 16  | 800316 | -0.02    | -109.46   | -1.2      | 1         |
| 11 | 11  | 80   | 3     | 17  | 800317 | -0.02    | -109.46   | -0.1      | 0.7       |
| 12 | 12  | 80   | 3     | 18  | 800318 | -0.02    | -109.46   | -1.2      | 2.3       |
| 13 | 13  | 80   | 3     | 19  | 800319 | -0.02    | -109.46   | -4.1      | -0.3      |
| 14 | 14  | 80   | 3     | 20  | 800320 | -0.02    | -109.46   | -4.8      | -0.8      |
| 15 | 15  | 80   | 3     | 21  | 800321 | -0.02    | -109.46   | -5.2      | 2         |
| 16 | 16  | 80   | 3     | 22  | 800322 | -0.02    | -109.46   | -2.7      | 2.7       |
| 17 | 17  | 80   | 3     | 23  | 800323 | -0.02    | -109.46   | -4.4      | 1.1       |
| 18 | 18  | 80   | 3     | 24  | 800324 | -0.02    | -109.46   | -4.3      | 0.7       |
| 19 | 19  | 80   | 3     | 25  | 800325 | -0.02    | -109.46   | -3.8      | 0.5       |
| 20 | 20  | 80   | 3     | 26  | 800326 | -0.02    | -109.46   | -3        | 0.2       |
| 21 | 21  | 80   | 3     | 27  | 800327 | -0.02    | -109.46   | -3.2      | -0.2      |
| 22 | 22  | 80   | 3     | 28  | 800328 | -0.02    | -109.46   | -1.9      | 0.7       |
| 23 | 23  | 80   | 3     | 29  | 800329 | -0.02    | -109.46   | -0.8      | 0.3       |
| 24 | 24  | 80   | 8     | 11  | 800811 | 0        | -109.56   | -3.3      | 1.5       |
| 25 | 25  | 80   | 8     | 12  | 800812 | 0        | -109.56   | -3.5      | 0.8       |
| 26 | 26  | 80   | 8     | 13  | 800813 | 0        | -109.56   | -4.9      | 1.9       |
| 27 | 27  | 80   | 8     | 14  | 800814 | 0        | -109.56   | -1.2      | 2.1       |

Comme tout fichier, les fichiers de données SAS ont des **extensions** qui leur sont propre sous l'environnement Windows. Afin que vous puissiez reconnaître ces fichiers, voici leurs extensions :

**Extension des fichiers de données SAS**

| Version de SAS            | Extension des fichiers de données ( <i>Dataset</i> ) |
|---------------------------|--|
| Version 8 et plus récente | <b>Sas7bdat</b> (ex: <i>elnino.sas7bdat</i> )        |
| Versions antérieures à 8  | <b>sd2</b> (ex : <i>elnino.sd2</i> )                 |

Les données contenues à l'intérieur d'un fichier de données SAS se présentent sous forme d'une **table**, concept que nous allons immédiatement aborder.

## Les tables, variables et observations

Les **tables** sont des **structures d'entreposage** bidimensionnelles (**Ligne/colonne**) permettant de stocker des données. Les bases de données sont généralement constituées d'une ou de plusieurs tables. Par exemple, le fichier de données **El Nino** sur lequel nous travaillons représente une base de données constituée d'une seule table. Revenons à l'illustration précédente montrant la zone de données du fichier de données **El Nino** où les données sont évidemment organisées **en lignes et colonnes**, donc sous forme d'une **table**. Les **colonnes d'une table de données représentent les variables** et les **lignes représentent les observations**.

|               |     |      | Variable | Variable |        |          |           |           |           |  |
|---------------|-----|------|----------|----------|--------|----------|-----------|-----------|-----------|--|
|               | obs | year | month    | day      | date   | latitude | longitude | zon_winds | mer_winds |  |
|               | 1   | 80   | 3        | 7        | 800307 | -0.02    | -109.46   | -6.8      | 0.7       |  |
|               | 2   | 80   | 3        | 8        | 800308 | -0.02    | -109.46   | -4.9      | 1.1       |  |
| Observation → | 3   | 80   | 3        | 9        | 800309 | -0.02    | -109.46   | -4.5      | 2.2       |  |
|               | 4   | 80   | 3        | 10       | 800310 | -0.02    | -109.46   | -3.8      | 1.9       |  |
|               | 5   | 80   | 3        | 11       | 800311 | -0.02    | -109.46   | -4.2      | 1.5       |  |
|               | 6   | 80   | 3        | 12       | 800312 | -0.02    | -109.46   | -4.4      | 0.3       |  |
|               | 7   | 80   | 3        | 13       | 800313 | -0.02    | -109.46   | -3.2      | 0.1       |  |
| Observation → | 8   | 80   | 3        | 14       | 800314 | -0.02    | -109.46   | -3.1      | 0.6       |  |
|               | 9   | 80   | 3        | 15       | 800315 | -0.02    | -109.46   | -3        | 1         |  |
|               | 10  | 80   | 3        | 16       | 800316 | -0.02    | -109.46   | -1.2      | 1         |  |
|               | 11  | 80   | 3        | 17       | 800317 | -0.02    | -109.46   | -0.1      | 0.7       |  |
|               | 12  | 80   | 3        | 18       | 800318 | -0.02    | -109.46   | -1.2      | 2.3       |  |
|               | 13  | 80   | 3        | 19       | 800319 | -0.02    | -109.46   | -4.1      | -0.3      |  |
|               | 14  | 80   | 3        | 20       | 800320 | -0.02    | -109.46   | -4.8      | -0.8      |  |
|               | 15  | 80   | 3        | 21       | 800321 | -0.02    | -109.46   | -5.2      | 2         |  |
|               | 16  | 80   | 3        | 22       | 800322 | -0.02    | -109.46   | -2.7      | 2.7       |  |
|               | 17  | 80   | 3        | 23       | 800323 | -0.02    | -109.46   | -4.4      | 1.1       |  |
|               | 18  | 80   | 3        | 24       | 800324 | -0.02    | -109.46   | -4.3      | 0.7       |  |
|               | 19  | 80   | 3        | 25       | 800325 | -0.02    | -109.46   | -3.8      | 0.5       |  |
|               | 20  | 80   | 3        | 26       | 800326 | -0.02    | -109.46   | -3        | 0.2       |  |
|               | 21  | 80   | 3        | 27       | 800327 | -0.02    | -109.46   | -3.2      | -0.2      |  |
|               | 22  | 80   | 3        | 28       | 800328 | -0.02    | -109.46   | -1.9      | 0.7       |  |
|               | 23  | 80   | 3        | 29       | 800329 | -0.02    | -109.46   | -0.8      | 0.3       |  |
|               | 24  | 80   | 8        | 11       | 800811 | 0        | -109.56   | -3.3      | 1.5       |  |
|               | 25  | 80   | 8        | 12       | 800812 | 0        | -109.56   | -3.5      | 0.8       |  |
|               | 26  | 80   | 8        | 13       | 800813 | 0        | -109.56   | -4.9      | 1.9       |  |
|               | 27  | 80   | 8        | 14       | 800814 | 0        | -109.56   | -1.2      | 2.1       |  |

Mais qu'est-ce que les variables et les observations? Le premier est un contenant servant à stocker des données d'un même type et pouvant prendre plusieurs valeurs différentes et le deuxième est un regroupement des valeurs d'une ou de plusieurs variables.

Prenons par exemple la base de données **El Nino**. La variable **Longitude** contient des enregistrements de l'emplacement longitudinal de la bouée d'observation lorsque celle-ci

a recueillie une observation. La variable **S\_S\_Temp**, (*qui n'apparaît pas dans les illustrations partielles de la table elnino se trouvant aux pages 9 et 10*), quant à elle, contient des valeurs de température à la surface de la mer. Ainsi, chacune des variables de la table contient des données d'un même type.

Lorsque la bouée effectue un enregistrement, elle capte des données sur la **longitude**, la **latitude**, la **température** et la **direction du vent**, c'est-à-dire une valeur pour chacune des variables de la table. Ces données recueillies lors d'un enregistrement forme une observation.

Au niveau des variables, celles-ci peuvent être de deux types différents: **caractère** ou **numérique**. Le type de variable détermine le type de valeur qui peut être contenue dans celle-ci. Ainsi, une variable de type caractère contient des valeurs sous forme de chaîne de caractères (*Lettre, chiffre ou caractère spécial*) et une variable de type numérique contient des valeurs sous forme de chiffre et de nombre.

La différence entre les chiffres et les nombres d'une variable de type caractère et d'une variable de type numérique est que ceux du type numérique sont des quantités tandis que ceux du type caractère sont simplement une suite de caractères. **Nous pouvons donc faire des calculs, comme une moyenne et un écart-type, avec les variables de type numérique mais pas avec les variables de type caractère.**

## Les valeurs manquantes dans SAS

Les valeurs manquantes sont, comme leur appellation l'indique, des valeurs qui sont manquantes pour des variables à l'intérieur d'une ou plusieurs observations. Les valeurs manquantes sont une réalité dans le monde des bases de données et peuvent représenter un grand problème à surmonter dans les analyses et l'interprétation des résultats

À l'intérieur de SAS, les valeurs manquantes sont représentées différemment selon le type de variable pour laquelle la valeur est manquante.



## Représentation des valeurs manquantes

| Type de variable | Valeur manquante |
|------------------|------------------|
| Caractère        | Cellule vide     |
| Numérique        | . (un point)     |

## Valeur manquante - Numérique

|       |       |
|-------|-------|
| 26.08 | 25.38 |
| 26.24 | .     |
| 26.05 | .     |
| 25.67 | .     |
| 25.39 | .     |

## Valeur manquante - Caractère

|   |
|---|
| B |
| h |
| . |
| h |

## Les librairies SAS

Une **librairie SAS** est simplement un contenant servant à stoker des fichiers de données SAS. Une façon de visualiser les choses est de considérer une librairie comme un tiroir où l'on met des fichiers de données.

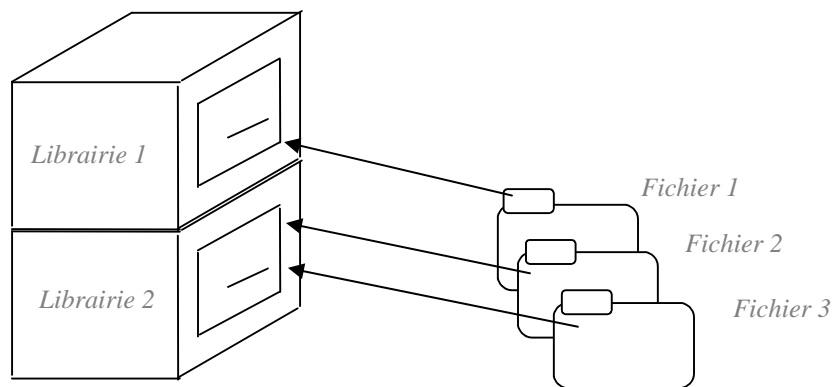
**Remarque!**

De façon concrète, dans le système de fichier Windows, une librairie est simplement un répertoire (**folder**) où vous conservez vos fichiers de données **sas7bdat** ou **sd2**. En d'autres termes, une librairie est la représentation logique dans SAS d'un répertoire de Windows.

Il existe **deux types de librairies**: la **librairie temporaire** dont les fichiers sont effacés lors de la fermeture de SAS et les **librairies permanentes** qui conservent les fichiers de données, dans un ou plusieurs répertoires que vous aurez spécifiés, même après la fermeture de SAS.

La librairie temporaire existe par défaut dans SAS et se nomme **WORK**. Ainsi, tous les fichiers de données qui se retrouvent dans cette librairie sont effacés à la fermeture de la session **SAS**. Toutes autres librairies, créés par défaut dans SAS ou par l'utilisateur sont des librairies permanentes.

### Schéma conceptuel des fichiers et des librairies



À l'intérieur des programmes SAS, que nous allons aborder un peu plus loin, la spécification d'un fichier de données SAS doit se faire de la façon suivante: **nom de librairie.nom de fichier**.

Par exemple, un fichier de données nommé **Analyse.sas7bdat** qui se trouve dans la librairie **Science** doit être spécifié de la façon suivante: **Science.Analyse** et un fichier de données nommé **Dimension.sas7bdat** qui se trouve dans la librairie temporaire **Work** doit être spécifié **Work.Dimension**.

#### **Important!**

*Dans le cas où vous omettez le nom de la librairie lorsque vous spécifiez un fichier dans un programme, SAS assigne automatiquement ce fichier à la librairie temporaire **Work**.*

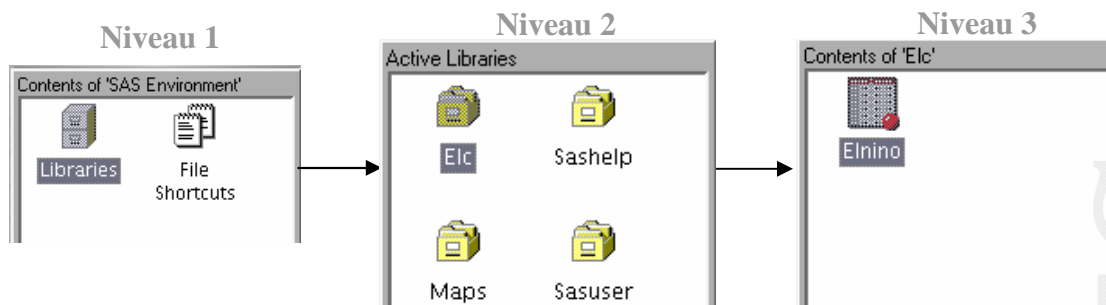
#### **Important!**

*Une librairie doit être créée avant que vous puissiez faire référence à elle dans un programme SAS et notez que la librairie temporaire **Work** est automatiquement créée lorsque vous ouvrez SAS.*


## Navigation dans les librairies et fichiers de données SAS

Il n'est pas très difficile, dans la fenêtre **Explorer**, de naviguer à travers les fichiers et les librairies. Premièrement, il faut afficher la liste des librairies disponibles en double-cliquant sur l'icône **Librairies** de la fenêtre **Explorer**. Par la suite, afin d'accéder aux

fichiers qui se trouve à l'intérieur d'une librairie, il suffit de double-cliquer sur l'icône de celle-ci.



Finalement, si vous double-cliquez sur l'icône d'un fichier de données, la table de données contenue dans ce fichier s'affichera **en mode lecture** dans une fenêtre **VIEWTABLE**. Vous ne pouvez donc pas apporter de modification à une table à travers une fenêtre **VIEWTABLE**. Remarquez que nous nous retrouvons devant une situation où il y a trois niveaux différents. Bien sûr, l'illustration ci-dessus porte sur le fichier de données **elnino** (illustré à la page 9) se trouvant dans la librairie **ELC**.

Si vous vous trouvez au niveau des fichiers d'une librairie et que vous voulez revenir à la liste des librairies, il suffit simplement de cliquer sur le bouton **Up one level**  que vous trouvez sur la barre d'outil lorsque vous êtes dans la fenêtre **Explorer**. En cliquant une deuxième fois sur ce même icône, vous reviendrez à la fenêtre où se trouve l'icône **Librairies**.

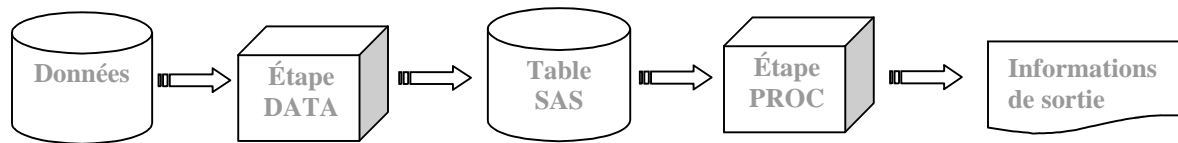
## Les programmes SAS

Les programmes SAS sont des ensembles d'instructions ou séquences d'étapes qui servent à traiter des données et à en extraire de l'information. Un programme SAS est généralement composé de quatre éléments:

1. Les données d'entrée (Table SAS ou « données brutes »)
2. Une étape **DATA** (Traite, manipule les données)
3. Une étape **PROC** (Analyse les données)
4. Les informations de sortie (Résultats)

Voici un schéma illustrant ces éléments et l'ordre habituel dans lequel ils sont traités :

### Schéma de traitement des données en information



*\*\* Les données d'entrée peuvent être des tables SAS ou des données brutes*

La rédaction d'un programme SAS doit se faire selon certaines règles strictes:

- Les étapes **DATA** et **PROC** commencent respectivement par une instruction **DATA** et **PROC**.
- Une étape se termine par l'instruction **RUN**.
- Chaque instruction d'un programme doit commencer par un **mot clé**, mot spécifique du langage de programmation SAS ayant une fonction précise et prenant la couleur **bleu**.
- Chaque instruction doit se terminer par un **point-virgule (;)**. Les instructions peuvent être étalés sur plusieurs lignes. SAS reconnaît la fin d'une instruction par le **point-virgule (;)**.

#### Remarque!

*Bien que nous présentons l'ordre général dans lequel se suivent les éléments d'un programme SAS, notez que les étapes, qu'elles soient des étapes **DATA** ou **PROC**, sont indépendantes une de l'autre. Vous pouvez donc éditer un programme contenant seulement une étape **DATA** et faire exécuter celle-ci sans avoir à faire d'étapes **PROC** et vice-versa. Pour des fins de lisibilité, dans la fenêtre **Editor**, une ligne sépare chacune des étapes **DATA** et **PROC** d'un programme SAS.*

Comme dans la plupart des éditeurs de programme, il est possible de laisser des commentaires dans le code du programme SAS. Les raisons premières des commentaires sont, tout d'abord, d'avoir une description de ce que le programme effectue et deuxièmement, d'indiquer des informations importantes à la compréhension du code. Afin d'inscrire un commentaire, il suffit d'encadrer un texte par une barre oblique suivie

d'une étoile et une étoile suivie d'une barre oblique (/\* et \*/). Les textes mis en commentaires prennent la couleur **verte** et n'ont aucune incidence sur le programme SAS. Voici un exemple de programme SAS et ses différents éléments:

```

/*Ce programme crée la table comptable à partir de la table
staff se trouvant dans la librairie liste et affiche les données de
cette table dans la fenêtre Output*/

data liste.comptable;
set liste.staff;
if emploi='Comptable';
keep nom prénom;
run;

proc print data=liste.comptable;
run;

```

*Commentaires*

*Mot clé*

*Point-virgule*

*Étape DATA*

*Étape PROC*


## Comment exécuter, examiner les résultats et sauvegarder un programme SAS

```

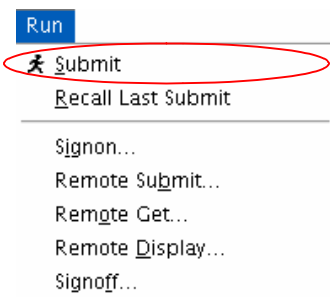
data liste.comptable;
set liste.staff;
if emploi='Comptable';
keep nom prénom;
run;

proc print data=liste.comptable;
run;



```

Comme expliqué dans la présentation de l'interface graphique de SAS, les programme SAS sont édités dans la fenêtre **Editor**. Une fois que le code est entré, il suffit de cliquer sur le bouton **Submit**  qui se trouve sur la barre d'outil

(généralement dans la partie supérieure de la fenêtre SAS) ou d'utiliser la commande **Submit** du menu **Run** pour exécuter le programme.

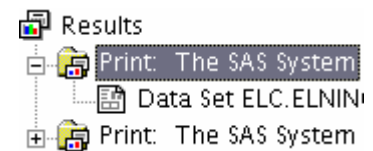


Sas exécutera toutes les instructions qui sont inscrites dans la fenêtre **Editor**. Il est également possible d'exécuter seulement certaines instructions en sélectionnant (*mettre en*

*surbrillance*) celles-ci avant de cliquer sur le bouton **Submit**. . Dans le cas où vous devez arrêter l'exécution d'un programme SAS, il suffit de cliquer sur le bouton **Break**  qui se trouve également sur la barre d'outil de SAS.

Lorsque vous exécutez un programme SAS, les résultats de ce programme sont affichés dans la fenêtre **Output** et des notes diverses sont inscrit dans la fenêtre **Log**. Bien sûr, ce ne sont pas tous les programmes qui ont des résultats à afficher. Par exemple, un programme qui manipule les données d'un fichier n'affiche pas de résultat simplement parce que les résultats de celui-ci sont les changements apportés au fichier de données.

Lorsqu'il y a des résultats qui sont affichés, la navigation dans ceux-ci peut se faire en défilant la fenêtre **Output**. Par contre, il est également possible de naviguer dans les résultats à l'aide de la



fenêtre **Results**. Dans cette fenêtre, vous trouverez un arbre qui est composé des titres des résultats. Ainsi, pour aller directement à un résultat, vous n'avez qu'à double-cliquer sur le titre de celui-ci. La navigation à partir de la fenêtre **Results** est conseillée lorsqu'il y a un grand nombre de résultats affichés dans la fenêtre **Output**.

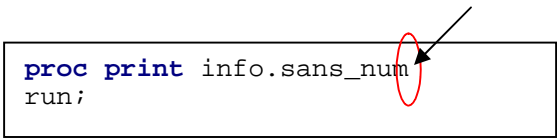
Finalement, il vous est possible de faire la sauvegarde de vos programmes SAS. Il suffit, lorsque vous êtes dans la fenêtre **Editor**, d'utiliser les commandes **Save** ou **Save As** du menu **File** et SAS vous demandera de donner un nom et un emplacement pour sauvegarder le fichier de programme. Notez que, sous Windows, les fichiers de programme SAS possèdent l'extension **.SAS**.

## Aperçu du débogage des programmes SAS – Fenêtre Log

Il est très probable, surtout au début, que vous fassiez des erreurs lors de l'écriture de vos programmes SAS. Des erreurs courantes sont l'oubli d'un point-virgule à la fin d'une instruction ou l'utilisation d'une mauvaise syntaxe de programmation.

Lorsque vous faites des erreurs dans l'édition de vos programmes, SAS refusera d'exécuter les commandes que vous lui présentez et vous indiquera les erreurs ainsi qu'une description de ces erreurs dans la fenêtre **Log**. Cette fenêtre est donc très importante pour le débogage des programmes SAS.

Par exemple, face à l'oubli du point virgule à la fin de la première instruction du programme suivant:



```
proc print info.sans_num
run;
```

SAS affichera ceci dans la fenêtre **Log** et souligne l'endroit où il a trouvé une erreur.

NOTE: The SAS System stopped processing this step because of errors.

NOTE: PROCEDURE PRINT used:  
 real time 36.58 seconds  
 cpu time 0.01 seconds

NOTE: SCL source line.

11 proc print info.sans\_num  
 -----  
 22

NOTE: SCL source line.

12 run;  
 ---  
 202

ERROR 22-322: Syntax error, expecting one of the following: ;, DATA, DOUBLE, N, NOOBS, OBS, ROUND, ROWS, SPLIT, STYLE, UNIFORM, WIDTH.

ERROR 202-322: The option or parameter is not recognized and will be ignored.

Similairement, si nous n'écrivons pas le mot clé **PROC** dans la première instruction du même programme, SAS affichera ceci dans la fenêtre **Log**

NOTE: SCL source line.

5 print data=elc.elnino;  
 -----  
 180

ERROR 180-322: Statement is not valid or it is used out of proper order.

6 run;

Soyez donc très alerte face à ce qui est indiqué dans la fenêtre **Log** de SAS car les informations qui s'y trouvent vous aideront à corriger vos erreurs de programmation.

**EL-Nino! 1**


- **Créer et nommer une librairie,**
- **Inclure un fichier de données dans une librairie**
- **Écrire et exécuter un programme SAS simple**
- **Examiner la sortie SAS**
- **Sauvegarder un programme SAS**

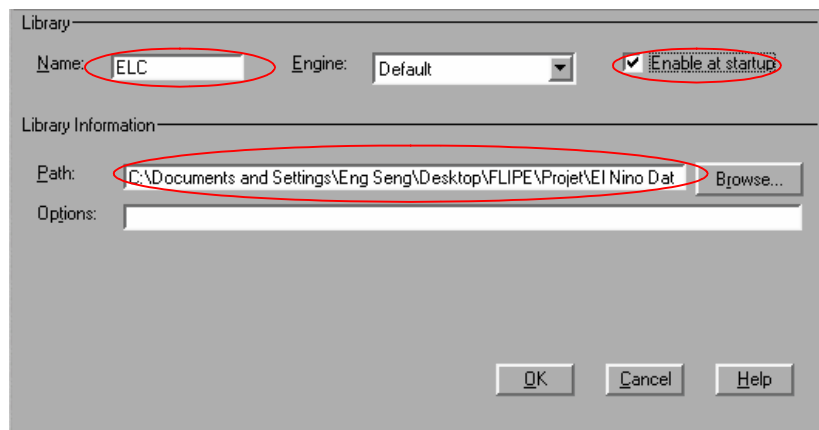
Afin de commencer ses analyses, l'analyste en chef d'**ELC** vous demande de créer une librairie dans laquelle sera stocké le fichier de données **El Nino**. De plus, afin d'avoir une idée du déroulement de l'exécution d'un programme SAS, il vous demande également de créer, d'exécuter un programme simple et de sauvegarder celui-ci.

Pour répondre à ses demandes, suivez les étapes suivantes.

**Étape 1:** Démarrez le logiciel **SAS**.

**Étape 2:** Créer une nouvelle librairie et nommez la **ELC** en cliquant sur le bouton *New*

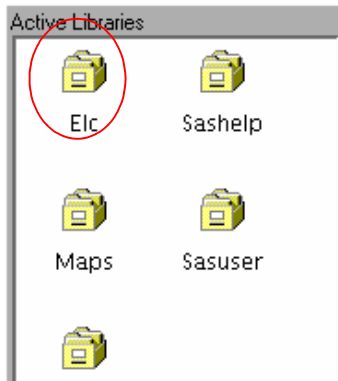
**Library**  qui se trouve sur la barre d'outil. La fenêtre de création de librairie apparaîtra.



Dans l'espace réservé au nom de librairie dans le champ « *Name* », inscrivez **ELC**. Indiquez le chemin d'accès de la librairie sur votre disque dur dans le champ « *Path* » ; c'est à cet endroit (*répertoire*) que les fichiers de données de la librairie **ELC** seront stockés sur votre disque dur. Finalement cochez l'option « *Enable at startup* » qui



chargera la librairie à chaque ouverture de SAS. Dans le cas où vous ne cochez pas cette option, la librairie ne sera pas disponible lors de vos sessions SAS ultérieures. Elle ne sera donc disponible que pour la session SAS en cours et il vous faudra la recréer lors des sessions suivantes. Cliquez sur **OK**, une fois que vous avez terminé ces manipulations.

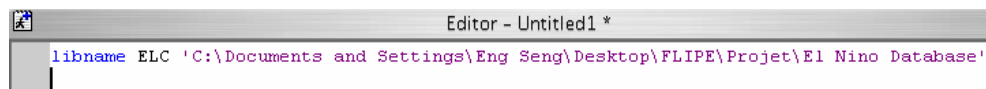


Vous venez donc de créer une nouvelle librairie du nom d'**ELC** et qui sera chargée à chaque fois que vous démarrerez SAS.

Observez que l'icône de cette librairie a été ajoutée dans la liste des librairies de la fenêtre **Explorer**.

Il existe une autre méthode pour la création de librairies. Elle consiste à utiliser l'instruction *libname* à l'intérieur d'un

programme SAS. La création d'une librairie à travers cette méthode se fait de la façon suivante: **libname** *nom\_librairie* '*chemin d'accès au répertoire contenant les fichiers de données*'. Pour notre exemple, la création de la librairie **ELC** se ferait avec l'instruction **libname ELC** '*C:\Documents and Settings\Eng Seng\Desktop\FLIPE\Projet\El Nino Database*'.



Notez qu'une librairie créée avec cette méthode n'est disponible que pour la session en cours. Si vous désirez utiliser cette méthode, nous suggérons d'inclure une telle instruction au début de vos programmes SAS. Ainsi, la librairie dont vous avez besoin à l'intérieur d'un programme sera créée lorsque vous exécutez celui-ci.

**Étape 3:** Nous allons insérer le fichier **elnino.sas7bdat**, qui est fourni avec ce guide, dans la librairie **ELC**. Il suffit de mettre ce fichier dans le répertoire qui est rattaché à cette librairie (*c'est le répertoire que vous avez inscrit dans le champ **path** lors de la création de la librairie*). Une fois que vous avez inséré le fichier dans ce répertoire, affichez la liste des fichiers de données de la librairie **ELC** et celui-ci apparaîtra. (*S'il n'apparaît*

*pas, appuyer sur la touche **F5** du clavier afin de rafraîchir l'affichage du contenu de la fenêtre **Explorer**.)*

Maintenant que la librairie et le fichier de données sont en place, nous allons maintenant créer un programme SAS qui affiche les données du fichier **elnino** dans la fenêtre **Output**.

**Étape 4:** Cliquez dans la fenêtre **Editor** afin d'éditer le programme. Pour identifier ce programme, nous allons mettre en commentaire une description de ce que celui-ci effectue. Inscrivez donc les instructions suivantes dans la fenêtre **Editor**:

```
/*Ce programme nous affiche, dans la fenêtre Output, les données du fichier elnino de la
librairie ELC*/

proc print data=elc.elnino;
run;
```

Par la suite, cliquez sur le bouton **Submit**  et allez dans la fenêtre **Output** pour observer les résultats et dans la fenêtre **Log** pour constater les notes laissées par SAS.

### Sortie partielle - fenêtre Output

| Obs | obs | year | month | day | date   | latitude | longitude | zonal_winds | meridional_winds | humidity | air_temp | s_s_temp |
|-----|-----|------|-------|-----|--------|----------|-----------|-------------|------------------|----------|----------|----------|
| 1   | 1   | 80   | 3     | 7   | 800307 | -0.02    | -109.46   | -6.8        | 0.7              | .        | 26.14    | 26.24    |
| 2   | 2   | 80   | 3     | 8   | 800308 | -0.02    | -109.46   | -4.9        | 1.1              | .        | 25.66    | 25.97    |
| 3   | 3   | 80   | 3     | 9   | 800309 | -0.02    | -109.46   | -4.5        | 2.2              | .        | 25.69    | 25.28    |
| 4   | 4   | 80   | 3     | 10  | 800310 | -0.02    | -109.46   | -3.8        | 1.9              | .        | 25.57    | 24.31    |
| 5   | 5   | 80   | 3     | 11  | 800311 | -0.02    | -109.46   | -4.2        | 1.5              | .        | 25.30    | 23.19    |
| 6   | 6   | 80   | 3     | 12  | 800312 | -0.02    | -109.46   | -4.4        | 0.3              | .        | 24.72    | 23.64    |

*\*\* Les variables humidity, air\_temp et s\_s\_temp dans la sortie ci-dessus ne sont pas visibles sur les illustrations de la table elnino se trouvant aux pages 9 et 10, celles-ci n'étant que des illustrations partielles de la table elnino.*

### Sortie partielle – fenêtre Log

```
9 /*Ce programme nous affiche, dans la fenêtre Output, les données du
9 ! librairie ELC*/
10
11 proc print data=elc.elnino;
12 run;

NOTE: There were 10000 observations read from the data set ELC.ELNINO.
NOTE: PROCEDURE PRINT used:
      real time          0.57 seconds
      cpu time           0.53 seconds
```

**Étape 5:** Nous allons sauvegarder ce programme SAS. Premièrement, cliquez dans la fenêtre **Editor** pour activer celle-ci. Ensuite, cliquer sur la commande **Save As** du menu **File**. Dans la fenêtre qui apparaît, donner un nom et un emplacement au fichier de programme SAS qui sera créé.

Et voilà, nous venons de réaliser un programme SAS !!! Maintenant que nous comprenons de quoi est constitué un programme, attaquons nous aux détails des étapes **DATA** et **PROC**.



# Section 2

Les étapes DATA

# Étapes DATA - Traitement des données

## Qu'est-ce qu'une étape Data?

Les **étapes DATA** sont des ensembles d'instructions SAS principalement utilisés pour **créer des fichiers de données SAS**. Elles débutent par une **instruction DATA** suivie d'instructions diverses qui traitent et manipulent les données contenues dans un fichier de données SAS ou provenant d'une source externe.

De façon plus détaillée, nous pouvons grâce aux étapes **DATA**:

- **Modifier et créer des variables, observations et valeurs dans une table**
- **Créer un nouveau fichier de données en « mode entrée des données »**
- **Créer un nouveau fichier de données à partir d'un fichier existant**
- **Créer un fichier SAS à partir de fichiers de données brutes**
- **Extraire des données**
- **Gérer les fichiers**

Dans ce guide, nous allons surtout nous attarder à la **création de fichiers de données et de variables** à l'aide d'étapes **DATA**. Les deux méthodes de création de fichier de données que nous allons aborder sont la création de fichier par **entrée de données** et la création de fichier **à partir d'un fichier existant**.

### **Important!**

*Comme expliqué dans la section sur les concepts fondamentaux, les données d'un fichier de données SAS sont structurés sous la forme d'une table. C'est ainsi que la création d'un fichier de données passe nécessairement par la création de la table de données qui est contenue dans celui-ci. Nous allons donc créer, dans les sections qui suivent, des fichiers de données en passant par la création des tables de ceux-ci. Notez également que dans certains ouvrages, on ne différencie pas le fichier de données de la table qui est contenue dans celui-ci.*

## Création de fichier de données SAS

Traisons tout d'abord de la création d'un fichier de données en « *mode entrée de données* ». Ce que nous entendons par « *mode entrée de données* » est que nous allons simplement donner à SAS les valeurs que nous voulons qu'il inscrive dans la table du fichier de données.

Cette méthode de création de fichier de données SAS se fait donc en spécifiant les **variables à créer** et les **valeurs à insérer** dans une table. La syntaxe générale d'une étape **DATA** permettant la création d'un fichier de données en « *mode entrée de données* » est la suivante:

### Syntaxe générale d'une étape DATA pour la création d'un fichier en « mode entrée de données »

```
DATA nom_librairie.nom_fichier;  
INPUT nom_variable-1 <$> nom_variable2- <$> ... nom_variable-n;  
DATALINES;  
valeur_variable-1 valeur_variable-2...valeur_variable-n  
valeur_variable-1 valeur_variable-2...valeur_variable-n  
valeur_variable-1 valeur_variable-2...valeur_variable-n  
...  
valeur_variable-1 valeur_variable-2...valeur_variable-n  
RUN;
```

*Explications des instructions de cette syntaxe SAS :*

**Instruction 1:** l'instruction **DATA** suivie d'un nom de librairie et d'un nom de fichier indique à SAS le nom du fichier de données à créer ainsi que la librairie à l'intérieur de laquelle entreposer celui-ci.

**Instruction 2:** l'instruction **INPUT** suivi du ou des noms des variables indique à SAS les variables à créer à l'intérieur de la table de données. Chacun des noms de variable doit être séparé par un espace. Par défaut, des variables de type numérique sont créées. Afin de créer une variable de type caractère, le nom de cette variable doit être suivi d'un espace et d'un signe \$.

**Instruction 3:** l'instruction **DATALINES** et les valeurs que nous retrouvons en dessous sous forme de ligne et de colonne, donc sous forme de table, indiquent à SAS les valeurs à inscrire dans la table de données. Ces valeurs doivent également être séparées par un espace.

**Instruction 4:** l'instruction **RUN** termine l'étape **DATA**.

**Exemple:** voici un exemple de création d'un fichier de données contenant des informations sur la ville, le nom, le prénom, le département, l'âge, le numéro de téléphone et le salaire de plusieurs employés d'une certaine entreprise. Le programme qui réalise cela est le suivant:

```
data info.employees;
input Ville $ Nom $ Prenom $ Dept $ Age Num_tel $ Salaire;
datalines;
Montreal Lapierre Alain mark 35 5145485948 35000
Montreal Solo Yan fin 45 4505483209 42000
Quebec Gagné Martine rh 21 5145435489 35000
Montreal Tong Jean mark 25 8134395490 36000
Quebec Lam Chris fin 24 5145489902 53000
Quebec Free Man fin 45 5478935873 36000
Toronto Alan Fred mark 35 7463829934 36000
Ottawa Aime Fils rh 55 8753733928 62000
Montreal Gucci Charles compt 45 5145489340 55000
Toronto Gagnon Marc mark 32 8903452345 28000
Ottawa Charest Jean compt 56 3494328902 56000
Montreal Cote Yvon mark 34 5147643645 47000
Toronto Recci Mark fin 37 4169483291 32000
Toronto Sylvan Roy mark 45 4164833849 26000
Montreal Vigé Anne rh 28 5145478392 34000
Run;
```

Dans ce fichier qui sera nommé **employees** et qui sera sauvegardé dans la librairie **info**, SAS créera les variables de type caractère **Ville**, **Nom**, **Prénom** et **Dept** et les variables de type numérique **Age**, **Num\_tel** et **Salaire**. Par la suite, SAS lira chacune des lignes de valeurs inscrites sous l'instruction **DATALINES**. La première valeur de chacune des lignes sera assignée à la variable **Ville**, la deuxième valeur à la variable **Nom** et ainsi de suite. SAS lira chacune des lignes qui suivent l'instruction **DATALINES** jusqu'à ce qu'il rencontre l'instruction **RUN**.

Voici la table créée par cette étape **DATA**:

|    | Ville    | Nom      | Prenom  | Dept  | Age | Num_tel  | Salaire |
|----|----------|----------|---------|-------|-----|----------|---------|
| 1  | Montreal | Lapierre | Alain   | mark  | 35  | 51454859 | 35000   |
| 2  | Montreal | Solo     | Yan     | fin   | 45  | 45054832 | 42000   |
| 3  | Quebec   | Gagné    | Martine | rh    | 21  | 51454354 | 35000   |
| 4  | Montreal | Tong     | Jean    | mark  | 25  | 81343954 | 36000   |
| 5  | Quebec   | Lam      | Chris   | fin   | 24  | 51454899 | 53000   |
| 6  | Quebec   | Free     | Man     | fin   | 45  | 54789358 | 36000   |
| 7  | Toronto  | Alan     | Fred    | mark  | 35  | 74638299 | 36000   |
| 8  | Ottawa   | Aime     | Fils    | rh    | 55  | 87537339 | 62000   |
| 9  | Montreal | Gucci    | Charles | compt | 45  | 51454893 | 55000   |
| 10 | Toronto  | Gagnon   | Marc    | mark  | 32  | 89034523 | 28000   |
| 11 | Ottawa   | Charest  | Jean    | compt | 56  | 34943289 | 56000   |
| 12 | Montreal | Cote     | Yvon    | mark  | 34  | 51476436 | 47000   |
| 13 | Toronto  | Recci    | Mark    | fin   | 37  | 41694832 | 32000   |
| 14 | Toronto  | Sylvan   | Roy     | mark  | 45  | 41648338 | 26000   |
| 15 | Montreal | Vigé     | Anne    | rh    | 28  | 51454783 | 34000   |

Il est également possible de créer un fichier de données en créant une nouvelle table en mode entrée de données à l'aide des menus de commandes SAS. Pour ce faire, vous n'avez qu'à suivre les étapes suivantes:

**Étape 1:** Cliquez dans la fenêtre **Explorer** (*soyez certain d'être sous l'onglet **Explorer** et non **Results***), allez au niveau de la liste de librairies disponibles et entrez dans n'importe quelle librairie en double-cliquant sur son icône.



**Étape 2:** Placez votre curseur de souris dans la fenêtre **Explorer**, cliquez sur le bouton droit de votre souris et dans le menu qui apparaît choisissez la commande **New**.

**Étape 3:** Dans la fenêtre qui apparaît sélectionnez **Table** et cliquez sur **OK**.



|   | Nom | Prenl | C | D |
|---|-----|-------|---|---|
| 1 |     |       |   |   |
| 2 |     |       |   |   |
| 3 |     |       |   |   |
| 4 |     |       |   |   |

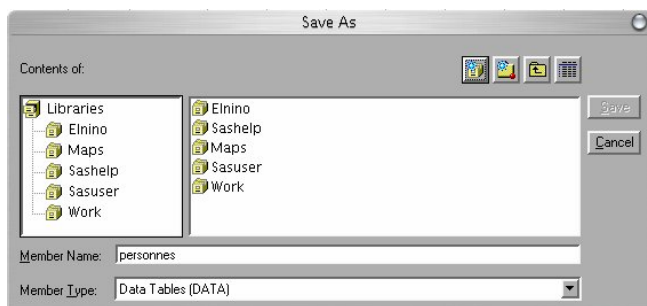
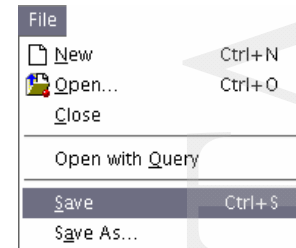
**Étape 4:** Une table vide sera créée. Vous pouvez changer les noms des variables de cette table en double-cliquant sur l'indice de chacune des colonnes.



**Étape 5:** Saisissez simplement, dans les cellules appropriées, les valeurs que vous désirez avoir dans la table. Le type de données que vous insérez déterminera le type de la variable.

|   | Nom      | Pren  | Age | Num_tel |  |
|---|----------|-------|-----|---------|--|
| 1 | Lapierre | Alain | 35  | 514     |  |
| 2 |          |       |     |         |  |

**Étape 6:** Une fois que vous avez terminé l'entrée des données, cliquez sur le menu **File** et ensuite sur la commande **Save**. La fenêtre qui apparaît vous demandera de donner un nom au fichier et d'indiquer dans quelle librairie sauvegarder celui-ci. Voilà!!



Maintenant que nous savons comment créer un fichier de données en « *mode entrée de données* », traitons de la création d'un fichier de données à partir d'un fichier de données existant. Ce que nous entendons par « *créer un fichier à partir d'un fichier existant* » est d'appliquer des modifications sur les données d'un fichier existant et de sauvegarder ce fichier modifié sous un autre nom résultant donc en un nouveau fichier.

Pour réaliser cela, il suffit d'indiquer à SAS **un fichier source**, **un fichier de sortie** et les **manipulations à faire sur les données du fichier source** afin d'obtenir le fichier de sortie. Parmi les manipulations possibles, nous retrouvons **l'élimination** ou la **conservation** d'une ou de plusieurs variables de la table du fichier source. Les instructions pour réaliser ces manipulations sont respectivement, **DROP** et **KEEP**. Traitons de ceux-ci.

## Instructions DROP et KEEP

Comme expliqué plus tôt, à l'intérieur d'une **étape DATA**, l'**instruction DATA** est suivi du nom et de l'emplacement du fichier à créer, c'est-à-dire le fichier de sortie. Pour ce qui est de la spécification de l'emplacement et du nom du fichier source, une instruction **SET** doit être utilisée. Voici la syntaxe générale d'une étape **DATA** permettant la création d'un fichier de données à partir d'un fichier de données existant.

**Syntaxe générale d'une étape DATA pour la création d'un fichier de données à partir d'un fichier existant.**

```
DATA nom_librairie.nom_fichier;  
SET nom_librairie.nom_fichier;  
DROP ou KEEP nom_variable-1 nom_variable-2 ... nom_variable-n;  
RUN;
```

*Explications des instructions de cette syntaxe SAS :*

**Instruction 1:** L'instruction **DATA** suivie d'un nom de librairie et d'un nom de fichier indique à SAS le nom du fichier de données qu'il doit créer ainsi que la librairie dans laquelle entreposer celui-ci.

**Instruction 2:** L'instruction **SET** suivie d'un nom de librairie et d'un fichier indique à SAS le nom du fichier source dont les données seront modifiées et dans quelle librairie il se trouve.

**Instruction 3:** Les instructions **DROP** et **KEEP** suivie du ou des noms de variables indique à SAS les variables à éliminer ou conserver de la table du fichier source pour obtenir la table du fichier de sortie. Les noms des variables doivent être séparés par un espace.

**Instruction 4:** L'instruction **RUN** termine l'étape **DATA**.

### **Remarque!**

*Notez que la table du fichier source n'est pas altérée lors de cette manipulation. Les modifications n'affectent que la table du fichier de sortie. Ne vous souciez donc pas de perdre les informations du fichier source.*

**Exemple:** Reprenons le fichier de données de l'exemple précédent, c'est-à-dire le fichier **employes** de la librairie **info**. Créons un nouveau fichier en éliminant la variable **Num\_tel** et nommons ce nouveau fichier **employes2**. Le programme exécutant cela est le suivant:

```
/*Ce programme élimine la variable
Num_tel du fichier employes de la
librairie info et sauvegarde les données
modifiées dans le fichier employes2*/

data info.employes2;
set info.employes;
drop num_tel;
run;
```

Nous passons donc de ceci...

|   | Ville    | Nom      | Prenom  | Dept | Age | Num_tel  | Salaire |
|---|----------|----------|---------|------|-----|----------|---------|
| 1 | Montreal | Lapierre | Alain   | mark | 35  | 51454859 | 35000   |
| 2 | Montreal | Solo     | Yan     | fin  | 45  | 45054832 | 42000   |
| 3 | Quebec   | Gagné    | Martine | rh   | 21  | 51454354 | 35000   |
| 4 | Montreal | Tong     | Jean    | mark | 25  | 81343954 | 36000   |
| 5 | Quebec   | Lam      | Chris   | fin  | 24  | 51454899 | 53000   |
| 6 | Quebec   | Free     | Man     | fin  | 45  | 54789358 | 36000   |
| 7 | Toronto  | Alan     | Fred    | mark | 35  | 74638299 | 36000   |
| 8 | Ottawa   | Aime     | File    | rh   | 55  | 87537339 | 62000   |

à ceci...

|   | Ville    | Nom      | Prenom  | Dept | Age | Salaire |
|---|----------|----------|---------|------|-----|---------|
| 1 | Montreal | Lapierre | Alain   | mark | 35  | 35000   |
| 2 | Montreal | Solo     | Yan     | fin  | 45  | 42000   |
| 3 | Quebec   | Gagné    | Martine | rh   | 21  | 35000   |
| 4 | Montreal | Tong     | Jean    | mark | 25  | 36000   |
| 5 | Quebec   | Lam      | Chris   | fin  | 24  | 53000   |
| 6 | Quebec   | Free     | Man     | fin  | 45  | 36000   |
| 7 | Toronto  | Alan     | Fred    | mark | 35  | 36000   |
| 8 | Ottawa   | Aime     | File    | rh   | 55  | 62000   |

## Création de nouvelles variables à l'aide d'expressions

Nous avons vu qu'il faut spécifier les variables à créer lors de la création d'un fichier de données SAS. Ainsi, nous avons nommé et indiqué le type de variable que nous voulions avoir. Il existe une autre méthode de création de variable qui consiste à utiliser des expressions.

Qu'est-ce qu'une expression? Une expression est une suite d'**opérateurs** et d'**opérandes** qui produit généralement une valeur. Les opérandes sont des **constantes** ou des **variables** tandis que les opérateurs sont des symboles qui peuvent être de nature **arithmétique**, **logique** ou **comparatif**. Voyons quelques exemples d'expression :

### Exemples d'expression

- **Salaire + Prime**
- **Alpha/beta**
- **-1\*X**

Comment créons nous des variables à l'aide d'expressions? Cela se fait à l'aide d'une instruction d'assignation. Il est possible, puisqu'une expression retourne une valeur, d'assigner cette valeur à une variable en utilisant un symbole **égale (=)**. Ainsi, avec le premier des trois exemples d'expression, nous pouvons créer la variable **Total** et lui assigner une valeur qui est la somme des valeurs des variables **Salaire** et **Prime** écrivant l'instruction **Total=Salaire+Prime**.

Il suffit donc, pour créer une nouvelle variable lors d'une étape **DATA**, d'utiliser une expression afin de produire une valeur qui sera assignée à une cette nouvelle variable.

**Exemple:** Voici un exemple d'une étape **DATA** où nous créons deux nouvelles variables à l'aide d'expressions. Dans cet exemple, nous utilisons encore le fichier **employes** de la librairie **info**. Nous allons y créer deux nouvelles variables c'est-à-dire **Prime** qui est

égale à **10%** du salaire et **Total** qui est la somme du **Salaire** et de la **Prime**. Le programme pour créer ces variables est le suivant:

```
/*Ce programme crée les variables Prime et Total
dans la table du fichier employes de la
librairie info*/

data info.employes;
set info.employes;
Prime=0.1*Salaire;
Total=Salaire+Prime;
run;
```

Nous utilisons donc la même syntaxe que pour la création d'un fichier à partir d'un autre fichier c'est-à-dire qu'il nous faut un fichier source et un fichier de sortie. Par contre, le fichier source est le même que le fichier de sortie dans cet exemple. Ceci revient à faire directement des modifications sur les des données du fichier source.

Nous passons donc de ceci...

|   | Ville    | Nom      | Prenom  | Dept | Age | Num_tel  | Salaire |
|---|----------|----------|---------|------|-----|----------|---------|
| 1 | Montreal | Lapierre | Alain   | mark | 35  | 51454859 | 35000   |
| 2 | Montreal | Solo     | Yan     | fin  | 45  | 45054832 | 42000   |
| 3 | Quebec   | Gagné    | Martine | rh   | 21  | 51454354 | 35000   |
| 4 | Montreal | Tong     | Jean    | mark | 25  | 81343954 | 36000   |
| 5 | Quebec   | Lam      | Chris   | fin  | 24  | 51454899 | 53000   |
| 6 | Quebec   | Free     | Man     | fin  | 45  | 54789358 | 36000   |
| 7 | Toronto  | Alan     | Fred    | mark | 35  | 74638299 | 36000   |
| 8 | Ottawa   | Δime     | File    | rh   | 55  | 87537339 | 62000   |

À ceci...

|   | Ville    | Nom      | Prenom  | Dept | Age | Num_tel  | Salaire | Prime | Total |
|---|----------|----------|---------|------|-----|----------|---------|-------|-------|
| 1 | Montreal | Lapierre | Alain   | mark | 35  | 51454859 | 35000   | 3500  | 38500 |
| 2 | Montreal | Solo     | Yan     | fin  | 45  | 45054832 | 42000   | 4200  | 46200 |
| 3 | Quebec   | Gagné    | Martine | rh   | 21  | 51454354 | 35000   | 3500  | 38500 |
| 4 | Montreal | Tong     | Jean    | mark | 25  | 81343954 | 36000   | 3600  | 39600 |
| 5 | Quebec   | Lam      | Chris   | fin  | 24  | 51454899 | 53000   | 5300  | 58300 |
| 6 | Quebec   | Free     | Man     | fin  | 45  | 54789358 | 36000   | 3600  | 39600 |
| 7 | Toronto  | Alan     | Fred    | mark | 35  | 74638299 | 36000   | 3600  | 39600 |
| 8 | Ottawa   | Δime     | File    | rh   | 55  | 87537339 | 62000   | 6200  | 68200 |

## Les opérateurs dans les expressions

Il est important, premièrement, d'indiquer qu'il existe deux types d'opérateur dans SAS.

Le **préfixe**, qui précède l'opérande et qui n'affecte que l'opérande qui le suit et les **infixes** qui s'applique aux opérandes se situant de chaque côtés de celui-ci.

Voici un exemple de chacun des types d'opérateur: **Préfixe: - X** et **Infixe: Y\*X**

De plus, comme mentionné précédemment, les opérateurs dans SAS peuvent être de différentes natures: **arithmétique**, **comparatif** et **logique**.

Les opérateurs de nature arithmétique permettent naturellement d'appliquer des opérations arithmétiques sur les opérandes tandis que les opérateurs comparatifs et logiques permette de faire des vérifications sur les données et servent énormément lors des traitements conditionnels que nous aborderons prochainement. Voici une série de tableaux présentant les opérateurs et leurs utilités.

**Tableau des opérateurs arithmétiques**

| Opérateurs | Utilité        | Exemple d'application     |
|------------|----------------|---------------------------|
| **         | Exponentiel    | $Y = X^{**}3$             |
| +          | Addition       | $Z = X + Y$               |
| -          | Soustraction   | $Z = X - 3$               |
| *          | Multiplication | $T = 1.564 * \text{Prix}$ |
| /          | Division       | $W = \text{Age} / 5$      |

**Tableau des opérateurs de comparaison**

| Opérateur | Utilité                 | Exemple d'application |
|-----------|-------------------------|-----------------------|
| =         | Égalité                 | $Y = X$               |
| ^=        | Non égalité à           | $Y \neq 3$            |
| >         | Plus grand que          | $Y > X$               |
| <         | Plus petit que          | $Y < X$               |
| >=        | Plus grand ou égale que | $Y \geq X$            |
| <=        | Plus petit ou égale que | $Y \leq X$            |

Tableau des opérateurs logiques

| Opérateur | Utilité | Exemple d'application |
|-----------|---------|-----------------------|
| &         | ET      | a>5 & b<6             |
|           | OU      | x<5   x>10            |
| ^         | NON     | ^nom='Sammy'          |

Nous ne donnons pas plus de détails sur ces opérateurs car, comme nous l'avons spécifié dans l'introduction de ce guide, nous supposons que vous possédez les connaissances de base à propos de ceux-ci.

**Remarque!**

*Notez que l'opérateur **égale** (=) dans une expression est un opérateur de comparaison. Celui-ci a donc une nature différente du **égale** que l'on retrouve dans une instruction d'assignation (celui que nous avons utilisé pour créer des variables à l'aide d'expressions).*

## Les opérandes dans les expressions: constantes et variables

Comme nous l'avons expliqué précédemment, les variables peuvent être de type **caractère** ou **numérique**. C'est également le cas pour les constantes mais, bien sûr, les constantes, selon leur définition, ont des valeurs fixes.

L'utilisation des constantes de type caractère se fait en utilisant des chaînes de caractères entre guillemets. Voici un exemple dans lequel nous utilisons un opérateur de comparaison afin de comparer la valeur de la variable Nom à une constante de type caractère: **Nom= 'Jean'**.

**Important!**

*N'oubliez pas les guillemets lors de l'utilisation d'une constante de type caractère car il est très important de différencier les constantes de type caractère et les noms des variables. Si l'on écrit **Nom= 'Jean'**, on compare le contenu de la variable Nom à la valeur constante Jean. Par contre, si l'on écrit **Nom=Jean** on compare la valeur de la variable Nom à la valeur de la variable Jean.*

Pour ce qui est des constantes de type **numérique**, il suffit d'utiliser directement la valeur désirée. Par exemple: **Age = 65**

Outre ces détails, il est important de signaler d'utiliser, dans les expressions, des opérandes de même type c'est-à-dire des opérandes de type caractère entre elles et des opérandes de type numérique entre elles.

Lorsque vous construisez des expressions avec des opérandes de type caractère et numérique ensemble, SAS procédera à des conversions de caractère à numérique ou vice-versa mais nous n'aborderons pas cet élément dans ce guide, ces notions dépassant les limites de celui-ci.

## Le traitement conditionnel des données

SAS nous offre la possibilité, similairement à la plupart des langages de programmation, d'effectuer du traitement conditionnel c'est-à-dire d'exécuter des traitements seulement lorsque certaines conditions sont respectées. C'est ici que les opérateurs de nature comparatif et logique servent énormément car nous nous servons des expressions pour créer des conditions à respecter.

Dans SAS, les traitements conditionnels se font à l'aide des ensembles d'instructions **IF-THEN**, **ELSE** et **DO-END**. Voici la syntaxe générale d'une étape **DATA** dans laquelle nous utilisons des instructions de traitement conditionnel:

### Syntaxe générale d'une étape **DATA** incluant des traitements conditionnels

```
DATA nom_librairie. nom_fichier;  
SET nom_librairie.nom_fichier;  
IF « expression » THEN « instruction de traitement »;  
ELSE « instructions de traitement »;  
RUN;
```

*Explications des instructions de cette syntaxe SAS :*

**Instruction 1:** L'instruction **DATA** suivie d'un nom de librairie et d'un nom de fichier indique à SAS le nom du fichier à créer avec les traitements conditionnels.



**Instruction 2:** L'instruction **SET** suivie d'un nom de librairie et d'un nom de fichier indique à SAS le nom du fichier sur lequel il doit appliquer les instructions.

**Instruction 3:** Le mot clé **IF** suivie d'une expression stipule une condition à respecter. Le mot clé **THEN** indique à SAS le traitement à appliquer si la condition suivant le mot clé **IF** est respectée.

**Instruction 4:** L'instruction **ELSE** indique le traitement à appliquer dans le cas où la condition suivant le mot clé **IF** n'est pas respectée.

**Instruction 5 :** L'instruction **RUN** termine l'étape **DATA**.

**Exemple:** Un cas souvent rencontré d'utilisation de traitement conditionnel est le **recodage de variable**. Supposons que nous voulons savoir, à l'aide d'une variable binaire que nous nommerons **bin**, quelles personnes de notre fichier **employes** a un salaire plus élevé que 35 000 et est âgé de plus de 24 ans. La variable **bin** prendra la valeur **1** pour ceux qui respectent ces conditions et **0** sinon. Le programme nécessaire à la création de cette variable binaire est le suivant:

```
/*Ce programme vérifie à l'aide d'une variable
binaire si un employé a un salaire plus élevé
que 35000 et est âgé de plus de 24 ans*/

data info.employes;
set info.employes;
if salaire>35000 & age>24 then bin=1;
else bin=0;
run;
```

Nous avons donc le résultat suivant:

| Age | Num_tel  | Salaire | Prime | Total | bin |
|-----|----------|---------|-------|-------|-----|
| 35  | 51454859 | 35000   | 3500  | 38500 | 0   |
| 45  | 45054832 | 42000   | 4200  | 46200 | 1   |
| 21  | 51454354 | 35000   | 3500  | 38500 | 0   |
| 25  | 81343954 | 36000   | 3600  | 39600 | 1   |
| 24  | 51454899 | 53000   | 5300  | 58300 | 0   |
| 45  | 54789358 | 36000   | 3600  | 39600 | 1   |
| 35  | 74638299 | 36000   | 3600  | 39600 | 1   |
| 55  | 87537339 | 62000   | 6200  | 68200 | 1   |
| 45  | 51454893 | 55000   | 5500  | 60500 | 1   |
| 32  | 89034523 | 28000   | 2800  | 30800 | 0   |
| 56  | 34943289 | 56000   | 5600  | 61600 | 1   |
| 34  | 51476436 | 47000   | 4700  | 51700 | 1   |
| 37  | 41694832 | 32000   | 3200  | 35200 | 0   |
| 45  | 41648338 | 26000   | 2600  | 28600 | 0   |
| 28  | 51454783 | 34000   | 3400  | 37400 | 0   |

Encore une fois, le fichier source est le même que le fichier de sortie dans cet exemple ce qui revient à appliquer directement des modifications sur le fichier source.

Bien que très utiles, **une seule instruction peut être exécuté dans chacune des instruction IF-THEN ou ELSE**. Afin de pouvoir traiter un groupe d'instruction de façon conditionnelle, il nous faut utiliser les instructions **IF-THEN** et **ELSE** avec des instructions **DO** et **END**. Voici la syntaxe générale d'un traitement conditionnel pour un groupe d'instructions:

**Syntaxe générale d'une étape DATA incluant des traitements conditionnels  
(groupe d'instructions)**

```
DATA nom_librairie.nom_fichier;  
SET nom_librairie.nom_fichier;  
IF « expression » THEN DO;  
  « Instruction de traitement-1 »;  
  « Instruction de traitement-2 »;  
  ...  
  « Instruction de traitement-n »;  
END;  
ELSE DO;  
  « Instruction de traitement-1 »;  
  « Instruction de traitement-2 » ;  
  ...  
  « Instruction de traitement-n »;  
END;  
RUN;
```

L'explication du programme est la même que pour le traitement conditionnel avec les instructions **IF-THEN** et **ELSE** à la différence que plusieurs instructions (les instructions entre les **DO** et les **END** - groupe d'instructions) sont exécutées lorsque la condition est respectée.

Nous terminons bien sûr cette section par une sous-section **El Nino!**. Cependant, mentionnons que nous traitons, dans la **Section 4** (*Autres notions utiles*), de deux manipulations qui se font à l'intérieur d'étapes **DATA** qui sont la **concaténation** et la **fusion** de tables. Nous recommandons de lire les sous-sections portant sur celles-ci.

**El-Nino! 2**

- **Création d'un nouveau fichier de données**
- **Utilisation d'un traitement conditionnel**
- **Création de variables à l'aide d'expressions**

Notre analyste chez **ELC** pense que les variables **date**, **month** et **day** du fichier **elnino** sont inutiles et aimerait les éliminer. Par contre, il désire garder une copie du fichier originale **elnino**. Il vous demande donc de lui écrire un programme qui crée un nouveau fichier à partir du fichier **elnino** mais en éliminant les trois variables inutiles. Ce nouveau fichier doit s'appeler **elnino\_utile** et être entreposé dans la librairie **ELC**. De plus il veut pouvoir identifier, à l'aide d'une variable binaire (0/1) les observations dont la longitude est entre -109 et -120 (*variable longitude*).

Finalement, il s'est rendu compte que les mesures de température de l'air dans le fichier **elnino** sont en degrés Fahrenheit et il aimerait avoir ces mesures en degrés Celsius. La formule de conversion de Fahrenheit à Celsius est la suivante: **degré Celsius = (5/9)\*(degré Fahrenheit -32)**. Le programme que vous devez écrire est donc...

```
/*Ces instructions créent le fichier elnino_utile à partir du fichier
elnino en éliminant les variables date, month et day*/

data elc.elnino_utile;
set elc.elnino;
drop date month day;
run;

/*Ce programme recode la longitude en une variable binaire longitude_bin*/

data elc.elnino_utile;
set elc.elnino_utile;
if -120<=longitude<=-109 then longitude_bin=0;
else longitude_bin=1;
run;

/*Ce programme créer la variable air_temp_C dans laquelle sont les mesures
de température de la variable air_temp mais en degrés Celsius*/

data elc.elnino_utile;
set elc.elnino_utile;
air_temp_C = (5/9)*(air_temp-32);
run;
```



# Section 3

Les étapes PROC

# Étapes PROC - Analyses de données

## Qu'est-ce qu'une étape PROC?

Les **étapes Proc** sont des ensembles d'instructions SAS principalement utilisés pour **analyser les fichiers de données SAS**. Suivant une structure similaire aux étapes **DATA**, elles débutent par une **instruction PROC** suivie d'instructions diverses qui traitent et manipulent les données contenues dans un fichier de données SAS.

Il existe une panoplie de procédure à l'intérieur de SAS. Nous avons donc choisi les procédures de base qui nous semblent les plus utiles pour le début de votre parcours c'est-à-dire les procédures **CONTENTS**, **PRINT**, **SORT**, **TRANSPOSE**, **MEANS**, **FREQ** et **UNIVARIATE**.

## Procédure CONTENTS – Examiner la zone descriptive d'un fichier de données

Comme expliqué dans la première partie de ce guide, un fichier SAS est composé de deux parties: la zone descriptive et la zone de données. La procédure **CONTENTS** nous permet d'afficher cette dernière dans la fenêtre **Output** de SAS. Voici la syntaxe d'utilisation de la procédure **CONTENTS**:

### Syntaxe générale d'utilisation d'une procédure CONTENTS

```
PROC CONTENTS DATA=nom_librairie.nom_fichier;  
RUN;
```

*Explication des instructions de cette syntaxe SAS:*

**Instruction 1:** L'instruction de procédure **PROC CONTENTS** suivie de l'option **DATA=**, d'un nom de librairie et d'un nom de fichier indique à SAS l'emplacement et le nom du fichier dont nous voulons afficher la zone descriptive.

**Instruction 2:** L'instruction **RUN** termine l'étape **PROC**.

**Exemple:** si nous reprenons notre fichier de données sur les employés et voulons afficher sa zone descriptive, nous devons utiliser le programme suivant.

```
/*Ce programme affiche la zone descriptive
du fichier employes qui se trouve dans la
librairie info*/

proc contents data=info.employes;
run;
```

Voici la sortie qui s'affiche dans la fenêtre **Output**.

**The CONTENTS Procedure**

|                |                                |                       |    |
|----------------|--------------------------------|-----------------------|----|
| Data Set Name: | INFO.EMPLOYES                  | Observations:         | 15 |
| Member Type:   | DATA                           | Variables:            | 10 |
| Engine:        | V8                             | Indexes:              | 0  |
| Created:       | 16:55 Thursday, April 13, 2006 | Observation Length:   | 80 |
| Last Modified: | 16:55 Thursday, April 13, 2006 | Deleted Observations: | 0  |
| Protection:    |                                | Compressed:           | NO |
| Data Set Type: |                                | Sorted:               | NO |
| Label:         |                                |                       |    |

**-----Engine/Host Dependent Information-----**

|                             |   |
|-----------------------------|---|
| Data Set Page Size:         | 8192  |
| Number of Data Set Pages:   | 1   |
| First Data Page:            | 1   |
| Max Obs per Page:           | 101   |
| Obs in First Data Page:     | 15  |
| Number of Data Set Repairs: | 0   |
| File Name:                  | C:\Documents and Settings\Eng Seng\Desktop\FLIPE\Projet\Guide d'introduction à SAS\info\employes.sas7bdat |
| Release Created:            | 8.0202M0  |
| Host Created:               | WIN_PRO   |

**-----Alphabetic List of Variables and Attributes-----**

| #  | Variable | Type | Len | Pos |
|----|----------|------|-----|-----|
| 5  | Age      | Num  | 8   | 0   |
| 4  | Dept     | Char | 8   | 64  |
| 2  | Nom      | Char | 8   | 48  |
| 6  | Num_tel  | Char | 8   | 72  |
| 3  | Prenom   | Char | 8   | 56  |
| 8  | Prime    | Num  | 8   | 16  |
| 7  | Salaire  | Num  | 8   | 8   |
| 9  | Total    | Num  | 8   | 24  |
| 1  | Ville    | Char | 8   | 40  |
| 10 | bin      | Num  | 8   | 32  |

## PROC PRINT - Examiner la zone de données d'un fichier SAS

Il est possible d'afficher la zone de données d'un fichier à l'intérieur de la fenêtre **OUTPUT** à l'aide de la procédure **PRINT**. La syntaxe générale d'utilisation de cette procédure est la suivante :

### Syntaxe générale d'utilisation d'une procédure PRINT

```
PROC PRINT DATA=nom_librairie.nom_fichier;  
RUN;
```

*Explication des instructions de cette syntaxe SAS:*

**Instruction 1:** L'instruction de procédure **PROC PRINT** suivie de l'option **DATA=**, d'un nom de librairie et d'un nom de fichier indique à SAS l'emplacement et le nom du fichier dont nous voulons afficher la zone de données.

**Instruction 2:** L'instruction **RUN** termine l'étape **PROC**.

**Exemple:** Si vous voulez afficher la zone de données du fichier **employees** dans la fenêtre **Output**, le programme nécessaire est:

```
/*Le programme suivant affiche la zone de  
données du fichier employees dans la fenêtre  
output*/  
  
proc print data=info.employees;  
run;
```

Voici une sortie partielle des résultats de ce programme dans la fenêtre **Output**:

| Obs | Ville    | Nom      | Prenom  | Dept | Age | Num_tel  | Salaire | Prime | Total | bin |
|-----|----------|----------|---------|------|-----|----------|---------|-------|-------|-----|
| 1   | Montreal | Lapierre | Alain   | mark | 35  | 51454859 | 35000   | 3500  | 38500 | 1   |
| 2   | Montreal | Solo     | Yan     | fin  | 45  | 45054832 | 42000   | 4200  | 46200 | 1   |
| 3   | Quebec   | Gagné    | Martine | rh   | 21  | 51454354 | 35000   | 3500  | 38500 | 0   |
| 4   | Montreal | Tong     | Jean    | mark | 25  | 81343954 | 36000   | 3600  | 39600 | 1   |
| 5   | Quebec   | Lam      | Chris   | fin  | 24  | 51454899 | 53000   | 5300  | 58300 | 0   |
| 6   | Quebec   | Free     | Man     | fin  | 45  | 54789358 | 36000   | 3600  | 39600 | 1   |

Bien qu'il est possible, comme nous l'avons vu dans la première section du guide, d'inspecter la zone de données d'un fichier en double-cliquant sur l'icône d'un fichier directement, l'utilisation de la procédure **PRINT** nous offre des avantages au niveau du contrôle des données affichées.

Nous pouvons ainsi choisir les variables et les observations que nous voulons afficher dans la fenêtre **Output**. Pour ce faire, il suffit d'accompagner la procédure **PRINT** d'une

instruction **VAR** pour choisir les variables à afficher **et/ou** d'une instruction **WHERE** afin de choisir les observations qui nous intéressent.

Afin de sélectionner les variables à afficher, il nous faut simplement utiliser l'instruction **VAR** de la manière suivante :

```
PROC PRINT DATA=nom_librairie.nom_fichier;  
VAR nom_variable-1 nom_variable-2 ... nom_variable-n;  
RUN;
```

Le nom des variables suivant l'instruction **VAR** indique à SAS les variables dont les données seront à afficher. Ces noms de variable doivent être séparés par un espace.

Pour ce qui est de l'instruction **WHERE**, celle-ci nous permet de spécifier une ou des conditions, à l'aide d'expressions, sur le choix des observations. Ainsi, seulement les observations qui respectent ces conditions seront affichées. L'utilisation de l'instruction **WHERE** se fait suivant la syntaxe suivante :

```
PROC PRINT DATA=nom_librairie.nom_fichier;  
WHERE « expression »;  
RUN;
```

Bien sûr, il est possible d'utiliser les instructions **VAR** et **WHERE** en même temps.

**Exemple:** Supposons que, de la table **employes**, nous voulons seulement afficher les données de la variable **salaire** pour les observations dont le prénom est **Jean** et le nom **Tong**. Le programme qui réalise cela est :

```
/*Ce programme affiche le salaire de Jean Tong à  
partir de la table employes de la librairie info*/  
  
proc print data=info.employes;  
var nom prenom salaire;  
where nom='Tong' & prenom='Jean';  
run;
```



Et voici le résultat:

| Obs | Nom  | Prenom | Salaire |
|-----|------|--------|---------|
| 4   | Tong | Jean   | 36000   |

## PROC SORT - Tri des données

Une autre procédure qui vous sera utile est la procédure **SORT** qui permet de trier les observations d'une table en ordre croissant ou décroissant. La syntaxe générale d'utilisation de cette procédure est la suivante:

### Syntaxe générale d'utilisation de la procédure SORT

```
PROC SORT DATA=nom_bibliotheque.nom_fichier;  
BY <DESCENDING> variable-1 <DESCENDING> variable-2 ... <DESCENDING> variable-n;  
RUN;
```

*Explication des instructions de cette syntaxe SAS:*

**Instruction 1:** L'instruction **PROC SORT** suivie de l'option **DATA=**, d'un nom de bibliothèque et d'un nom de fichier indique à SAS le fichier de données dont les données seront triées.

**Instruction 2:** L'instruction **BY** suivie d'un ou de plusieurs noms de variables indique à SAS la variable ou les variables par rapport auxquelles les observations doivent être triées. Ce sont donc les critères de tri. SAS utilisera la première variable comme critère et dans le cas où il y a des égalités dans le tri des observations, SAS utilisera la deuxième variable et ainsi de suite. Par défaut, la procédure **SORT** trie les observations en ordre croissant. Pour trier les observations en ordre décroissant, il suffit d'inscrire **DESCENDING** devant le nom de la ou des variables qui servent de critère de tri.

**Instruction 3:** L'instruction **RUN** termine l'étape **PROC**.

### Remarque!

*Vous pouvez également utiliser des variables de type caractère comme critère de tri. Les observations seront simplement ordonnées en ordre alphabétique si vous demandez un tri croissant, et en ordre alphabétique inverse si vous demandez un tri décroissant.*

**Exemple:** Supposons que nous voulons trier les observations de la table du fichier **employes** par ordre de niveau de salaire. Le programme pour effectuer ce tri est le suivant :

```
/*Ce programme trie en ordre croissant
de salaire les observations de la table
du fichier employes*/

proc sort data=info.employes;
by salaire;
run;
```

Le résultat est le suivant:

|   | Ville    | Nom      | Prenom  | Dept | Age | Num_tel  | Salaire | Prim |
|---|----------|----------|---------|------|-----|----------|---------|------|
| 1 | Toronto  | Sylvan   | Roy     | mark | 45  | 41648338 | 26000   |      |
| 2 | Toronto  | Gagnon   | Marc    | mark | 32  | 89034523 | 28000   |      |
| 3 | Toronto  | Recci    | Mark    | fin  | 37  | 41694832 | 32000   |      |
| 4 | Montreal | Vigé     | Anne    | rh   | 28  | 51454783 | 34000   |      |
| 5 | Montreal | Lapierre | Alain   | mark | 35  | 51454859 | 35000   |      |
| 6 | Quebec   | Gagné    | Martine | rh   | 21  | 51454354 | 35000   |      |
| 7 | Montreal | Tong     | Jean    | mark | 25  | 81343954 | 36000   |      |
| 8 | Quebec   | Free     | Man     | fin  | 45  | 54789358 | 36000   |      |
| 9 | Toronto  | Alan     | Fred    | mark | 35  | 74638299 | 36000   |      |

## PROC TRANSPOSE -Transposition des données d'une table SAS

Il est parfois nécessaire de transposer les données d'une table, c'est-à-dire de restructurer celles-ci de façon à ce que les lignes deviennent les colonnes et les colonnes deviennent les lignes. La procédure SAS nous permettant de transposer les données d'une table est **TRANSPOSE**. La syntaxe générale de son utilisation est la suivante:

### Syntaxe générale d'utilisation de la procédure Transpose

```
PROC TRANSPOSE DATA=nom_librairie.nom_fichier OUT= nom_librairie.nom_fichier;  
RUN;
```

*Explication des instructions de cette syntaxe:*

**Instruction 1:** L'instruction de procédure **PROC TRANSPOSE** suivie des options **DATA=** et **OUT=**, qui sont elles mêmes suivies de noms de librairies et de noms de fichiers, indique respectivement à SAS le fichier dont la table est à transposer et le fichier dans lequel sauvegarder la table transposée.

**Instruction 2:** L'instruction **RUN** termine l'étape **PROC**.

Voici une illustration de l'effet d'une transposition de table effectuée par SAS:

Table d'origine

|   | Nom      | Prenom  | Age | Num_tel    | Salaire |
|---|----------|---------|-----|------------|---------|
| 1 | Lapierre | Alain   | 35  | 5145485948 | 35000   |
| 2 | Solo     | Yan     | 45  | 4505483209 | 43000   |
| 3 | Gagné    | Martine | 21  | 5145435489 | 40000   |
| 4 | Tong     | Jean    | 25  | 8134395490 | 37000   |

Table transposée

|   | NAME OF FORMER VARIABLE | COL1       | COL2       | COL3       | COL4       |
|---|-------------------------|------------|------------|------------|------------|
| 1 | Nom                     | Lapierre   | Solo       | Gagné      | Tong       |
| 2 | Prenom                  | Alain      | Yan        | Martine    | Jean       |
| 3 | Age                     | 35         | 45         | 21         | 25         |
| 4 | Num_tel                 | 5145485948 | 4505483209 | 5145435489 | 8134395490 |
| 5 | Salaire                 | 35000      | 43000      | 40000      | 37000      |

### Remarque!

*Comme vous pouvez le constater, SAS crée lors de la transposition, une colonne supplémentaire appelé **NAME OF FORMER VARIABLE** qui indique à quelle variable ou colonne de la table d'origine correspond une ligne de la table transposée.*

Il est également possible de ne transposer que les données de certaines variables, les données des autres variables étant éliminées. Pour ce faire, il suffit de spécifiez dans une instruction **VAR** la ou les variables à garder et à transposer. Ceci se fait selon la syntaxe suivante :

```
PROC TRANSPOSE DATA=nom_librairie.nom_fichier OUT= nom_librairie.nom_fichier;
VAR variable-1 variable-2 ... variable-n;
RUN;
```

**Remarque!**

Lorsque vous utilisez la procédure **TRANPOSE** sans spécifier les variables à transposer, celle-ci ne transpose que les valeurs des variables de type numérique. Ainsi, si vous voulez transposer les valeurs de variables de type caractère, il vous faut spécifier le nom de ces variables à l'intérieur d'une instruction **VAR**.

Finalement, il est aussi possible d'appliquer une transposition des données d'une table par rapport aux valeurs d'une variable choisie à l'aide d'une instruction **BY**. C'est ce qu'on appelle la transposition par groupe. Cette transposition se fait selon la syntaxe suivante :

```
PROC TRANSPOSE DATA=nom_bibliothèque.nom_fichier OUT= nom_bibliothèque.nom_fichier;  
BY variable de groupement;  
RUN;
```

**Remarque!**

Pour effectuer une transposition par groupe, il est nécessaire que les valeurs de la variable par rapport à laquelle vous voulez transposer soient en ordre croissant. Il vous faudra donc appliquer une procédure **SORT** avant la procédure **TRANPOSE**.

**Exemple:** Supposons que nous voulons transposer les valeurs de la table **employees** selon la ville de l'employé. Il nous faut premièrement mettre les observations de la table **employees** en ordre croissant de ville avec le programme suivant :

```
/*Ce programme trie en ordre croissant  
de ville les observations de la table du  
fichier employees*/  
  
proc sort data=info.employees;  
by ville;  
run;
```

Le résultat est que les observations de la table **employees** sont maintenant groupées par ville comme illustré ci-dessous.

|    | Ville    | Nom      | Prenom  | Dept  | Age | N   |
|----|----------|----------|---------|-------|-----|-----|
| 1  | Montreal | Vigé     | Anne    | rh    | 28  | 514 |
| 2  | Montreal | Lapierre | Alain   | mark  | 35  | 514 |
| 3  | Montreal | Tong     | Jean    | mark  | 25  | 813 |
| 4  | Montreal | Solo     | Yan     | fin   | 45  | 450 |
| 5  | Montreal | Cote     | Yvon    | mark  | 34  | 514 |
| 6  | Montreal | Gucci    | Charles | compt | 45  | 514 |
| 7  | Ottawa   | Charest  | Jean    | compt | 56  | 349 |
| 8  | Ottawa   | Aime     | Fils    | rh    | 55  | 875 |
| 9  | Quebec   | Gagné    | Martine | rh    | 21  | 514 |
| 10 | Quebec   | Free     | Man     | fin   | 45  | 547 |

Lorsque nous appliquons la transposition par groupe à cette table, avec la variable **ville** comme variable de groupement, SAS va transposer les observations par groupe de ville. Les observations dont la ville est Montréal seront les premières à être transposées suivies des observations dont la ville est Ottawa et ainsi de suite. De plus, une colonne indiquant à quelle ville les observations transposées appartiennent est créée.

À partir de la table précédente, nous obtenons donc ceci :

|    | Ville    | NAME OF FORMER VARIABLE | COL1     | COL2     | COL3     | COL4     | COL5   |
|----|----------|-------------------------|----------|----------|----------|----------|--------|
| 1  | Montreal | Nom                     | Vigé     | Lapierre | Tong     | Solo     | Cote   |
| 2  | Montreal | Prenom                  | Anne     | Alain    | Jean     | Yan      | Yvon   |
| 3  | Montreal | Dept                    | rh       | mark     | mark     | fin      | mark   |
| 4  | Montreal | Age                     | 28       | 35       | 25       | 45       | 34     |
| 5  | Montreal | Num_tel                 | 51454783 | 51454859 | 81343954 | 45054832 | 514764 |
| 6  | Montreal | Salaire                 | 34000    | 35000    | 36000    | 42000    | 47000  |
| 7  | Montreal | Prime                   | 3400     | 3500     | 3600     | 4200     | 4700   |
| 8  | Montreal | Total                   | 37400    | 38500    | 39600    | 46200    | 51700  |
| 9  | Montreal | bin                     | 1        | 1        | 1        | 1        | 1      |
| 10 | Ottawa   | Nom                     | Charest  | Aime     |          |          |        |
| 11 | Ottawa   | Prenom                  | Jean     | Fils     |          |          |        |

Le programme nécessaire à ce traitement est:

```
/*Ce programme transpose les valeurs des variables de la table employes par
groupe de ville et sauvegarde ces valeurs transposées dans le fichier
employes_trans*/

proc transpose data=info.employes out=info.employes_trans;
var nom prenom dept age num_tel salaire prime total bin;
by ville;
run;
```

## PROC MEANS, FREQ et UNIVARIATE - Statistiques descriptives

Certaines procédures SAS permettent également d'analyser les données et de calculer des statistiques descriptives de base. Nous allons aborder trois procédures qui vous permettent d'appliquer des analyses statistiques simples.

La première, la procédure **MEANS**, calcule entre autres le **nombre d'observations**, la **moyenne**, l'**écart type**, le **minimum** et le **maximum** pour les variables de type **numérique** d'un fichier. La syntaxe générale de la procédure **MEANS** est la suivante:

### Syntaxe générale d'utilisation de la procédure MEANS

```
PROC MEANS DATA=nom_librairie.nom_fichier;
RUN;
```

*Explications des instructions de cette syntaxe:*

**Instruction 1:** L'instruction **PROC MEANS** suivie de l'option **DATA=** d'un nom de librairie et d'un nom de fichier indique à SAS le fichier à partir duquel les statistiques doivent être calculées.

**Instruction 2:** L'instruction **RUN** termine l'étape **PROC**.

Voici un exemple d'utilisation de la procédure **MEANS** :

```
/* Ce programme génère des statistiques simples pour les
variables numériques du fichier de données employes*/

proc means data=info.employes;
run;
```

et voici les résultats de ce programme:

| The MEANS Procedure |    |            |            |            |            |
|---------------------|----|------------|------------|------------|------------|
| Variable            | N  | Mean       | Std Dev    | Minimum    | Maximum    |
| Age                 | 15 | 37.4666667 | 10.7893245 | 21.0000000 | 56.0000000 |
| Salaire             | 15 | 40866.67   | 11063.88   | 26000.00   | 62000.00   |
| Prime               | 15 | 4086.67    | 1106.39    | 2600.00    | 6200.00    |
| Total               | 15 | 44953.33   | 12170.27   | 28600.00   | 68200.00   |
| bin                 | 15 | 0.7333333  | 0.4577377  | 0          | 1.0000000  |

Il est également possible de choisir les variables par rapport auxquelles nous voulons avoir ces statistiques. Il suffit d'utiliser une instruction **VAR** suivie des noms des variables dont nous voulons avoir les statistiques. De plus, il est aussi possible d'indiquer à SAS de calculer des statistiques par rapport aux valeurs d'une variable avec l'instruction **CLASS**.

```
PROC MEANS DATA=nom_librairie.nom_fichier;
VAR variable-1 variable-2 ... variable-n;
CLASS variable;
RUN;
```

**Exemple:** Supposons que nous voulons avoir la moyenne et l'écart type des salaires des employés par rapport à chacune des villes du fichier **employees**. Pour ce faire, il nous faut utiliser une instruction **VAR** pour sélectionner seulement la variable **salaire** et l'instruction **CLASS** pour spécifier à SAS que les statistiques de cette variable doivent être calculées pour chacune des valeurs de la variable **ville**. Le programme qui nous donnera ces résultats est donc:

```
/*Ce programme génère des statistiques simples pour la
variable salaire selon les valeurs de la variable ville
du fichier de données employees*/

proc means data=info.employees;
var salaire;
class ville;
run;
```

Le résultat de ce programme est le suivant :

| The MEANS Procedure         |          |   |          |          |          |          |
|-----------------------------|----------|---|----------|----------|----------|----------|
| Analysis Variable : Salaire |          |   |          |          |          |          |
| Ville                       | N<br>Obs | N | Mean     | Std Dev  | Minimum  | Maximum  |
| Montreal                    | 6        | 6 | 41500.00 | 8264.38  | 34000.00 | 55000.00 |
| Ottawa                      | 2        | 2 | 59000.00 | 4242.64  | 56000.00 | 62000.00 |
| Quebec                      | 3        | 3 | 41333.33 | 10115.99 | 35000.00 | 53000.00 |
| Toronto                     | 4        | 4 | 30500.00 | 4434.71  | 26000.00 | 36000.00 |

**Remarque!**

La procédure **MEANS** peut aussi donner quelques autres statistiques comme la médiane et les quantiles.

La deuxième procédure que nous abordons, la procédure **FREQ**, permet d'obtenir des distributions de fréquences (**fréquences, fréquences cumulées, fréquences relatives et fréquences relatives cumulées**) à une dimension et plusieurs dimensions. Les tableaux de fréquences à deux dimensions sont également appelées **tableaux croisés ou tableaux de contingence**. La syntaxe d'utilisation de base de la procédure **FREQ** est la suivante:

**Syntaxe d'utilisation de la procédure FREQ**

```
PROC FREQ DATA = nom_librairie.nom_fichier;  
TABLES variable-1 variable-2 ... variable-n;  
RUN;
```

*Explication des instructions de cette syntaxe:*

**Instruction 1:** L'instruction de procédure **PROC FREQ** suivie de l'option **DATA=**, d'un nom de librairie et d'un nom de fichier indique à SAS le fichier à partir duquel il doit calculer les statistiques.

**Instruction 2:** L'instruction **TABLES** suivie du ou des noms de variables indique à SAS la ou les variables par rapport auxquelles il doit calculer les statistiques.

**Instruction 3:** L'instruction **RUN** termine l'étape **PROC**.

Pour spécifier à SAS de construire des tables à n dimensions, il suffit d'utiliser la syntaxe suivante dans l'instruction **TABLES**: variable-dimension-1\*variable-dimension-2\*...\*variable-dimension-n. Par exemple pour construire un tableau croisé à deux dimensions, les variables dans l'instruction **TABLES** doivent être spécifiées de la manière suivante: **TABLES** variable-1\*variable-2.

Dans le cas où l'on omet l'instruction **TABLES**, cette procédure calcule les statistiques pour chacune des variables d'un fichier.



**Exemple:** Voici un exemple d'utilisation de la procédure **FREQ** dans lequel nous affichons la distribution de la variable **Dept** de la table **employees**.

```
/*Ce programme donne la distribution des fréquences de
la variable dept du fichier employees*/

proc freq data=info.employees;
table dept;
run;
```

Le résultat de ce programme est le suivant :

**The FREQ Procedure**

| Dept  | Frequency | Percent | Cumulative<br>Frequency | Cumulative<br>Percent |
|-------|-----------|---------|-------------------------|-----------------------|
| compt | 2         | 13.33   | 2                       | 13.33                 |
| fin   | 4         | 26.67   | 6                       | 40.00                 |
| mark  | 6         | 40.00   | 12                      | 80.00                 |
| rh    | 3         | 20.00   | 15                      | 100.00                |

Voici un deuxième exemple dans lequel nous affichons un tableau croisé entre les variables département et ville.

```
/*Ce programme donne la distribution des fréquences du fichier de données
employees par rapport aux valeurs des variables dept et ville */

proc freq data=info.employees;
table dept*ville;
run;
```

**The FREQ Procedure**  
**Table of Dept by Ville**

| Dept      | Ville                        |                             |                              |                              |              |
|-----------|------------------------------|-----------------------------|------------------------------|------------------------------|--------------|
|           | Montreal                     | Ottawa                      | Quebec                       | Toronto                      | Total        |
| Frequency |                              |                             |                              |                              |              |
| Percent   |                              |                             |                              |                              |              |
| Row Pct   |                              |                             |                              |                              |              |
| Col Pct   |                              |                             |                              |                              |              |
| compt     | 1<br>6.67<br>50.00<br>16.67  | 1<br>6.67<br>50.00          | 0<br>0.00<br>0.00<br>0.00    | 0<br>0.00<br>0.00<br>0.00    | 2<br>13.33   |
| fin       | 1<br>6.67<br>25.00<br>16.67  | 0<br>0.00<br>0.00<br>0.00   | 2<br>13.33<br>50.00<br>66.67 | 1<br>6.67<br>25.00<br>25.00  | 4<br>26.67   |
| mark      | 3<br>20.00<br>50.00<br>50.00 | 0<br>0.00<br>0.00<br>0.00   | 0<br>0.00<br>0.00<br>0.00    | 3<br>20.00<br>50.00<br>75.00 | 6<br>40.00   |
| rh        | 1<br>6.67<br>33.33<br>16.67  | 1<br>6.67<br>33.33<br>50.00 | 1<br>6.67<br>33.33<br>33.33  | 0<br>0.00<br>0.00<br>0.00    | 3<br>20.00   |
| Total     | 6<br>40.00                   | 2<br>13.33                  | 3<br>20.00                   | 4<br>26.67                   | 15<br>100.00 |

**Remarque!**

La procédure **FREQ** est la procédure qui vous permet de réaliser des tests de l'égalité des proportions pour les tableaux de fréquences à une dimension et des tests d'indépendance pour les tableaux croisés à deux dimensions.

## PROC UNIVARIATE - Statistiques descriptives plus détaillées

Une troisième procédure fournissant des statistiques descriptives est la procédure **UNIVARIATE**. Celle-ci fournit, en plus des statistiques tels que la moyenne et l'écart type, des informations sur **les quantiles et les valeurs extrêmes**. La syntaxe d'utilisation générale de la procédure **UNIVARIATE** est la suivante :

### Syntaxe d'utilisation de la procédure UNIVARIATE

```
PROC UNIVARIATE DATA=nom_librairie.nom_fichier;  
VAR variable-1 variable-2 ... variable-n;  
RUN;
```

*Explication des instructions de cette syntaxe:*

**Instruction 1:** L'instruction de procédure **PROC UNIVARIATE** suivie de l'option **DATA=**, d'un nom de librairie et d'un nom de fichier indique à SAS, le fichier à partir duquel il doit calculer les statistiques.

**Instruction 2:** L'instruction **VAR** suivie des noms des variables indique à SAS les variables par rapport auxquelles il doit calculer les statistiques.

**Instruction 3:** L'instruction **RUN** termine l'étape **PROC**.

**Exemple:** Voici un exemple dans lequel nous affichons les statistiques descriptives de la variable **age** du fichier **employees** à l'aide de la procédure **UNIVARIATE**

```

/*Ce programme utilise la procédure UNIVARIATE afin
d'afficher des statistiques descriptives portant sur la
variable age*/

proc univariate data=info.employees;
var age;
run;

```

Les résultats (*partiels*) de ce programme est le suivant:

**The UNIVARIATE Procedure**  
**Variable: Age**

**Moments**

|                 |            |                  |            |
|-----------------|------------|------------------|------------|
| N               | 15         | Sum Weights      | 15         |
| Mean            | 37.466667  | Sum Observations | 562        |
| Std Deviation   | 10.7893245 | Variance         | 116.409524 |
| Skewness        | 0.23215411 | Kurtosis         | -0.8087095 |
| Uncorrected SS  | 22686      | Corrected SS     | 1629.73333 |
| Coeff Variation | 28.7971295 | Std Error Mean   | 2.78579161 |

**Basic Statistical Measures**

| Location |          | Variability         |           |
|----------|----------|---------------------|-----------|
| Mean     | 37.46667 | Std Deviation       | 10.78932  |
| Median   | 35.00000 | Variance            | 116.40952 |
| Mode     | 45.00000 | Range               | 35.00000  |
|          |          | Interquartile Range | 17.00000  |

**Tests for Location: Mu0=0**

| Test        | -Statistic- | -----p Value----- |        |
|-------------|-------------|-------------------|--------|
| Student's t | t 13.4492   | Pr >  t           | <.0001 |
| Sign        | M 7.5       | Pr >=  M          | <.0001 |
| Signed Rank | S 60        | Pr >=  S          | <.0001 |

**Quantiles (Definition 5)**

| Quantile   | Estimate |
|------------|----------|
| 100% Max   | 56       |
| 99%        | 56       |
| 95%        | 56       |
| 90%        | 55       |
| 75% Q3     | 45       |
| 50% Median | 35       |
| 25% Q1     | 28       |
| 10%        | 24       |
| 5%         | 21       |
| 1%         | 21       |
| 0% Min     | 21       |

La procédure **UNIVARIATE** calcule et nous affiche également des résultats sur divers tests statistiques. Nous n'aborderons pas ceux-ci, la compréhension de ces résultats nécessitant certaines connaissances statistiques précises.

**EL-Nino! 3**

- **Afficher des données dans la fenêtre Output**
- **Construire des tableaux de fréquences**
- **Calculer des statistiques simples**

L'analyste en chef vous demande de lui créer une liste des observations de la température à la surface de la mer (variable **s\_s\_temp**) qui ont été recueillies sur **latitude** entre -0.01 et 0.005 et sur une **longitude** entre -109.5 et -109. Il veut que vous affichiez cette liste dans la fenêtre **Output**.

Il aimerait également avoir la fréquence des observations effectuées par les bouées selon les années et finalement, il veut aussi avoir la moyenne et l'écart-type de la température à la surface de la mer (ne tenez pas compte de l'unité de mesure de la température) pour chacune des années d'observation. Écrivez le programme qui lui donnera ces informations. Le programme qui vous donne ces informations est donc...

```
/*Cette procédure affiche une liste des observations pour la
variables s_s_temps qui ont été recueillies sur une latitude
entre -0.01 et 0.005 et une longitude
entre -109.5 et -109*/

proc print data=elc.elnino;
var s_s_temp;
where -0.01<=latitude<=0.005 & -109.5<=longitude<=-109;
run;

/*Cette procédure produit un tableau de fréquence des
observations du fichier de données elnino selon les années
d'observations*/

proc freq data=elc.elnino;
table year;
run;

/*Ce procédure calcule la moyenne et l'écart-type de la
variable air_temp selon les années d'observation*/

proc means data=elc.elnino;
var air_temp;
class year;
run;
```



# Section 4

Création de graphiques

# Les procédures SAS – Création de graphiques

## Création de graphiques dans SAS

Les étapes **PROC** permettent également de créer des graphiques à l'intérieur de SAS. Nous allons traiter deux procédures SAS qui nous permettent de créer respectivement des diagrammes à barres horizontales ou verticales, des diagrammes en pointes de tarte et des diagrammes en nuage de points. Ceux-ci permettent d'illustrer la distribution des fréquences d'une variable ou la relation entre les valeurs de deux variables.

## PROC GCHART - Diagramme à barres et en pointes de tarte

La procédure **GCHART** permet de créer des diagrammes à barres verticales ou horizontales ainsi que des diagrammes en pointes de tarte. La syntaxe d'utilisation de cette procédure est la suivante.

### Syntaxe d'utilisation de la procédure GCHART

```
PROC GCHART DATA=nom_lib.nom_fichier;  
HBAR ou VBAR ou PIE variable;  
RUN;
```

*Explication des instructions de cette syntaxe:*

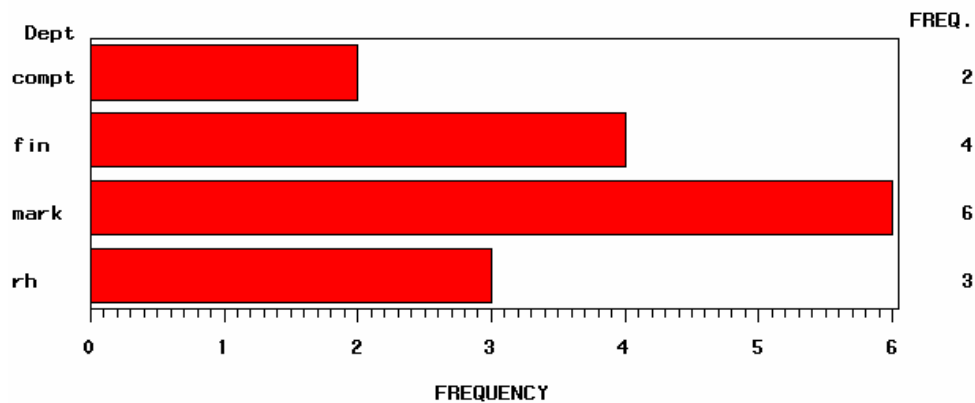
**Instruction 1:** L'instruction de procédure **PROC GCHART** suivie de l'option **DATA=**, d'un nom de librairie et d'un nom de fichier spécifie à SAS le fichier de données à partir duquel construire les graphiques.

**Instruction 2:** L'instruction **HBAR**, **VBAR** ou **PIE** suivie du nom d'une variable permet de choisir le type de graphique qui sera construit et la variable à partir de laquelle construire celui-ci. **HBAR** vous crée un diagramme à barres horizontales, **VBAR** vous crée un diagramme à barres verticales et **PIE** vous crée un graphique en pointes de tarte.

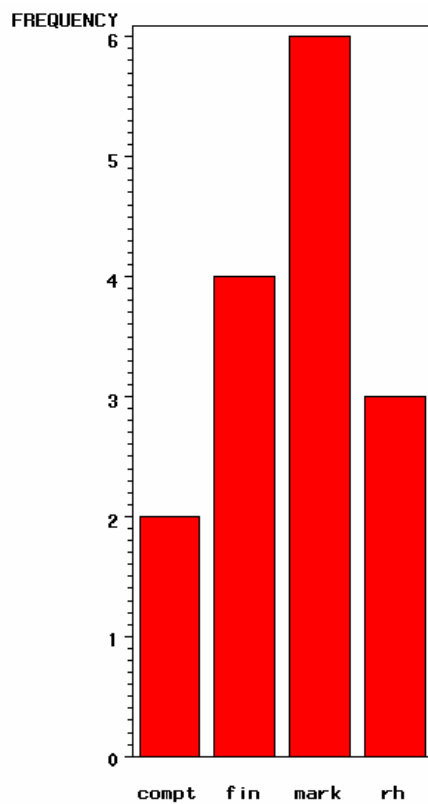
**Instruction 3:** L'instruction **RUN** termine l'étape PROC.

**Exemple:** Voici les graphiques à barres horizontales, verticales et en pointes de tarte construits à partir de de la variable **Dept** du fichier de données **employes**. Ces graphiques donnent la distribution des fréquences des observations selon les départements des employés.

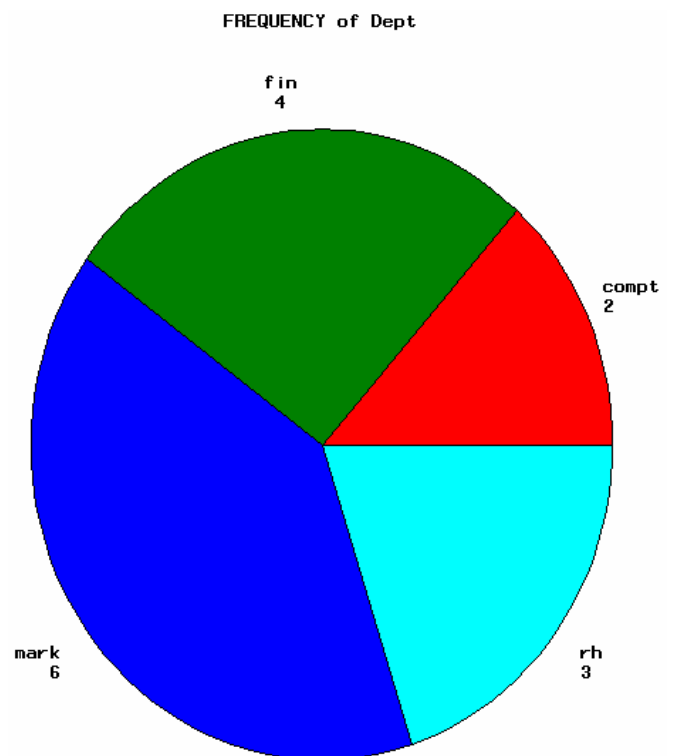
Graphique à barres horizontales - HBAR



Graphique à barres verticales– VBAR



Graphique en pointes de tarte – PIE



Le graphique à barres horizontales est obtenu à partir du programme suivant. Bien sûr, pour obtenir les diagrammes à barres verticales et en pointes de tarte, il suffit de changer l'instruction **HBAR** en **VBAR** ou **PIE**.

```
/*Ce programme crée un graphique à barres horizontales à  
partir des données de la variable Dept*/  
  
proc gchart data=info.employees;  
HBAR dept;  
run;
```

## PROC GPLOT - Diagramme en nuage de points (ou de dispersion)

La procédure **GPLOT**, quant à elle, vous permet de créer des graphiques en nuage de points à deux dimensions. La syntaxe générale d'utilisation de cette procédure est la suivante.

### Syntaxe d'utilisation de la procédure GPLOT

```
PROC GPLOT DATA=nom_libririe.nom_fichier;  
PLOT variable-axe-verticale*variable-axe-horizontale;  
RUN;
```

*Explication des instructions de cette syntaxe SAS:*

**Instruction 1:** L'instruction de procédure **PROC GPLOT** suivie de l'option **DATA=**, d'un nom de librairie et d'un nom de fichier spécifie à SAS le fichier de données à partir duquel construire le graphique.

**Instruction 2:** L'instruction **PLOT** suivie des noms des variables spécifie les variables à assigner à l'axe vertical et l'axe horizontal.

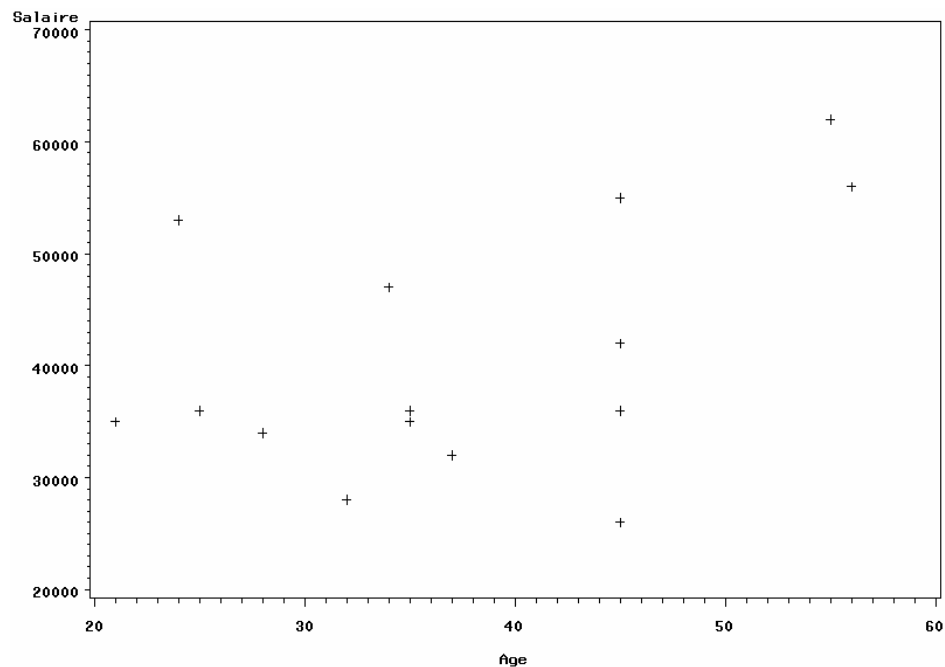
**Instruction 3:** L'instruction **RUN** termine l'étape **PROC**.



**Exemple:** Créons un graphique en nuage de points à partir des variables salaire et age du fichier de données employes. Le programme pour créer ce graphique est:

```
/*Ce programme crée un graphique en nuage de points  
de la variable salaire par rapport à la variable  
age*/  
  
proc gplot data=info.employes;  
plot salaire*age;  
run;
```

et voici le graphique créé :



## EL-Nino! 4

### - Création de graphiques

Encore une fois, l'analyste en chef vous demande de l'aide. Il veut que vous lui créiez deux graphiques. Le premier est un graphique en pointes de tarte pour illustrer la

distribution des fréquences des observations du fichier **elnino** selon les années d'observation des bouées et le deuxième est un graphique en nuage de points ayant la variable **s\_s\_temp** comme axe vertical et la variable **air\_temp** comme axe horizontal. Éditer le programme qui créera ces deux graphiques.

Le programme que vous devez éditer est donc...

```
/*Ce programme crée un graphique en pointes de tarte
au niveau des fréquences des observations par année
du fichier elnino*/

proc gchart data=elc.elnino;
pie year;
run;

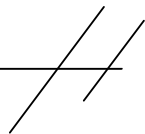
/*Ce programme crée un graphique en nuage de points
de la variables s_s_temp par rapport à la variable
air_temp du fichier de données elnino*/

proc gplot data=elc.elnino;
plot s_s_temp*air_temp;
run;
```



# Section 5

Autres notions utiles



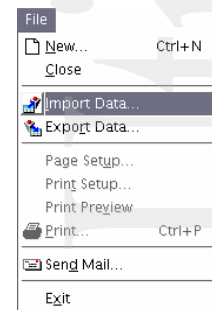
## Autres notions utiles

Dans cette avant dernière section, nous allons aborder des éléments qui vous seront certainement utiles à travers vos cours.

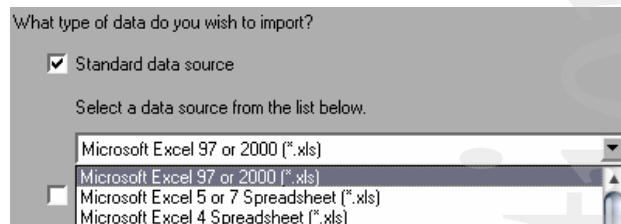
### Importation de fichiers de données Excel dans SAS

Les fichiers de données, dans la majorité des cours où vous ferez utilisation de SAS, vous seront remis sous forme de fichiers Excel (*et possiblement dans d'autres formats non natifs à SAS*). Il vous faudra donc les importer à l'intérieur de SAS et voici comment.

**Étape 1:** cliquez sur le menu **File** et choisissez la commande **Import Data**.



**Étape 2:** La fenêtre d'assistant d'importation de fichier apparaîtra. Choisissez **Microsoft Excel 97 or 2000**, dans la section « **Select a data source from the list below** » et cliquez sur le bouton **Next**. (L'option « **Standard data source** » doit être cochée afin d'avoir accès à cette liste.)

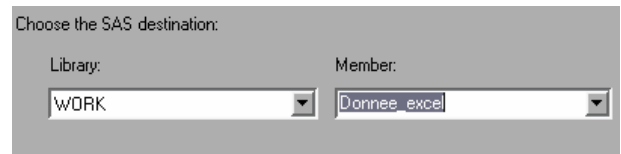


**Étape 3:** SAS vous demandera de sélectionner le fichier Excel dont vous voulez importer les données. Il s'agit de mettre le chemin d'accès à ce fichier. Si vous ne connaissez pas ce chemin d'accès, vous pouvez utiliser le bouton **Browse** afin de naviguer dans vos fichiers. Une fois que vous avez indiqué le chemin d'accès de votre fichier de données, cliquez sur le bouton **Next**.



**Étape 4:** La fenêtre suivante vous demande de donner un nom au fichier SAS qui va être créé à partir des données inclues dans le fichier Excel et

d'indiquer dans quelle librairie entreposer celui-ci. Une fois que vous avez spécifié ces informations, cliquez sur **Finish** pour terminer l'importation du fichier Excel.



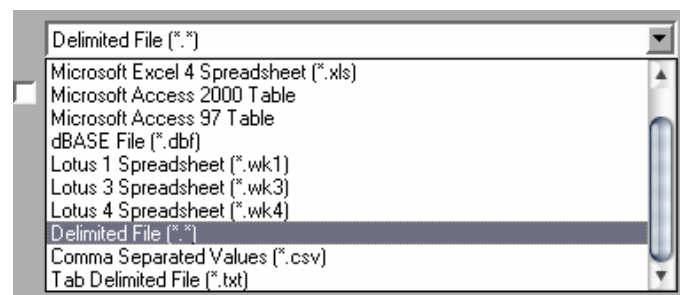
## Importation de fichiers de données brutes dans SAS

Vous avez sûrement remarqué qu'il est possible d'importer des fichiers provenant de plusieurs logiciels différents, incluant Excel, lorsque vous avez défilé la liste de sélection du type de fichier dans les manipulations précédentes.

Cependant, beaucoup de fichiers de données sont des fichiers de données brutes. Les données à l'intérieur de ceux-ci ne sont pas formatées sous les normes d'aucun logiciel en particulier. Par contre, nous retrouvons quand même une certaine structure de base à l'intérieur des données de ces types de fichier. Chacune des données du fichier est séparée par un « délimiteur » qui peut, entre autre, être un espace ou une tabulation.

L'assistant d'importation de fichiers de SAS a la capacité d'importer ces fichiers de données brutes grâce à ces « délimiteurs ».

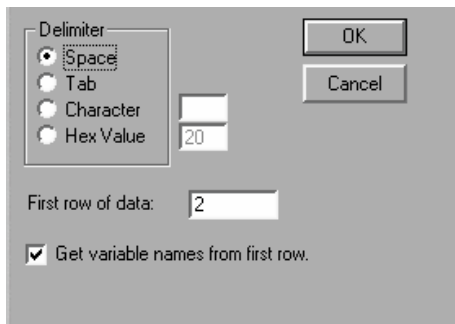
Ainsi, pour importer un fichier de données brutes, qui ont différentes extensions dont .dat, .raw et .txt, il suffit de choisir dans la liste du type de fichier de l'assistant d'importation, le type



**Delimited File** (ou **Tab delimited File** pour les fichiers .txt).

Par la suite, lorsque vous arrivez à la fenêtre qui demande le chemin d'accès du fichier, indiquez le chemin d'accès du fichier de données brutes et cliquez sur le bouton **Options**. Dans cette fenêtre, vous pouvez choisir le type de « délimiteur » qui est utilisé dans le

fichier et également indiquer si la première ligne du fichier contient les noms des variables ou non.



## Utilisation des listes de variables

Il est très courant d'avoir à traiter plusieurs variables en même temps, c'est-à-dire d'appliquer le même traitement à plusieurs variables ou d'appliquer un traitement qui nécessite plusieurs variables. Dans ces cas, lorsque le nombre de variables à traiter est grand, il devient très fastidieux d'inscrire le nom de chacune de celles-ci.

Afin de contrer ce problème, nous pouvons avoir recours à ce qu'on appelle des listes de variables. Il existe deux types de liste de variables, celles qui sont composées de **variables numérotées** et celles qui sont composées de variables ayant tous des noms différents. Traitons tout d'abord le premier type.

Si vous avez une longue liste de cent vingt-quatre variables à créer soit **x1, x2, x3, ..., x124**. Au lieu de toutes les énumérer dans votre programme, il est possible d'utiliser une liste de variables de la façon suivante: une liste de variable **x1, x2, ..., x124** peut s'écrire **x1-x124** dans SAS. Ainsi, voici un exemple

```
/*Ce programme calcule la moyenne ainsi que d'autres statistiques  
descriptives des variables x1 à x124 en utilisant une liste de  
variables (x1-x124)*/  
  
proc means data=exemple.liste1;  
var x1-x124;  
run;
```

Traisons maintenant du deuxième type de liste de variables. Dans le cas où votre liste de variables est composée de variables ayant différents noms, il suffit d'utiliser une liste de variables de cette façon : **nom de la première variable--nom de la dernière variable**.

Par exemple, si vous voulez traiter quatre variables nommées **temps**, **air**, **mer**, **vent** sans les énumérer une par une, il suffit de créer une liste de cette façon : **temps--vent**.

```
/*Ce programme calcule la moyenne ainsi que d'autres
statistiques descriptives des variables temps, air, mer et
vent utilisant une liste de variables (temps--vent)*/

proc means data=exemple.liste2;
var temps--vent;
run;
```

## Concaténation et fusion de tables

La concaténation et la fusion de tables sont deux méthodes différentes pour grouper les données de différentes tables donc de différents fichiers. Nous allons regarder les cas simples de concaténation et de fusion de tables. Commençons la concaténation de table.

La concaténation de tables permet, dans le cas le plus simple, de rassembler les observations de tables ayant les mêmes variables pour former une nouvelle table. La concaténation de tables se fait dans une étape **DATA** et sa syntaxe d'utilisation est la suivante:

### Syntaxe d'une étape DATA permettant de concaténer n tables

```
DATA nom_librairie.nom_fichier;
SET nom_librairie.nom_fichier-1 nom_librairie.nom_fichier-2 ... nom_librairie.nom_fichier-n;
RUN;
```

*Explications des instructions de cette syntaxe SAS :*

**Instruction 1:** L'instruction **DATA** suivie d'un nom de librairie et d'un nom de fichier indique à SAS le nom du fichier de données qu'il doit créer, résultant de la concaténation de tables de plusieurs fichiers de données, et dans quelle librairie assigner celui-ci.

**Instruction 2:** L'instruction **SET** suivie des différents noms de librairie et de fichiers indique à SAS les fichiers de données dont les tables sont à concaténer.

**Instruction 3:** L'instruction **RUN** termine l'étape **DATA**.

Illustrons une concaténation de tables afin de permettre une meilleure compréhension du traitement effectué. Supposons que nous avons les deux tables suivantes:

Table 1

|   | Nom    | Prenom   | Age |
|---|--------|----------|-----|
| 1 | Dubé   | Jean     | 33  |
| 2 | Sanson | Caroline | 28  |

Table 2

|   | Nom     | Prenom  | Age |
|---|---------|---------|-----|
| 1 | Leclair | Francis | 30  |
| 2 | Gucci   | Nadia   | 24  |

Ces deux tables ont des observations sur les mêmes variables **Nom**, **Prénom** et **Age**.

Avec l'aide d'une concaténation nous pouvons créer une troisième table qui sera le rassemblement des observations de ces deux tables.

Voici la table que nous voulons créer :

Table 3

|   | Nom     | Prenom   | Age |
|---|---------|----------|-----|
| 1 | Dubé    | Jean     | 33  |
| 2 | Sanson  | Caroline | 28  |
| 3 | Leclair | Francis  | 30  |
| 4 | Gucci   | Nadia    | 24  |

Le programme SAS qui effectue ce traitement est le suivant :

```
/*Ce programme concatène la table1 et la  
table2 afin de créer la table3*/  
  
data exemple.table3;  
set exemple.table1 exemple.table2;  
run;
```

La concaténation des tables utilise donc les noms des variables comme point d'encrage afin de rassembler les données de plusieurs tables.



Le deuxième type de rassemblement des données de différentes tables est la fusion. Cette méthode rassemble les données de plusieurs tables selon les valeurs d'une variable ou de quelques variables communes aux tables; ces valeurs étant les mêmes pour les différentes tables. La fusion de tables se fait également à l'intérieur d'une étape **DATA** et à l'aide de l'instruction **MERGE**. La syntaxe d'utilisation de la fusion est la suivante :

#### Syntaxe d'une étape **DATA** permettant de fusionner n tables

```
DATA nom_librairie.nom_fichier;  
MERGE nom_librairie.nom_fichier-1 nom_librairie.nom_fichier-2 ... nom_librairie.nom_fichier-n;  
BY nom_variable_commune_aux_tables;  
RUN;
```

*Explication des instructions de cette syntaxe SAS :*

**Instruction 1:** L'instruction **DATA** suivie d'un nom de librairie et d'un nom de fichier indique à SAS le nom du fichier de données qu'il doit créer et dans quelle librairie ce fichier sera contenu.

**Instruction 2:** L'instruction **SET** suivie des différents noms de fichiers indique à SAS les fichiers de données dont les tables sont à fusionner.

**Instruction 3:** L'instruction **BY** suivit d'un nom de variable indique à SAS la variable par rapport à laquelle la fusion des tables doit être faite. Le rassemblement des données des tables sera effectué selon les valeurs de cette variable.

**Instruction 4:** L'instruction **RUN** termine l'étape **DATA**.

#### **Remarque!**

*Une condition doit être respectée afin que la fusion de tables fonctionne: les observations de chacune des tables à fusionner doivent être triées, en ordre croissant, par rapport à la variable de l'instruction **BY**, c'est-à-dire la variable par rapport à laquelle la fusion est effectuée. Il faut donc utiliser une procédure **SORT** pour chacune des tables avant la fusion.*

Illustrons une fusion de tables afin de permettre une meilleure compréhension du traitement effectué. Supposons que nous avons les trois tables suivantes:

Table 1

|   | Nom     | Age |
|---|---------|-----|
| 1 | Dubé    | 33  |
| 2 | Gucci   | 24  |
| 3 | Leclair | 30  |
| 4 | Sanson  | 28  |

Table 2

|   | Nom     | Salaire |
|---|---------|---------|
| 1 | Dubé    | 30000   |
| 2 | Gucci   | 25000   |
| 3 | Leclair | 60000   |
| 4 | Sanson  | 40000   |

Table 3

|   | Nom     | Dept |
|---|---------|------|
| 1 | Dubé    | Mark |
| 2 | Gucci   | Mark |
| 3 | Leclair | Rh   |
| 4 | Sanson  | Gop  |

La variable commune aux trois tables est la variable **Nom** et les individus correspondants à cette variable sont les mêmes pour les trois tables. Remarquez bien que la variable **Nom** est triée par ordre alphabétique donc croissante. Nous voulons fusionner ces trois tables et obtenir ceci:

Table 4

|   | Nom     | Age | Salaire | Dept |
|---|---------|-----|---------|------|
| 1 | Dubé    | 33  | 30000   | Mark |
| 2 | Gucci   | 24  | 25000   | Mark |
| 3 | Leclair | 30  | 60000   | Rh   |
| 4 | Sanson  | 28  | 40000   | Gop  |

Le programme SAS qui effectue ce traitement est le suivant:

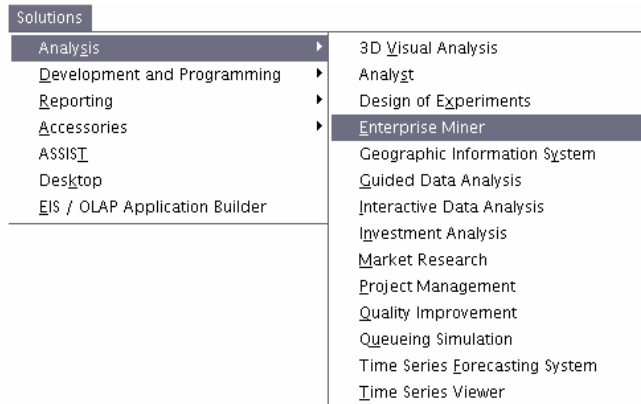
```
/*Ce programme fusionne les tables 1,2 et 3 afin de
créer la table 4*/

data exemple.table4;
merge exemple.table1 exemple.table2 exemple.table3;
by nom;
run;
```

## Introduction à la manipulation des fichiers de données dans le module SAS Entreprise Miner

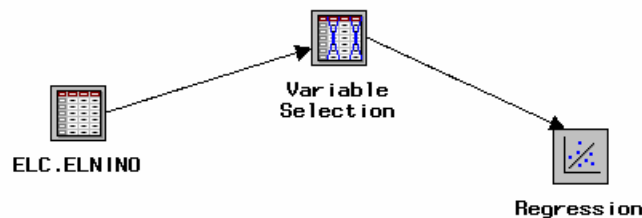
Nous allons consacrer cette dernière sous-section à un survol de la manipulation des fichiers de données à l'intérieur du module **SAS Entreprise Miner** qui est un module que vous utiliserez également dans vos cours.

Bien que **SAS Enterprise Miner** est externe à **SAS BASE**, nous jugeons qu'il est approprié de faire une introduction à celui-ci afin de vous donner une idée du fonctionnement de ce module.



Pour démarrer **SAS Enterprise Miner**, vous devez, à partir du menu de **SAS BASE**, cliquer sur la commande **Enterprise Miner** se trouvant dans le sous-menu **Analysis** du menu **Solutions**. Ce module sera démarré dans une nouvelle fenêtre.

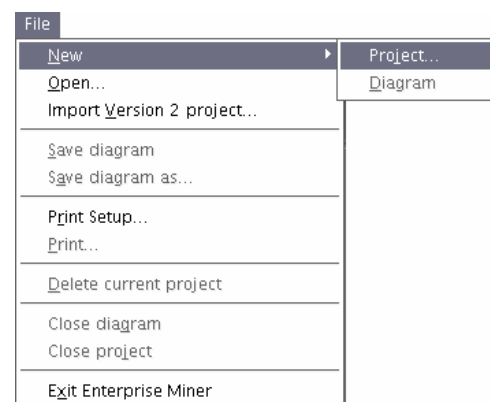
Le traitement des données dans ce module se fait à l'intérieur de **projets** dans lesquels l'utilisateur construit des **diagrammes** dont chacun des **nœuds** ont des tâches de traitement spécifiques. Voici un exemple de diagramme :



Pour commencer à travailler dans **Enterprise Miner**, il nous faut donc créer un projet.

Suivez les étapes suivantes afin de créer un projet.

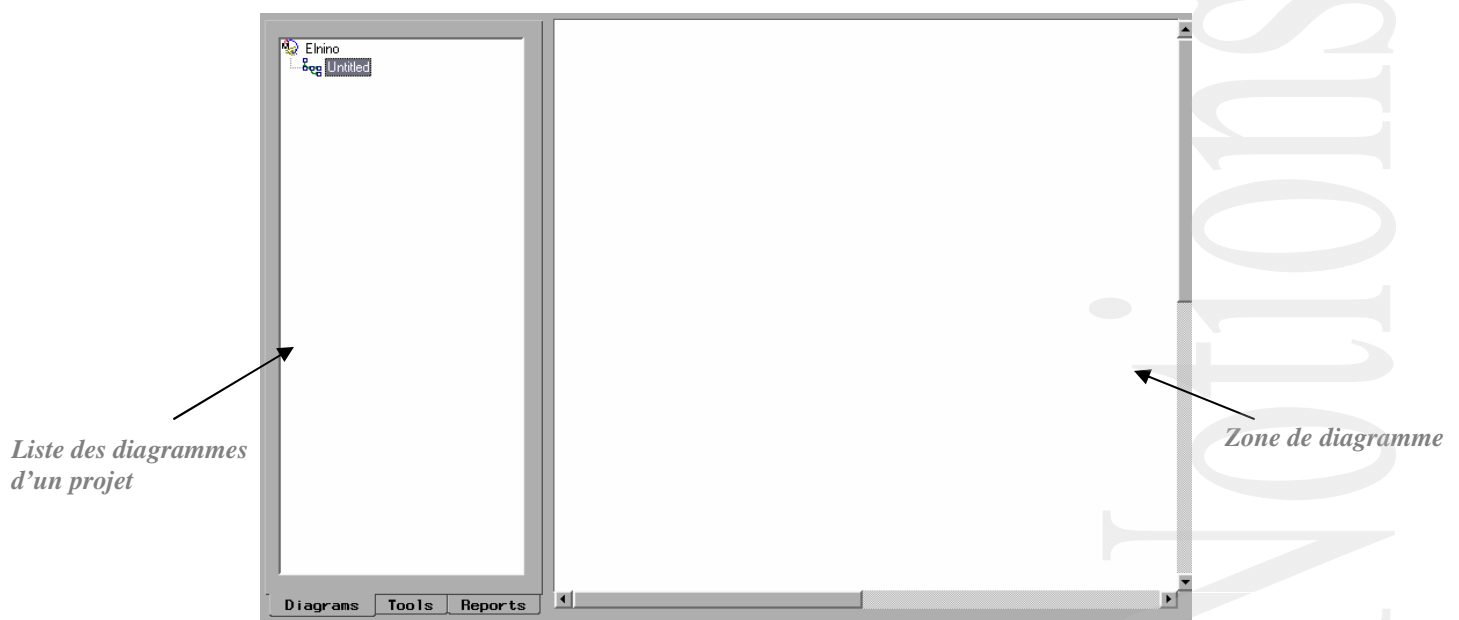
**Étape 1:** Lorsque vous êtes dans la fenêtre d'**Enterprise Miner**, cliquez sur le menu **File**, et sélectionnez la commande **Project** du sous-menu **New**.



**Étape 2:** Une fenêtre apparaît et vous demande de donner un nom au projet et de choisir un emplacement sur votre disque dur pour sauvegarder les fichiers de ce projet. Le répertoire utilisé par défaut dans SAS est: **C:\Documents and Settings\Nom \_utilisateur\My Documents\My SAS Files\V8\EM Projects\Nom du projet**. Par contre, vous pouvez le changer en cliquant sur le bouton **Browse**. Par exemple, si nous choisissons le nom **Elnino** pour un projet, SAS créera le répertoire suivant:



**C:\Documents and Settings\Nom d'utilisateur\My Documents\My SAS Files\V8\EM Projects\Elnino**. Donnez un nom et un emplacement au projet et cliquez sur **Create**. Voilà, vous venez de créer un projet et un fichier de projet (ayant **dmp** comme extension) est créé dans le répertoire que vous avez spécifié. Dans notre exemple, ce fichier est **Elnino.dmp**. Une fois que vous avez complété cette étape, une fenêtre comme celle illustrée ci-dessous apparaîtra.



La grande fenêtre du côté droit est la zone de diagramme. C'est donc dans cette fenêtre que vous aller dessiner et travailler vos diagrammes. Pour ce qui est de la fenêtre du côté gauche, c'est dans celle-ci qu'est affichée la liste des diagrammes d'un projet.

Nous pouvons maintenant créer un diagramme et les nœuds que nous allons survoler sont les nœuds suivants:



**Input Data Source:** Permet de spécifier le fichier de données à partir duquel nous voulons travailler.

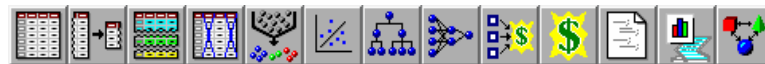


**Sampling:** Permet de créer un échantillon à partir des observations d'un fichier de données



**Data Partition:** Permet de créer des partitions, c'est -à-dire de séparer les observations d'un fichier sur plusieurs fichiers différents.

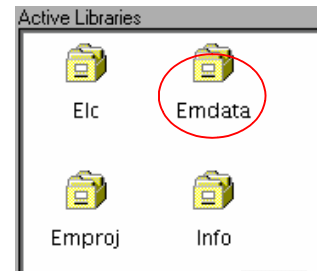
Ces nœuds sont généralement accessibles à partir d'une barre de nœuds qui se trouve dans la partie supérieure de la fenêtre d'**Entreprise Miner**. Ainsi, pour utiliser un nœud dans un diagramme, il suffit de glisser-déposer (*drag and drop*) ce nœud dans la zone de diagramme.



Avant de manipuler ces nœuds et créer un diagramme, nous aimerions traiter des librairies nommées **EMDATA**. Lorsque vous exécutez certains nœuds, dont les trois nœuds que nous traitons, des fichiers de données sont créés et ces fichiers de données de sortie sont entreposés dans une librairie permanente **EMDATA**.

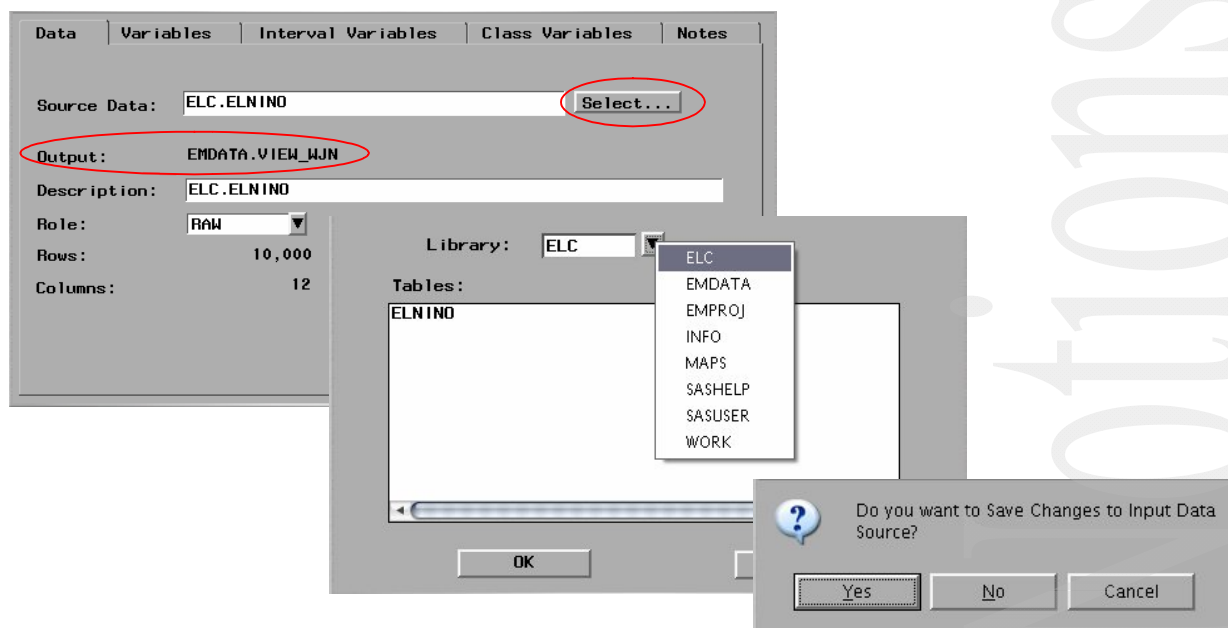
Une librairie permanente **EMDATA** est créée par défaut par SAS pour chaque nouveau projet que vous créez et le répertoire relié par défaut à cette librairie est **C:\Documents and Settings\Nom\_utilisateur\My Documents\My SAS Files\V8\EM Projects\Nom\_projet\emdata\**

Vous pouvez également avoir accès, comme pour toutes bibliothèques de SAS, aux fichiers de la bibliothèque **EMDATA** d'un projet à partir de la fenêtre **Explorer**. Cette bibliothèque s'affichera à chaque fois que vous créez ou ouvrez un projet.



Nous allons maintenant créer un diagramme et pour ce faire, suivre les étapes suivantes :

**Étape 1:** Glissez-déposez un nœud **Input DATA** dans la zone de diagramme et double-cliquez sur celui-ci pour accéder à ses commandes. Dans la fenêtre qui apparaît, vous retrouverez sous l'onglet **Data** qui vous permet de choisir le fichier de données à partir duquel vous voulez travailler. Pour choisir ce fichier, cliquez sur le bouton **Select** du champ **Source Data**. Il vous apparaîtra une nouvelle fenêtre vous donnant accès à la liste des bibliothèques SAS et à leurs fichiers de données. Vous n'avez qu'à sélectionner le fichier de données que vous voulez et à cliquer sur le bouton **OK**.



Une fois que vous avez choisi votre fichier (*dans cet exemple, le fichier **elnino** dans la bibliothèque **ELC***), vous n'avez qu'à fermer la fenêtre avec le bouton de fermeture se trouvant au coin supérieur droit de toute fenêtre Windows. SAS vous demandera alors

une confirmation de la sélection du fichier de données. Cliquez sur le bouton **Yes** à ce moment. Nous venons donc d'exécuter le nœud **Input Data**.

### Remarque!

*Si vous ne retrouvez pas une librairie, dans la liste des librairies disponibles du nœud Input, que vous avez créée dans une session antérieure de SAS, il est probable que vous n'ayez pas coché l'option «**Enable at Startup**» lors de la création de cette librairie ou que vous l'avez créée avec l'instruction **libname**. Pour retrouver cette librairie, il suffit de la recréer, avec la méthode de votre choix, avant d'exécuter le nœud Input.*

L'exécution de ce nœud crée un fichier de sortie qui est inséré dans la librairie **EMDATA** du projet et vous pouvez voir le nom de celui-ci dans les informations de l'onglet **Data** (*view\_wjn.sas7bview*). Remarquez que l'extension de ce fichier est **sas7bview** ce qui indique que celui-ci n'est pas un fichier de données SAS. En fait, ce fichier est ce qu'on appelle un «SAS data view» qui est un fichier qui pointe vers le fichier de données que vous avez choisi. Dans la plupart des cas, un «SAS data view» peut être utilisé comme s'il s'agissait d'un fichier de données SAS.

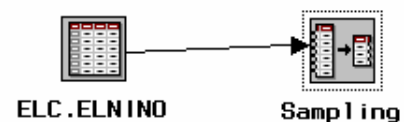
### Répertoire Emdata du projet elnino avant l'exécution du nœud Input DATA

| d Settings\Eng Seng\My Documents\My SAS Files\V8\EM Projects\Elnino\emdata |      |      |
|--|------|------|
| Na...  | Size | Type |
|  |      |      |

### Répertoire Emdata du projet elnino après l'exécution du nœud Input DATA

| tings\Eng Seng\My Documents\My SAS Files\V8\EM Projects\Elnino\emdata |      |            |
|---|------|------------|
| Na...   | Size | Type       |
| view_wjn.sas7bview  | 5 KB | SAS System |

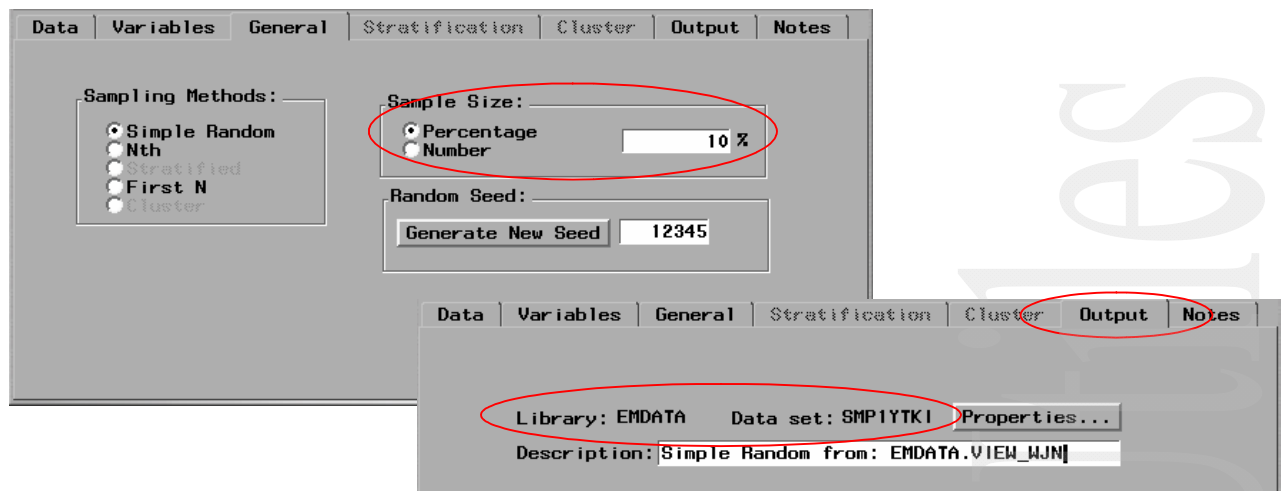
**Étape 2:** Glissez-déposez le nœud **Sampling** dans la zone de diagramme. Il faut maintenant relier le nœud **Input Data** au nœud **Sampling**. Pour faire cela, placer




le curseur de la souris aux alentours du nœud **Input Data**. Lorsque celui-ci prend la forme d'une croix, appuyez, gardez le bouton de la souris enfoncée et tirez une flèche



vers le nœud **Sampling**. Cela fera en sorte que le nœud **Sampling** utilisera le fichier du nœud **Input Data** et créera un échantillon à partir de celui-ci.

Double-cliquez maintenant sur le nœud **Sampling** pour accéder à ses commandes et dans l'onglet **General** de la fenêtre qui apparaît, indiquez la taille de l'échantillon que vous voulez créer. Ceci peut être spécifié en nombre d'observations ou en pourcentage d'observations du fichier source. SAS créera donc un échantillon et assignera ce fichier à la librairie **EMDATA** du projet. Vous pouvez voir le nom de ce fichier en cliquant sur l'onglet **Output** (*smp1ytki.sas7bdat* dans cet exemple).



Une fois que vous avez spécifié la taille de l'échantillon que vous voulez créer, cliquez sur le bouton **Submit**  qui se trouve sur la barre d'outil de la fenêtre SAS pour exécuter le nœud **Sampling**. SAS vous demandera si vous voulez inspecter l'échantillon qui est créé. Cliquez sur **Yes** si vous désirez l'inspecter ou **No** sinon. Fermer ensuite la fenêtre de commandes du nœud **Sampling**.

### Répertoire Emdata du projet elnino après l'exécution du nœud Sampling

| tings\Eng Seng\My Documents\My SAS Files\V8\EM Projects\Elnino\emdata                                  |        |              |
|--|--------|--------------|
| Na...  | Size   | Type         |
|  smp1ytki.sas7bdat  | 209 KB | SAS System I |
|  view_z9s.sas7bview | 5 KB   | SAS System I |



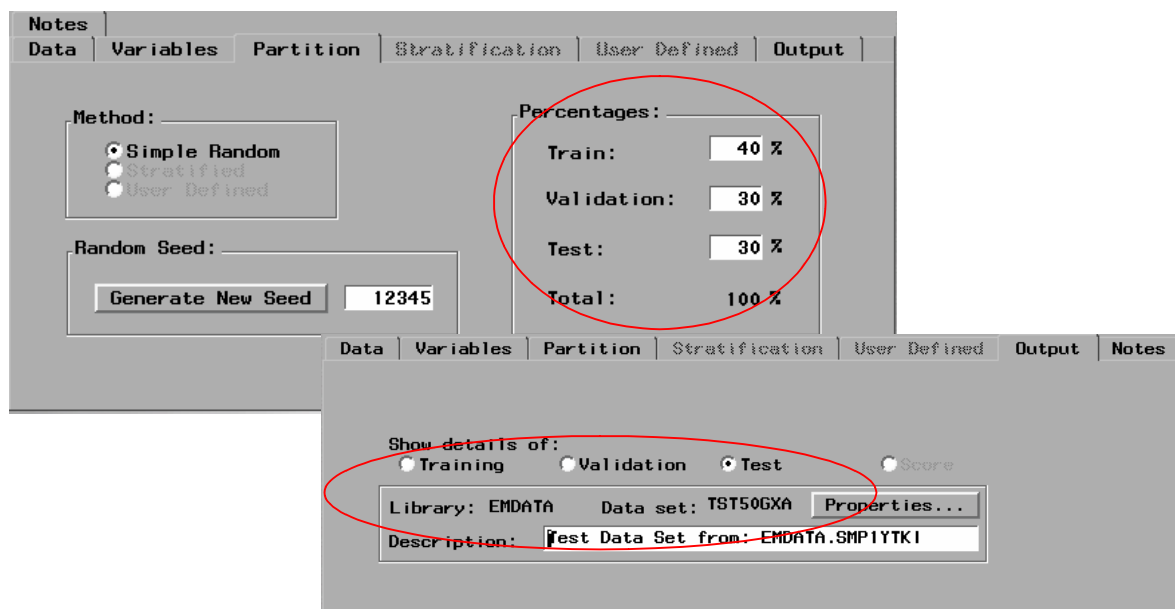
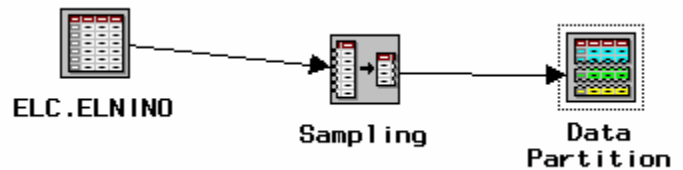
**Étape 3:** Finalement, glissez-déposez le nœud **Partition** dans la zone de diagramme et tirez une flèche partant du nœud

**Sampling** et allant au nœud **Partition**.

Ceci fait en sorte que le nœud **Partition**

utilisera le fichier de données produit par

le nœud **Sampling** pour créer des partitions, c'est-à-dire séparer les observations de ce fichier en deux ou trois fichiers différents. Double-cliquez sur le nœud partition pour accéder aux commandes de celui-ci. Dans la fenêtre qui apparaît, vous vous trouverez dans l'onglet **Partition** à partir duquel vous pouvez spécifier les proportions du fichier source à allouer à trois différentes partitions (*fichier Train, fichier Validation et fichier Test*). Le total de ces proportions doit éga100%. Il vous est donc possible de séparer le fichier de données sur, au plus, trois fichiers différents. Ces fichiers de données seront bien sûr assignés à la bibliothèque **EMDATA** du projet et pour connaître les noms de ceux-ci, vous pouvez cliquer sur l'onglet **OUTPUT** et sélectionnez **Training, Validation** ou **Test** afin d'afficher le nom de chacun de ces fichiers respectivement.

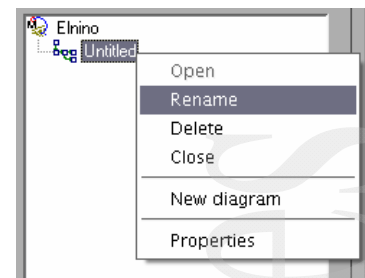



## Répertoire Emdata du projet elnino après l'exécution du nœud Partition

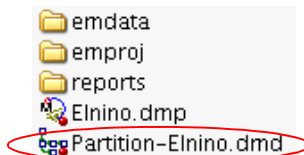
tings\Eng Seng\My Documents\My SAS Files\V8\EM Projects\Elnino\emdata

| Na...              | Size   | Type       |
|--------------------|--------|------------|
| smp1ytki.sas7bdat  | 209 KB | SAS System |
| view_z9s.sas7bview | 5 KB   | SAS System |
| valpo4yp.sas7bdat  | 65 KB  | SAS System |
| trnzeg72.sas7bdat  | 81 KB  | SAS System |
| tst50gxa.sas7bdat  | 65 KB  | SAS System |

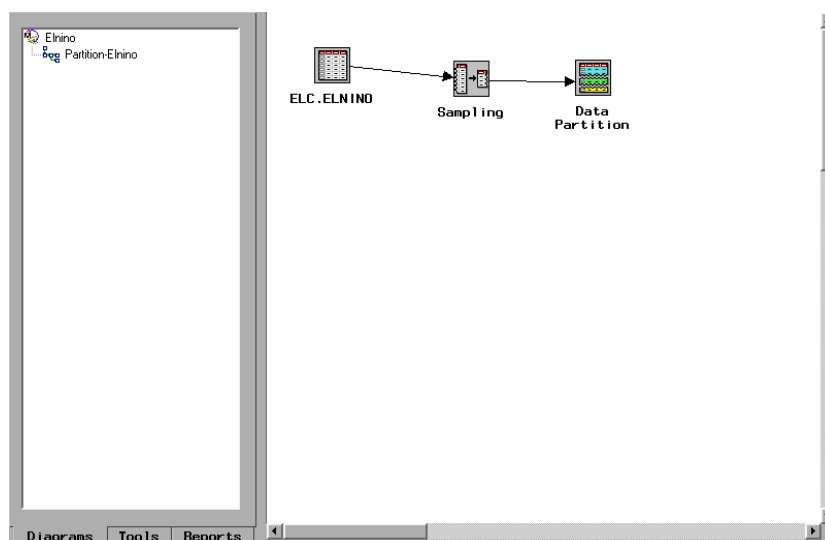
Terminons cette sous-section en sauvegardant le diagramme que nous avons créé. Tout d'abord, donnons un nom à ce diagramme car en ce moment, le diagramme est encore intitulé **Untitled**. Pour changer ce nom, placer le curseur de la souris sur **Untitled** et cliquer sur le bouton droit de la souris. Du menu apparaissant, choisissez la commande **Rename** et inscrivez le nom que vous désirez. Par exemple, inscrivons le nom **Partition-Elnino**.



Par la suite, sauvegardez le diagramme en cliquant sur le bouton **Save**  ou en cliquant sur la commande **Save diagram** du menu **File**. Un fichier de diagramme (ayant une extension **dmd**) sera créé dans le répertoire du projet. Dans notre cas, ce fichier est **Partition-Elnino.dmd**.



Le résultat final de toutes ces manipulations est le suivant:





# Section 6

Ressources à consulter



## Quelques documents à consulter

Nous allons terminer ce guide en vous référant à quelques documents que vous pouvez consulter si vous désirez approfondir vos connaissances à propos de SAS.

### Livres à consulter

Voici une liste non exhaustive des documents que vous pouvez consulter à la bibliothèque Myriam et J.-Robert Ouimet située à HEC Montréal.

**The little SAS book : a primer / Lora D. Delwiche and Susan J. Slaughter. – [Delwiche, Loran D.](#), Cary, N.C. : SAS Publ., 1996.**

**A handbook of statistical analyses using SAS / Geoff Der and Brian S. Everitt. [Der, Geoff.](#), Boca Raton: Chapman & Hall/CRC, c2002.**

**SAS programming for researchers and social scientists / Paul E. Spector. – [Spector, Paul E.](#), Thousand Oaks: Sage Publications, c2001.**

### Liens Internet

De plus, voici une liste de sites Internet qui sont des tutoriaux en ligne de SAS

<http://www.umanitoba.ca/centres/mchp/teaching/sasmanual/index.shtml>

<http://www.itc.virginia.edu/research/sas/training/v8/>

<http://www.utexas.edu/cc/stat/tutorials/sas8/sas8.html>

Finalement, vous pouvez également consulter la documentation officielle de SAS en ligne pour les versions 8 et 9 de SAS.

SAS version 8 - <http://v8doc.sas.com/sashtml/>

SAS version 9 - <http://support.sas.com/onlinedoc/912/docMainpage.jsp>

